Proceedings of the 25th Computer Vision Winter Workshop Conference

Alan Lukežič, Domen Tabernik, Klemen Grm (eds.)

Rogaška Slatina, February 3-5, 2020

Proceedings of the 25th Computer Vision Winter Workshop Conference February 3-5, 2020, Rogaška Slatina, Slovenia

Volume Editors: Alan Lukežič, Domen Tabernik, Klemen Grm
https://cvww2020.vicos.si/proceedings

Publisher
Slovenian Pattern Recognition Society, Ljubljana 2020
Electronic edition
Slovenian Pattern Recognition Society, Ljubljana 2020

CIP zapis:
Kataložni zapis o publikaciji (CIP) pripravili v
Narodni in univerzitetni knjižnici v Ljubljani
COBISS.SI-ID=303769344
ISBN 978-961-90901-9-0 (pdf)

Contents

Preface

We would like to welcome you to the 25th Computer Vision Winter Workshop (CVWW2020). This year the workshop is organized by the Slovenian Pattern Recognition Society (SPRS), and held in Rogaška Slatina, Slovenia, from February 3rd to February 5th, 2020. We hope that your experience at CVWW is both professionally and personally rewarding!

The Computer Vision Winter Workshop (CVWW) is an annual international meeting of several computer vision research groups, located in Ljubljana, Prague, Vienna, and Graz. The aim of the workshop is to foster interaction and exchange of ideas among researchers and PhD students. The focus of the workshop spans a wide variety of computer vision and pattern recognition topics, such as image analysis, medical imaging, 3D vision, human-computer interaction, vision for robotics, machine learning, as well as applied computer vision and pattern recognition.

CVWW 2020 received a total of 30 submissions from six countries. The paper selection was coordinated by the Program Chairs, and included a rigorous double-blind review process. The international Technical Program Committee consisted of 33 renowned computer vision experts, who conducted the review. Each submission was examined by three experts, who were asked to comment on the strengths and weaknesses of the papers and justify their recommendation for accepting or rejecting a submission. The Program Chairs used the reviewers' comments to render the final decision on each paper. As a result of this review process, 8 original papers were accepted for publications. These have been presented at the workshop as oral presentations. Workshop also included 16 invited presentations of on-going works. The Program Chairs would like to thank all reviewers for their high-quality and detailed comments, which served as a valuable source of feedback for all authors, and most of all for their time and effort, which helped to make the CVWW2020 a success.

The workshop program included an invited talk by assoc. prof. dr. Tatiana Tommasi (Department of Control and Computer Engineering, Politecnico di Torino), to whom we thank for her participation. We also extend our thanks to the Slovenian Pattern Recognition Society, through which the workshop was organized, and we want to acknowledge and thank our supporters from the Faculty of Computer and Information Science, University of Ljubljana for their contributions. To the sponsor and their representatives in attendance, thank you!

We hope that the 25th iteration of the Computer Vision Winter Workshop is a productive and enjoyable meeting for you and your colleagues, and inspires new ideas that can advance your professional activities.

Welcome and thank you for your participation!

Official sponsor

University *of Ljubljana*
Faculty *of Computer and Information Science*

Committees

Workshop Chairs

Alan Lukežič, University of Ljubljana
Domen Tabernik, University of Ljubljana
Klemen Grm, University of Ljubljana

Program Committee

Dániel Baráth, Czech Technical University in Prague
Horst Bischof, Graz University of Technology
Borja Bovcon, University of Ljubljana
Jan Cech, Czech Technical University in Prague
Luka Cehovin, University of Ljubljana
Ondrej Chum, Czech Technical University in Prague
Matej Dobrevski, University of Ljubljana
Ziga Emersic, University of Ljubljana
Klemen Grm, University of Ljubljana
Jiří Hladůvka, TU Wien
Tomas Hodan, Czech Technical University in Prague
Marija Ivanovska, University of Ljubljana
Matej Kristan, University of Ljubljana
Janez Krizaj, University of Ljubljana
Walter Kropatsch, TU Wien
Alan Lukezic, University of Ljubljana
Jiri Matas, Czech Technical University in Prague
Blaž Meden, University of Ljubljana
Dmytro Mishkin, Czech Technical University in Prague
Jon Muhovic, University of Ljubljana
Janez Perš, University of Ljubljana
Horst Possegger, Graz University of Technology
Peter Rot, University of Ljubljana
PETER ROTH, Graz University of Technology
Robert Sablatnig, TU Wien
Danijel Skocaj, University of Ljubljana
Julia Skovierova, Czech Technical University in Prague
Vitomir Struc, University of Ljubljana
Radim Šára, Czech Technical University in Prague
Jaka Šircelj, University of Ljubljana
Domen Tabernik, University of Ljubljana
Giorgos Tolias, Czech Technical University in Prague
Vitjan Zavrtanik, University of Ljubljana

Invited Talk

Learning to Generalize with Self-Supervision

Tatiana Tommasi
Department of Control and Computer Engineering, Politecnico di Torino
tatiana.tommasi@polito.it

Although deep networks have significantly increased the performance of visual recognition methods, it is still challenging to achieve the robustness across visual domains that is necessary for real-world applications. In many practical tasks collecting annotated samples may be very costly, but at the same time using models trained from data belonging to a different domain will produce only poor results. To tackle this issue, research on Domain Adaptation (DA) and Generalization (DG) has flourished over the last decade with several approaches based on feature alignment, generative and adversarial solutions. In this talk I will present a new point of view on the DA and DG settings that considers self-supervision as an auxiliary powerful tool to adapt and generalize across domains. Specifically the talk will show how solving a jigsaw puzzle or recognizing the orientation of an image can improve robustness and support generalization of models learned on photos, cartoon, paintings or sketches. We will also see how this beneficial effect extends from object classification to detection and may also be applied when the shift across domains involve different label sets (partial domain adaptation) or when the target domain reduces to a single test sample.

Original Contributions

25^{th} Computer Vision Winter Workshop
Domen Tabernik, Alan Lukežič, Klemen Grm (eds.)
Rogaška Slatina, Slovenia, February 3–5, 2020

On Learning Vehicle Detection in Satellite Video

Roman Pflugfelder[1,2], Axel Weissenfeld[1], Julian Wagner[2]
[1]AIT Austrian Institute of Technology, Center for Digital Safety & Security
[2]TU Wien, Institute of Visual Computing & Human-Centered Technology
{roman.pflugfelder|axel.weissenfeld}@ait.ac.at, e1326108@student.tuwien.ac.at

Abstract. *Vehicle detection in aerial and satellite images is still challenging due to their tiny appearance in pixels compared to the overall size of remote sensing imagery. Classical methods of object detection very often fail in this scenario due to violation of implicit assumptions made such as rich texture, small to moderate ratios between image size and object size. Satellite video is a very new modality which introduces temporal consistency as inductive bias. Approaches for vehicle detection in satellite video use either background subtraction, frame differencing or subspace methods showing moderate performance (0.26 - 0.82 F_1 score). This work proposes to apply recent work on deep learning for wide-area motion imagery (WAMI) on satellite video. We show in a first approach comparable results (0.84 F_1) on Planet's SkySat-1 LasVegas video with room for further improvement.*

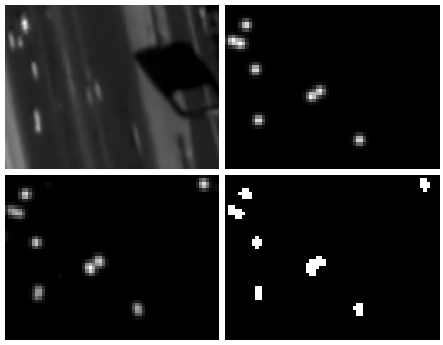


Figure 1. Results of the proposed method. Top, left: video frame of the SkySat-1 LasVegas video showing a city highway with multiple cars. Top, right: vehicle labelling provided by Zhang et al. [34, 33]. Bottom, left: the method's response (heat) map. Bottom, right: the final segmentation result. The network detects all labelled cars and even a bus or truck at the right image border.

1. Introduction

Object detection, i.e. the recognition and localisation of objects, in visual data is a very important and still unsolved problem. For example, the problem becomes challenging in aerial imaging and remote sensing as the data and scenes differ significantly from the case considered usually in computer vision [6, 25].

Such remote detection is important in surveillance, as demanding applications let surveillance currently undergo a transition from near to mid distances (as with security cameras) to sceneries such as whole cities, traffic networks, forests, and green borders. Beside coverage new, low orbit satellite constellations[1] will allow multiple daily revisits and constantly falling costs per image. Such applications can be found e.g. in urban planning, traffic monitoring, driver behaviour analysis, and road verification for assisting both scene understanding and land use classification. Civilian and military security is another area to benefit with applications including military reconnaissance, detection of abnormal or dangerous behaviour, border protection, and surveillance of restricted areas.

Although remotely acquired data shows great reduction of occlusion and perspective distortion due to the overhead view, new difficulties arise. Typical aerial and satellite images are very large in resolution and data size. For example, wide-area motion imagery (WAMI) provides instead of a few megapixel (MP) typical for security cameras up to 400 MP per image frame and three image frames per second (2.2 TB/s for 16 bit per px). Satellite video gives today 4K RGB video with 30 frames per second (759.3 MB/s). Satellite images capture large sceneries, usually dozens of square kilometers which introduce instead of a few visually large objects, thou-

[1]https://earthi.space, 11/03/2019

sands of tiny objects coming from hundreds of categories in a single image. At the same time these objects reduce in pixel size by orders of magnitude from 10^4 px to 10^2 px, to even 10 px for satellite video [34], depending on the camera's ground sample distance (GSD)[2].

This severe magnification of scenery and reduction of object size to very tiny appearances have consequences. Object detection becomes very ambiguous and sensitive to noise and nuisances and the search space dramatically increases and becomes very sparse. Inferred labels of data usually capture instead of the bounding box or contour sole positions, as the extent of objects is even for humans, e.g. in WAMI or satellite video, unrecognizable. All this leads to major difficulties if not inapplicability of vanilla methods [17]. Manual labelling of data is furthermore very tedious, for many cases impossible, hence, research on object detection in satellite video relies currently on background subtraction and frame differencing [16, 28, 32, 18, 4, 3].

Recent literature [22, 23, 35, 15, 31, 8, 27, 17, 34, 13, 2, 29, 30] also suggests to apply deep learning on aerial and satellite high resolution RGB single images, however, the work shows moderate performance for GSD larger than 15 cm [23]. All work is also tested with rather narrow datasets of very different sceneries which makes the validity of the results questionable and the comparison of methods difficult. It is therefore unclear, if deep learning on high resolution images will further improve, given the limitations of the data.

Another problem of still images is the impossibility to capture the dynamic behaviour of vehicles which is essential for many applications. For example, vehicle heading and speed are important indicators in traffic models. Although rapid retargeting for multi-angular image sequences with Worldview-2 is possible [21], the time interval of around one minute between consecutive images is too large for reasonable analysis.

For these reasons the paper addresses the problem of vehicle detection in satellite video. Such video was introduced 1999 by DLR-TubSat, since 2013 Planet's SkySat-1 delivers up to 120 s, 30 Hz, 2K panchromatic video covering two areas of 1.1 km^2 with up to 80 cm GSD. China's Jilin programme launched 2015, now provides even 4 MP color video.

[2]GSD is the spatial distance of two adjacent pixels on the image measured on the ground.

To the best of our knowledge this is the first work on using neural networks and deep learning to directly regress positions of vehicles *in satellite video*. Inspired by recent work on WAMI [17] this paper proposes to exploit the temporal consistency in satellite video by using a neural network and deep learning instead of using background subtraction or frame differencing, by this improving over the state-of-the-art in vehicle detection with satellite video. To overcome shortage of labelled video, this work follows in this context the novel idea of transfer learning by recognising similarity of WAMI and satellite video data.

To summarise, the contributions of this work are

- the confirmation of results in LaLonde et al. [17] which shows clearly improvement in vehicle detection (from 0.79 to 0.93 in F_1 score) when using a spatiotemporal convolutional network,
- empirical results showing the applicability of FoveaNet [17] to reduced resolution (0.91 F_1 score for 40% of the original image resolution and 0.79 F_1 score for 20%), yielding sizes of up to 3.6×1.8 px for vehicles which simulates satellite video and finally,
- a transfer learning approach that uses labelled WAMI data to train a detector for satellite video with 0.84 F_1 score which is comparable to the currently best (subspace) method E-LSD[33] with 0.83 F_1 score on the same data.

2. Related Work

Deep learning significantly improved previously handcrafted methods of object recognition [6]. Neural networks and back-propagation allow a learning formalism, where features and inference are jointly learnt from data in a neat end-to-end framework. Object detection is designed either as direct regression of bounding box image coordinates [24] or by using the idea of object proposals as intermediate step [25].

These developments triggered also work on deep learning for object detection in remote sensing [22, 23, 35, 15, 31, 8, 27, 17, 34, 13, 2, 29, 30]. Applying deep learning for remote sensing is challenging, as labels are very expensive for satellite data and good augmentation, transfer learning or even unsupervised methods circumventing this problem are currently unknown [38, 20]. Besides deep learning, object detection in remote sensing can be categorised

according to the approach taken as well as the sensor modality, i.e. satellite image, sequence of multi-angular satellite images, satellite video, aerial image and WAMI.

Applying a classifier on top of a sliding window is one possible approach. Using a convolutional neural network in combination with hard negative mining showed by a F_1 score of 0.7 reasonable results with 15 cm GSD on aerial images [15]. Following the golden standard [25], adapted variants of the base feature, region proposal and Fast R-CNN network have been proposed such as using skip connections in the base and focal loss [31], or using a dilated, multi-scale VGG16 as base in combination with hard negative mining [8] which gives AP and Recall larger than 0.8 in their experiments. Guo et al. [27] introduces proprietary base, region proposal and detection networks, but did not show results on vehicles. This approach is useful with aerial images, but fails entirely for 1 m GSD video as shown by [34] (F_1 score of 0.5). Results on high resolution satellite images are still unknown in literature.

Another idea is to pixel-wise classify vehicle vs. background (semantic segmentation), e.g. by combining Inception and ResNet to give a heatmap. Assuming a fixed vehicle size and using non-maxima suppression gives excellent results [23] (F_1 score larger than 0.9). Imbert proposes a generative U-Net in combination with hard negative mining for satellite images but kept unfortunately results in absolute F_1 scores confidential.

Spatiotemporal information is a further cue important in object detection, especially with WAMI and satellite video. The standard is to use background subtraction (BGS) [35, 16, 28, 32, 1] and frame differencing (FD) [18, 4, 3], except Al-Shakarji et al. [2] who combined YOLO with spatiotemporal filtering on WAMI (F_1 score of 0.7), and Mou and Zhu [22] who use KLT tracking on video with a SegNet on overlapping multispectral data, however, they did not show results for vehicles. Zhang and Xiang [35] apply a ResNet classifier trained on CIFAR on proposals from a mixture of Gaussians foreground model, but did not show a proper evaluation.

The standard here is to apply connected component analysis [16, 28], saliency analysis, segmentation [32, 18], distribution fitting [4, 3] followed by morphology. F_1 scores of larger than 0.9 for ships and scores between 0.6 and 0.8 for vehicles on the Burji Khalifa [32], Valencia [4, 3] and Las Vegas [16] videos suggest BGS, FD for larger objects. Both BGS and FD depend heavily on registration and parallax correction, hence, these methods introduce various nuisances for vehicles which are difficult to handle. Evaluation on single, selective scenes is further too narrow to draw a final conclusion.

Very recent work [33] suggests a subspace approach for discriminating vehicles and background. The idea shows potential with F_1 score results of larger than 0.8 on the simple Las Vegas video, which therefore needs further evaluation with more complex traffic patterns.

Another problem is the sparsity of vehicle occurrences in very large images as in WAMI which has been tackled by clustering the large images to draw attention to certain parts of the image and then to apply convolutional neural networks on single images [29][30] or multiple video frames [17] for final detection. Such clustering combined with deep spatiotemporal analysis shows excellent results on WAMI (F_1 score larger than 0.9) [17].

Also very recently tracking of airplanes, trains and vehicles has been considered for satellite video [10, 9, 26, 12], either by using optical flow [10, 9], correlation trackers (KLT) [26] or a combination of correlation and Kalman filters [12].

3. Methodology

With our goal of detecting moving vehicles in satellite videos, we were inspired by the work of Lalonde et al. [17], who designed two neural networks, denoted as ClusterNet and FoveaNet, to detect vehicles in WAMI. The ClusterNet proposes regions of objects (ROOBI) based on areas of interest (AOI), which are input to the FoveaNet. Instead of using the ClusterNet to determine ROOBIs we split the AOI into square tiles (ROOBIs) with size $N \times N$; e.g. N=128 px. The object detection based on the FoveaNet consists of two steps as depicted in Fig. 2.

3.1. FoveaNet and thresholding

The FoveaNet is a fully convolutional neural network (CNN) and consists of eight convolutional layers. The number of filters per convolution are 32, 32, 32, 256, 512, 256, 256 and 1. Their filter sizes are summarized in Tab. 3. After the first convolution a 2×2 max pooling is carried out. Moreover, during training the 6^{th} and 7^{th} convolutional layers have a 50% dropout. The heatmap is generated by the final 1×1 convolutional layer where each neuron gives a

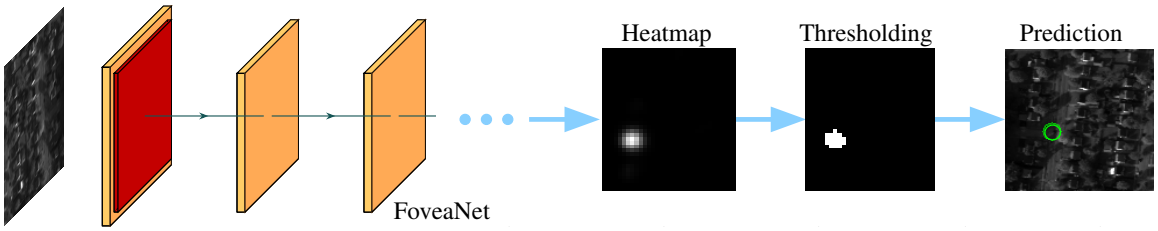


Figure 2. The object detection process consists of two steps [17]: The FoveaNet predicts a heatmap, which indicates the likelihood that an object is at a given image coordinate. Vehicles are detected by thresholding the heatmap.

vote of the likelihood of a moving vehicle at pixel level.

The input to the network is a stack of frames with size $N \times N \times c$, where $N \times N$ is the ROOBI size and c depicts the number of consecutive adjoining frames in a stack. Hereinafter we refer to c as channels. Thereby, the CNN shall learn to predict the positions of the objects of the central frame. We believe the FoveaNet is capable to learn spatiotemporal features by feeding the network with stacks of multiple frames (e.g. c=5), which are especially important in lower resolution images as existing in satellite videos.

The ground truth is based on heatmaps H, which are created by superimposing Gaussian distributions, where the center of each distribution is the pixel position (x,y) of the vehicle in the image:

$$H(x,y) = \sum_{n=1}^{N} \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad (1)$$

where n are the downsampled ground-truth coordinates provided in pixel positions and σ is the variance of the Gaussian blur. During training the network learns to minimize the Euclidean distance between the network output and the generated ground truth heatmaps.

The original FoveaNet uses ReLUs as activation functions. We discovered, however, the problem known as the "Dying ReLU" problem[3]. During training, a weight update triggered by a large gradient flowing through a ReLU can make the neuron inactive. If this happens, the gradient flowing through this ReLU will always be zero and the network continues to give the same output. In our trainings we frequently discovered this phenomenon ($\sim$71% of the cases) using the Xavier initialization [11]. Hence, we replaced the ReLUs with either ELUs (Exponential Linear Unit) or Leaky ReLUs.

The second step processes the predicted heatmap to determine the objects' positions. For this, the heatmaps are converted into segmentation maps via OTSU thresholding [17]. If the segmented area is larger than a threshold α, then the center of the area is defined as the object position.

3.2. Transfer learning

To the best of our knowledge there are currently no annotated datasets of satellite videos publicly available. In contrast, there are some labeled WAMI datasets accessible; e.g. the WPAFB dataset[4] contains over 160.000 annotated moving vehicles. WAMI and satellite images, however, differ considerably, among other things due to the different GSD. For instance, the WPAFB images have about four times higher GSD than the LasVegas video. Our core idea is to use transfer learning for a domain transfer from WAMI to satellite images. For this, we train our CNN based on the WPAFB dataset. Afterwards we fine-tune the CNN on satellite video data.

4. Experimental Evaluation and Results

Our network was trained from scratch using PyTorch - we used Adam with a learning rate of 1e-5 and a batch size of 32. Data preparation includes frame registration to compensate camera motion.

We conducted three experiments. In the first experiment we carry out a baseline evaluation to reproduce the results of [17]. For the second experiment, we reduce the image resolution (GSD) of the WPAFB dataset. Thereby, the vehicle size in these low-resolution images is in the same order as in satellite videos. In the third experiment, we carry out a fine-tuning and evaluate the FoveaNet on satellite data.

[3] http://cs231n.github.io/neural-networks-1, 11/03/2019

[4] https://www.sdms.afrl.af.mil/index.php?collection=wpafb2009, 11/03/2019

Figure 3. AOI 40 contains a lot of dense traffic passing the intersection. On the contrary, AOI 41 contains mostly single vehicles driving on the road. Traffic patterns of AOI 34 are a combination of AOI 40 and AOI 41.

Detections are considered true positives if they are within a certain distance θ of a ground truth coordinate. If multiple detections are within this radius, the closest one is taken and the rest, if they do not have any other ground truth coordinates within the distance θ, are marked as false positives (FP). Any detections that are not within θ of a ground truth coordinate are also marked as FP. Ground truth coordinates which have no detections within θ are marked as false negatives. Quantitative results are compared in terms of precision, recall, and F_1 measure.

To compare our results with LaLonde et al. [17] we selected three of their AOIs (area of interest) - 34, 40 and 41. The contents of the AOIs 40 and 41 with respect to traffic patterns widely differ as displayed in Fig. 3. Whereas AOI 40 contains a lot of dense traffic at a main intersection, AOI 41 mainly consists of single vehicles on the road. AOI 34 is a combination of both traffic patterns. Data was split into training and testing in the following manner: AOI 34 was trained on AOIs 40 and 41. AOI 40 was trained on AOIs 34 and 41 and AOI 41 was trained on 34 and 40. In contrast to [17], we omitted AOI 42 for training as it is a sub-region of AOI 41.

For training and evaluation based on the WPAFB dataset, only frames with moving vehicles were included. We excluded frames without moving vehicles as our approach focuses solely on the detection and omits the region proposal part (ClusterNet) of [17]. A vehicle is defined as moving if it moves at least ω pixel within 5 frames.

4.1. Experiment 1: Baseline evaluation

In the first experiment we reproduced the results in [17]. For this, we set the following parameters: N=100 px (ROOBI edge length), σ=2 (variance of Gaussian blur), θ=40 px (evaluation threshold), ω=15 px (threshold for removing stationary cars) and α=15 px (threshold to disregard small segments). Tab. 1 indicates that our results are in the same order of magnitude than [17]. For instance, we achieve a F_1 score of 0.90 in AOI 34 (c=5), whereas Lalonde et al. have a F_1^* score of 0.93. The difference in the results is most likely due to the implementation differences of the second step, where we i.a. do not split connected regions into multiple detections. This presumption is confirmed looking at the evaluation results of AOI 40, where the differences of the F_1 score are greatest. AOI 40 contains a lot of dense traffic at the intersection resulting in connected regions, which cause false negative detections (Fig. 4). Furthermore, the results in Tab. 1 confirm that the network is learning spatiotemporal features which improve the overall performance comparing single versus multi-channels. For instance, the precision of AOI 34 increases from 0.73 (c=1) to 0.87 (c=5).

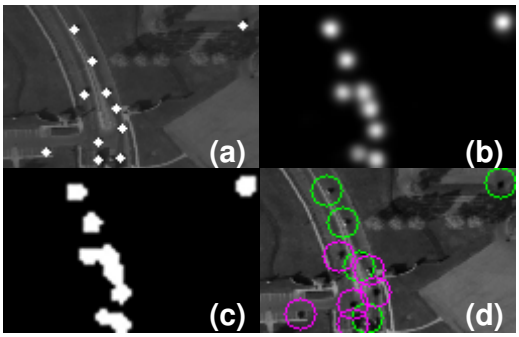


Figure 4. The detection of vehicles in crowded scenes is error-prone. Detection results of a ROOBI with reduced resolution (SF=0.2): (a) ground truth, (b) predicted heatmap, (c) after thresholding, (d) detected vehicles (green: true positives, pink: false negatives)

Table 1. Results are based on three AOIs of the WPAFB dataset with various channel sizes (c). For comparison, F_1^* scores of [17] are provided. Results of the second experiment include two scaling factors - 0.4 and 0.2.

		Experiment 1				Experiment 2					
		Full Resolution				Scaling factor 0.4			Scaling factor 0.2		
AOI	c	**Prec.**	**Rec.**	F_1	F_1^*	**Prec.**	**Rec.**	F_1	**Prec.**	**Rec.**	F_1
34	1	0.73	0.88	0.79		0.55	0.55	0.55	0.39	0.35	0.37
40	1	0.73	0.82	0.77		0.55	0.48	0.51	0.20	0.21	0.20
41	1	0.76	0.90	0.82		0.60	0.72	0.65	0.28	0.42	0.34
34	3	0.86	0.94	0.90		0.93	0.77	0.84	0.80	0.61	0.69
40	3	0.92	0.89	0.90		0.95	0.69	0.80	0.93	0.56	0.70
41	3	0.93	0.93	0.93		0.97	0.84	0.90	0.89	0.69	0.77
34	5	0.87	0.93	0.90	0.93	0.94	0.78	0.85	0.91	0.63	0.74
40	5	0.92	0.89	0.90	0.98	0.96	0.70	0.81	0.90	0.57	0.70
41	5	0.93	0.92	0.93	0.93	0.97	0.85	0.91	0.90	0.70	0.79

4.2. Experiment 2: Downscaled WPAFB dataset

For the second experiment we reduced the images by a scaling factor (SF) of 0.4 and 0.2 resulting in 40% and 20% of the original image resolution, respectively. We selected a SF of 0.2, because this factor reduces the typical vehicle object size in the WPAFB dataset from the order of 18×9 px to 3.6×1.8 px, which is like the vehicle size in satellite videos. The following parameters were set for the experiments: SF=0.4 with N=100 px, σ=2, θ=16 px, ω=6 px, α=15 px and SF=0.2 with N=100 px, σ=1, θ=8 px, ω=3 px, α=3.5 px. Comparing results of detections based on c=1 (Tab. 1) indicate that the performance significantly decreases with lower image resolutions; e.g. the F_1 score of AOI 40 decreases from 0.77 to 0.20 (SF=0.2). In contrast, the detection results significantly improve if the number of channels is increased. These results confirm our hypothesis that the learned spatiotemporal features are of great importance for detecting tiny objects such as vehicles under low resolution.

One of the main problems with low resolution images is the small distance between neighboring vehicles as displayed in Fig. 4. In this case the FoveaNet creates a heatmap with a large number of connected regions, which result in a large number of false negative detections. To deal with small distances between neighboring cars we reduced the variance σ of Eq. 1, which improved the detection results. Otherwise, this issue has not been addressed in this work, although enhancing step 2 of the object detection will most likely improve the results.

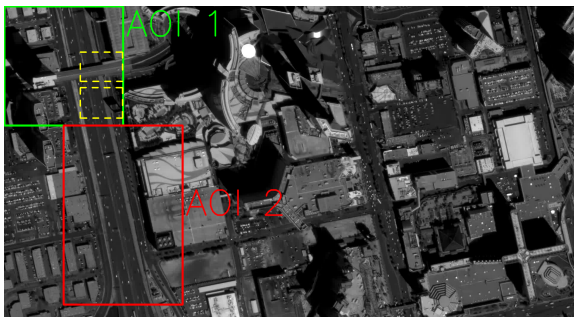


Figure 5. Example of the SkySat-1 LasVegas video in which both AOIs are shown. AOI 1 (400x400 px) is used for evaluation and AOI 2 (600x400 px) for training. Two ROOBIs are sketched as yellow dashed rectangles.

4.3. Experiment 3: Satellite video

The third experiment is conducted to evaluate the detection performance of the FoveaNet on the panchromatic satellite SkySat-1 LasVegas video[5] consisting of 700 frames, whose GSD is $\sim$1.0 m and its frame rate is 30 fps. We defined two AOIs as illustrated in Fig. 5. While AOI 2 is mainly composed of straight parallel roads, AOI 1 contains additionally a bridge which results in more complex traffic patterns. The ground truth which was kindly shared by [33] consists of bounding boxes for moving vehicles. We used the center points of those bounding boxes as ground truth analogous to the WPAFB ground truth.

For training and evaluation we set θ=8 px, α=4 px, σ=1, c=5 and N=128 px. Additionally, we set SF=0.2 and ω=3 px for training the WPAFB dataset. We observed in this experiment higher efficiency in training by replacing the ELUs with Leaky ReLUs.

Tab. 2 shows the results of nine individual experiments using FoveaNet with different filter sizes in the respective convolutional layers (Tab. 3). FoveaNet is trained on the 80 % reduced WPAFB and directly applied to the LasVegas video. We observe high recall ($>$0.8) but average precision which proves applicability of transfer learning.

[5] https://www.youtube.com/watch?v=lKNAY5ELUZY

In contrast to LaLonde et al. [17], we do not observe large influence of the filter size to the final performance of the network. The argument that large filter sizes in the first layer are needed for spatial contextual information seems to be misleading, as context is introduced in higher layers of a deep network by the network's receptive field. We argue that the filter size depends on the pixel distance of vehicles in consecutive frames so that the spatiotemporal network can exploit temporal information which is empirically confirmed by our experiments.

We then choose slightly smaller filter sizes (13-11-9-7-5-3-3-1) for the convolutional layers in FoveaNet, as this configuration shows best final results. We fine-tuned the network on AOI 2 which improved F_1 score from 0.55 to 0.84. A qualitative result of this experiment is shown in Fig. 1. The heat map of the network reconstructs amazingly well the ground truth. It detects not only cars but also buses and trucks which the network never saw before. Three experiments with varying filter sizes show further that filter sizes have minor influence on the result. We clearly see that our proposed method outperforms most methods for vehicle detection in satellite video except E-LSD[33] which is comparable to our results.

We then performed an experiment where we directly trained all layers of FoveaNet on AOI 2. Surprisingly, the overall results are only slightly worse which indicates that the learning problem is not as complex as for the WPAFB dataset. We conclude from all observations that FoveaNet learns to detect moving spots by characterising the slope of linear movement in spacetime which is a much simpler learning problem as learning spatiotemporal changes of visual appearance. However, pre-training on WPAFB is important for the network to generalise as can be seen in Fig. 6. Without pre-training the network is in this example not able to detect more complex motion patterns such as the moving vehicle on the bridge. It is an open question if such patterns could be learned by sole data augmentation.

Finally, we performed an experiment where we studied the effect of the frame rate of videos. Beside our baseline of considering every 10th image frame of the satellite video, we experimented with every 5th, 15th and 30th (1 fps) video frame. The results indicate less influence of higher frame rates on performance. This again supports our hypothesis that very simple features such as typical slopes of vehicle trajectories in spacetime are learned by the network. This presumption needs however further experiments and insight.

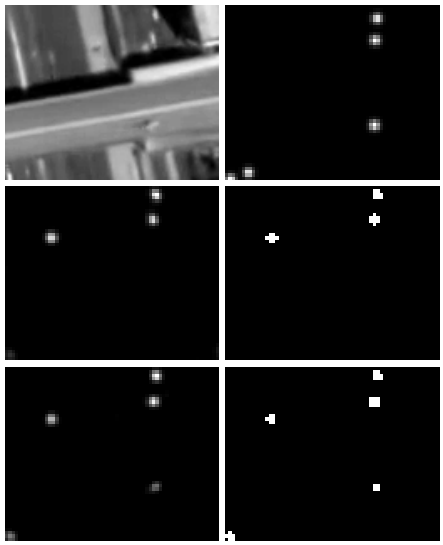


Figure 6. From left to right. Top: input image and ground truth. Middle: estimated and thresholded heatmap, FoveaNet trained with AOI 2. Bottom: estimated and thresholded heatmap, FoveaNet after fine-tuning.

5. Conclusion

This paper considers vehicle detection in satellite video. Vehicle detection in remote sensing is challenging as the objects usually appear tiny compared to the size of typical aerial and satellite images and discrimination of objects of interest from background is frequently ambiguous. Satellite video is a very new modality introduced 2013 by Skybox (now Planet) which might overcome the problem by introducing high temporal resolution. This allows to exploit temporal consistency of moving vehicles as inductive bias. Current state-of-the-art methods use either background subtraction, frame differencing or subspace learning in video, however, performance is currently limited (0.26 - 0.82 F_1 score).

The method in this paper is motivated by recent work in WAMI which exploits video in spatiotemporal convolutional networks[17]. We apply FoveaNet to the domain of satellite video by transfer learning the network with WPAFB and a small amount of available labelled video frames of the SkySat-1 LasVegas video which yields comparable results (0.84 F_1 score). Several ablation studies show minor influence of the filter sizes in the convolutional layers and minor influence of the frame rate (tempo-

WPAFB				LasVegas AOI 1				SOTA			
Conf.	**Prec.**	**Rec.**	F_1		**Prec.**	**Rec.**	F_1		**Prec.**	**Rec.**	F_1
1	0.56	0.67	0.61	scratch	0.87	0.80	0.83	ViBe[5]	0.58	0.17	0.26
2	0.46	0.76	0.57	**fine-tuning**				GMMv2[39]	0.65	0.27	0.38
3	0.40	0.79	0.53	1	0.84	0.82	0.83	GMM[14]	0.46	0.50	0.48
4	**0.42**	**0.81**	**0.55**	**4**	**0.86**	**0.82**	**0.84**	Fast-RCNN-LRP[34]	0.58	0.44	0.50
5	0.43	0.85	0.58	9	0.76	0.85	0.80	GoDec[36]	0.95	0.36	0.52
6	0.47	0.80	0.60	skip 5	0.84	0.83	0.84	RPCA-PCP[7]	0.94	0.41	0.57
7	0.46	0.82	0.59	**skip 10**	**0.86**	**0.82**	**0.84**	Decolor[37]	0.77	0.59	0.67
8	0.46	0.83	0.59	skip 15	0.85	0.81	0.83	LSD[19]	0.87	0.71	0.78
9	0.45	0.70	0.55	skip 30	0.83	0.82	0.83	**E-LSD[33]**	**0.85**	**0.79**	**0.82**

Table 2. Left: Evaluation results of nine different filter size configurations (see Tab. 3) of the FoveaNet. Middle: Results of the FoveaNet trained from scratch, fine-tuned with different filter sizes and different fps (conf. 4). Right: Evaluation results of state-of-the-art (SOTA) methods are presented.

conf.	**filter size**	**conf.**	**filter size**
1	19-17-15-13-11-9-7-1	6	9-7-5-3-3-3-3-1
2	17-15-13-11-9-7-5-1	7	7-5-3-3-3-3-3-1
3	15-13-11-9-7-5-3-1	8	5-3-3-3-3-3-3-1
4	13-11-9-7-5-3-3-1	9	3-3-3-3-3-3-3-1
5	11-9-7-5-3-3-3-1		

Table 3. Filter size configurations of the various experiments. Conf. 3 corresponds to the filter sizes suggested by LaLonde et al. [17].

ral resolution) on the overall result. This indicates a much simpler learning problem than for the original high-resolution WAMI data, however, we show that temporal information is essential for a good detection performance. Improvements of FoveaNet, e.g. including the final segmentation of the heat map into the network, are left for future work.

Acknowledgment

This research was supported by the Austrian Research Promotion Agency (FFG) under grant MiTrAs-867030 and by the European Union's H2020 programme under grant FOLDOUT-787021. The imagery and video used in this work is in courtesy of the U.S. air force research laboratory sensors directorate layered sensing exploitation division and Planet Inc. under creative common CC-BY-NC-SA[6].

References

[1] S. A. Ahmadi, A. Ghorbanian, and A. Mohammadzadeh. Moving vehicle detection, tracking and traffic parameter estimation from a satellite video: a perspective on a smarter city. *IJRS*, 40(22):8379–8394, 2019. 3

[2] N. Al-Shakarji, F. Bunyak, H. Aliakbarpour, G. Seetharaman, and K. Palaniappan. Multi-cue vehicle detection for semantic video compression in georegistered aerial videos. In *CVPR*, June 2019. 2, 3

[3] W. Ao, Y. Fu, X. Hou, and F. Xu. Needles in a haystack: Tracking city-scale moving vehicles from continuously moving satellite. *IEEE TIP*, preprint:1–1, 2019. 2, 3

[4] W. Ao, Y. Fu, and F. Xu. Detecting tiny moving vehicles in satellite videos. *CoRR*, abs/1807.01864, 2018. 2, 3

[5] O. Barnich and M. Van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *IEEE TIP*, 20(6):1709–1724, June 2011. 8

[6] R. Benenson, M. Mathias, T. Tuytelaars, and L. Van Gool. Seeking the strongest rigid detector. In *CVPR*, June 2013. 1, 2

[7] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, June 2011. 8

[8] P. Ding, Y. Zhang, W.-J. Deng, P. Jia, and A. Kuijper. A light and faster regional convolutional neural network for object detection in optical remote sensing images. *ISPRS J. P&RS*, 141:208 – 218, 2018. 2, 3

[9] B. Du, S. Cai, C. Wu, L. Zhang, and D. Tao. Object tracking in satellite videos based on a multi-frame optical flow tracker. *CoRR*, abs/1804.09323, 2018. 3

[10] B. Du, Y. Sun, S. Cai, C. Wu, and Q. Du. Object tracking in satellite videos by fusing the kernel correlation filter and the three-frame-difference algorithm. *IEEE Geoscience and Remote Sensing Letters*, 15(2):168–172, Feb 2018. 3

[11] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 4

[6] https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode

[12] Y. Guo, D. Yang, and Z. Chen. Object tracking on satellite videos: A correlation filter-based tracking method with trajectory correction by kalman filter. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(9):3538–3551, Sep. 2019. 3

[13] J. Imbert. Fine-tuning of fully convolutional networks for vehicle detection in satellite images: Data augmentation and hard examples mining. Master's thesis, KTH, 2019. 2

[14] P. KaewTraKulPong and R. Bowden. *Video-Based Surveillance Systems*, chapter An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection, pages 135–144. Springer, 2002. 8

[15] Y. Koga, H. Miyazaki, and R. Shibasaki. A cnn-based method of vehicle detection from aerial images using hard example mining. *Remote Sensing*, 10, Jan. 2018. 2, 3

[16] G. Kopsiaftis and K. Karantzalos. Vehicle detection and traffic density monitoring from very high resolution satellite video data. In *IGARSS*, July 2015. 2, 3

[17] R. LaLonde, D. Zhang, and M. Shah. Clusternet: Detecting small objects in large scenes by exploiting spatio-temporal information. In *CVPR*, June 2018. 2, 3, 4, 5, 6, 7, 8

[18] H. Li, L. Chen, F. Li, and M. Huang. Ship detection and tracking method for satellite video based on multiscale saliency and surrounding contrast analysis. *Journal of Applied Remote Sensing*, 13(2):1 – 17, 2019. 2, 3

[19] X. Liu, G. Zhao, J. Yao, and C. Qi. Background subtraction based on low-rank and structured sparse decomposition. *IEEE TIP*, 24(8):2502–2514, Aug 2015. 8

[20] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS J. P&RS*, 152:166 – 177, 2019. 2

[21] L. Meng and J. P. Kerekes. Object tracking using high resolution satellite imagery. *IEEE J. of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(1):146–152, 2 2012. 2

[22] L. Mou and X. X. Zhu. Spatiotemporal scene interpretation of space videos via deep neural network and tracklet analysis. In *IGARSS*, July 2016. 2, 3

[23] T. Mundhenk, G. Konjevod, W. Sakla, and K. Boakye. A large contextual dataset for classification, detection and counting of cars with deep learning. In *ECCV*, Oct. 2016. 2, 3

[24] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 2

[25] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 2, 3

[26] J. Shao, B. Du, C. Wu, J. Wu, R. Hu, and X. Li. Vcf: Velocity correlation filter, towards space-borne satellite video tracking. In *ICME*, July 2018. 3

[27] G. Wei, Y. Wen, Z. Haijian, and H. Guang. Geospatial object detection in high resolution satellite images based on multi-scale convolutional neural network. *Remote Sensing*, 10(1), 2018. 2, 3

[28] A. Xu, J. Wu, G. Zhang, S. Pan, T. Wang, Y. Jang, and X. Shen. Motion detection in satellite video. *Journal of Remote Sensing and GIS*, 6(2):1–9, 2017. 2, 3

[29] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling. Clustered object detection in aerial images. *CoRR*, abs/1904.08008, 2019. 2, 3

[30] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling. Clustered object detection in aerial images. In *ICCV*, 2019. 2, 3

[31] M. Y. Yang, W. Liao, X. Li, and B. Rosenhahn. Vehicle detection in aerial images. *CoRR*, abs/1801.07339, 2018. 2, 3

[32] T. Yang, X. Wang, B. Yao, J. Li, Y. Zhang, Z. He, and W. Duan. Small moving vehicle detection in a satellite video of an urban area. *Sensors*, 16(9):1528, Sept. 2016. 2, 3

[33] J. Zhang, X. Jia, and J. Hu. Error bounded foreground and background modeling for moving object detection in satellite videos. *CoRR*, 2019. 1, 2, 3, 6, 7, 8

[34] J. Zhang, X. Jia, and J. Hu. Local region proposing for frame-based vehicle detection in satellite videos. *Remote Sensing*, 11(20):2372, Oct. 2019. 1, 2, 3, 8

[35] X. Zhang and J. Xiang. Moving object detection in video satellite image based on deep learning. In *LIDAR Imaging Detection and Target Recognition*, volume 10605, pages 1149 – 1156. SPIE, 2017. 2, 3

[36] T. Zhou and D. Tao. Godec: Randomized low-rank & sparse matrix decomposition in noisy case. In *ICML*, 2011. 8

[37] X. Zhou, C. Yang, and W. Yu. Moving object detection by detecting contiguous outliers in the low-rank representation. *IEEE TPAMI*, 35(3):597–610, March 2013. 8

[38] X. X. Zhu, D. Tuia, L. Mou, G. Xia, L. Zhang, F. Xu, and F. Fraundorfer. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36, Dec 2017. 2

[39] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *CVPR*, Aug. 2004. 8

25th Computer Vision Winter Workshop
Domen Tabernik, Alan Lukežič, Klemen Grm (eds.)
Rogaška Slatina, Slovenia, February 3–5, 2020

CNN-CASS: CNN for Classification of Coronary Artery Stenosis Score in MPR Images

Mariia Dobko, Bohdan Petryshak, Oles Dobosevych
The Machine Learning Lab, Ukrainian Catholic University, Lviv, Ukraine
{dobko_m, petryshak, dobosevych}@ucu.edu.ua

Abstract. *To decrease patient waiting time for diagnosis of the Coronary Artery Disease, automatic methods are applied to identify its severity using Coronary Computed Tomography Angiography scans or extracted Multiplanar Reconstruction (MPR) images, giving doctors a second-opinion on the priority of each case. The main disadvantage of previous studies is the lack of large set of data that could guarantee their reliability. Another limitation is the usage of handcrafted features requiring manual preprocessing, such as centerline extraction. We overcome both limitations by applying a different automated approach based on ShuffleNet V2 network architecture and testing it on the proposed collected dataset of MPR images, which is bigger than any other used in this field before. We also omit centerline extraction step and train and test our model using whole curved MPR images of 708 and 105 patients, respectively. The model predicts one of three classes: 'no stenosis' for normal, 'non-significant' — 1-50% of stenosis detected, 'significant' — more than 50% of stenosis. We demonstrate model's interpretability through visualization of the most important features selected by the network. For stenosis score classification, the method shows improved performance comparing to previous works, achieving 80% accuracy on the patient level. Our code*[1] *is publicly available.*

1. Introduction

According to the American Heart Association report [2], approximately 17.6 million deaths were attributed to Cardiovascular Diseases (CVD) globally in 2016, making it the leading cause of death in the world. By 2030 it is estimated that CVD will be responsible for over 23.6 million deaths [2], and thus the ability to get early diagnosis becomes crucial. Cardiovascular diseases affect the heart or blood vessels. They include Coronary Heart Disease, or Coronary Artery Disease (CAD), which occurs when plaque (a combination of cholesterol, calcium, fat and other substances) builds up in the arteries and clogs them. This narrowing of the arteries, called stenosis, interferes in the healhy blood flow by hindering oxygen-rich blood cells transportation to the heart.

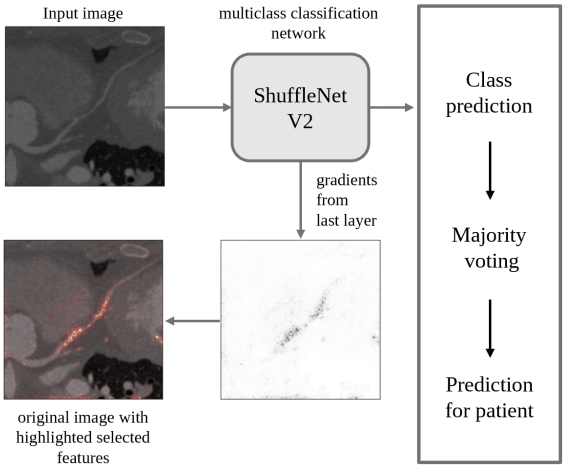


Figure 1: The pipeline of stenosis classification on an MPR image. For each 2D image, ShuffleNet V2 predicts probabilities of stenosis score. Activation regions of the last layer in the model are shown and overimposed on the input image.

Coronary Computed Tomography Angiography (CCTA) is one of the components used by radiologists in diagnostics of the coronary artery disease. In the recent studies, CCTA was proven to be clinically effective in combination with functional test-

[1] https://github.com/ucuapps/CoronaryArteryStenosisScoreClassification/

ing (SCOT-HEART [1]), or even as an alternative to it (PROMISE [7]). Coronary CTA helps doctors in evaluation of the degree of artery stenosis. For patients in the risk group, it is vital to get the diagnosis in a short period of time, however, nowadays it takes up to two weeks to receive analysis results after the scanning procedure. In order to decrease the waiting time, the previous works [12], [22] applied various semi-automated algorithms to identify the severity of disease in medical images, giving doctors a second-opinion on the priority of each case. Such automated computer-aided systems are capable to increase the access to diagnostics and eventually reduce mortality by faster recognition of critical cases.

Due to recent advances in applications of machine learning to medical domain, it is now possible to use neural networks for assessment of the severity of coronary artery stenosis. Among the main disadvantages of previous approaches are the lack of large set of data that could guarantee their reliability and usage of handcrafted features during the preprocessing steps. Our contribution to this field lies in application of a different approach and testing it on the created dataset, which is bigger than any other used in this field before. We also propose a fully automated method to classify stenosis score, that utilizes whole curved Multiplanar Reconstruction (MPR) images without manual preprocessing or centerline extraction, see Figure 1.

2. Related Works

The problem of stenosis score classification on CCTA images of coronary arteries is insufficiently studied. The major difficulty is the absence of publicly available structured and professionally labeled sets of data. Another one is domain specificity, which requires certain expertise in analysing medical data.

Datasets of coronary arteries are usually formed by CCTA scans, from which MPR images can be extracted and transformed to either straight or curved representation of arteries. In the related paper [3], CCTA scans of only 163 patients were collected, and the proposed network was trained and tested using images of 98 and 65 patients, respectively. The authors first straightened MPR images by applying the centerline extraction technique [20] and then used the transformed data to simplify classification of stenosis level. The centerline extraction step used in this approach requires manual placement of a single seed point in the artery of interest, so that the method is not fully automated. 3D convolutional neural network was utilized to extract features which are used by recurrent neural network for classification. While the achieved accuracy shows good performance and feasibility of deep learning methods for stenosis score classification, the reliability of obtained results can not be justified on such small data sample. Another drawback is its poor stability: as the authors admit, even small errors in centerline extraction may essentially increase the overall error.

Centerline extraction is a common preprocessing method [15] although it often requires manual assistance. It is used in the previous study [22], where user interaction is needed to localize the artery by annotating the start and end points of the vessel. As the authors described, the start point was placed in the coronary ostium of the corresponding arterial tree and the end point was placed at the most distal point inside the vessel. Some parameters were chosen manually, for example, contrast filled (foreground) regions are defined by empirically determining a lower and upper bound values of intensity. Other handcrafted features include the mean and dispersion values of the artery radius, as well as the mean and dispersion of a rotation angle corresponding to a typical location of the artery.

In order to avoid errors caused by centerline extraction, in our approach we use curved MPR images instead. These MPRs are generated from CCTA with the help of radiologist assistant during a general pipeline of the coronary artery diagnosis. Thus, our method does not require any handcrafted features, but utilizes the whole MPR image, where artery is curved.

3. Data

For training our algorithms, we used curved MPR images of the coronary artery with stenosis levels annotated by professional radiologists from a well-renowned Future Medical Imaging Group (FMIG) in Australia. Our current dataset consists of 160,000 MPR images which were extracted from CCTA scans of 828 unique patients; see data statistics in Table 1.

3.1. MPR generation process

The CCTA stack of images representing the patient's heart, is produced by the CT scanner. The raw CT scans require thorough and long analysis as the coronary artery is represented in each slice as a small circular region, similar to a dot. To increase the in-

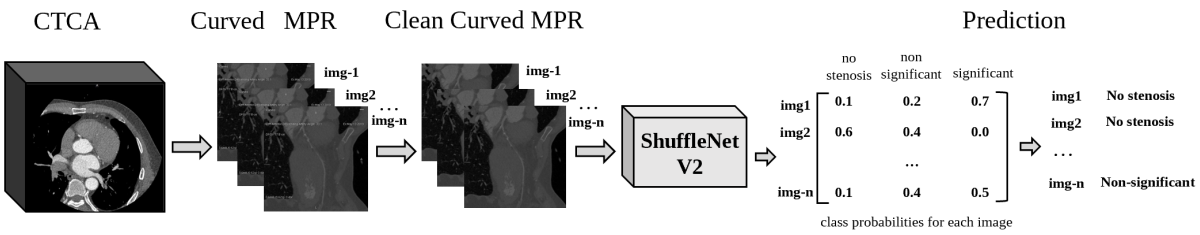


Figure 2: **Prediction process.** First, the whole CCTA is converted into sets of MPR images for each artery branch (completed by a radiologist assistant). Then, our method automatically cleans the image from the text and meta information, and feeds the obtained preprocessed images to the ShuffleNet V2. As an output, the probability predictions of each class are produced.

Arteries	Arteries	Sections	Sections
LAD	824	**LAD**	822
		D-1	729
		D-2	356
		D-3	68
LCX	722	**LCX**	639
		PLV-LCX	15
		PDA-LCX	17
RCA	721	**RCA**	91
		OM	6
		OM-1	81
		OM-2	281
		OM-3	75
		PLV-RCA	609
		PDA-RCA	71

Table 1: **Collected dataset statistics.** Number of cases containing certain arteries and branches for all 828 patients.

terpretability of the data, the clinicians use the MPR technique [10, 5]. MPR is the process of using the data from axial CT images to create a more anatomical representation of the coronary artery by tracking the whole specific artery branch along the CT volume and generating its two-dimensional image.

Each branch is represented by 50 MPR images, where one image corresponds to the specific view angle (180 degrees in total). The reason for that is that plaque might be located anywhere along the vessel, and be invisible only from some view angles.

3.2. Labels extraction from medical reports

The condition of coronary artery of each patient is described by the report. It contains the meta information about the person (age, gender, heart rate, etc.), characterization of stenosis score, type of the plaque and calcium score to all artery sections and branches. The raw reports are not suitable for training classification machine learning algorithms as they do not have any specific category attached to the particular image or at least to a stack of images.

We created the parsing pipeline, which takes the report of the patient as an input and extracts all information relevant to our task. The parsed data include the description of all important artery sections and branches with corresponding stenosis categories. The latter are grouped according to the standard defined by the Society of Cardiovascular Computed Tomography (SCCT) and Coronary Artery Disease - Reporting and Data System (CAD-RADS) [4]: 0% - Normal, 1-24% - Minimal stenosis or plaque with no stenosis, 25-49% Mild stenosis, 50-69% - Moderate stenosis, 70-99% - Severe stenosis, 100% - Total Occlusion. In the reports, three main artery sections are presented: LAD (Left Anterior Descending Artery) with D-1, D-2, D-3 branches; RCA (Right Coronary Artery) with PDA-RCA, PLV-RCA branches; LCx (Left Circumflex Artery) with OM-1, OM-2, OM-3, LCx-PDA, LCx-PLV branches.

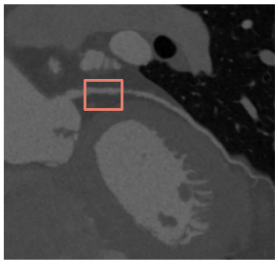
3.3. Data labeling process

Specific recommendation for further patient treatment depends largely on the identified level of stenosis [4]. No further cardiac investigation is required unless moderate (50-69%) or higher stage was reported. Preventive therapy and risk factor modification is suggested for minimal or mild stenosis. Due to these specific regulations, we assign one of the three classes for each MPR image: 'no stenosis' for normal cases, 'non-significant' — 1-50% of stenosis detected, 'significant' — critical cases where more than 50% of stenosis is present and instant doctor's atten-

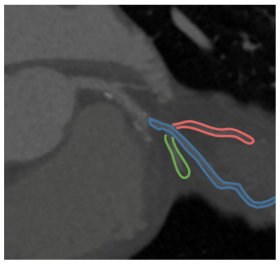
tion is required.

3.4. Challenges in the dataset

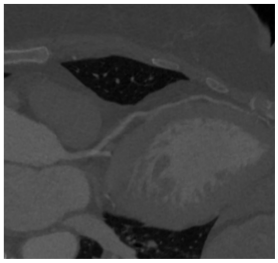
After all described preprocessing steps, we obtained a structured labeled dataset, but it is still incomplete and contains noise. One of the main issues is the appearance of several branches on the same image, see example in Figure 3 (b). This increases the risk of mislabeled data as each set of images representing one branch has just one label, which is then assigned to every single image in the set. For example, for the healthy LAD branch all the 50 images retrieved from different viewpoint angles are labeled as being normal. However, if on some of those images a neighbouring vessel (e.g., LCx) with over 50% score of stenosis is partly present, then the model may detect stenosis and misclassify a healthy LAD.



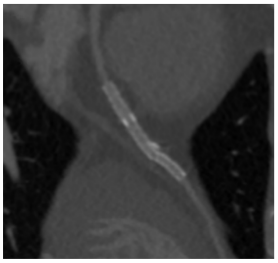
(a) Small portion of useful signal



(b) Several branches on one MPR



(c) Myocardial bridging



(d) Stent

Figure 3: **Examples of difficult images.** Examples of the hard cases which are present in collected dataset. **(a)** The amount of pixels responsible for region of stenosis is many times smaller than the entire sample. **(b)** While the label for one MPR image corresponds to only one artery segment, several of them might be present on the image. **(c)** Physically natural narrowing can be visually similar to stenosis called myocardial bridging. **(d)** Example of artery with inserted stent, which can be mistakenly classified as stenosis with plaques.

Another issue is an inconsistency between medical reports written by doctors and the labeling system prescribed by CAD-RADS. The problem arises when the annotation provided by a radiologist is on the borderline between two classes from the CAD-RADS system. For example, the doctor might mark a specific branch by a "50%" of the stenosis. While it is satisfactory in medical terms, it becomes a challenge for us to choose which group this annotation belongs to - whether it should be considered as a significant or non-significant case.

Also we pre-process MPR images before inference. Each MPR image contains text information (meta-information about the picture), which has the highest intensity on the image. We remove it by assigning the average intensity of the neighbor pixels. Other examples of data challenges and difficulties are represented in Figure 3.

4. Experiments

4.1. General pipeline

We took 708 unique patients for training, 15 for validation and 105 for testing. We include the MPR images of every coronary artery and its branches described in Subsection 3.2 in training and testing phases.

We fed the MPR images into an optimized network architecture ShuffleNet V2. Each branch of the artery in most of the cases is represented by 50 MPR images (see Subsection 3.1). Thus for one branch, we get 50 predictions describing its stenosis score. Then using the majority rule, we assign the final stenosis score for the branch. The prediction pipeline is illustrated in Figure 2, and it is followed by evaluation technique, shown in Figure 4.

4.2. Methodology

The technique which is widely used for optimizing the neural network architectures is 1x1 convolution. The authors of ShuffleNet [21] approached it and managed to reduce the time for this operation. The main idea behind ShuffleNet was to use separable depthwise convolutions [3], grouped convolutions on 1x1 convolution layers - pointwise group, followed by channel shuffle operation. In this paper, we use an improved version of this architecture - ShuffleNet V2 [14], which is not only faster, but also more accurate.

We use ShuffleNet V2, pretrained on ImageNet [6], to extract features from curved MPR images and classify them into three classes: ‘no stenosis’, ‘non-significant stenosis’, ‘significant stenosis’.

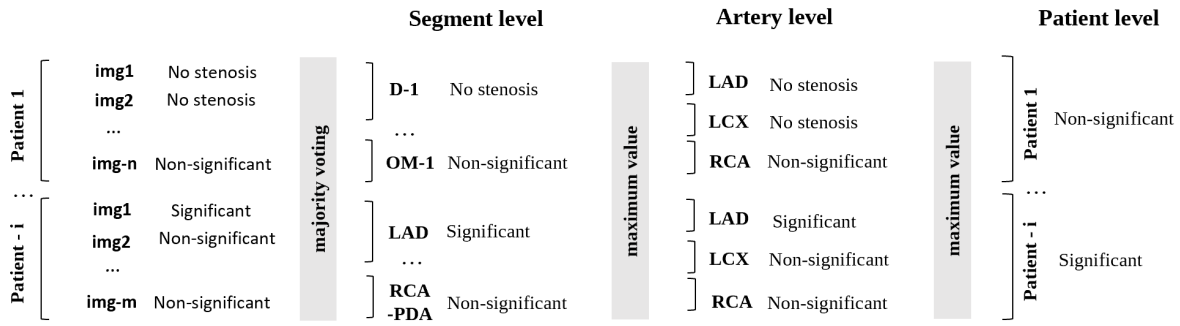


Figure 4: **Evaluation process.** The performance of proposed method is evaluated as suggested in [19] on segment-, artery- and patient-levels using F_1 score and accuracy. For each image, we predict one of the three classes, and then assign the class for the segment (branch) by applying the majority rule to all 50 images corresponding to it. The maximal (most critical) class of all of the branches from one artery is assigned to this artery. The final prediction is calculated for each patient by choosing the maximal class out of all patient's arteries.

The structure of the basic building block of ShuffleNet V2 [14] with residual is displayed in Figure 5. There are several building blocks which are stacked to construct the network. The input of feature channels is split into two branches at the beginning. In each block one branch directly goes through the block and joins the next one. The other branch has three convolution layers with the same input and output channels. Only one from the three 1x1 convolutions is group-wise. The two branches are concatenated after convolutions. For spatial downsampling the block is modified by the removal of split operator. The Channel Shuffle improves accuracy by enabling information communication between different groups of channels.

We trained the model on our dataset using Adam optimizer [11] with 10^{-4} learning rate. We chose the best value of learning rate for our model using LR Range Test [17]. It is a method that implies running the model for a few iterations with initially very small learning rate and then increasing it linearly between low and high learning rate values after each epoch. This allows to estimate the minimum and maximum boundary learning rates. The gradient accumulation and batch normalization [9, 8] were used to increase the batch size and provide a better direction towards a minimum of the loss function.

To introduce robustness properties and desired invariance in our model, we employed standard data augmentation techniques. In the case of MPR images, we primarily need a scale, rotate, blur, brightness, and transpose invariance.

4.3. Evaluation metrics

The CAD-RADS classification is applied on a per-patient basis and represents the highest grade of stenosis from the coronary tree [19]. Taking this into account, we evaluate our method perfor-

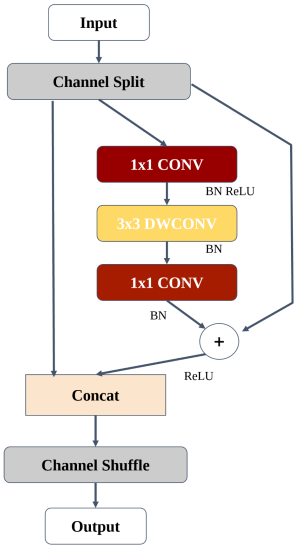


Figure 5: Structure of the basic building block of ShuffleNet V2 with residual [14]. CONV: convolution layer. DWCONV: depthwise convolution. BN: batch normalization. Channel Shuffle: crucial operation for ShuffleNet architectures.

mance on segment-, artery- and patient-levels, and define stenosis score according to the maximum value found at the current level. On the artery level, the highest grade of stenosis is selected out of all the grades on segments. On the patient level, the maximal stenosis stage is chosen among all the arteries. The accuracy and F_1 score for multiclass classification are computed on each level.

We apply weighted averaging for F_1 score measure because we deal with multiclass labels. For each label metrics are calculated and then average weighted by the number of true instances for each class:

$$\sum_{i=1}^{3} F_1(class_i) * W_i$$

where i - iterator over the 3 classes, F_1 is $F_1 - score$ and W - weight for the current class .

Results	Accuracy		F_1 score	
	Our	RNN-based[22]	Our	RNN-based[22]
Segment-level	**0.81**	0.80	**0.81**	0.75
Artery-level	**0.81**	0.76	**0.82**	0.77
Patient-level	**0.80**	0.75	**0.80**	0.75

Table 2: Accuracy and F_1 score on test sets: 105 patients (approximately 25,000 MPR images in total) in our test, 65 patients in Zreik et al.[22]

The final results and comparison to previous study are reported in Table 2. We also display confusion matrices in Table 3 for every level separately in order to show results across the classes. This allows to observe the model's sensitivity (True Positive rate) - cases where the model correctly predicts the positive class, and specificity (True Negative rate) - cases where the model correctly predicts the negative class. For medical problems the False Positive error is always less dangerous than False Negative, these metrics are shown on confusion matrices. None of the patients with significant stenosis were classified as having no stenosis, and none of the healthy patients were put in the significant stenosis category. The model makes mistakes between 'no stenosis' and 'non-significant' classes, as well as between 'non-significant' and 'significant', which could be caused by the noise in data (see Figure 3) and weak labels.

4.4. Results interpretability

In order to achieve model interpretability we use Captum [13] library containing implemented meth-

		Predicted		
	Segment level	No stenosis	Non-Significant	Significant
Actual	No stenosis	0.92	0.07	0.01
	Non-Significant	0.32	0.6	0.08
	Significant	0.18	0.26	0.55

		Predicted		
	Artery level	No stenosis	Non-Significant	Significant
Actual	No stenosis	0.91	0.08	0.01
	Non-Significant	0.24	0.67	0.09
	Significant	0.13	0.26	0.61

		Predicted		
	Patient level	No stenosis	Non-Significant	Significant
Actual	No stenosis	0.81	0.19	0.00
	Non-Significant	0.07	0.80	0.13
	Significant	0.00	0.21	0.79

Table 3: **Confusion matrices.** For each level: segment, artery and patient we calculate confusion matrix to see the number of False Positives, False Negatives and compare them.

ods that identify which training features are important for the model. We visualized the features from the last layer of our model to understand which image regions have the largest impact on the model. The attribution of the network prediction to its input features was performed by applying axiomatic attribution method – Integrated Gradients [18] implemented in Captum [13]. In this approach, the integral of the gradients of the output prediction for the specified class is computed with respect to the input image pixels. We observe that for all types of arteries and all levels of stenosis the model pays attention mostly to the artery zone, while the background with noise does not play a role in classification. Some examples of the features visualization using heatmaps are shown in Figure 8. It is noticeable that for images which contain stenosis, model is more confident in the regions where plaques are located. This demonstrates model reliability, since plaques presence directly correlates with stenosis.

Although our model is capable of handling the background noise information and mainly takes into account only relevant areas, there are some corner cases. One of the most common types of plaque is calcified plaque. In computed tomography scans, it is represented as the pixels with a high level of intensity. There are structures and tissues like calcified ribs, sternum, costal cartilages, ventricle walls, etc., which are in the same range of radiodensity. The visualization showed that our models pay attention to these regions and associate them with the stenosis presence, which causes the lower specificity of our

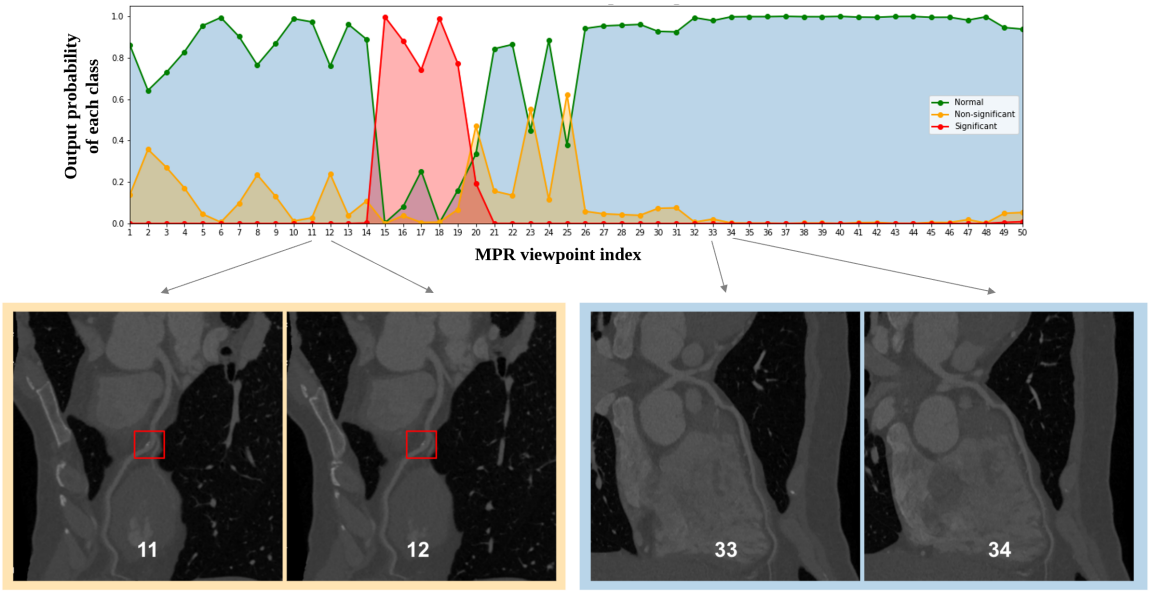


Figure 6: **Prediction for one segment.** The particular example has 25% of stenosis in LAD artery. The labels for all 50 images representing this segment are the same - 'non-significant'. While it is true for some of the MPR view points (see two images at the bottom left), from most of the angles stenosis is not seen either by a human eye, nor by a model (see two images at the bottom right). Thus, it is a direct illustration of weak labeling. It is important to treat every segment as a set of 50 images which are related.

algorithms (see Figure 7).

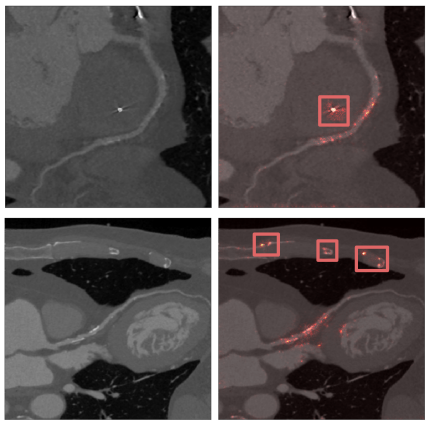


Figure 7: **Visualization of the model's confusion.** Scans with the structures, which are similar to the calcified plaque. **(a)** The anomaly, with the high level intensity pixels. **(b)** Calcified ribs, which caused the model confusion.

5. Conclusion

We propose a simple automated framework, which is capable of detecting the stenosis score in curved MPR images. Our method shows improvements over the previous results [22] reporting 80% accuracy for the multiclass classification of stenosis level.

Our main contribution lies in creating new dataset of Cardiac CT scans of more than 800 patients, which is larger than any previous dataset, and suggesting a new approach for stenosis level classification. The proposed method omits centerline extraction and does not require any handcrafted features. Furthermore, we obtain explainable results and display features which impacted network's decisions.

The model interpretability through visualization of feature importance is very helpful in medical imaging as radiology specialists may use it to build trust in the model's predictions, refine classification of borderline cases, as well as gather observations for future testing.

6. Discussion

There are several ways to improve our approach. The curved MPR images, which were used as input in this work, contain not only artery, but also the background, where there is no useful information for determining the level of stenosis. We believe that the results of the proposed approach can be improved by using segmentation as an additional step during pre-

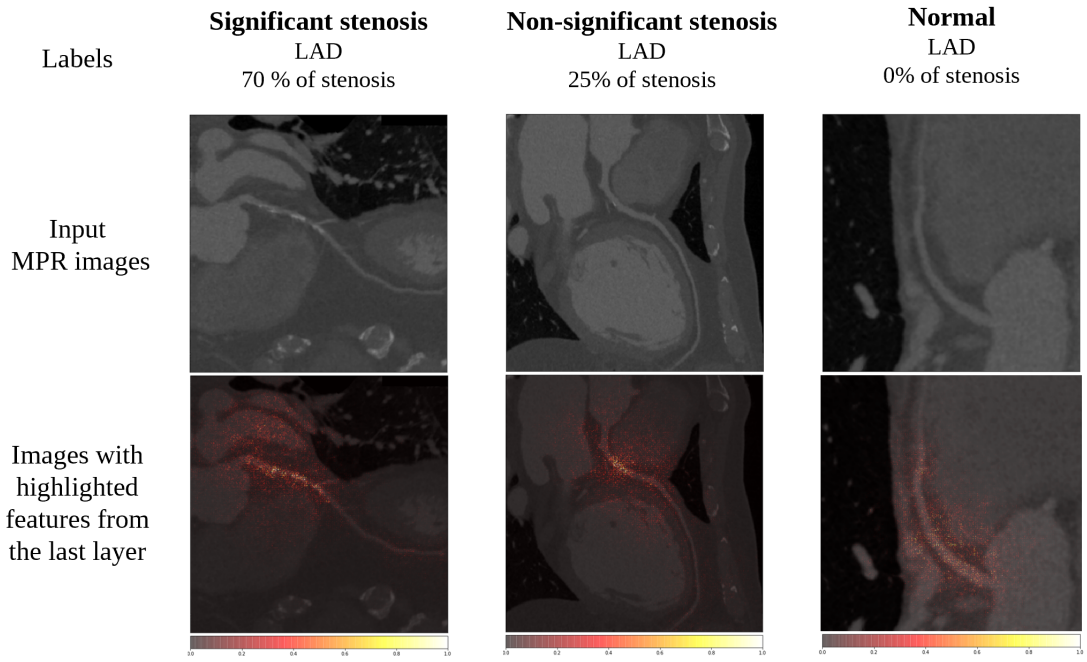


Figure 8: **Visualizations of the last layer's most important features for correctly predicted cases**. These were created by using Integrated Gradients [18]. Each class is represented for three examples of Left Anterior Descending artery: significant, non-significant, no stenosis. **Top**: The input two-dimensional MPR images of three different patients generated from CCTA scans. **Bottom**: The gradient visualization of most impactful features for the model. The brighter the pixels are, the more importance they have in prediction.

processing and feeding the network with segmented images, where only the artery region is present.

Taking into account the problems shown on Figure 7, we might improve the performance of our models by adding the attention gate [16] to the current network architecture. It will automatically learn the relevant areas for our task and suppress the unrelated target structures.

To obtain the final stenosis score for one branch, we take 50 predictions of our network for each corresponding MPR image representing the artery and assign the prevailing class. With this approach, we do not take into account the spatial relationship between MPR images. One possible improvement might be to apply the 3D CNNs to catch patterns across three spatial dimensions. One of the options to skip the step of MPR extraction is to create a new method, which would directly use 3D images of CCTA for stenosis score classification.

Due to the difficulty in collecting reliable labels for medical data, unsupervised or weakly-supervised approaches should be considered. We believe that one of the possible ways of implementing such a solution is to train autoencoder exclusively on normal images with noise, then, to decide whether the particular MPR image represents the normal case based on its distance to the corresponding image generated by the model.

The other area for research is the extraction of MPR images from CCTA. This task is handled semi-manually by a radiologist at the clinics, therefore, it is costly and takes long time. Our dataset already contains extracted MPR data. We think this process can be simplified by application of deep learning in building multiplanar reconstruction images based only on data from CCTA scans.

Acknowledgements

This research was supported by Faculty of Applied Sciences at Ukrainian Catholic University. The authors thank Future Medical Imaging Group for providing the access to their data, Andrew Dobrotwir for constructive discussions and sharing medical expertise, Rostyslav Hryniv for helpful insights, and Jan Kybic for valuable feedback.

References

[1] CT coronary angiography in patients with suspected angina due to coronary heart disease (SCOT-HEART): an open-label, parallel-group, multicentre trial. *The Lancet*, 385(9985):2383–2391, June 2015. 2

[2] E. J. Benjamin, P. Muntner, A. Alonso, M. S. Bittencourt, C. W. Callaway, A. P. Carson, A. M. Chamberlain, A. R. Chang, S. Cheng, S. R. Das, F. N. Delling, L. Djousse, M. S. Elkind, J. F. Ferguson, M. Fornage, L. C. Jordan, S. S. Khan, B. M. Kissela, K. L. Knutson, T. W. Kwan, D. T. Lackland, T. T. Lewis, J. H. Lichtman, C. T. Longenecker, M. S. Loop, P. L. Lutsey, S. S. Martin, K. Matsushita, A. E. Moran, M. E. Mussolino, M. O'Flaherty, A. Pandey, A. M. Perak, W. D. Rosamond, G. A. Roth, U. K. Sampson, G. M. Satou, E. B. Schroeder, S. H. Shah, N. L. Spartano, A. Stokes, D. L. Tirschwell, C. W. Tsao, M. P. Turakhia, L. B. VanWagner, J. T. Wilkins, S. S. Wong, and S. S. V. and. Heart disease and stroke statistics — a report from the american heart association. *Circulation*, 139(10), Mar. 2019. 1

[3] F. Chollet. Xception: Deep learning with depthwise separable convolutions, 2016. 4

[4] R. C. Cury, S. Abbara, S. Achenbach, A. Agatston, D. S. Berman, M. J. Budoff, K. E. Dill, J. E. Jacobs, C. D. Maroules, G. D. Rubin, F. J. Rybicki, U. J. Schoepf, L. J. Shaw, A. E. Stillman, C. S. White, P. K. Woodard, and J. A. Leipsic. CAD-RADSTM coronary artery disease – reporting and data system. an expert consensus document of the society of cardiovascular computed tomography (SCCT), the american college of radiology (ACR) and the north american society for cardiovascular imaging (NASCI). endorsed by the american college of cardiology. *Journal of Cardiovascular Computed Tomography*, 10(4):269–281, July 2016. 3

[5] N. C. Dalrymple, S. R. Prasad, M. W. Freckleton, and K. N. Chintapalli. Introduction to the language of three-dimensional imaging with multidetector CT. *RadioGraphics*, 25(5):1409–1428, Sept. 2005. 3

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 4

[7] P. S. Douglas, U. Hoffmann, M. R. Patel, D. B. Mark, H. R. Al-Khalidi, B. Cavanaugh, J. Cole, R. J. Dolor, C. B. Fordyce, M. Huang, M. A. Khan, A. S. Kosinski, M. W. Krucoff, V. Malhotra, M. H. Picard, J. E. Udelson, E. J. Velazquez, E. Yow, L. S. Cooper, and K. L. Lee. Outcomes of anatomical versus functional testing for coronary artery disease. *New England Journal of Medicine*, 372(14):1291–1300, Apr. 2015. 2

[8] J. Hermans, G. Spanakis, and R. Möckel. Accumulated gradient normalization, 2017. 5

[9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. 5

[10] A. Kanitsar, D. Fleischmann, R. Wegenkittl, P. Felkel, and E. Groller. CPR - curved planar reformation. In *IEEE Visualization, 2002. VIS 2002.* IEEE. 3

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014. 5

[12] Kirişli et al. Standardized evaluation framework for evaluating coronary artery stenosis detection, stenosis quantification and lumen segmentation algorithms in computed tomography angiography. *Medical image analysis*, 17(8):859–876, 2013. 2

[13] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, J. Reynolds, A. Melnikov, N. Lunova, and O. Reblitz-Richardson. Pytorch captum. `https://github.com/pytorch/captum`, 2019. 6

[14] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun. ShuffleNet v2: Practical guidelines for efficient CNN architecture design. In *Computer Vision – ECCV 2018*, pages 122–138. Springer International Publishing, 2018. 4, 5

[15] C. T. Metz, M. Schaap, A. C. Weustink, N. R. Mollet, T. van Walsum, and W. J. Niessen. Coronary centerline extraction from CT coronary angiography images using a minimum cost path approach. *Medical Physics*, 36(12):5568–5579, Nov. 2009. 2

[16] J. Schlemper, O. Oktay, M. Schaap, M. Heinrich, B. Kainz, B. Glocker, and D. Rueckert. Attention gated networks: Learning to leverage salient regions in medical images, 2018. 8

[17] L. N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Mar. 2017. 5

[18] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328, 2017. 6, 8

[19] A. R. van Rosendael, I. J. van den Hoogen, A. M. Bax, S. J. Al'Aref, O. A. H. Alawamlh, D. Larine, and J. K. Min. CT and calcification. In *Coronary Calcium*, pages 83–123. Elsevier, 2019. 5

[20] J. M. Wolterink, R. W. van Hamersvelt, M. A. Viergever, T. Leiner, and I. Išgum. Coronary artery centerline extraction in cardiac CT angiography using a CNN-based orientation classifier. *Medical Image Analysis*, 51:46–60, Jan. 2019. 2

[21] X. Zhang, X. Zhou, M. Lin, and J. Sun. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *2018 IEEE/CVF Conference*

on Computer Vision and Pattern Recognition. IEEE, June 2018. 4

[22] Zreik et al. A recurrent CNN for automatic detection and classification of coronary artery plaque and stenosis in coronary CT angiography. *IEEE Transactions on Medical Imaging*, 38(7):1588–1598, 2019. 2, 6, 7

25^{th} Computer Vision Winter Workshop
Domen Tabernik, Alan Lukežič, Klemen Grm (eds.)
Rogaška Slatina, Slovenia, February 3–5, 2020

Towards Data-driven Multi-target Tracking for Autonomous Driving

Christian Fruhwirth-Reisinger, Georg Krispel, Horst Possegger, Horst Bischof
Graz University of Technology
Institute of Computer Graphics and Vision
{christian.reisinger, georg.krispel, possegger, bischof}@icg.tugraz.at

Abstract. *We investigate the potential of recurrent neural networks (RNNs) to improve traditional online multi-target tracking of traffic participants from an ego-vehicle perspective. To this end, we build a modular tracking framework, based on interacting multiple models (IMM) and unscented Kalman filters (UKF). Following the tracking-by-detection paradigm, we leverage geometric target properties provided by publicly available 3D object detectors. We then train and integrate two RNNs: A state prediction network replaces hand-crafted motion models in our filters and a data association network finds detection-to-track assignment probabilities. In our extensive evaluation on the publicly available KITTI dataset we show that our trained models achieve competitive results and are significantly more robust in the case of unreliable object detections.*

1. Introduction

Multi-target tracking (MTT) aims at jointly estimating the number of targets and their current states from a sequence of unreliable measurements. It is one of the fundamental visual perception tasks for autonomous driving (AD) [21] which allows, for example, reactive navigation or motion planing.

In this work, we address the problem of tracking robustness despite unreliable detections, which strongly degrade tracking performance. This requires, on the one hand, precise state predictions for frames without proper detections and, on the other hand, reasonable track-to-detection assignments. Hence, we train and integrate two recurrent neural networks (RNNs) for these purposes, as illustrated in Fig. 1. The advantage as shown in our evaluations is that data-driven models generalize better on numerous different situations.

Most recent online multi-target tracking approaches, *e.g.* [16, 30, 44, 50] follow the tracking-by-detection paradigm and thus, assume a detected set of possible targets in each frame. For AD, these detections are usually obtained from state-of-the-art object detectors [29, 39, 45] which estimate 3D bounding boxes from LiDAR data or RGB images.

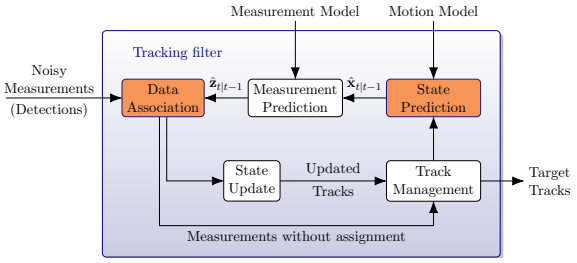


Figure 1. Tracking-by-detection scheme. We exchange the highlighted modules (*i.e.* State Prediction and Data Association) with data-driven recurrent neural networks.

Simultaneously tracking multiple targets is commonly handled by applying a single target tracker (STT) for each object instance, realized by probabilistic filters [1, 6]. These filters predict the current state usually relying on hand-crafted motion models and update the predictions with assigned detections to estimate the *posterior* distribution of each track. A data association step assigns the most reasonable detection to each track and consequently allows the filter updates. Finally, MTT requires track management to initialize and terminate trajectories.

Hence, the main challenges in multi-target tracking are the assignment of detections to tracks and determining whether a track exists or not. The former strongly depends on the detection quality. Many false positive (FP) or false negative (FN, *i.e.* missed) detections require additional knowledge of the tracked targets, *e.g.* their motion behavior, to produce a reasonable trajectory. Additionally, targets at high speed, moving sensors and interacting targets hamper

accurate associations. The question whether a track exists or not, on the other hand, is even harder to answer. Occluded targets and false detections can lead to missing or wrongly initialized tracks, respectively.

Our main contribution is to investigate several tracking aspects for AD within a combined Bayesian filter-based MTT framework. To track traffic participants in 3D world coordinates, we leverage the interacting multiple model (IMM) [8] approach combined with an unscented Kalman filter (UKF) [23, 47], to allow multiple, potentially non-linear motion models. For our investigation, we focus on two tasks: State prediction and data association. We exchange these traditionally hand-crafted parts with learned RNN models and evaluate their impact on the KITTI dataset [17]. Our experiments demonstrate that our data-driven models significantly improve the tracking performance, especially in the case of unreliable or missing detections. Furthermore, in contrast to most recent works which focus exclusively on well-represented object classes (*i.e*. cars and pedestrians), we consider all available object classes. This allows more meaningful conclusions about the tracking capabilities for real-world AD scenarios.

2. Related Work

Because of the large diversity of tracking methods, we focus mainly on online filtering-based and deep learning approaches which have been proposed for traffic scenarios. For a more extensive survey we refer the interested reader to the recent works of Krebs *et al.* [27] and Vo *et al.* [46].

Bayesian Filtering: Most MTT algorithms following the tracking-by-detection paradigm are modeled as parallel STT approaches joined by a data association step. However, even for the simple case of a single target, well-known filtering approaches, *e.g.* linear Kalman filter (KF) [24] or UKF, can not be applied directly [46]. The reason for this are detection origin uncertainties, FP and FN detections.

A simple solution to this problem is the nearest neighbor (NN) filter [3, 9], which uses the closest detection in terms of spatial distance to each predicted state, *e.g.* [1]. However, such a setup is prone to lose tracks in case of wrong detection-to-track assignments due to FPs and FNs. An improved version is the probabilistic data association (PDA) filter [2, 4]. It uses assignment probabilities of certain detections in each frame and applies the state estimation filter with weighted detections to all targets individually [3, 46]. This improves the results in cluttered environments. However, both filters, NN and PDA, are designed for STT and should be used in MTT problems with clearly separable targets only.

In contrast to this local data assocation, global strategies consider all detections and tracked targets in every frame. The most common approaches are global nearest neighbor (GNN) and joint probabilistic data association (JPDA) [15]. While the former solves a minimization problem on given costs w.r.t. distance, intersection over union (IoU) or likelihood, the latter is an extension to the PDA filter. It performs a weighted update including all detections within a certain gating region simultaneously regarding all tracks. For example, Choi *et al.* [10] combined GNN association with a linear KF. Their association criterion is based on a weighted sum of target distance and size. In contrast, Sharma *et al.* [44] proposed a tracker without filtering which solves the association problem via the Hungarian [37] algorithm. They devised several costs from 3D cues, which are directly learned from monocular images.

Commonly used filtering approaches typically employ a single linear motion model, *e.g.* constant velocity (CV). However, traffic participants do not always act in a linear way. Hence, the interacting multiple model (IMM) [8] approach enables switching various models representing different motion patterns. This includes, for example, the *coordinated turn* modeled by the constant turn rate velocity (CTRV) model and static or slow movement modeled by the random model [31]. Rachman [40] proposed a tracker for traffic scenarios based on unscented Kalman filtering with an IMM and JPDA, which inspired our baseline MTT framework. However, his evaluation is restricted to a specific environmental scenario with few selected KITTI sequences and thus, complicates deducing insights on the general applicability for autonomous driving.

The main drawback of these approaches is the missing ability to handle a variable number of tracks. Stacking all tracks in a single state vector, on the one hand, restricts trackers to a fixed number of targets. Initiating a new filter for each target, on the other hand, requires an external track management. Alternatively, the multiple hypothesis tracker (MHT) [7, 41] builds hypotheses for all track-to-measurement assignments over time including the possibility of initiating and terminating tracks. However, the complexity of MHT grows exponentially with each time

step, which requires complexity reduction, *e.g.* via hypothesis pruning or track merging [11, 12, 25].

Sequential Monte Carlo (SMC) methods [14, 13, 36], like the particle filter [18, 34] can also be directly used for state estimation. This filter usually performs better in non-linear/non-Gaussian environments because it approximates the *posterior* PDF by a finite set of particles. One drawback of these approaches, however, is the high computational complexity.

Deep Learning: A more recent research direction is to leverage deep neural networks (DNNs) for MTT. They have gained a lot of attention within the last years because of their impressive performance in many computer vision challenges, especially object detection and classification [28].

Despite the fact that AD requires tracking in 3D world coordinates because vehicles depend on spatial information, a lot of tracking approaches are designed and evaluated in the image space. Sharma *et al.* [44], for example, leverage 2D and 3D cues from monocular images to perform tracking in the 2D image space. Another 2D approach was proposed by Gündüz and Acarman [19]. They exploit image features to find similarities between consecutive frames. In contrast, Zhang *et al.* [50] used 3D information from point clouds fused with image features even though they perform tracking in 2D.

On the other hand, one way to track in 3D world coordinates from monocular images only, is to estimate the distances between ego-vehicle and detected targets [42]. End-to-end trainable models using 3D LiDAR data [33] and additional RGB images [16] were also proposed. Hu *et al.* [20] estimate 3D bounding boxes from a sequence of images and track them with a trained long short-term memory (LSTM) network. A universal tracking approach based on RNNs was proposed by Milan *et al.* [35]. Their end-to-end trainable network represents equal structures like well-known Bayesian filters and processes simple bounding box inputs.

In contrast to these, we leverage the best of both worlds, *i.e.* combining learned motion/association models and well understood filtering techniques.

3. Multiple Object Tracker

Given a sequence of noisy 3D bounding boxes, we want to track all objects of interest trough time. Each target is represented by its current state which is modeled by a random variable. Such a task can be seen as a dynamic state estimation problem. For this purpose, we model a single tracker for each target with the state-space approach which allows state estimation from noisy detections. Thus, a dynamical model describes the state transition over time and a measurement model relates detections to the state.

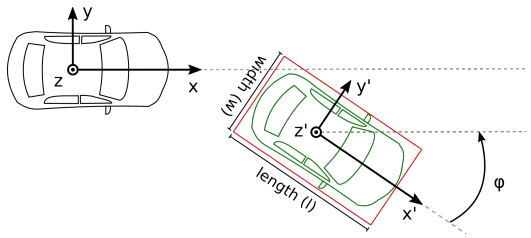


Figure 2. Sensor model [17] with ego-vehicle in black and an exemplary target car in green.

A well-developed framework for this problem is the Bayesian filter for which computational traceable solutions (*i.e.* KF, UKF) exist. It can be applied in a recursive manner to handle incoming detections for each time frame which is crucial for an online tracker. In the following we discuss the specific combination of such filters we exploit for target tracking. Additionally, we replace parts of the filters/trackers with data-driven RNN models.

3.1. Bayesian Tracking Framework

Representations: Object detections are represented by 3D bounding boxes, relative to the ego-vehicle's center coordinates x, y, z as illustrated in Fig. 2.

We further define the state vector at time t as $\mathbf{x}_t = (\text{pos}_t, \omega_t, v_t, \dot{\omega}_t, \text{bb}_t, \varphi_t)^T$, where pos_t denotes the center coordinates x_t, y_t and z_t, bb_t denotes the bounding box dimension, *i.e.* width w_t, length l_t and height h_t, and φ_t denotes the bounding box orientation. Additionally, the state vector contains non-observable parameters: Steering angle ω_t, velocity v_t and turn rate $\dot{\omega}_t$. Notice, we use different parameters for steering angle and bounding box orientation to support scenarios where ego-motion measurements are not available.

The measurement vector $\mathbf{z}_t$ contains observable parameters which can be obtained from the object detector at time t, *i.e.* $\mathbf{z}_t = (x_t, y_t, z_t, w_t, l_t, h_t, \varphi_t)^T$.

Unscented Kalman Filter (UKF): is a computationally tractable solution of the Bayesian filter which allows non-linear dynamical models. The main idea of the UKF is to propagate a fixed number of appropriately chosen weighted sample points – so-called sigma points – through a non-linear func-

tion by using the Unscented Transformation (UT). This process does not need an analytical derivation of dynamic and measurement functions. We leverage the scaled UT [22] which ensures a positive semi-definite covariance matrix.

Thus, we first need to determine $2n+1$ sigma points $\mathcal{X}_{i,t-1} \in \mathbb{R}^n$ with $i \in \{0,\dots,2n\}$ for n state variables, weights $\mathbf{w}^{(m)} \in \mathbb{R}^n$ for state mean $\hat{\mathbf{x}}_{t-1|t-1}$, and weights $\mathbf{w}^{(c)} \in \mathbb{R}^n$ for the state covariance matrix $\hat{\mathbf{P}}_{t-1|t-1}$. These weights depend on the scaling factor $\lambda = \alpha^2(n+\kappa) - n$, where α controls the spread of the sigma points around the mean. The remaining parameters κ and β represent another scaling and prior knowledge about the state distribution, respectively. To avoid sampling non-local effects under strong nonlinearities, the parameter should be $0 \leq \alpha \leq 1$. Furthermore, positive semi-definiteness can be guaranteed by choosing the parameter $\kappa \geq 0$. A typically good choice for state estimation problems is $\kappa = 0$. Finally, for Gaussian distributions $\beta = 2$ is optimal, otherwise it should be non-negative.

We further use the scaling parameters to sample scaled sigma points $\mathcal{X}_{i,t-1} \in \mathbb{R}^n$ with $i \in \{0,\dots,2n\}$ as in [22] from the previous *posterior* state $\hat{\mathbf{x}}_{t-1|t-1}$ and covariance $\mathbf{P}_{t-1|t-1}$. Afterwards, we propagate the sigma points through a potentially non-linear function $f(\cdot)$ representing the dynamic model and use the propagated sigma points

$$\mathcal{X}_{i,t|t-1} = f(\mathcal{X}_{i,t-1}) \quad \forall i \in \{0,\dots,2n\}, \tag{1}$$

to calculate the predicted mean $\hat{\mathbf{x}}_{t|t-1}$ and covariance $\mathbf{P}_{t|t-1}$ as

$$\hat{\mathbf{x}}_{t|t-1} = \sum_{i=0}^{2n} \mathbf{w}_i^{(m)} \mathcal{X}_{i,t|t-1}, \quad \text{and} \tag{2a}$$

$$\mathbf{P}_{t|t-1} = \sum_{i=0}^{2n} \mathbf{w}_i^{(c)} \mathbf{v}_x^T \mathbf{v}_x^T + \mathbf{Q}_t, \tag{2b}$$

with innovation $\mathbf{v}_x = \left(\mathcal{X}_{i,t|t-1} - \hat{\mathbf{x}}_{t|t-1}\right)$ and process noise covariance matrix $\mathbf{Q}_t \in \mathbb{R}^{n\times n}$.

The update step requires a new set of $2n+1$ sigma points $\mathcal{X}_{i,t} \in \mathbb{R}^n$ with $i \in \{0,\dots,2n\}$ for n state variables and weights for predicted state mean $\hat{\mathbf{x}}_{t|t-1}$ and corresponding covariance matrix $\hat{\mathbf{P}}_{t|t-1}$. Afterwards, we propagate these sigma points through the measurement function $h(\cdot)$

$$\mathcal{Z}_{i,t|t-1} = h\left(\mathcal{X}_{i,t}\right) \quad \forall i \in \{0,\dots,2n\}, \tag{3}$$

and build a weighted sum to obtain predicted *a priori* measurements $\hat{\mathbf{z}}_{t|t-1}$, the corresponding innovation covariance matrix $\mathbf{S}_t$ and the cross covariance $\mathbf{C}_t$ as

$$\hat{\mathbf{z}}_{t|t-1} = \sum_{i=0}^{2n} \mathbf{w}_i^{(m)} \mathcal{Z}_{i,t|t-1}, \quad \text{and} \tag{4a}$$

$$\mathbf{S}_t = \sum_{i=0}^{2n} \mathbf{w}_i^{(c)} \mathbf{v}_z \mathbf{v}_z^T + \mathbf{R}_t, \quad \text{and} \tag{4b}$$

$$\mathbf{C}_t = \sum_{i=0}^{2n} \mathbf{w}_i^{(c)} \left(\mathcal{X}_{i,t|t-1} - \hat{\mathbf{x}}_{t|t-1}\right) \mathbf{v}_z^T, \tag{4c}$$

with innovation $\mathbf{v}_z = \left(\mathcal{Z}_{i,t|t-1} - \hat{\mathbf{z}}_{t|t-1}\right)$ and measurement noise covariance matrix $\mathbf{R}_t \in \mathbb{R}^{q\times q}$.

Finally, we compute the *posterior* mean $\hat{\mathbf{x}}_{t|t}$ and covariance matrix $\mathbf{P}_{t|t}$ as

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{C}_t \mathbf{S}_t^{-1} (\mathbf{z}_t - \hat{\mathbf{z}}_{t|t-1}), \quad \text{and} \tag{5a}$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{C}_t \mathbf{S}_t^{-1} \mathbf{C}_t^T. \tag{5b}$$

Interacting Multiple Model: The original UKF implementation supports only a single dynamic model which is often also referred to as motion model. Mostly, a single model is not able to cover the motion behavior of various traffic participants. One solution to this problem is the interacting multiple model (IMM). It is a traceable approximation to the intractable multiple model optimal Bayes filter [43], which is modeled as jump Markov non-linear system. Besides the states of a system, such a filter estimates mode probabilities, which defines how likely a motion model matches the system's behavior.

The multiple model optimal Bayes filter and its approximation assumes a fixed set $\mathcal{M} = \{M_j\}_{j=1}^r$ of r models, each processed by a recursive filter, *e.g.* linear KF or UKF. Model state transitions within the IMM are modeled by a first-order Markov chain represented by a state transition probability matrix $\Pi = [p_{i,j}] \in \mathbb{R}^{r\times r}$, where $p_{i,j}$ denotes the probability of a state transition from model i to model j. Hence, the main diagonal $p_{i,i}$ contains the probabilities to stay in the same model state.

Basically, a full cycle of the recursive IMM filter contains four steps: interaction, prediction, update and combination. First, we perform a probabilistic mixing with the *posterior* state estimate $\hat{\mathbf{x}}_{i,t-1|t-1}$ and covariance estimate $\mathbf{P}_{i,t-1|t-1}$ of the previous stage for each filter j as

$$\hat{\mathbf{x}}^{\star}_{j,t-1|t-1} = \sum_{i=1}^{r} \mu_{i|j,t-1}\, \hat{\mathbf{x}}_{i,t-1|t-1}, \quad \text{and} \tag{6a}$$

$$\mathbf{P}^{\star}_{j,t-1|t-1} = \sum_{i=1}^{r} \mu_{i|j,t-1} \left(\mathbf{P}_{i,t-1|t-1} + \mathbf{v}_i \mathbf{v}_i^T\right), \tag{6b}$$

with innovation $\mathbf{v}_i = (\hat{\mathbf{x}}_{i,t-1|t-1} - \hat{\mathbf{x}}^{\star}_{j,t-1|t-1})$. This results in a single initial state $\hat{\mathbf{x}}^{\star}_{j,t-1|t-1}$ and covariance $\mathbf{P}^{\star}_{j,t-1|t-1}$. The mixing probabilities $\mu_{i|j,t-1}$ can be calculated as

$$\mu_{i|j,t-1} = \frac{p_{i,j}\,\mu_{i,t-1}}{\mu^{-}_{j,t}}, \text{ with } \mu^{-}_{j,t} = \sum_{i=1}^{r} p_{i,j}\mu_{i,t-1}, \quad (7)$$

where $\mu^{-}_{j,t}$ is the predicted mode probability for filter j at the current time step, $\mu_{i,t-1}$ the mode probability of the previous time step, and $p_{i,j}$ the transition probability to switch from model i to j. In summary, the previous filters with their mode probability and the transition probability directly influence the initial state of each filter.

Afterwards, each of the j filters performs a separate prediction step as in Eq. (2) to obtain predicted states $\hat{\mathbf{x}}_{j,t|t-1}$ and corresponding covariance matrices $\mathbf{P}_{j,t|t-1}$. Additionally, this step yields predicted measurements $\hat{\mathbf{z}}_{j,t|t-1}$ and corresponding innovation covariance matrices $\mathbf{S}_{j,t}$ (see Eq. (4)). In order to obtain the *posterior* state $\hat{\mathbf{x}}_{j,t|t}$ and covariance matrix $\mathbf{P}_{j,t|t}$ each filter updates its state individually (see Eq. (5)). Within an IMM filter cycle, we then update the mode probabilities

$$\mu_{j,t} = \frac{\mathcal{L}_{j,t}\,\mu^{-}_{j,t}}{\sum_{i=1}^{r}\mathcal{L}_{i,t}\,\mu^{-}_{i,t}}, \quad \mathcal{L}_{j,t} = \mathcal{N}\left(\mathbf{z}_t; \hat{\mathbf{z}}_{j,t|t-1}, \mathbf{S}_{j,t}\right), \quad (8)$$

where $\mathcal{L}_{j,t}$ denotes the likelihood of a model fitting the assigned measurement $\mathbf{z}_t$.

Finally, we obtain the *posterior* state $\hat{\mathbf{x}}_{t|t}$ and its covariance $\mathbf{P}_{t|t}$ by combining the output of each filter, weighted by the mode probability

$$\hat{\mathbf{x}}_t = \sum_{j=1}^{r} \mu_{j,t}\,\hat{\mathbf{x}}_{j,t|t}, \quad \text{and} \quad (9a)$$

$$\mathbf{P}_t = \sum_{j=1}^{r} \mu_{j,t|t}\left(\mathbf{P}_{j,t|t} + \mathbf{v}_j\mathbf{v}_j^T\right), \quad (9b)$$

with innovation $\mathbf{v}_j = (\hat{\mathbf{x}}_{j,t|t} - \hat{\mathbf{x}}_{t|t})$. Note that this final result is not part of the filter recursion itself.

Within our tracking framework each tracker is initialized with three motion models as in [40]. First, the constant velocity (CV) model for straight motion. Second, the constant turn rate velocity (CTRV) model for coordinated turns, *e.g.* at cross ways. And third, the random (RAND) model represents static or slowly moving targets.

3.2. Data Association and Track Management

Following the tracking-by-detection scheme, our approach requires an association mechanism which joins tracks and detections in each time frame. To this end, we leverage two different approaches. A global exclusive method, *i.e.* Hungarian [37] algorithm, on the one hand, and a joint probabilistic approach, *i.e.* JPDA [2, 15] on the other hand.

For the global exclusive approach we use the negative intersection over union value of targets and detections to fill a cost matrix. Afterwards, the Hungarian algorithm finds associations by minimizing the total cost. Tracks without assigned detection and vice versa are managed by the track management.

The JPDA approach takes all tracks and detections of a certain time frame into account. First, it performs a gating mechanism based on the Mahalanobis distance between tracks and detections. Afterwards, it calculates association probabilities for each detection within a certain gating region. Finally, the used filter performs an update weighted by these association probabilities. Detections which do not belong to a gating region and tracks without assigned detections are also managed by the track management.

A simple track manager takes care of unassigned detections and tracks. Based on fixed thresholds, a track missing τ_m updates gets terminated. On the other hand, detections without assignment are used to initialize new tracks. Such newly initialized tracks are considered active after receiving τ_u updates.

3.3. RNN Models

Our modular framework allows to exchange different parts with data-driven models. This is, on the one hand, the IMM with hand-crafted motion models which can be replaced by a RNN trained for state prediction. Data association, on the other hand, depends on the intersection over union between tracks and detections. Hence, we train an encoder-decoder RNN to find association probabilities.

State Prediction: We leverage a two-layer LSTM network with 256 hidden units each and a fully connected output layer with two nodes and linear activation as shown in Table 1. The network takes the

Layer	Type	Input	Output	Activation
1	LSTM	4	256	-
2	LSTM	256	256	-
3	FC	256	2	identity

Table 1. Prediction network architecture.

center coordinates x_t, y_t of the bounding box ground area and the ego-vehicle movement in x and y direction as an input. The output is then the predicted

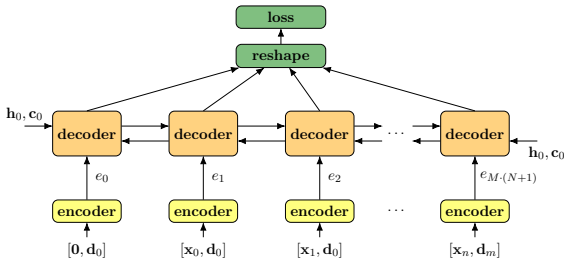


Figure 3. Encoder-Decoder structured network for data association using a bidirectional LSTM decoder [49].

center position of the bounding box ground area.

The model is trained with sequences of length 5 and optimized by the Adam [26] optimizer with a learning rate of 0.003 and the mean squared error loss. For inference, we replace the IMM by propagating the sigma points of the UKF through the trained model (see Eq. (1)).

Data Association: Matching a variable number of tracks and detections can be handled by an encoder-decoder model [49]. We adapted this approach for our purpose and thus, replace the encoder with a RNN and the MSE loss with a cross entropy error loss. Furthermore, we also learned the initial internal states $\mathbf{h}_0$ and $\mathbf{c}_0$ of the decoder while training.

The input to this network are permutations of all matching combinations between N tracks and M detections including the case that a detection belongs to no track. Fig. 3 shows the model structure. Each permutation contains the last 5 states $\mathbf{x}$ of a track and one detection $\mathbf{d}$. All entries contain the center position x, y of the bounding box ground area and its top-view dimension, *i.e.* width w and length l.

The encoded permutations $\mathbf{e}_i$ of size 64 are the input of the decoder, which is composed of one bidirectional LSTM layer with 64 hidden units and two fully-connected layers resulting in a single output for each permutation pair. After softmax activation and reshaping the output, we get a cost matrix representing the detection-to-track assignment probability.

Because the network is not able to learn the one-to-one constraint between tracks and detections, we apply the Hungarian algorithm on the cost matrix at inference time. Furthermore, we remove assignments with a probability < 0.5 and assignments to the dummy track with probability > 0.5.

Training is also done with the Adam optimizer and a learning rate of 0.003. To avoid overfitting, we apply early stopping and data augmentation, *i.e.* mirroring along the x-axis and adding noise.

4. Experimental Results

Dataset: To demonstrate our multi-target tracking framework for AD scenarios, we evaluate it on the publicly available KITTI dataset [17] which provides various environment categories, *i.e.* *City*, *Residential* and *Road*. The dataset contains RGB image sequences and LiDAR data with corresponding 3D bounding box annotations for eight different object classes, *i.e.* *Car*, *Pedestrian*, *Cyclist*, *Van*, *Tram*, *Person sitting*, *Truck* and *Misc*.

The KITTI dataset contains two partly overlapping datasets: The *tracking* dataset and the *raw* dataset. The former consists of 21 training sequences and 29 test sequences and the latter contains 38 sequences, sorted by environment categories. For evaluations on the *tracking* dataset, we apply the widely used train/validation split [38] since there are no public annotations for the corresponding test data. For evaluations on the *raw* dataset, we carefully select a set of sequences[1] containing all environmental categories as well as under-represented object classes, *i.e.* *Cyclist*, *Truck* and *Tram*. We further ensure that no training sequence is part of the validation set.

Performance measures: We employ the widely used CLEAR measures [5], namely Multiple Object Tracking Precision (MOTP) and Accuracy (MOTA). MOTP reflects the tracker's precision *wrt.* object locations and dimensions, whereas MOTA states the overall tracking ability. MOTA can be described as the consistent labeling of objects over time and takes false positive trajectories (FP), false negative trajectories (FN) and identity switches (IDS) into account. Additionally, we report the common track quality measures [32] which describe the coverage of tracks as either mostly tracked (MT), partly tracked (PT) or mostly lost (ML).

Baseline: In addition to our combined filter model, we implement a 3D version of SORT [6]. Concurrently to our work, a similar SORT extension [48] was submitted to KITTI[2] (approximately 5–6 % lower MOTA than the current leaders). This allows us to compare our evaluations to state-of-the art approaches listed in this leaderboard.

Notation: For each experiment, we denote a tracker configuration by the respective data associa-

[1] We use sequences 0001, 0005, 0014, 0018, 0060, 0084, 0020, 0039, 0064, and 0070 of KITTI *raw* as validation set.

[2] `http://www.cvlibs.net/datasets/kitti/eval_tracking.php`

	IDS	MOTA	MOTP	FPS
3D SORT	53	76.5%	**75.5%**	**412**
GNN	85	75.9%	74.6%	74
JPDA	99	72.1%	73.4%	50
RNN_{DA}	**48**	76.2%	74.3%	33
RNN_{Pr} – GNN	68	**77.2%**	71.5%	40
RNN_{Pr} – RNN_{DA}	78	75.6%	71.5%	34

Table 2. Evaluation of all tracker configurations on KITTI *raw* dataset regarding precision, accuracy, ID switches and runtime (without the object detection step) in frames per second (FPS). **Bold** scores denote the best results.

tion scheme, *e.g.* GNN denotes the IMM-UKF baseline with GNN [37]. RNN_{DA} denotes the learned data association network. Configurations which use the data-driven state prediction model are denoted by the additional prefix RNN_{Pr}.

4.1. Realistic Traffic Scenario

MTT for AD must perform well for all kinds of object classes and environments. Hence, we evaluate our trackers on the KITTI *raw* dataset, considering all traffic participants for which annotated ground truth is available. Note that we use the corresponding training set to optimize all parameters which are then fixed for all further experiments.

KITTI *raw* dataset: The results in Table 2 show that data-driven models improve the tracking performance in different ways. The learned data association model, on the one hand, produces the lowest number of ID switches and the prediction model outperforms the baseline by approx. 0.7% regarding MOTA on the other hand. Fig. 4 reveals that the

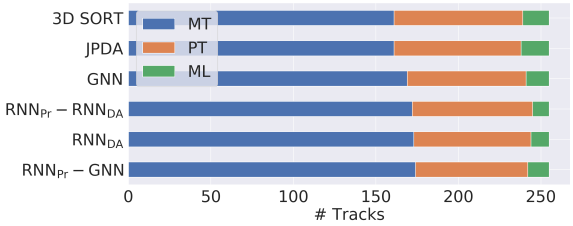


Figure 4. Track coverage of all approaches on the selected sequences of the KITTI *raw* dataset.

learned prediction model ensures also a high coverage of tracks. Additionally, it shows that the learned association model minimizes the number of totally lost tracks. Notwithstanding, we observe that the threshold values regarding initiation and termination of tracks highly influence the number of MT targets, although this only slightly affects the overall MOTA.

	IDS	MOTA	MOTP	FPS
3D SORT	77	63.1%	73.1%	**657**
GNN	46	71.6%	**73.2%**	98
JPDA	63	60.0%	67.8%	77
RNN_{DA}	**43**	**72.1%**	71.2%	54
RNN_{Pr} – GNN	65	72.0%	66.8%	52
RNN_{Pr} – RNN_{DA}	72	71.7%	65.9%	48

Table 3. Evaluation on KITTI *raw* dataset, where we omit detections of every second frame.

Fig. 5 shows qualitative results for $\text{RNN}_{\text{Pr}} - \text{RNN}_{\text{DA}}$.

We observed that evaluating solely on the main object classes (*i.e.* cars, pedestrians and cyclists) shows only a minor performance improvement (overall 1–2 % MOTA). Nevertheless, we evaluate on all object classes since they are all important for reliable perception in AD.

Dropping Detections: In order to evaluate the state prediction quality of all trackers, we drop detections for selected frames. Because of periodically missing detections, updates in consecutive time frames are not possible. Thus, we adapt the thresholds for initialization and termination of tracks within our track management. We set the threshold for initialization to 1 which causes an immediate initialization and increase the threshold for termination by the number of frames without detection.

Table 3 shows the results for omitting all detections in every second frame. We notice a significant decrease w.r.t. MOTA for the baseline approach and the tracker with JPDA. While the former is limited

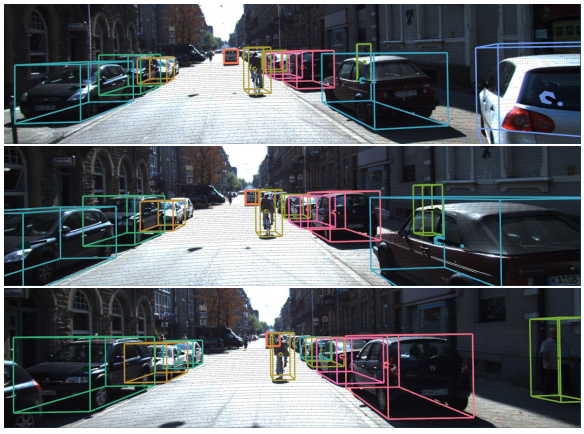


Figure 5. Qualitative results of sequence '0005' from the KITTI *raw* dataset. The illustration shows colored 3D bounding boxes, each color representing a track instance.

	IDS	MOTA	MOTP	FPS
3D SORT	65	55.7%	**72.5%**	**826**
GNN	**28**	63.0%	71.5%	112
JPDA	54	48.4%	61.7%	107
RNN_{DA}	44	67.2%	68.9%	73
$\text{RNN}_{\text{Pr}} - \text{GNN}$	48	63.8%	61.1%	60
$\text{RNN}_{\text{Pr}} - \text{RNN}_{\text{DA}}$	53	**67.3%**	58.7%	56

Table 4. Evaluation on KITTI *raw* dataset, where we omit detections of every second and third frame.

by a linear motion model, the latter suffers from its static configuration. In contrast, configurations with our learned components lose only $\approx 4\% - 5\%$ while SORT loses more than 13% accuracy. This results in an increase of ML tracks as illustrated in Fig. 6.

Table 4 shows the results for omitting all detections in every second and third frame. Again, SORT and the JPDA tracker perform worse and the number of ML tracks for these trackers is two times higher as for our best performing approach.

4.2. Different Detectors

KITTI *tracking* dataset: For a better comparability to state-of-the-art approaches, we also evaluate on the widely used validation split [38] of the KITTI *tracking* dataset. Note, however, that a direct comparison is not possible as most approaches evaluate in 2D image space and only consider the three most well-represented object classes (cars, pedestrians and cyclists). Table 5 shows results for all object classes which are approximately 9% worse in comparison to our previous evaluations. This can be contributed to the larger number of highly crowded pedestrian scenes which cause frequent detection errors due to heavy occlusions.

So far, all our reported results leverage Frustum PointNets [39] detections. Table 6 demonstrates the effect of using PointRCNN [45] instead. The results are similar to Frustum PointNets (Table 5). However,

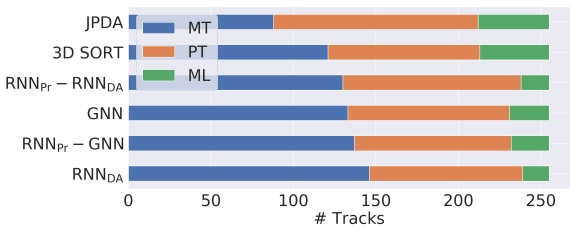


Figure 6. Track coverage of all approaches on the selected sequences of the KITTI *raw* dataset assuming omitted detections of each second frame.

	IDS	MOTA	MOTP	FPS
CIWT [38] (cars)	26	74.38%	82.85%	-
CIWT [38] (ped.)	41	61.87%	78.85%	-
3D SORT	160	67.2%	71.6%	**358**
GNN	191	**68.9%**	71.0%	50
JPDA	**103**	57.8%	**72.2%**	34
RNN_{DA}	185	64.5%	70.3%	27
$\text{RNN}_{\text{Pr}} - \text{GNN}$	229	68.7%	67.0%	30
$\text{RNN}_{\text{Pr}} - \text{RNN}_{\text{DA}}$	240	64.0%	66.1%	27

Table 5. Evaluation on the validation split of the KITTI *tracking* dataset. Note that [38] only evaluate on selected, well-represented object classes.

	IDS	MOTA	MOTP	FPS
3D SORT	64	66.4%	80.9%	**400**
GNN	67	**68.5%**	81.2%	62
JPDA	**48**	54.8%	**81.4%**	45
RNN_{DA}	77	60.2%	**81.4%**	31
$\text{RNN}_{\text{Pr}} - \text{GNN}$	91	68.2%	76.7%	36
$\text{RNN}_{\text{Pr}} - \text{RNN}_{\text{DA}}$	128	62.5%	76.3%	32

Table 6. Evaluation on the validation split of the KITTI *tracking* dataset using PointRCNN [45] detections.

we observe a performance decrease for models with RNN_{DA}. This can be contributed to the model training, since RNN_{DA} was trained using Frustum PointNets detections. Additionally, the overall improved MOTP results reveal a significantly better bounding box orientation estimation of PointRCNN compared to Frustum PointNets.

5. Conclusion

In this paper, we investigate several tracking aspects for AD within a combined Bayesian filter-based MTT approach. In particular, we leverage the IMM combined with UKF, as well as different data association methods. In order to increase tracking robustness despite unreliable detections, we exchange the state prediction and data association with data-driven models. Our evaluations show competitive results to state-of-the-art approaches and an improved robustness on the challenging KITTI dataset.

Acknowledgements

This project was partially funded by the Austrian Research Promotion Agency (FFG) under the project DGT (860820) and the Christian Doppler Laboratory for Embedded Machine Learning.

References

[1] A. Asvadi, P. Peixoto, and U. Nunes. Detection and Tracking of Moving Objects Using 2.5D Motion Grids. In *Proc. ITSC*, 2015. 1, 2

[2] Y. Bar-Shalom, F. Daum, and J. Huang. The Probabilistic Data Association Filter: Estimation in the presence of measurement origin uncertainty. *IEEE Control Sys.*, 29(6):82–100, 2009. 2, 5

[3] Y. Bar-Shalom and X.-R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, first edition, 1995. 2

[4] Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11(5):451–460, 1975. 2

[5] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *JIVP*, 2008(1):1–10, 2008. 6

[6] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple Online and Realtime Tracking. In *Proc. ICIP*, 2016. 1, 6

[7] S. S. Blackman. Multiple Hypothesis Tracking for Multiple Target Tracking. *IEEE MAES*, 19(1 II):5–18, 2004. 2

[8] H. Blom and Y. Bar-Shalom. The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE TAC*, 33(8):780–783, 1988. 2

[9] S. Challa, M. R. Morelande, D. Musicki, and R. J. Evans. *Fundamentals of Object Tracking*. Cambridge University Press, first edition, 2011. 2

[10] J. Choi, S. Ulbrich, B. Lichte, and M. Maurer. Multi-Target Tracking using a 3D-Lidar sensor for Autonomous Vehicles. In *Proc. ITSC*, 2013. 2

[11] I. J. Cox. A review of statistical data association techniques for motion correspondence. *IJCV*, 10(1):53–66, 1993. 3

[12] I. J. Cox and S. L. Hingorani. Efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE TPAMI*, 18(2):138–150, 1996. 3

[13] B. Fortin, R. Lherbier, and J.-C. Noyer. A Model-Based Joint Detection and Tracking Approach for Multi-Vehicle Tracking with Lidar Sensor. *IEEE TITS*, 16(4):1883–1895, 2015. 3

[14] B. Fortin, J. C. Noyer, and R. Lherbier. A Particle Filtering Approach for Joint Vehicular Detection and Tracking in LiDAR data. In *Proc. I2MTC*, 2012. 3

[15] T. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *J. Ocean. Eng.*, 8(3):173–184, 1983. 2, 5

[16] D. Frossard and R. Urtasun. End-to-end Learning of Multi-sensor 3D Tracking by Detection. In *Proc. ICRA*, 2018. 1, 3

[17] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *IJRR*, 32(11):1231–1237, 2013. 2, 3, 6

[18] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F*, 140(2):107, 1993. 3

[19] G. Gündüz and T. Acarman. A Lightweight Online Multiple Object Vehicle Tracking Method. In *Proc. IV*, 2018. 3

[20] H.-N. Hu, Q.-Z. Cai, D. Wang, J. Lin, M. Sun, P. Krähenbühl, T. Darrell, and F. Yu. Joint Monocular 3D Detection and Tracking. In *Proc. ICCV*, 2019. 3

[21] J. Janai, F. Güney, A. Behl, and A. Geiger. Computer Vision for Autonomous Vehicles: Problems, Datasets and State-of-the-Art. *arXiv CoRR*, abs/1704.05519, 2017. 1

[22] S. J. Julier. The scaled unscented transformation. In *Proc. ACC*, 2002. 4

[23] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proc. IEEE*, 92(3):401–422, 2004. 2

[24] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.*, 82(1):35, 1960. 2

[25] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg. Multiple Hypothesis Tracking Revisited. In *Proc. ICCV*, 2015. 3

[26] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *Proc. ICLR*, 2015. 6

[27] S. Krebs, B. Duraisamy, and F. Flohr. A survey on leveraging deep neural networks for object tracking. In *Proc. ITSC*, 2017. 2

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proc. NeurIPS*, 2012. 3

[29] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. PointPillars: Fast Encoders for Object Detection From Point Clouds. In *Proc. CVPR*, 2019. 1

[30] P. Lenz, A. Geiger, and R. Urtasun. FollowMe: Efficient online min-cost flow tracking with bounded memory and computation. In *Proc. ICCV*, 2011. 1

[31] X. R. Li and V. P. Jilkov. Survey of maneuvering target tracking. Part I: Dynamic models. *IEEE TAES*, 39(4):1333–1364, 2003. 2

[32] Y. Li, C. Huang, and R. Nevatia. Learning to associate: HybridBoosted Multi-target Tracker for Crowded Scene. In *Proc. CVPR*, 2009. 6

[33] 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 *Proc. CVPR*, 2018. 3

[34] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proc. ICCV*, 1999. 3

[35] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler. Online Multi-Target Tracking Using Recurrent Neural Networks. In *Proc. AAAI*, 2017. 3

[36] N. Morales, J. Toledo, L. Acosta, and J. Sanchez-Medina. A Combined Voxel and Particle Filter-Based Approach for Fast Obstacle Detection and Tracking in Automotive Applications. *IEEE TITS*, 18(7):1824–1834, 2017. 3

[37] J. Munkres. Algorithms for the Assignment and Transportation Problems. *JSIAM*, 5(1):32–38, 1957. 2, 5, 7

[38] A. Osep, W. Mehner, M. Mathias, and B. Leibe. Combined Image- and World-Space Tracking in Traffic Scenes. In *Proc. ICRA*, 2017. 6, 8

[39] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum PointNets for 3D Object Detection from RGB-D Data. In *Proc. CVPR*, 2018. 1, 8

[40] A. S. A. Rachman. 3D-LIDAR Multi Object Tracking for Autonomous Driving. Master's thesis, Delft University of Technology, Center for Systems and Control, 2017. 2, 5

[41] D. Reid. An algorithm for tracking multiple targets. *IEEE TAC*, 24(6):843–854, 1979. 2

[42] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granstrom. Mono-Camera 3D Multi-Object Tracking Using Deep Learning Detections and PMBM Filtering. In *Proc. IV*, 2018. 3

[43] M. Schreier. *Bayesian environment representation, prediction, and criticality assessment for driver assistance systems*. PhD thesis, Technische Universität Darmstadt, Department of Electrical Engineering and Information Technology, 2017. 4

[44] S. Sharma, J. A. Ansari, J. K. Murthy, and K. M. Krishna. Beyond Pixels: Leveraging Geometry and Shape Cues for Online Multi-Object Tracking. In *Proc. ICRA*, 2018. 1, 2, 3

[45] S. Shi, X. Wang, and H. Li. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. In *Proc. CVPR*, 2019. 1, 8

[46] B.-N. Vo, M. Mallick, Y. Bar-shalom, S. Coraluppi, R. Osborne, R. Mahler, and B.-T. Vo. *Multitarget Tracking*, pages 1–15. John Wiley & Sons, Inc., first edition, 2015. 2

[47] E. A. Wan and R. van der Merwe. The unscented Kalman filter for nonlinear estimation. In *Proc. AS-SPCC*, 2000. 2

[48] X. Weng and K. Kitani. A Baseline for 3D Multi-Object Tracking. *arXiv CoRR*, abs/1907.03961, 2019. 6

[49] K. Yoon, D. Y. Kim, Y. C. Yoon, and M. Jeon. Data association for multi-object tracking via deep neural networks. *Sensors*, 19(3):1–15, 2019. 6

[50] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi, and C. C. Loy. Robust Multi-Modality Multi-Object Tracking. In *Proc. ICCV*, 2019. 1, 3

25^{th} Computer Vision Winter Workshop
Domen Tabernik, Alan Lukežič, Klemen Grm (eds.)
Rogaška Slatina, Slovenia, February 3–5, 2020

A new semi-supervised method improving optical flow on distant domains

Tomáš Novák, Jan Šochman, Jiří Matas
Center for Machine Perception, Department of Cybernetics, FEE CTU in Prague
Technická 2, 166 27 Prague 6, Czech Republic
{novakt34,jan.sochman}@fel.cvut.cz

Abstract. *We propose a semi-supervised approach to learning by formulating the optimization as constrained gradient descent on a loss function that includes unsupervised terms. The method is demonstrated on semi-supervised optical flow training that promotes photo-consistency and smoothness of the flow. We show that the unsupervised objective significantly improves the estimation on a distant domain while maintaining the performance on the original domain. As a result, we achieve state-of-the-art results on the Creative Flow+ dataset among CNN-based methods that did not train on any samples from the dataset.*

1. Introduction

Supervised learning of CNN methods achieves state of the art results on all major optical flow datasets. However, when presented with samples that are distant to their training set, they often produce inconsistent estimates. We find that unsupervised optical flow methods, possibly due to their less domain-dependent objective, perform better in this setting. However, they fall short of supervised methods when enough labeled training data is available. We aim to combine the performance of the supervised methods with the robustness of the unsupervised methods.

This paper presents a new semi-supervised training approach that combines supervised and unsupervised objectives. The training optimization is formulated as a constraint gradient descent that takes gradients from both losses; however, skips all unsupervised samples that lead to worse performance on the supervised samples i.e., all unsupervised gradients that have a negative dot product with the supervised gradient are omitted. The method is tested on optical flow estimation, and it is shown that it makes the network perform close to the unsupervised methods on data from a distant domain while maintaining the performance on the labeled domain.

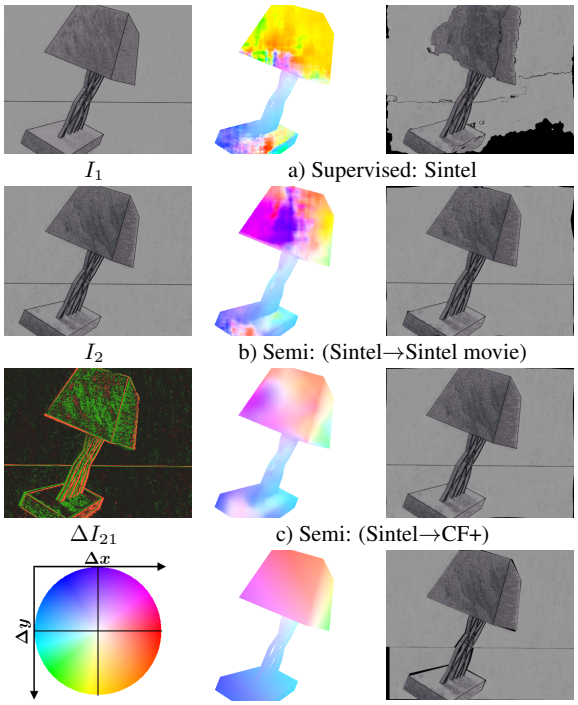


Figure 1: **Optical flow on a distant domain without/with semi-supervision**. Left: A sample pair of images I_1, I_2 from the Creative Flow+ (CF+) dataset, their difference ΔI_{21} and optical flow color coding wheel. Notice texture changes on both the object and background. Middle: Foreground optical flow for supervised and semi-supervised models with GT. Right: I_2 warp to I_1 showing geometric consistency of the flow. The Sintel fine-tuned model (a) corrupts optical flow in major parts. The proposed constrained semi-supervision on Sintel domain alone (b) improves the estimates. Further improvement is achieved by adding unlabeled CF+ samples (c).

More specifically, we demonstrate this behavior on a recently published Creative Flow+ dataset [24]. The dataset features artistic-like scenes with untextured regions or objects with changing texture. All

supervised CNN-based approaches fine-tuned on another domain (Sintel) produce highly inaccurate estimates in this setting. The method is able to retrain a supervised model mitigating this effect. We demonstrate that even without using the distant domain samples, we already get a significant performance gain. Upon introducing the images from the distant domain (with no GT), we are able to bring the error on the distant domain even lower.

The contributions of this paper are the following. First, a novel method to combine supervised and unsupervised objectives is presented. The training is formulated as constrained gradient descent on a loss function that includes terms from unsupervised training - i.e., in the optical flow estimation photoconsistency, smoothness, and forward-backward consistency.

Second, we demonstrate that when supervised training leads to abrupt estimates on a distant domain, introducing the unsupervised objective using the proposed semi-supervised method improves results on the distant domain. However, the model performance on the supervised domain does not drop. This effect is observed even without using any samples from the other domain.

Finally, we show that adding unlabeled samples from the distant domain improves the results on the distant domain even more.

2. Related work

Supervised training. *FlowNet* [7] was the first work to introduce end-to-end supervised training of optical flow. The authors proposed two CNN architectures as well as a large synthetic dataset *FlyingChairs* that was needed to train the network in a supervised fashion. This work demonstrated that neural networks are able to act as an optical flow estimator.

Many other architectures and training techniques were proposed since [12, 10, 28, 21, 27, 11] improving results on standard optical flow benchmarks [4, 8, 20] and surpassing the classical approaches.

Though our method applies to any end-to-end trainable network, we chose to build our experiments on *PWC-Net* [28] architecture, since it is a popular choice among current approaches. It combines a pyramidal approach with correlation cost volume on each level. Furthermore, the correlation is done on encoder features instead of images.

Unsupervised/self-supervised training. There is also a class of unsupervised or self-supervised techniques that aim to train the optical flow network without any ground truth, just from frame pairs (or more frames) themselves [1, 33, 23, 2]. This means they do not rely on any labeling, which in the optical flow context is nontrivial to obtain, and can thus be trained on potentially unlimited size of data.

They apply the same principles from the famous Horn–Schunck method [9] or many related [26] to create a training signal for the network. The main task is to assess the optical flow quality without any ground-truth. This is mostly done by measuring the photometric difference between the source image and the back-warped target image. Other objectives, such as smoothness or consistency between forward and backward flow, are added.

This work is further developed by adding occlusion reasoning [30, 19, 13] and so-called *data distillation* [16, 17]. Furthermore, attempts to train algorithms that combine optical flow with other tasks were done [32, 22, 14].

Fully unsupervised training is, however, not able to compete with the supervised training on the conventional optical flow datasets. They struggle with photometric deviations like occlusions, motion blur, reflections, et cetera. Even the ability to use much more training data than supervised approaches does not compensate.

Semi-supervised training. If we do omit cases of unsupervised pre-training and supervised fine-tuning, there were only a few attempts in the optical flow context to create a combination of supervised and unsupervised training.

A simple supervised and unsupervised loss combination was presented in [31, 34]. Lai et al. [15] present an approach based on a Generative Adversarial Network. The discriminator is trained to recognize the photometric difference map between the source and target image back-warped by either ground truth or estimated optical flow. Further, endpoint error loss is applied alongside the adversarial loss for all labeled data.

3. Method

The goal is to combine supervised and unsupervised training. In this section, the proposed constrained semi-supervision method is first introduced, then the loss terms used throughout the work are listed.

3.1. Semi-supervision: constraint gradient descent

At each iteration during training, the network is evaluated on one pair of frames with ground-truth (supervised sample) and N pairs without (unsupervised samples). The gradient from the supervised sample poses a reasonable (but not optimal) constraint that skips all unsupervised samples leading to worse performance on the supervised samples.

Let Θ be the network parameters. By back-propagation, the gradient

$$\mathbf{G}_s = \nabla L_{sup}(\Theta) \tag{1}$$

is computed for the supervised sample and

$$\mathbf{G}_u^n = \nabla L_{un}(\Theta) \tag{2}$$

for n-th unsupervised sample[1].

$\mathbf{G}_s$ is used as the constraining vector. Positive dot product with the constraining vector ensures that the added $\mathbf{G}_u^i$ does not have an orientation opposite to $\mathbf{G}_s$. Thus, the parameter update vector is defined as:

$$\mathbf{G} = \mathbf{G}_s + \sum_{\forall i: \mathbf{G}_u^i \cdot \mathbf{G}_s > 0} \lambda_M \mathbf{G}_u^i \tag{3}$$

Thus, by updating the parameters by $\mathbf{G}$, the value of L_{sup} linearized at Θ will not rise. However, some updates from unsupervised loss are still considered.

3.2. Loss terms

Let I_1, I_2 be two consecutive frames and $\mathbf{f}_{GT,1\to2}$ ground truth forward flow. Let $l = 1 \dots 5$ be the flow pyramid scale from the largest ¼ to the smallest ¹⁄₆₄ of the input image size. Let $\mathbf{f}^l_{1\to2}, \mathbf{f}^l_{2\to1}$ be the estimated forward and backward flow on the scale l. By I^l and $\mathbf{f}^l_{GT}$ we denote an image resp. flow down-sampled to the scale l.

Supervised loss is the standard L2 endpoint-error loss [28]:

$$L_{sup}(\mathbf{f}_{1\to2}) = \sum_{l=1}^{5} \alpha_l \sum_{\mathbf{x}\in P} \left\| \mathbf{f}^l_{1\to2}(\mathbf{x}) - \mathbf{f}^l_{GT,1\to2}(\mathbf{x}) \right\|_2 . \tag{4}$$

Data term. The data term is based on [19]; however, we drop the occlusion-awareness since it has not proven beneficial in our setting. The term is defined as

$$\begin{aligned} L^l_D(\mathbf{f}^l_{1\to2}, \mathbf{f}^l_{2\to1}) = \\ \sum_{\mathbf{x}\in P} \rho\Big(f_D\big(I^l_1(\mathbf{x}), I^l_2(\mathbf{x} + \mathbf{f}^l_{1\to2}(\mathbf{x}))\big)\Big) + \\ \rho\Big(f_D\big(I^l_2(\mathbf{x}), I^l_1(\mathbf{x} + \mathbf{f}^l_{2\to1}(\mathbf{x}))\big)\Big), \end{aligned} \tag{5}$$

where $\rho(x) = (x^2 + \epsilon^2)^\gamma$ (default $\gamma = 0.45$) is the Charbonnier penalty [26] that increases robustness to outliers. f_D measures the photometric difference between two pixels. The experiments are done with both brightness constancy constraint (per channel) [33] and the ternary census transform adjusted for loss function in [19].

Smoothness term. Second order smoothness constraint is employed as in [19], since it has been proved to be beneficial in classical flow estimation methods , [29]. To decrease over-smoothing on object edges, we combine it with edge awareness [13].

$$\begin{aligned} L^l_S(\mathbf{f}^l_{1\to2}, \mathbf{f}^l_{2\to1}) = \\ \sum_{\mathbf{x}\in P} \sum_{(\mathbf{s},\mathbf{r})\in N(x)} \sigma\big(I^l_1, \mathbf{f}^l_{1\to2}, \mathbf{s}, \mathbf{x}, \mathbf{r}\big) + \\ \sigma\big(I^l_2, \mathbf{f}^l_{2\to1}, \mathbf{s}, \mathbf{x}, \mathbf{r}\big), \end{aligned} \tag{6}$$

where $N(\mathbf{x})$ contains horizontal, vertical and both diagonal neighborhoods of $\mathbf{x}$ and σ measures the edge-aware smoothness:

$$\begin{aligned} \sigma(I, \mathbf{f}, \mathbf{s}, \mathbf{x}, \mathbf{r}) = \\ \rho\big(\mathbf{f}(\mathbf{s}) - 2\mathbf{f}(\mathbf{x}) + \mathbf{f}(\mathbf{r})\big) \cdot \\ \cdot \exp\big(-\|I(\mathbf{x}) - I(\mathbf{s})\|_2\big) \cdot \\ \cdot \exp\big(-\|I(\mathbf{x}) - I(\mathbf{r})\|_2\big). \end{aligned} \tag{7}$$

We assume $\rho(\cdot)$ computes the average over the penalties from each component.

FW-BW consistency. Adding the forward-backward consistency term also proved to help with learning the flow [19]:

$$\begin{aligned} L^l_C(\mathbf{f}^l_{1\to2}, \mathbf{f}^l_{2\to1}) = \\ \sum_{\mathbf{x}\in P} \rho\Big(\mathbf{f}^l_{1\to2}(\mathbf{x}) - \mathbf{f}^l_{2\to1}\big(\mathbf{x} + \mathbf{f}^l_{1\to2}(\mathbf{x})\big)\Big) + \\ \rho\Big(\mathbf{f}^l_{2\to1}(\mathbf{x}) - \mathbf{f}^l_{1\to2}\big(\mathbf{x} + \mathbf{f}^l_{2\to1}(\mathbf{x})\big)\Big). \end{aligned} \tag{8}$$

Unsupervised loss is defined as a weighted sum over loss terms and pyramid levels:

$$L_{un} = \sum_{l=1}^{5} \alpha_l \big(L^l_D + \lambda_S L^l_S + \lambda_C L^l_C\big) \tag{9}$$

where α_l is the pyramid scale weight.

[1] To ease the notation, we omit some obvious arguments from the loss function.

4. Experiments

This section describes the structure of experiments and the technical details. Results are discussed in the next section.

Overall, the experiments examine the domain transfer ability of supervised, unsupervised and semi-supervised training from Sintel dataset [4] to Creative Flow+ [24]. First, supervised and unsupervised models are tested and their performance is observed. Afterwards, the proposed constrained semi-supervision is put to the test in two settings - limited to samples from Sintel domain or also including unlabeled frames from CF+. For comparison, we try to pose the semi-supervision as a simple loss combination and also test a baseline supervised on both Sintel and CF+. All experiments were done with the popular PWC-Net [28] architecture.

To denote the experiments, a system of abbreviations in the format "[*training method*]: ([*datasets*])" is used. Training method is either supervised (Sup), unsupervised (Unsup) or semi-supervised (Semi). Plus sign "+" denotes the training was done on a combination of two datasets. With semi-supervised training, arrow "$\rightarrow$" separates a dataset serving as the source of supervised samples from a dataset of unsupervised samples.

4.1. Datasets

In the experiments, we use the following datasets. The letter in the bracket next to the dataset name is the abbreviation used in the experiments.

Sintel (S) [4]. To avoid complicated online evaluation, a 90-10 split of the publicly-available data to training and testing parts was created yielding 1562 train and 2×87 test samples (separately clean and final pass). In training, both *Clean* and *Final* passes are combined.

Sintel movie (Sm). All frames from the original movie [5] were extracted for unsupervised and semi-supervised training, similarly to [17]. To cope with compression artifacts, we downscaled the 4K resolution images to 1152×648. Cuts between scenes, where no optical flow exists, were avoided with PySceneDetect [6]. Moreover, too dim (typical for fade ins/outs) or too similar consecutive images were detected using pixel-wise brightness resp. brightness difference and excluded. Altogether, 9372 samples were created.

KITTI 2015 (K) [20]. Testing is done on all 200 annotated samples. Unsupervised methods train on 13K samples from the multiview extensions of KITTI'15 and '12 [8]. Frames from the annotated pairs are excluded.

Creative Flow+ (CF+) [24] is a recently introduced dataset with artistic-like scenes and ground truth optical flow. Tests are done on the 10K sample list provided by the authors. Some of the experiments also use the set of 153K *mixamo* train frames. Full resolution images (1500×1500) are used. Note that it is more meaningful to observe performance on the foreground areas since optical flow on the background is often not well defined.

4.2. Supervised training distant domain performance

First, to establish an overview of how supervised models perform on a distant domain, their performance is tested on CF+, similarly to [24]. The two pre-trained *PWC-Net* models made available by authors [28] are evaluated. One was trained on FlyingChairs (C) [7] and FlyingThings3D (T) [18] datasets, the second was fine-tuned for the Sintel [4] dataset. The experiments are denoted as *Sup: (C,T)* and *Sup: (C,T,S)*.

4.3. Unsupervised training distant domain performance

Next, we make a similar overview for the unsupervised training and its distant domain transfer ability. Two unsupervised models are trained, one with per-channel brightness constancy constraint, another with census transform data term. We name the models *Unsup [brightness]: (C,K+S)* and *Unsup [Census]: (C,K+S)* respectively.

Tests with different parameter settings and training protocols resulted in the following training procedure. To initialize the models, a pre-training phase consisting of 240K iterations on FlyingChairs dataset [12] is performed with unsup. loss L_{un}, regularization $\lambda_S = 3.0$, no forward-backward consistency ($\lambda_C = 0$) and f_D as a brightness or difference. Learning rate starts with $1e-4$ and is halved every 100K iterations. Input image size is 512×384. Fine-tuning is done on all KITTI and Sintel samples with the same setting, apart from activated consistency term $\lambda_C = 0.3$ and f_D as brightness or census difference respectively. With the brightness difference, convergence is reached after 455K iterations, 746K iterations are needed for the census difference. Images are cropped to 896×320.

4.4. Semi-supervision on single domain

The previous overview shows that unsupervised models have a better distant domain transfer ability, but suffer from low accuracy on the original domain. We therefore attempt to introduce the transfer ability of unsupervised methods to a well-performing supervised model using the proposed semi-supervision method.

The *PWC-Net* model trained (supervised) for Sintel dataset by the authors [28] is fine-tuned using the constrained semi-supervision method taking supervised samples from Sintel and unsupervised samples from Sintel movie dataset. We refer to this experiment as *Semi: (S→Sm)*.

In order to establish a control experiment, we also continue training with supervised loss only (labeled as *Sup: (C,T,S) - modif. supervision*).

In the experiments we tested multiple hyperparameter settings and ended with the following one: One supervised and six unsupervised samples are fed to the method at each iteration. We set $\lambda_M = 0.1$, f_D as per-channel brightness constancy constraint. Frames are cropped to 768×384. To warm-up the optimization, first three epochs are performed just with supervised loss and are followed by 2 semi-supervised epochs with small learning rate 1e-7. Afterwards, we perform 133K iterations with learning rate 1e-5 that is halved after 30K, 50K, 70K, 90K, 105K and 120K iterations.

4.5. Semi-supervision including distant domain

Next, the idea of the previous experiment is developed further by taking unlabeled samples from the distant domain.

The network is trained in the same way as in the previous experiment with the only difference that the unsupervised samples are taken from the training part of the CF+ dataset (i.e., frames only, no GT flow). We name the experiment as *Semi: (S→CF)*.

4.6. Unconstrained semi-supervision

To test the need for the constrained semi-supervision method, an experiment without any constraining takes place. The loss is simply defined as a combination of supervised and unsupervised terms

$$L_{comb} = L_{sup} + \lambda_U L_{un} \tag{10}$$

as e.g. in [31].

We refer to this experiment as *Uncons. semi: (S)*. Again, the experiment starts with the Sintel fine-tuned network as in previous sections. The network is trained with L_{comb} as a loss function on the Sintel dataset with $\lambda_S = 3.0, \lambda_C = 0.3$ and f_D as a brightness constancy constraint.

We test three settings of the unsupervised loss weight λ_U = 0.1, 1 and 2. In all three cases, a CF+ test error drop occurs in the first 30K iterations, however, it is followed by a rise even above the control (*Sup: (C,T,S) - modif. supervision*) experiment. At the same time, with all three λ_U settings, both terms of the loss L_{sup} and L_{un} are decreasing during training. This suggests that L_{comb} leads to an over-fitting on Sintel in unsupervised objective.

In the final results table, we state the situation before the error rise for $\lambda_U = 0.1$ and 1.

4.7. Supervised training

To establish a supervised comparison, we also fine-tune the PWC-Net model for the CF+ dataset in a supervised manner. We refer to the experiment as *Sup: (C,T,S,S+CF)*.

In each training epoch, we train on all Sintel training samples and the same number of randomly chosen CF+ samples. We train for 171K iterations starting with learning rate 1e-5 that is gradually halved.

4.8. Common technical details

This subsection describes the common technical details of the training.

In all experiments, Adam optimizer is used with default $\beta_1 = 0.9, \beta_2 = 0.999$. Batch size is four with the exception of semi-supervised experiments. As in the original PWC-Net paper [28], the pyramid weights are $\alpha_1 = 0.005$, $\alpha_2 = 0.01$, $\alpha_3 = 0.02$, $\alpha_4 = 0.08$, $\alpha_5 = 0.32$.

Census photometric difference is computed on different window sizes at each pyramid scale, from the largest to the smallest scale it is: $7 \times 7, 7 \times 7, 5 \times 5, 3 \times 3, 3 \times 3$.

For data augmentation, both common and relative (between frames in a pair) geometric transforms are used: random rotation, translation, scale, squeeze, flip, and crop. Photometric transforms are also included: random gamma, brightness, contrast, and relative color channel brightness changes.

Error measures. *EPE* refers to an average endpoint error

$$\frac{1}{\sum_{P \in S} |A(P)|} \sum_{P \in S} \sum_{x \in A(P)} \left\| \mathbf{f}^P_{1\to 2}(\mathbf{x}) - \mathbf{f}^P_{GT,1\to 2}(\mathbf{x}) \right\|_2, \tag{11}$$

Method	CF+ AEPE [px] ALL	CF+ AEPE [px] median ALL	CF+ AEPE [px] median FG	Sintel AEPE [px] Clean	Sintel AEPE [px] Final	KITTI 2015 [%] Fl-all
Horn-Schunck [9]	8.34	3.49	12.17	8.73∗	9.61∗	–
Classic+NLfast [25]	13.35	7.05	9.27	9.12∗	10.08∗	–
Brox2011 [3]	9.05	3.27	8.28	7.56∗	9.11∗	–
Sup: (C,T) [28]	66.97	41.88	22.77	2.44	3.82	34.3
Sup: (C,T,S) [28]	74.23	33.54	18.21	1.78	2.41	10.6
Sup: (C,T,S) - modif. supervision	30.44	14.73	11.30	1.69	2.22	14.7
Unsup [brightness]: (C,K+S)	10.60	4.80	7.99	5.23	6.18	40.2
Unsup [Census]: (C,K+S)	15.06	9.05	8.65	4.22	5.19	25.1
Uncons. semi: (S) $\lambda_U = 0.1$	25.76	15.19	10.63	1.79	2.19	12.2
Uncons. semi: (S) $\lambda_U = 1$	24.91	15.32	9.95	2.54	3.10	22.0
Semi: (S→Sm)	17.36	8.41	8.91	1.81	2.49	16.9
Semi: (S→CF)	7.88	3.79	6.65	1.79	2.25	18.9
Sup: (C,T,S,S+CF)	8.19	3.54	5.62	1.81	2.24	17.4

Table 1: **Main results table.** All numbers except columns marked median and Fl-all, are mean endpoint errors over all test samples. Fl-all denotes outlier ratio (>3px and >5% EPE), median is computed across individual sample average EPEs. Dataset abbreviations: C: Flying Chairs [12], T: FlyingThings3D [18], S: Sintel [4], Sm: Sintel movie, CF: Creative Flow+[24], K: KITTI unlabeled multiview extension [8, 18]. For classical methods, we list the results from [24]. Results marked with a star (*) come from the official test benchmark.

	Creative Flow+ AEPE [px]									
		median		Style, FG				Speeds, FG		
	ALL	ALL	FG	flat	toon	tex	stylit	<1%	1-3%	>3%
Sup: (C,T) [28]	66.97	41.88	22.77	41.18	10.86	16.09	23.67	23.17	17.84	32.73
Sup: (C,T,S) [28]	74.23	33.54	18.21	24.71	7.03	17.46	21.77	17.50	15.18	30.94
Sup: (C,T,S) - modif. supervision	30.44	14.73	11.30	7.79	6.42	13.62	14.76	8.48	12.36	28.22
Unsup [brightness]: (C,K+S)	10.60	4.80	7.99	7.67	5.90	9.01	8.85	4.90	9.54	25.51
Unsup [Census]: (C,K+S)	15.06	9.05	8.65	7.83	5.93	9.94	9.99	5.47	9.81	27.79
Uncons. semi: (S) $\lambda_U = 0.1$	25.76	15.19	10.63	11.14	5.99	12.35	12.19	8.26	10.99	26.19
Uncons. semi: (S) $\lambda_U = 1$	24.91	15.32	9.95	11.24	5.72	11.75	10.86	7.76	10.24	24.5
Semi: (S→Sm)	17.36	8.41	8.91	7.20	5.66	10.66	10.79	5.95	10.18	26.23
Semi: (S→CF)	7.88	3.79	6.65	6.85	5.32	8.94	6.19	3.47	8.68	23.58
Sup: (C,T,S,S+CF)	8.19	3.54	5.62	5.84	4.61	9.10	4.36	2.94	7.21	20.20

Table 2: **Detailed results of the presented methods on CF+.** We list the same metrics as in the original paper [24]. All numbers except column marked median, are average endpoint errors. Median is computed across individual sample average EPEs. Performance is broken down into All (full frame) and FG (foreground) as well as by style and speed (<1% ground-truth optical flow length less than 1% of the frame size i.e. 15 px, 1-3% between 15 and 45 px, >3% over 45 px).

where S is a set of test samples, $A(P)$ defines the area of interest (whole image, foreground pixels etc.) and $\mathbf{f}^P_{1\to 2}$ is the flow estimated on sample P scaled to original image size.

Fl-all is an error measure proposed for the KITTI'15 dataset, where there is an uncertainty in optical flow measurements. It is defined as the percentage of optical flow outliers i.e., flow end-point error is $> 3px$ and $> 5\%$ of GT flow.

5. Results and discussion

This section discusses the results of the experiments described in the previous section. The results of the experiments are listed in Table 1, qualitative assessment is presented in Figure 2. Extended evaluation on the Creative Flow+ dataset is shown in Table 2.

Supervised training. First, we observe that the supervised methods fail on the CF+ dataset, see *Sup: (C,T)* and *Sup: (C,T,S)* in Table 1. Figure 2 indicates abruptly outlying estimates on constant intensity regions. Problems also occur on object texture changes. We get slightly different results to [24], possibly due to a different framework, however, the conclusion is the same.

Unsupervised training. With unsupervised training, the models do not suffer from the distant domain transfer issues - the performance on the CF+ dataset is significantly better, as shown in Table 1, *Unsup [brightness]: (C,K+S)* and *Unsup [Cen-*

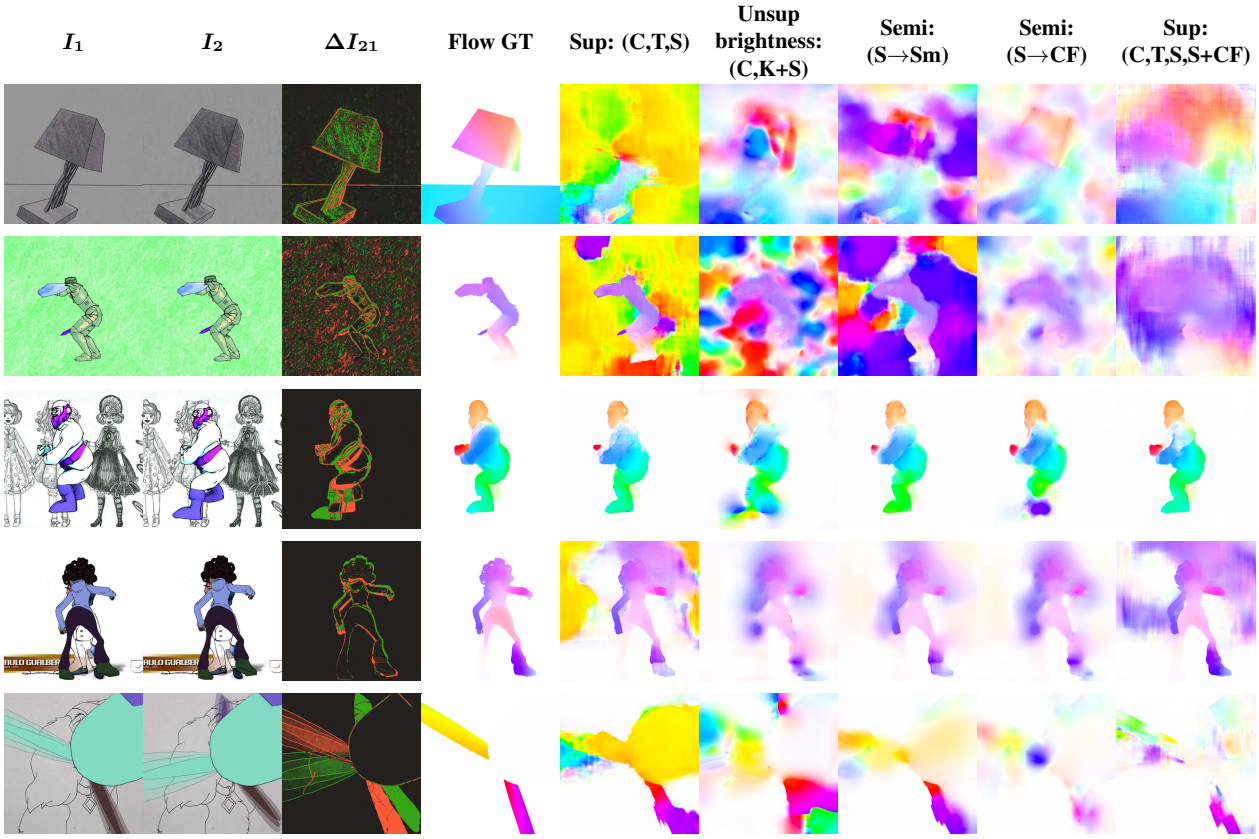


Figure 2: **Qualitative assessment.** Input images (first two columns) with a color coded difference visualization (third column); the ground truth flow and flow estimates for selected methods (following columns).

sus]: (C,K+S). Figure 2 shows that the estimated flow field is smoother, with no abrupt outliers. However, the test errors on Sintel and KITTI dataset stay far behind the supervised models.

We hypothesize that although the unsupervised objective is unable to properly handle the effects of occlusions, motion blur, local ambiguities, etc., yet, it is more universal than training for a supervised objective on a single domain. Therefore, we expect it to perform better on a distant domain.

Semi-supervision on single domain. Semi-supervision attempts to combine the observed distant domain transfer ability of unsupervised models with the accuracy of supervised models on Sintel and KITTI.

Table 1, *Semi: (S→Sm)*, shows that constrained semi-supervision training significantly drops the test error on CF+ while the error on Sintel changes just slightly. Curiously, this is done without introducing any CF+ samples.

We attribute the increased CF+ accuracy partially to our way of supervised training, which seems to decrease the error on CF+ as shown by our control experiment *Sup: (C,T,S) - modif. supervision*. It is most likely caused by differences in augmentations, probably skipping additive white noise in our setting.

However, semi-supervision leads to a significant decrease, suggesting that adding the unsupervised loss with the proposed method makes the model perform closer to unsupervised methods on a distant domain with only minor changes on the Sintel domain.

Semi-supervision including distant domain. When the semi-supervised model is explicitly presented with the samples from CF+, the error on this distant domain drops significantly to the level of the unsupervised methods (Table 1 - *Semi: (S→CF)*). Note that the error is also significantly below the semi-supervision on a single domain. Again, the error on Sintel stays virtually the same.

We hypothesize that since the images from the other domain are presented, the network starts to recognize it and optimize the unsupervised criterion specifically on these samples. However, the super-

vised constraint prevented to apply the same criterion on the supervised samples.

Unconstrained semi-supervision. Unconstrained semi-supervision tested the need for the proposed constrained semi-supervision method by formulating the training as a simple linear combination of supervised and unsupervised losses.

As Table 1 *Uncons. semi: (S)* shows, the performance on CF+ is similar for both λ_U settings, especially on the foreground regions. On Sintel and KITTI, low $\lambda_U = 0.1$ preserves the accuracy of the initial model; however, a significant error rise is observed with higher $\lambda_U = 1$.

The observations correspond to the expectations - with small unsupervised term weight, the training is not able to introduce the unsupervised objective to the model. When we attempt to promote it more with higher λ_U, the accuracy on the supervised domain is lost.

Supervised CF training. Supervised training on CF+ is able to improve the performance on the dataset while maintaining the accuracy on Sintel (see Table 1, *Sup: (C,T,S,S+CF)*). Evaluated on the whole frames, it does not surpass constrained semi-supervision. However, as it was already mentioned, the background flow is often not well defined; thus, this metric is not as relevant.

The performance margin to a constrained semi-supervision on the foreground areas is not as large as e.g., the margin between supervised and unsupervised methods on Sintel, suggesting that CF+ features complicated scenes that are hard to solve even with supervision.

6. Conclusion

In this paper, we propose a semi-supervision method by constraining the unsupervised update by the supervised gradient.

The experiments show that the proposed constrained semi-supervision method leads to a better performance in distant domain transfer while maintaining the performance on the supervised (i.e., Sintel) domain. Some improvement is already observed when introducing the unsupervised objective only on a single domain, even better results are achieved when the unlabeled samples from the distant domain are included. Our control experiment was not able to prove that the same effect is achieved by an unconstrained formulation.

As it could be foreseen, supervised training on the distant domain improves the results even further, but the margin is not as significant as expected.

References

[1] A. Ahmadi and I. Patras. Unsupervised convolutional neural networks for motion estimation. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1629–1633, Sept. 2016. 2

[2] S. Alletto, D. Abati, S. Calderara, R. Cucchiara, and L. Rigazio. TransFlow: Unsupervised Motion Flow by Joint Geometric and Pixel-level Estimation. *arXiv:1706.00322 [cs]*, June 2017. arXiv: 1706.00322. 2

[3] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):500–513, 2010. 6

[4] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A Naturalistic Open Source Movie for Optical Flow Evaluation. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision – ECCV 2012*, Lecture Notes in Computer Science, pages 611–625. Springer Berlin Heidelberg, 2012. 2, 4, 6

[5] C. Levy (Director). Sintel. Blender Institute, 2010. 4

[6] B. Castellano. Breakthrough/PySceneDetect, Nov. 2019. [Online] https://github.com/Breakthrough/PySceneDetect. 4

[7] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox. FlowNet: Learning Optical Flow with Convolutional Networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 2, 4

[8] 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. 2, 4, 6

[9] B. K. Horn and B. G. Schunck. Determining Optical Flow. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 1980. 2, 6

[10] T.-W. Hui, X. Tang, and C. Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8981–8989, 2018. 2

[11] J. Hur and S. Roth. Iterative Residual Refinement for Joint Optical Flow and Occlusion Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5754–5763, 2019. 2

[12] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. FlowNet 2.0: Evolution of Optical

Flow Estimation with Deep Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 4, 6

[13] J. Janai, F. Guney, A. Ranjan, M. Black, and A. Geiger. Unsupervised Learning of Multi-Frame Optical Flow with Occlusions. pages 690–706, 2018. 2, 3

[14] H.-Y. Lai, Y.-H. Tsai, and W.-C. Chiu. Bridging Stereo Matching and Optical Flow via Spatiotemporal Correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1890–1899, 2019. 2

[15] W.-S. Lai, J.-B. Huang, and M.-H. Yang. Semi-Supervised Learning for Optical Flow with Generative Adversarial Networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 354–364. Curran Associates, Inc., 2017. 2

[16] P. Liu, I. King, M. R. Lyu, and J. Xu. DDFlow: Learning Optical Flow with Unlabeled Data Distillation. *arXiv:1902.09145 [cs]*, Feb. 2019. arXiv: 1902.09145. 2

[17] P. Liu, M. Lyu, I. King, and J. Xu. SelFlow: Self-Supervised Learning of Optical Flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4571–4580, 2019. 2, 4

[18] N. Mayer, E. Ilg, P. Hausser, 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*, pages 4040–4048, 2016. 4, 6

[19] S. Meister, J. Hur, and S. Roth. UnFlow: Unsupervised Learning of Optical Flow With a Bidirectional Census Loss. In *Thirty-Second AAAI Conference on Artificial Intelligence*, Apr. 2018. 2, 3

[20] M. Menze and A. Geiger. Object Scene Flow for Autonomous Vehicles. pages 3061–3070, 2015. 2, 4

[21] M. Neoral, J. Šochman, and J. Matas. Continual Occlusion and Optical Flow Estimation. In *Asian Conference on Computer Vision*, pages 159–174. Springer, 2018. 2

[22] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black. Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation. *arXiv:1805.09806 [cs]*, May 2018. arXiv: 1805.09806. 2

[23] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha. Unsupervised Deep Learning for Optical Flow Estimation. In *Thirty-First AAAI Conference on Artificial Intelligence*, Feb. 2017. 2

[24] M. Shugrina, Z. Liang, A. Kar, J. Li, A. Singh, K. Singh, and S. Fidler. Creative Flow+ Dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 4, 6

[25] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2432–2439, June 2010. ISSN: 1063-6919. 6

[26] D. Sun, S. Roth, and M. J. Black. A Quantitative Analysis of Current Practices in Optical Flow Estimation and the Principles Behind Them. *International Journal of Computer Vision*, 106(2):115–137, Jan. 2014. 2, 3

[27] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. Models Matter, So Does Training: An Empirical Study of CNNs for Optical Flow Estimation. *arXiv:1809.05571 [cs]*, Sept. 2018. arXiv: 1809.05571. 2

[28] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. pages 8934–8943, 2018. 2, 3, 4, 5, 6

[29] W. Trobin, T. Pock, D. Cremers, and H. Bischof. An Unbiased Second-Order Prior for High-Accuracy Motion Estimation. In *DAGM-Symposium*, 2008. 3

[30] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, and W. Xu. Occlusion Aware Unsupervised Learning of Optical Flow. pages 4884–4893, 2018. 2

[31] X. Xiang, M. Zhai, R. Zhang, Y. Qiao, and A. El Saddik. Deep Optical Flow Supervised Learning With Prior Assumptions. *IEEE Access*, 6:43222–43232, 2018. 2, 5

[32] Z. Yin and J. Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. pages 1983–1992, 2018. 2

[33] J. J. Yu, A. W. Harley, and K. G. Derpanis. Back to Basics: Unsupervised Learning of Optical Flow via Brightness Constancy and Motion Smoothness. In G. Hua and H. Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, Lecture Notes in Computer Science, pages 3–10. Springer International Publishing, 2016. 2, 3

[34] M. Zhai, X. Xiang, R. Zhang, N. Lv, and A. El Saddik. Learning Optical Flow Using Deep Dilated Residual Networks. *IEEE Access*, 7:22566–22578, 2019. 2

25^{th} Computer Vision Winter Workshop
Domen Tabernik, Alan Lukežič, Klemen Grm (eds.)
Rogaška Slatina, Slovenia, February 3–5, 2020

USACv20: robust essential, fundamental and homography matrix estimation

Maksym Ivashechkin[1], Daniel Barath[12], and Jiri Matas[1]
[1] Centre for Machine Perception, Czech Technical University in Prague, Czech Republic
[2] Machine Perception Research Laboratory, MTA SZTAKI, Budapest, Hungary
{ivashmak, matas}@cmp.felk.cvut.cz barath.daniel@sztaki.mta.hu

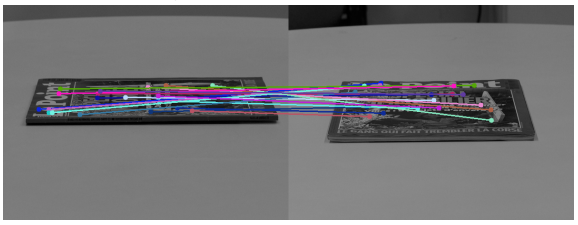
Abstract. *We review the most recent RANSAC-like hypothesize-and-verify robust estimators. The best performing ones are combined to create a state-of-the-art version of the Universal Sample Consensus (USAC) algorithm. A recent objective is to implement a modular and optimized framework, making future RANSAC modules easy to be included. The proposed method, USACv20, is tested on eight publicly available real-world datasets, estimating homographies, fundamental and essential matrices. On average, USACv20 leads to the most geometrically accurate models and it is the fastest in comparison to the state-of-the-art robust estimators. All reported properties improved performance of original USAC algorithm significantly. The pipeline will be made available after publication.*

1. Introduction

The RANdom SAmple Consensus (RANSAC) algorithm [12] has been one of the most widely used robust estimators in computer vision. RANSAC and many of its variants have been successfully applied to a wide range of vision tasks, for instance, short baseline stereo [37, 39], motion segmentation [37], detection of geometric primitives [31], wide baseline matching [27, 21, 22], in structure-from-motion [1, 40, 30] (SfM) or simultaneous localization and mapping [11, 23] (SLAM) pipelines, image mosaicing [14], and to perform [41] or initialize multi-model fitting [16, 26].

In this paper, we review some of the most recent RANSAC modifications, combine them together and propose a state-of-the-art variant of the Universal Sample Consensus [28] (USAC) algorithm. Also, an important objective is to make the implemented modular and optimized C++ framework publicly available, therefore, making future RANSAC modules easy to be combined with the proposed USACv20.

(a) `Community Photo Collection dataset` [40].



(b) `ExtremeView dataset` [22].

(c) `Tanks and Temples dataset` [17].

(d) `Piccadilly dataset` [40].

Figure 1. Example image pairs where USACv20 has lower error to ground truth inliers than OpenCV RANSAC and USAC [28] estimators.

In short, the RANSAC approach repeatedly creates minimal sets of randomly selected points and fits a model to them, e.g., a circle to three 2D points

or a homography to four 2D point correspondences. Next, the quality of the estimated model is measured, for example, by the cardinality of its support, i.e., the number of data points closer than a manually set inlier-outlier threshold. Finally, the model with the highest score, polished, e.g., by least squares fitting of all inliers, is returned.

Scoring function. Many modifications have been proposed since the publication of RANSAC, improving the components of the algorithm. For instance, in MAPSAC [36], the robust estimation is formulated as a process that estimates both the parameters of the data distribution and the quality of the model in terms of maximum a posteriori. MLESAC [38] estimates the model quality by a maximum likelihood process with all its beneficial properties, albeit under certain assumptions about data distributions. In practice, MLESAC results are often superior to the inlier counting of plain RANSAC, and are less sensitive to the inlier-outlier threshold defined manually.

Local Optimization. Observing that RANSAC requires in practice more samples than theory predicts, Chum et al. [8, 18] identified a problem that not all all-inlier samples are "good", i.e., lead to a model accurate enough to distinguish all inliers, e.g., due to poor conditioning of the selected random all-inlier sample. They addressed the problem by introducing the locally optimized RANSAC that augments the original approach with a local optimization step applied to the *so-far-the-best* model. This approach had been further improved in Graph-Cut RANSAC [3] considering the fact that real-world data often form spatially coherent structures. Graph-Cut RANSAC exploits the proximity of the points in the local optimization step, leading to results superior to LO-RANSAC in terms of geometric accuracy.

Sampling Strategies. Samplers NAPSAC [24] and PROSAC [6] modify the RANSAC sampling strategy to increase the probability of selecting an all-inlier sample early. PROSAC exploits an a priori predicted inlier probability rank of the points and starts the sampling with the most promising ones. PROSAC and other RANSAC-like samplers treat models without considering that inlier points often are in the proximity of each other. This approach is effective when finding a global model with inliers sparsely distributed in the scene, for instance, the rigid motion induced by changing the viewpoint in two-view matching. However, as it is often the case in real-world data, if the model is localized with inlier points close to each other, robust estimation can be significantly sped up by exploiting this in the sampling. NAPSAC assumes that inliers are spatially coherent. It draws samples from a hyper-sphere centered at the first, randomly selected, point. If this point is an inlier, the rest of the points sampled in its proximity are more likely to be inliers than the points outside the ball. Progressive NAPSAC [2] was proposed to combine NAPSAC-like localized sampling with PROSAC by drawing minimal samples from gradually growing neighborhoods.

Optimizing Model Verification. One of the most successful improvement for speeding up the verification is the optimal randomized model verification strategy [20, 7] (WaldSAC) based on Wald's theory of sequential decision making. When the level of outlier contamination is known a priori, the WaldSAC strategy is provably optimal. In practice, however, inlier ratios have to be estimated during the evaluation process and WaldSAC adjusted to the current so-far-the-best model. The performance of the SPRT test is not significantly affected by the imperfect estimation of these parameters.

Termination criterion. There were a number of different termination criteria proposed for RANSAC-like hypothesize-and-verify methods. The original criterion is based on the assumption that the inliers are noise-free. The number of iterations required is calculated from the inlier ratio and the number of points needed for the model estimation. This criterion was then relaxed by Progressive NAPSAC [2] by terminating if the probability of finding a model which has significantly more inliers than the previous best falls below a threshold. In [6], another criterion was proposed. The PROSAC algorithm terminates if the number of inliers satisfies the following conditions: (i) non-randomness – the probability that i^* out of n data points are by chance inliers to an arbitrary incorrect model is smaller than a threshold; (ii) maximality – the probability that a solution with more than i^* inliers exists and was not found after k samples is smaller than μ_0.

2. USACv20

The structure of the proposed framework is summarized in Algorithm 1. The standard RANSAC loop is executed between lines 2: and 27:. The implementation is modular, and each step of the algorithm allows a range of options.

In the version of USACv20 evaluated in the paper,

Algorithm 1 USACv20.

Input: $\mathcal{P}$ – points; η – confidence, t – maximum iterations, $\mathcal{T}$ – termination, ...
Output: $\hat{\theta}^*$ – the best found model

1: $\varepsilon^* \leftarrow \infty$
2: **while** ! *terminate* $(\mathcal{T}, \eta, t)$ **do**
3: $\quad \mathcal{S} \leftarrow$ *sampling* $(\mathcal{P})$
4: $\quad$ **if** ! *validate_sample* $(\mathcal{S})$ **then**
5: $\quad\quad$ ***continue***
6: $\quad \hat{\Theta} \leftarrow$ *estimate* $(\mathcal{S})$
7: $\quad$ **for** $\hat{\theta} \in \hat{\Theta}$ **do**
8: $\quad\quad$ **if** ! *validate_model* $(\hat{\theta}, \mathcal{S})$ **then**
9: $\quad\quad\quad$ ***continue***
10: $\quad\quad$ **if** ! *preemptive_verification*$(\hat{\theta})$ **then**
11: $\quad\quad\quad$ ***continue***
12: $\quad\quad \varepsilon \leftarrow$ *model_quality*$(\hat{\theta})$
13: $\quad\quad$ **if** $\varepsilon^* \prec \varepsilon$ **then**
14: $\quad\quad\quad \hat{\theta}' \leftarrow$ *recover_if_degenerate* $(\hat{\theta}, \mathcal{S})$
15: $\quad\quad\quad$ **if** $\hat{\theta}' =$ NULL **then**
16: $\quad\quad\quad\quad$ ***continue***
17: $\quad\quad\quad \varepsilon' \leftarrow$ *model_quality* $(\hat{\theta}')$
18: $\quad\quad\quad$ **if** $\varepsilon^* \prec \varepsilon'$ **then**
19: $\quad\quad\quad\quad \hat{\theta}_{LO} \leftarrow$ *local_optimization* $(\hat{\theta}')$
20: $\quad\quad\quad\quad \hat{\theta}_{LO} \leftarrow$ *recover* $(\hat{\theta}_{LO})$
21: $\quad\quad\quad\quad$ **if** $\hat{\theta}_{LO} \neq$ NULL **then**
22: $\quad\quad\quad\quad\quad \varepsilon_{LO} \leftarrow$ *model_quality* $(\hat{\theta}_{LO})$
23: $\quad\quad\quad\quad\quad$ **if** $\varepsilon' \prec \varepsilon_{LO}$ **then**
24: $\quad\quad\quad\quad\quad\quad \hat{\theta}', \varepsilon' \leftarrow \hat{\theta}_{LO}, \varepsilon_{LO}$
25: $\quad\quad\quad\quad \hat{\theta}^*, \varepsilon^* \leftarrow \hat{\theta}', \varepsilon'$
26: $\quad\quad\quad\quad \mathcal{T} \leftarrow$ *update* $(\hat{\theta}^*, \mathcal{I}_{\hat{\theta}^*})$
27: $\hat{\theta}^* \leftarrow$ *polish_final* $(\hat{\theta}^*)$

the chosen sampling method is Progressive NAPSAC, alg. 1, line 3. Other samplers are described in section 2.2. The pre-emptive model verification is SPRT, alg. 1, line 10. Other options could be none verification or $T_{d,d}$ test, see section 2.4. The termination condition, alg. 1, line 2 is combination of SPRT and P-NAPSAC since P-NAPSAC and SPRT are used. The measured quality of model is MSAC (sum of truncated errors), alg. 1, line 12. The MSAC quality could be also replaced by MLESAC or MAGSAC quality, see section 2.3. The local optimization step is done in the line 19 by graph-cut-based local optimization. Other modifications of local optimization are in the section 2.1.

The degeneracy of model (e.g., validation of epipolar oriented constraint [9]) is done in the alg. 1, line 8 and after finding so-far-the-best model in the line 14 (e.g., planarity of fundamental matrix [10]. In the end the output model is polished by least squares fitting on all inliers, alg. 1, line 27.

2.1. Local optimization

The options for local optimization are listed below. The one chosen in USACv20 is written in bold.

LO-RANSAC [8]	Refine each so-far-the-best model by an inner RANSAC.
FLO-RANSAC [18]	Improvement of LO-RANSAC.
Graph-Cut RANSAC [3]	Spatial coherence is considered when doing the inner RANSAC.
σ-consensus [4]	A part of the MAGSAC algorithm marginalizing over the noise-scale.

We chose Graph-Cut RANSAC since it is more accurate than LO-RANSAC and FLO-RANSAC and significantly faster than the σ-consensus which requires a number of least-squares fittings.

2.2. Sampling

The possible options for sampling are listed below. The one chosen in USACv20 is written in bold.

Uniform [12]	The default option.
NAPSAC [24]	Selecting the first points and, then, local sampling from its neighborhood.
PROSAC [6]	Sampling from the most promising samples first and progressively blending to the uniform sampler of RANSAC.
P-NAPSAC [2]	Combination of PROSAC and NAPSAC sampling from gradually growing neighborhoods.

We chose P-NAPSAC since it leads to finding a good-enough sample earlier than PROSAC when the sought model is localized. In case of having a global model, e.g. the background motion in two images, it is found not noticeably later than by PROSAC due to progressively blending into global sampling.

2.3. Quality

The options for the model quality calculation are listed below. The one chosen is written in bold.

RANSAC [12]	The number of inliers.
MSAC [38]	The sum of truncated errors.
MLESAC [38]	Likelihood of the model.
LMedS [29]	The least median of errors.
MAGSAC [4]	Sum of errors marginalized over the noise-scale.

We chose MSAC quality calculation since it is always more accurate than that of RANSAC; it does not require expensive calculations like MLESAC or MAGSAC; and does not need to know the outlier ratio a priori as LMedS does.

2.4. Pre-emptive verification

The options for the pre-emptive verification are listed below. The one chosen is written in bold.

$T_{d,d}$ [7]	If d out d points are inliers then model is good.
SPRT [7]	Verify model by sequential decision making based on Wald's theory.

The $T_{d,d}$ test can make many false-negatives (rejecting good models) when the inlier ratio is low. Therefore we chose SPRT verification.

2.5. Termination criterion

The options for the termination criterion are listed below. The one chosen is written in bold.

Standard [12]	Terminates if the probability of finding a model with more inliers than the previous best falls below a threshold with some confidence.
PROSAC [6]	Terminates when the maximality and non-randomness criteria are satisfied.
SPRT [7]	Termination based on a sequence of subsequent model validations.
P-NAPSAC [2]	The standard RANSAC criterion relaxed by requiring the new model to select significantly more inliers than the previous best.
MAGSAC [4]	Marginalization of the standard RANSAC criterion over the noise-scale σ.

The termination of SPRT and P-NAPSAC depends on different properties of the robust procedure. P-NAPSAC stops when the relaxed RANSAC criterion is triggered, meaning that the probability of finding a significantly better model than the previous best falls below a threshold. The SPRT criterion is triggered by the number of subsequent model verification sequences made. These two techniques can straightforwardly be combined. Thus, we stop when at least one of them is triggered.

2.6. Degeneracy

USACv20 framework includes different tests on degeneracy. DEGENSAC [10] is about detecting when the majority of the drawn sample originates from the same 3D plane. For fundamental and essential matrix estimation oriented epipolar constraint [9] is evaluated as well. For homography estimation the verification of samples by its orientation is included.

2.7. Other features

For PROSAC or Progressive NAPSAC, exploiting an a priori known quality of the input data points makes the finding of a good-enough model significantly earlier than by other samplers. However, such prior information usually is unknown, degrading PROSAC to being the entirely uniform sampler of RANSAC. In the proposed USACv20 framework, when such quality function is not available, we use the density of the points as the quality function. This reflects the fact real-world data often forms spatially coherent structures and, thus, good correspondences tend to be close.

The spatial coherence of points plays important role in the estimation. For instance, it is exploited in the graph-cut-based local optimization or in P-NAPSAC sampler. Consequently, the neighborhood graph must be computed. The efficient way to do this is using a multi-layer grid described in [2]. In USACv20 such neighborhood estimation is implemented and used in the experiments.

3. Experimental results

We compared the proposed USACv20 to three robust estimators, i.e., USAC [28] [1], GC-RANSAC [3] and the RANSAC implementation of OpenCV. The applied USACv20 consists of SPRT verification, DEGENSAC [10], P-NAPSAC sampler and the local

[1] http://wwwx.cs.unc.edu/~rraguram/usac/USAC-1.0.zip

optimization of GC-RANSAC. USAC estimator [28] includes SPRT verification, DEGENSAC, PROSAC sampler and the local optimization of the original LO-RANSAC. All estimators were tested using the same number of maximum iterations (10,000 for $\mathbf{H}$ and 1,000 for $\mathbf{F}$, $\mathbf{E}$ estimation) and confidence equals to 99%.

Fundamental matrix estimation was evaluated on the benchmark of [5]. The [5] benchmark includes: (1) the `TUM` dataset [35] consisting of videos of indoor scenes. Each video is of resolution 640×480. (2) The `KITTI` dataset [13] consists of consecutive frames of a camera mounted to a moving vehicle. The images are of resolution 1226×370. Both in `KITTI` and `TUM`, the image pairs are short-baseline. (3) The `Tanks and Temples` (`T&T`) dataset [17] provides images of real-world objects for image-based reconstruction and, thus, contains mostly wide-baseline pairs. The images are of size from 1080×1920 up to 1080×2048. (4) The `Community Photo Collection` (`CPC`) dataset [40] contains images of various sizes of landmarks collected from Flickr. In the benchmark, 1 000 image pairs are selected randomly from each dataset. SIFT [19] correspondences are detected, filtered by the standard SNN ratio test [19] and, finally, used for estimating the epipolar geometry.

The compared methods are USAC [28], GC-RANSAC [3], the RANSAC [12] implementation in OpenCV and the proposed USACv20. For all methods, the confidence was set to 0.99. For each method and problem, we chose the threshold maximizing the accuracy. The used error metric is Sampson distance. All methods were in C++.

The first four blocks of Table 1 report the median errors (ϵ_{med}, in pixels), the failure rates (f; in percentage) and processing times (t; in milliseconds) on the datasets used for fundamental matrix estimation. We report the median values to avoid being affected by the failures – which are also shown. A test is considered failure if the error of the estimated model is bigger than the 1% of the image diagonal. The best values are shown in red, the second best ones are in blue. It can be seen that *USACv20 leads to the lowest errors* on all datasets. Its failure ratio and processing time are always the lowest or the second lowest.

In Figures 4,5,7,6, the cumulative distribution functions (CDF) of the Sampson errors (left plot; horizontal axis) and processing times (right; in milliseconds) of the estimated fundamental matrices are shown. Being accurate or fast is interpreted by a curve close to the top. It can be seen that USACv20 is always amongst the top performing methods in terms of geometric accuracy. The only methods which are faster than USACv20 on any dataset, are significantly less accurate on that particular dataset. For instance, on `Tanks and Temples` (Fig. 7), USACv20 is the second fastest method (right plot) right after USAC which is the least accurate one (left).

For homography estimation, we downloaded `homogr` (12 pairs) and `EVD` (15 pairs) datasets [18]. They consist of image pairs of different sizes from 329×278 up to 1712×1712 with point correspondences and inliers selected manually. The `homogr` dataset contains mostly short baseline stereo images, whilst the pairs of `EVD` undergo an extreme view change, i.e., wide baseline or extreme zoom. In both datasets, the correspondences are assigned manually to one of the two classes, i.e., outlier or inlier of the most dominant homography present in the scene. All algorithms applied the normalized four-point algorithm [15] for homography estimation and were repeated 100 times on each image pair. To measure the quality of the estimated homographies, we used the RMSE re-projection error calculated from the provided ground truth inliers.

The fifth and sixth blocks of Table 1 report the median errors (ϵ_{med}, in pixels), the failure rates (f; in percentage) and processing times (t; in milliseconds) on the datasets used for homography estimation. We report the median values to avoid being affected by the failures – which are also shown. A test is considered failure if the error of the estimated model is bigger than the 1% of the image diagonal. The best values are shown in red, the second best ones are in blue. It can be seen that USACv20 is the most accurate method on the `Homogr` dataset and the second most accurate one on `ExtremeView`. Its failure ratio and processing time are always the lowest or the second lowest.

In Figures 2,3, the cumulative distribution functions (CDF) of the re-projection errors (left plot; horizontal axis) and processing times (right; in milliseconds) of the estimated homographies are shown. Being accurate or fast is interpreted by a curve close to the top. It can be seen that USACv20 is always amongst the most accurate methods. Its processing time is the second best on `Homogr` dataset by a margin of 2-3 ms. On `ExtremeView`, USACv20 is significantly faster than all the competitor robust esti-

mators.

For essential matrix estimation, we downloaded the `Strecha` (1359 pairs) dataset and the `Piccadilly` scene from the `1DSfM` dataset[2] [40]. For the images of `Strecha`, both the intrinsic camera parameters and the ground truth poses are provided. First, we detected SIFT correspondences [19], filtered them by the standard SNN ratio test [19] The intrinsic parameters were used for normalizing the point coordinates. The ground truth pose was used for validation purposes selecting the ground truth inlier correspondences from the detected ones. These selected inliers were then used for measuring the error of the estimated essential matrices. The `1DSfM` dataset consists of 13 scenes of landmarks with photos of varying sizes collected from the internet. It provides 2-view matches with epipolar geometries and a reference reconstruction from incremental SfM (computed with Bundler [32, 33]) for measuring error. We iterated through the provided 2-view matches, detected SIFT correspondences [19], filtered them by the standard SNN ratio test [19], and calculated the ground truth relative pose from the reference reconstruction made by the Bundler algorithm. Note that all image pairs were excluded from the evaluation where fewer than 20 correspondences were found. For the evaluation, we chose the largest scene, i.e. Piccadilly, consisting of $7,351$ images.

The last two blocks of Table 1 report the median errors (ϵ_{med}, in pixels), the failure rates (f; in percentage) and processing times (t; in milliseconds) on the datasets used for essential matrix estimation. The best values are shown in red, the second best ones are in blue. It can be seen that USACv20 is the most accurate method on both datasets while being the second fastest one.

In Figures 8,9, the cumulative distribution functions (CDF) of the SGD errors (left plot; horizontal axis) and processing times (right; in milliseconds) of the estimated homographies are shown. Being accurate or fast is interpreted by a curve close to the top. It can be seen that USACv20 is always amongst the most accurate methods while being marginally slower than USAC. However, since USAC does not have essential matrix solver so only fundamental matrices were estimated and then converted to essential matrix using ground truth intrinsic matrices. In general, 5-points algorithm [25] is much slower than

[2] http://www.cs.cornell.edu/projects/1dsfm/

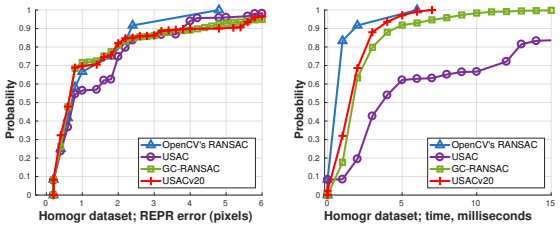


Figure 2. The cumulative distribution functions (CDF) of the Re-projection errors (left plot; horizontal axis) and processing times (right; milliseconds) of the estimated homographies on the `Homogr` dataset.

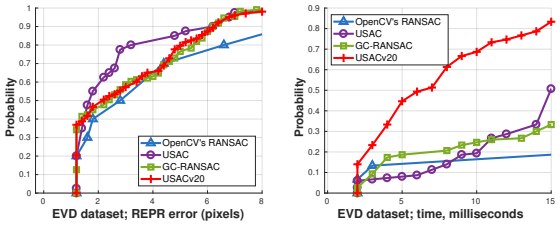


Figure 3. The cumulative distribution functions (CDF) of the Re-projection errors (left plot; horizontal axis) and processing times (right; milliseconds) of the estimated homographies on the `ExtremeView` dataset.

7-points algorithm which was used for F-estimation and number of output models for E ranges from 0 to 10 while number of estimated F matrices is at most 3; consequently all of these makes USAC framework faster.

In summary, the proposed USACv20 is, on all but one dataset (i.e., `ExtremeView`), more accurate than the original USAC algorithm while, usually, being faster. Even though USAC is more accurate on `ExtremeView`, it fails twice as often as USACv20.

The values reported in Table 1 are summarized in Table 2. It can be seen that the proposed algorithm is, on average, more accurate and faster than the compared state-of-the-art robust estimators. Its failure rate is the second best right behind GC-RANSAC.

4. Conclusion

In this paper, we reviewed some of the most recent RANSAC variants, combined them together and proposed a state-of-the-art variant, i.e. USACv20, of the Universal Sample Consensus [28] (USAC) algorithm. USACv20 is tested on 8 datasets, estimating homographies, fundamental and essential matrices. On average, it leads to the most geometrically accurate models and it is fastest compared to USAC, OpenCV's RANSAC and Graph-

	Fundamental matrix												Homography						Essential matrix					
	KITTI [13]			TUM [35]			T&T [17]			CPC [40]			Homogr [18]			EVD [18]			Strecha [34]			Piccadily [40]		
	ϵ_{med}	t	$f(\%)$	ϵ_{med}	t	f	ϵ_{med}	t	f	ϵ_{med}	t	f	ϵ_{med}	t	f	ϵ_{med}	t	f	ϵ_{med}	t	f	ϵ_{med}	t	f
USACv20	**0.2**	**1.9**	**0.2**	**0.3**	**2.1**	**8.4**	**0.6**	**5.6**	**12.9**	**0.5**	**5.3**	**43.0**	**0.7**	**2.2**	**0.0**	**2.3**	**8.5**	**31.3**	**0.4**	**8.1**	4.6	**0.9**	**7.3**	**2.2**
GC-RANSAC	**0.3**	2.3	**0.1**	**0.4**	3.1	8.6	**0.6**	8.8	**13.0**	**0.5**	7.2	**42.8**	**0.8**	2.8	**0.0**	2.5	**24.5**	**26.0**	**0.4**	**7.4**	**3.8**	**0.9**	14.5	3.1
USAC	0.4	3.3	0.3	0.6	**2.2**	9.2	1.4	**4.4**	15.0	0.8	**3.1**	44.0	0.9	10.0	**0.0**	**1.8**	25.0	73.3	0.8	9.1	**3.8**	1.3	**2.6**	3.1
OpenCV	0.4	**1.6**	**0.2**	0.5	4.4	**8.3**	0.8	18.5	**13.0**	0.7	14.9	45.2	0.9	**1.3**	**0.0**	3.5	136.0	33.3	0.5	69.2	**3.0**	1.0	121.0	**0.8**

Table 1. Median errors (ϵ_{med}), failure rates (f; as percentages) and avg. run-times (t, in milliseconds) are reported for each method on all tested problems and datasets. The error of the fundamental matrices is the Sampson distance from the ground truth. For homographies, the RMSE re-projection error from ground truth inliers is used. For essential matrix, the error is symmetric geometric distance (SGD) of normalized points. A test is considered a failure if the error is bigger than 1% of the image diagonal. For each method, the inlier-outlier threshold was set to maximize the accuracy (for fundamental matrix is 1 pixel, for homographies 2 pixels and for essential matrix, 1 pixel normalized by the intrinsic matrices) and the confidence to 0.99. The best values in each column are shown by red and the second best ones by blue.

	USACv20	GC-RANSAC	USAC	OpenCV
ϵ	**0.7**	**0.8**	1.0	1.0
t	**5.1**	8.8	**7.5**	45.9
f	**12.8**	**11.9**	18.6	13.0

Table 2. The avg. of the errors (ϵ; in pixels), processing times (t; in milliseconds) and failure rates (f; in percentages) in Table 1 are reported. The best values in each column are shown by red and the second best ones by blue.

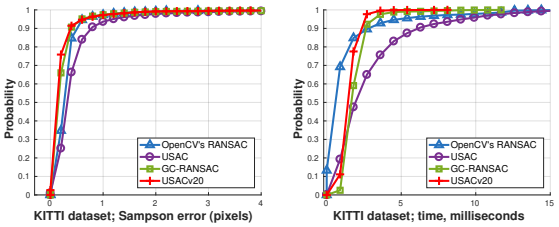


Figure 4. The cumulative distribution functions (CDF) of the Sampson errors (left plot; horizontal axis) and processing times (right; milliseconds) of the estimated fundamental matrices on the KITTI dataset.

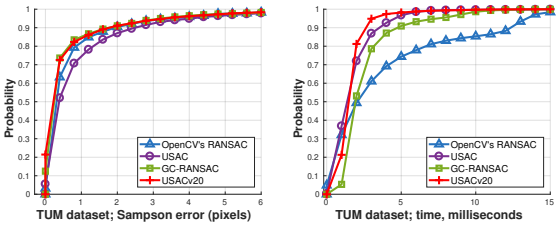


Figure 5. The cumulative distribution functions (CDF) of the Sampson errors (left plot; horizontal axis) and processing times (right; milliseconds) of the estimated fundamental matrices on the TUM dataset.

Cut RANSAC. Compared to the original USAC, all reported properties improved significantly. Also, an important objective was to implement a modular and optimized framework in C++ to make future RANSAC modules easy to be combined with. The

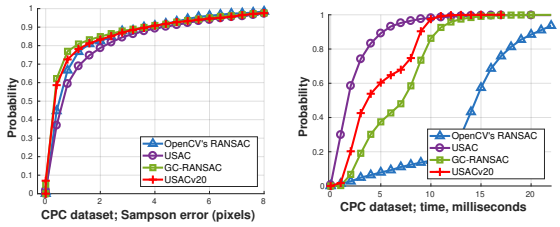


Figure 6. The cumulative distribution functions (CDF) of the Sampson errors (left plot; horizontal axis) and processing times (right; milliseconds) of the estimated fundamental matrices on the CPC dataset.

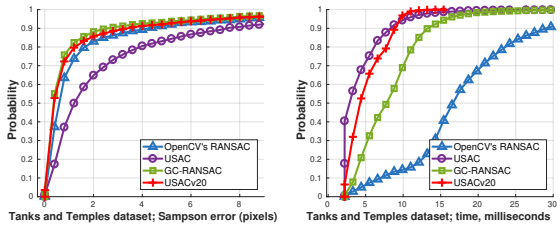


Figure 7. The cumulative distribution functions (CDF) of the Sampson errors (left plot; horizontal axis) and processing times (right; milliseconds) of the estimated fundamental matrices on the Tanks and temples dataset.

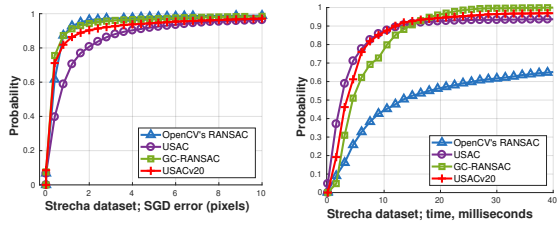


Figure 8. The cumulative distribution functions (CDF) of the Sampson errors (left plot; horizontal axis) and processing times (right; milliseconds) of the estimated essential matrices on the Strecha dataset.

pipeline will be made available after publication.

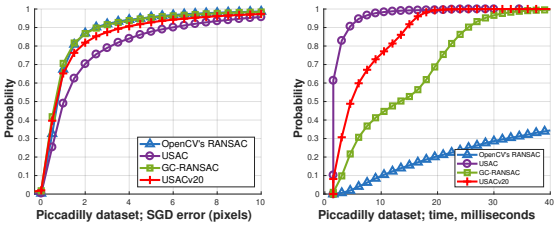


Figure 9. The cumulative distribution functions (CDF) of the Sampson errors (left plot; horizontal axis) and processing times (right; milliseconds) of the estimated essential matrices on the Piccadilly scene of the `1DSfM` dataset.

5. Acknowledgement

This research was supported by Czech Technical University student grant SGS OHK3-019/20.

References

[1] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski. Bundle adjustment in the large. In *European conference on computer vision*, pages 29–42. Springer, 2010. 1

[2] D. Barath, M. Ivashechkin, and J. Matas. Progressive NAPSAC: sampling from gradually growing neighborhoods. *arXiv preprint arXiv:1906.02295*, 2019. 2, 3, 4

[3] D. Barath and J. Matas. Graph-Cut RANSAC. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6733–6741, 2018. https://github.com/danini/graph-cut-ransac. 2, 3, 4, 5

[4] D. Barath, J. Noskova, and J. Matas. MAGSAC: marginalizing sample consensus. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. https://github.com/danini/magsac. 3, 4

[5] J.-W. Bian, Y.-H. Wu, J. Zhao, Y. Liu, L. Zhang, M.-M. Cheng, and I. Reid. An evaluation of feature matchers forfundamental matrix estimation. *arXiv preprint arXiv:1908.09474*, 2019. https://jwbian.net/fm-bench. 5

[6] O. Chum and J. Matas. Matching with PROSAC-progressive sample consensus. In *Computer Vision and Pattern Recognition*. IEEE, 2005. 2, 3, 4

[7] O. Chum and J. Matas. Optimal randomized RANSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1472–1482, 2008. 2, 4

[8] O. Chum, J. Matas, and J. Kittler. Locally optimized RANSAC. In *Joint Pattern Recognition Symposium*. Springer, 2003. 2, 3

[9] O. Chum, T. Werner, and J. Matas. Epipolar geometry estimation via RANSAC benefits from the oriented epipolar constraint. In *International Conference on Pattern Recognition*, 2004. 3, 4

[10] O. Chum, T. Werner, and J. Matas. Two-view geometry estimation unaffected by a dominant plane. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 772–779. IEEE, 2005. 3, 4

[11] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014. 1

[12] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 1, 3, 4, 5

[13] 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*, pages 3354–3361. IEEE, 2012. 5, 7

[14] D. Ghosh and N. Kaabouch. A survey on image mosaicking techniques. *Journal of Visual Communication and Image Representation*, 2016. 1

[15] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, USA, 2 edition, 2003. 5

[16] H. Isack and Y. Boykov. Energy-based geometric multi-model fitting. *International Journal of Computer Vision*, 2012. 1

[17] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun. Tanks and Temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):78, 2017. 1, 5, 7

[18] K. Lebeda, J. Matas, and O. Chum. Fixing the locally optimized RANSAC. In *British Machine Vision Conference*. Citeseer, 2012. http://cmp.felk.cvut.cz/wbs/. 2, 3, 5, 7

[19] D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer vision*. IEEE, 1999. 5, 6

[20] J. Matas and O. Chum. Randomized RANSAC with sequential probability ratio test. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1727–1732. IEEE, 2005. 2

[21] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 2004. 1

[22] D. Mishkin, J. Matas, and M. Perdoch. MODS: Fast and robust method for two-view matching. *Computer Vision and Image Understanding*, 2015. 1

[23] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam

system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 1

[24] D. R. Myatt, P. H. S. Torr, S. J. Nasuto, J. M. Bishop, and R. Craddock. NAPSAC: high noise, high dimensional robust estimation. In *In BMVC02*, pages 458–467, 2002. 2, 3

[25] D. Nistér. An efficient solution to the five-point relative pose problem. *Transactions on Pattern Analysis and Machine Intelligence*, pages 756–770, 2004. 6

[26] T. T. Pham, T.-J. Chin, K. Schindler, and D. Suter. Interacting geometric priors for robust multimodel fitting. *Transactions on Image Processing*, 2014. 1

[27] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *International Conference on Computer Vision*. IEEE, 1998. 1

[28] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm. USAC: a universal framework for random sample consensus. *Transactions on Pattern Analysis and Machine Intelligence*, 2013. 1, 4, 5, 6

[29] P. J. Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984. 4

[30] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 1

[31] C. Sminchisescu, D. Metaxas, and S. Dickinson. Incremental model-based estimation using geometric constraints. *Pattern Analysis and Machine Intelligence*, 2005. 1

[32] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *ACM SIGGRAPH 2006 Papers*, page 835–846, New York, NY, USA, 2006. Association for Computing Machinery. 6

[33] S. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International journal of computer vision*, 80(2):189–210, 2008. 6

[34] C. Strecha, R. Fransens, and L. Van Gool. Wide-baseline stereo from multiple views: a probabilistic account. In *Conference on Computer Vision and Pattern Recognition*. IEEE, 2004. 7

[35] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580. IEEE, 2012. 5, 7

[36] P. H. S. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *International Journal of Computer Vision*, 50(1):35–61, 2002. 2

[37] P. H. S. Torr and D. W. Murray. Outlier detection and motion segmentation. In *Optical Tools for Manufacturing and Advanced Automation*. International Society for Optics and Photonics, 1993. 1

[38] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 2000. 2, 4

[39] P. H. S. Torr, A. Zisserman, and S. J. Maybank. Robust detection of degenerate configurations while estimating the fundamental matrix. *Computer Vision and Image Understanding*, 1998. 1

[40] K. Wilson and N. Snavely. Robust Global Translations with 1DSfM. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 61–75. Springer, 2014. 1, 5, 6, 7

[41] M. Zuliani, C. S. Kenney, and B. S. Manjunath. The multi-RANSAC algorithm and its application to detect planar homographies. In *International Conference on Image Processing*. IEEE, 2005. 1

25^{th} Computer Vision Winter Workshop
Domen Tabernik, Alan Lukei, Klemen Grm (eds.)
Rogaka Slatina, Slovenia, February 3–5, 2019

Practical high-speed motion sensing: event cameras vs. global shutter

Ondřej Holešovský, Václav Hlaváč, Radoslav Škoviera
Czech Technical University in Prague
Czech Institute of Informatics, Robotics, and Cybernetics
160 00 Praha 6, Dejvice, Jugoslavských partyzánů 1580/3, Czech Republic
ondrej.holesovsky@cvut.cz https://people.ciirc.cvut.cz/holesond/

Roman Vítek
University of Defence in Brno
Faculty of Military Technology, Department of Weapons and Ammunition
662 10 Brno, Kounicova 65, Czech Republic
roman.vitek@unob.cz

Abstract. *We designed two maximally simplified and controlled experiments to test event-based and frame-based cameras in the price category affordable for an ordinary university research lab. First, we put common ArUco markers on a rotating disk and observed them by both types of cameras. We reconstructed the image from an event camera using a publicly available state-of-the-art algorithm and compared the ArUco marker recognition reliability. Surprisingly, our results suggest that the ability of the tested event camera to recognise quickly moving markers is inferior to an affordable* 1000 *fps frame-based camera. In the second experiment, we let the cameras observe a freely flying subsonic pistol projectile. A very expensive* 20000+ *fps camera provided ground-truth images, as the acquisition rate of the affordable frame-based camera was insufficient. Although event camera data was partially corrupt, it still allowed us to estimate the position of the small projectile every* 10 μ*s, when the projectile translated mostly along the event camera pixel rows.*

1. Introduction

Independent pixels of event cameras [1] generate asynchronous events in response to local log intensity changes. Each pixel performs a level-crossing sampling of the difference of logarithmic brightness sensed by the pixel. Each time the difference passes a preset threshold, the pixel emits a change detection (CD) event and resets its brightness reference to the current brightness. A CD event is characterised by its pixel coordinates, its precise timestamp in microsecond resolution, and the polarity of the brightness change. The advantages of event cameras over traditional cameras include lower sensor latency, higher temporal resolution, higher dynamic range (120 dB+ vs. 60 dB of traditional cameras), implicit data compression, and lower power consumption.

This work gives a partial answer to the fundamental question of applicability of event cameras: What are the applications, in which event cameras cannot be replaced by high-speed cameras capturing sequences of image frames? The answer to this question, however, is not so straightforward. Our initial hypothesis was that the superiority of event cameras comes from the fast capture of asynchronous events only from pixels where changes happen, as compared to the full frame readout in ordinary frame-based cameras. This phenomenon should be even stronger when changes in the scene are rather local.

Initially, we tested the limits of our ATIS event camera at a partner university in a very fast experiment when observing a flying bullet. Surprisingly, the motion speeds and scene complexities the event camera could record well were lower than we expected. Moreover, we observed strange phenomena in the event recordings.

In general, the speed limits of event cameras depend on two factors:

1. The pixel bandwidth and sensitivity restricts which light changes can be detected by a pixel.

2. The ability of the digital sensor logic to read out events from the pixel array correctly and in time restricts the spatial and temporal distribution of events in the pixel array.

We tested these limits in two experimental settings, namely on the detection of quickly rotating ArUco markers and on the position tracking of a flying projectile fired from a firearm. We compare event camera results to images captured by high-speed global shutter cameras.

2. Related work

There is a recent survey by Gallego et al. [2] mentioning several different event camera application areas. The mentioned areas are real-time interaction systems, object tracking, surveillance, object recognition, depth estimation, optical flow, 3D structured light scanning, high dynamic range (HDR) imaging, video compression, visual odometry, and image deblurring.

Event cameras attract growing attention, which was demonstrated at the Second International Workshop on Event-based Vision and Smart Cameras at CVPR in June 2019 [3].

To the best of our knowledge, however, the literature dealing with head-to-head comparisons of event and common frame-based cameras is quite limited.

A common (frame-based) monochrome camera provides as its output the sequence of grayscale images naturally. Reconstruction of images from an event camera is more complicated. The state-of-the-art approach to cope with this task was published in Rebecq et al. [4]. Among other things, the authors compare the quality of images reconstructed from events to standard camera frames. The reconstructed images better capture the dynamic range of the scene than the standard frames. The authors also compare visual-inertial odometry algorithms running on traditional camera frames and on images reconstructed from events. Event-based reconstructed intensity image results are reported to be on average superior not only to the results of traditional frames, but also to the state-of-the-art methods running on events directly. However, the first is no surprise as the chosen traditional camera frame rate was only 20 frames per second and the captured frames suffered from severe motion blur likely due to the too long exposure time.

Falanga et al. [5] analyse the response latency of obstacle avoidance of a quadrotor drone with a mounted camera. The obstacle size and shape was assumed to be known. Their event-based algorithm was able to detect the obstacle whenever a displacement of at least five pixels occurred. Given drones available at that time, the authors concluded that an event camera with resolution 320×262 pixels gave obstacle detection latencies comparable to standard stereo cameras running at 60 fps. Increasing the resolution of event cameras might make them a better solution, as the ability to sense a distant obstacle depends on sufficiently high spatial sensor resolution. However, most of the analysis was done theoretically. Experiments were only done with the event camera, not with the standard cameras.

Barrios-Avilés et al. [6], probably the closest work to ours, also test the object detection latency of event and standard cameras. Their vision system detects a black circular dot rotating on a white disk and estimates the position of the dot for control purposes. Surprisingly, the authors report latency differences between the two cameras in the order of 100 ms, despite the frame rate of the standard camera being 64 fps at VGA image resolution. It is unlikely that such long latency would be caused by the cameras or by the object detection algorithm based on image intensity thresholding running on the standard camera frames.

3. Method

We test the speed limits of an event camera on the task of reading ArUco markers [7] in motion. The name ArUco originates in a free software library [1] for processing the markers. The markers placed on a disk are rotated at a gradually increasing velocity, which is measured independently by a rotary encoder, see Figure 1.

Raw CD events on their own do not suffice for the detection of typical markers. To alleviate this problem, we utilise a state-of-the-art method for intensity image reconstruction from events. One such method, E2VID[2] was presented in [8] and [4].

The intensity image reconstruction method E2VID [4] utilises a recurrent convolutional neural network whose architecture is similar to UNet. In each iteration, the network computes a reconstructed intensity image as a function of a batch of events and a sequence of K previously reconstructed intensity images. The authors stored each event batch for the network input into a spatio-temporal voxel grid.

[1] https://www.uco.es/investiga/grupos/ava/node/26

[2] code: https://github.com/uzh-rpg/rpg_e2vid

The network was trained in a supervised mode on simulated event sequences and corresponding ground-truth intensity images.

We measure marker detection performance by two metrics, namely detection count and detection reliability. We define marker detection reliability r as

$$r = \frac{1}{N_r N_m} \sum_{i=0}^{N_r} M_i, \tag{1}$$

where M_i is the number of detected unique marker IDs within revolution i, N_r is the total number of recorded revolutions, N_m is the number of unique marker IDs printed on the disk. Detection count c is

$$c = \frac{1}{N_r N_m} \sum_{j=0}^{N_m} I_j, \tag{2}$$

where I_j is equal to the number of times the marker ID j is detected within N_r revolutions.

4. Implementation

We use the ATIS HVGA Gen3 event camera kit PSEE350EVK (from Prophesee[3]).We experimentally estimated the maximum data bandwidth to be approx. 22 million events per second, by rotating the camera with a telelens as fast as possible while looking at a densely textured scene. We mounted a lens with the focal length of 25 mm. The sensor generates both the standard change detection (CD) events, as well as asynchronous exposure measurement (EM) events. An older generation of the ATIS camera is described in [9]. In this work, we utilise only the simpler CD events, so that results can be applied to other event cameras such as [1] or [10]. The size of the photodiode of the change detector is not stated directly by the manufacturer. However, we expect this area to be at least 10% of the pixel area, which is the CD fill factor reported for the previous sensor generation in [9]. Thus, the size of the photodiode should be at least $20 \cdot 20 \cdot 0.1 = 40\ \mu m^2$. The change detector sensitivity parameter is in the range from 0 (least sensitive) to 100 (most sensitive).

All parameters in the reconstruction method [4] were kept at the defaults, except the number of events used for the reconstruction of a single intensity image. Unless noted otherwise, we set this number to 8640 events per frame, which enables good reconstruction of the markers we are using on plain white background. The effect of changing this parameter is investigated further in the experiments.

camera	ATIS	Photron	Basler
resolution $[px]$	480×360	1024×1024	480×360
pixel size $[\mu m]$	20×20	20×20	4.8×4.8
exposure $[\mu s]$	-	1	59*
frame rate [FPS]	-	20000	1000**

* exposure time in weak lighting conditions was 3000 μs
** 300 FPS was used during weak lighting conditions

Table 1: Camera parameters and experiment conditions

Our chosen fast global shutter camera is the Basler acA640-750um USB 3.0 camera [4]. Basler pixel area is at most 23 μm^2. Therefore, we may assume that the light-sensitive area of the ATIS pixels is at least two times larger than the same area in the Basler pixels.

Additionaly, a high speed camera Photron FastCam SA-Z[5] was used as a reference during tests with the projectiles at the ballistic laboratory of Department of Weapons and Ammunition, University of Defence in Brno. A list of basic parameters for all cameras is listed in Table 1.

ArUco markers were 3×3 mm in size. They were generated from a custom dictionary: 4×4 data squares in each marker, dictionary size 10, random seed 65536. Only marker IDs 1-9 were used. The centres of the nine markers were located on the circumference of the rotating disk, at the radius of 85 mm. A DC motor with controllable rotation speed rotated the disk. A rotary encoder mounted on the motor shaft provided independent angular velocity measurements.

A photo of our experimental setup for comparison of the ATIS and Basler cameras in different lighting conditions is in Figure 1. Both cameras have a similarly-sized field of view, see Figure 2. The Basler camera sees the marker as a square of size 34.1 pixels, the ATIS camera perceives the apparent marker size of 34.4 pixels. We deliberately chose as small markers as possible, in order to maximally reduce the number of events generated by the ATIS sensor per a single pass of a marker across the pixel array. The setup of the balistic experiment with the high-speed camera is shown in Figure 3.

[3] https://www.prophesee.ai/

[4] https://www.baslerweb.com

[5] https://photron.com/fastcam-sa-z/

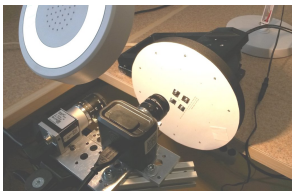
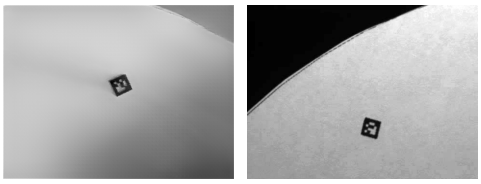


Figure 1: The experimental scene setup with strong lighting. The frame-based Basler (left) and the event-based ATIS (right) cameras are watching the white rotating disk with nine ArUco markers. A nonflickering LED lamp illuminates the scene from the top.



(a) ATIS (b) Basler

Figure 2: The field of view of each camera.

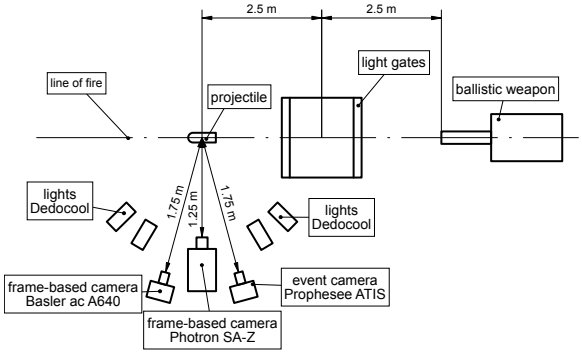


Figure 3: The scene setup of the balistic experiment.

5. Experiments

5.1. Strong lighting

Initially, we used the lamp shown in Figure 1 to strongly illuminate the scene, which is common in industrial settings. With exposure time 59 μs of the Basler camera, the white paper colour has mean intensity values of 61.3 ± 2.5, the black marker colour has mean intensity of 7.0 ± 1.5.

With the Basler camera, ArUco marker detection works reliably up to 25 revolutions per second (rps) or 150 kpx/s image velocity. A 50% detection reliability is reached around 34 rps, see Figure 6. See Figure 7 for sample marker images at different angular speeds.

The images reconstructed from the ATIS events enable reliable marker detection up to 4 rps or 24 kpx/s image velocity with sensitivity set to 40. 50% detection reliability is reached around 6 rps, see Figure 4. With growing rotational speed, the detection reliability of the ATIS decays faster when the sensitivity is set significantly lower or higher than 40, shown in Figure 5. However, the sensitivity value of 60 yields the most detected markers per revolution at low rotational speed, it detects the least markers at higher speeds. See Figure 8 for sample reconstructed marker images at different angular speeds.

Figure 9 shows two batches of recorded events 100 μs long visualised in the image plane. White pixels denote positive, black pixels negative polarity events. At the lower velocity of 3.5 rps, events are mostly read in time from all pixels which generate them. At the higher angular velocity of 6 rps, however, it is more common that no events are read from several pixel rows in the middle of the ArUco marker where events should have been present. Most of the time, the event rate measured in 50 μs intervals of the recording did not exceed 15 Mev/s in the faster case and 12 Mev/s in the slower case.

Reducing the event camera sensitivity setting did not help in raising the speed limits significantly. When the sensitivity is set too low, no events are generated at higher speeds. A sufficiently high sensitivity needs to be set so that the intensity reconstruction algorithm has enough data to reconstruct the intensity well. In the case of 4 rps, for example, lowering the sensitivity from 50 to 20 reduces the highest typical event rate from 15 to 10 million events per second. However, too few events are recorded, see Figure 10 and reliable intensity reconstruction becomes more challenging with very low sensitivity, see Figure 4. Altering the number of events used for the reconstruction of a single image does not help either, see Figure 11. Too few events and fine marker details are easily missed. Too many events and the edges become blurred.

5.2. Weak lighting

Weaker scene lighting is common in more natural or outdoor scenes, where artificial light cannot be conveniently used or is undesirable. We emulate weak lighting at night by a weaker LED lamp with all other light sources turned off, see Figure 12. With weak lighting and exposure time 3000 μs of the Basler camera, the white paper gives mean intensity values of 26.3 ± 4.5, the black marker colour has a

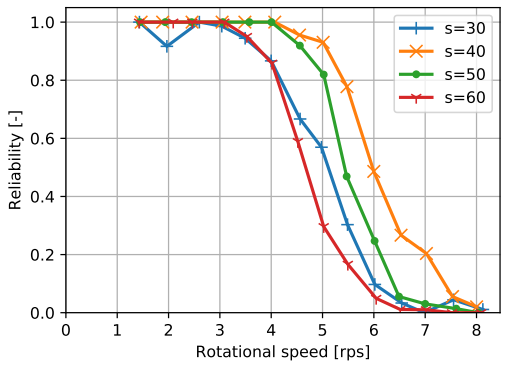


Figure 4: Reliability of marker detection with the ATIS as a function of rotational speed. For several sensitivity settings s. Strong lighting.

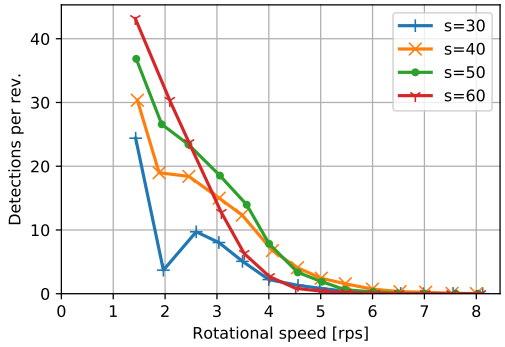


Figure 5: Detection count, i.e. mean number of single marker detections per revolution with the ATIS as a function of rotation speed. For several sensitivity settings s. Strong lighting.

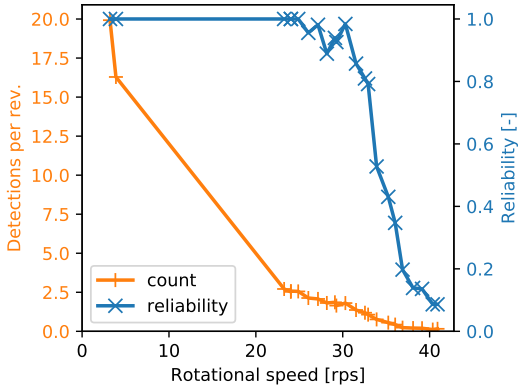
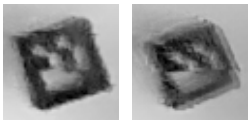


Figure 6: Reliability of marker detection and mean count of single marker detections per revolution as functions of rotational speed. Basler camera, strong lighting.

(a) 4.0 rps (b) 23 rps (c) 30 rps

Figure 7: Sample images of an ArUco marker captured by the Basler camera. Strong lighting, exposure time 59 μs. Different rotational speeds.

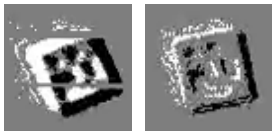


(a) 3.5 rps (b) 6.0 rps

Figure 8: Sample ATIS reconstructions of an ArUco marker from 8640 events per frame. Sensitivity 40, strong lighting.

(a) 3.5 rps. (b) 6.0 rps.

Figure 9: 100 μs of recorded ATIS events at different rotational velocities. Sensitivity 40, strong lighting.



(a) $s = 50$. (b) $s = 20$.

Figure 10: 100 μs of recorded ATIS events at 4 rps using different sensitivities s. Strong lighting.

(a) 2073 ev. (b) 4620 ev. (c) 8640 ev.

Figure 11: Unreliable ATIS reconstructions of an ArUco marker at 4 rps, sensitivity 20, strong lighting. Different number of events used for each reconstruction.

mean intensity of 2.6 ± 1.2.

With the Basler camera, ArUco marker detection works reliably up to 0.47 rps or 2.9 kpx/s image ve-

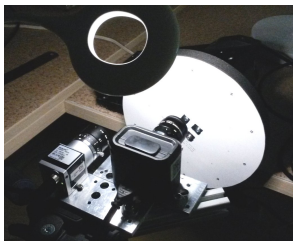


Figure 12: The experimental scene setup with weak lighting.

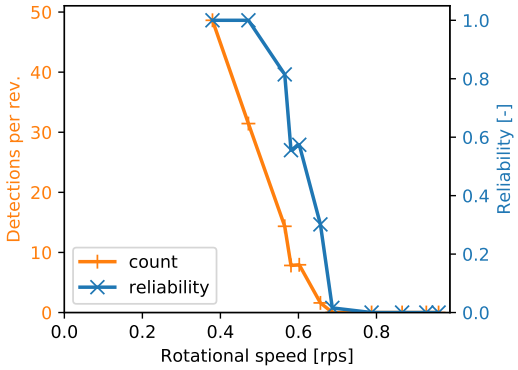


Figure 13: Reliability of marker detection with the Basler camera and mean count of single marker detections per revolution as functions of rotational speed. Weak lighting.

(a) 0.47 rps (b) 0.61 rps

Figure 14: Images of an ArUco marker captured by the Basler camera at weaker lighting. Exposure time 3000 μs. Different rotational speeds.

locity. 50% reliability is reached approximately for 0.61 rps or 3.7 kpx/s, see Figure 13. See Figure 14 for sample marker images at different angular speeds.

The signal-to-noise ratio of the ATIS event camera decreases with decreasing lighting intensity. To quantify this fact, we measured the ambient event rate generated by the event camera in a static scene, see Figure 15. In the case of sensitivity equal to 50, the noise event rate grows from 7000 events per second with strong lighting to 15000 events per second with weak lighting. This difference becomes even more pronounced with higher sensitivity settings.

The images reconstructed from the ATIS camera

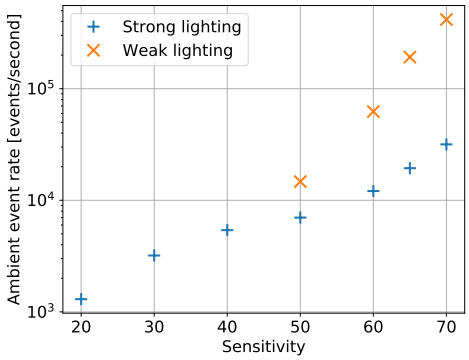


Figure 15: Maximum ambient (noise) event rate of the ATIS at the strong and weak lighting, as the function of contrast sensitivity. The event rate was measured on 10 ms long intervals.

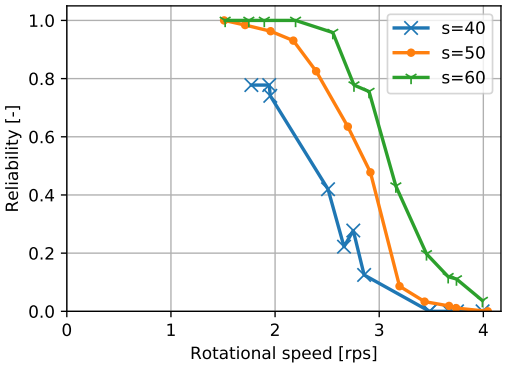


Figure 16: Reliability of marker detection using the ATIS as a function of rotational speed. For several sensitivity setting values s. Weak lighting.

events with the highest tested sensitivity setting of 60 enable reliable marker detection up to 2.2 rps or 13 kpx/s image velocity, see Figure 16. 3.1 rps or 19 kpx/s yield 50% reliability with the best performing sensitivity setting of 60. Lower sensitivities of 50 and 40 perform worse than 60. This time, unlike with the strong lighting, the same observation applies even to the detection count visualised in Figure 17. See Figure 18 for sample reconstructed marker images at different angular speeds and with different sensitivity settings.

5.3. Balistic experiment

To further test the ability of the ATIS sensor to read events from its pixel matrix, we recorded a 9 mm projectile freely flying at the speed of 300 m/s. The image velocity of the projectile was 220 kpx/s in

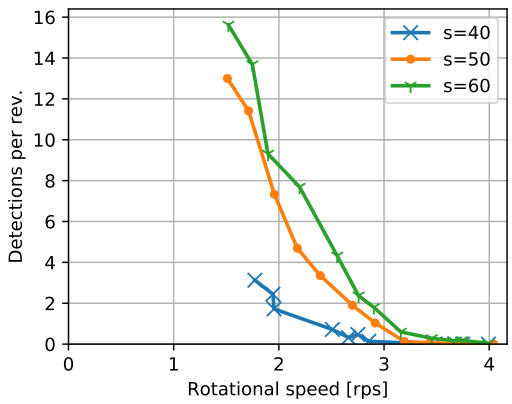


Figure 17: Mean count of single marker detections per revolution with the ATIS as a function of rotational speed. For several sensitivity setting values s. Weak lighting.

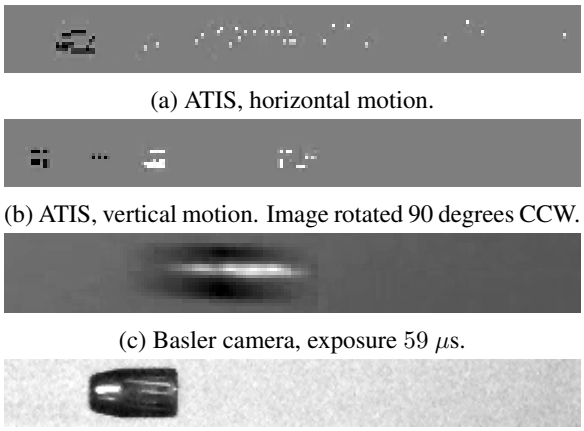
(a) 2.0 rps, $s = 40$ (b) 2.0 rps, $s = 60$ (c) 3.2 rps, $s = 60$

Figure 18: ATIS reconstructions of an ArUco marker from 8640 events per frame at weaker lighting. Different rotational speeds and sensitivities s.

ATIS, 620 kpx/s in Balser, 930 kpx/s in the Photron camera. ATIS sensitivity was set to 50. The resulting samples of event batches 20 μs long are in Figure 19, together with photos of the projectile taken by the Basler and the Photron cameras. The ATIS event rate did not exceed 6 million events per second. The projectile image from the Basler camera is significantly blurred, while the Photron image clearly shows the projectile shape.

When the projectile translated horizontally along the pixel rows of the ATIS, the sensor managed to read events from the entire projectile area in time. We also recorded the same flying projectile with the ATIS camera rotated by ninety degrees around the optical axis, to emulate vertical projectile translation. The visualisation of the vertical translation in Figure 19 suggests events omitted on entire pixel rows, resembling the results in Figure 9.

With the projectile moving mostly along the pixel rows, we are able to estimate the trajectory of the projectile directly from the ATIS events using a simple method. We process events in 10 μs long batches



(a) ATIS, horizontal motion.

(b) ATIS, vertical motion. Image rotated 90 degrees CCW.

(c) Basler camera, exposure 59 μs.

(d) Photron camera., exposure 1 μs.

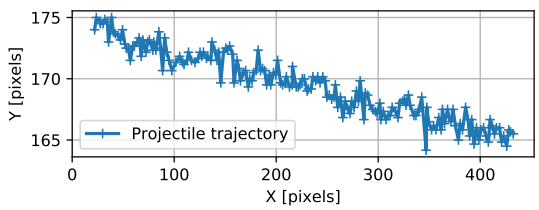
Figure 19: A 9 mm projectile freely flying at 300 m/s. 20 μs of recorded ATIS events and photos of the same scene from the Basler and Photron cameras.

and compute a bounding box of all negative polarity events within a batch. In the case of the Photron images, we threshold each image and compute the bounding box of its dark pixels. The centroid of a bounding box becomes the estimated projectile position.

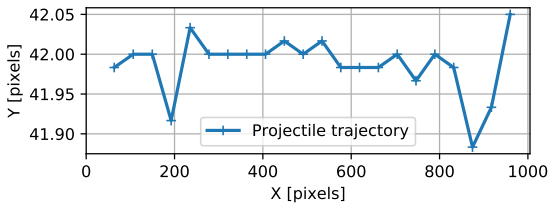
Figure 20 shows the estimated projectile trajectories. The estimates from the Photron camera are more precise than from the ATIS thanks to its higher sensor resolution. The ATIS provided estimates at the rate of 100 kHz and the Photron at 20 kHz. We note, however, that the Photron rate can be increased to 100 kHz for image resolutions comparable to the resolution of the ATIS. The image velocities observed by the two cameras do not have the same direction because their optical axes were not aligned. The results from the Basler camera are not included in the plots as its low 1000 Hz frame rate would fail to capture the image of the projectile more than twice per run, even if the image velocity was only 220 kpx/s.

6. Discussion

With strong lighting, the ATIS event camera enables ArUco marker detection and classification at approx. 6× lower image velocity than the fast global shutter Basler camera. This limit is likely caused by the limited capacity of the event readout circuitry of the ATIS. As the image velocity increases, the circuitry more likely tends to ommit reading events generated by pixels in a subset of pixel rows during



(a) ATIS camera.



(b) Photron camera.

Figure 20: Sample image trajectory of a freely flying 9 mm projectile estimated from ATIS events and Photron camera images.

a 100 μs long interval. This readout jitter corrupts event timestamps and makes intensity image reconstruction more challenging.

Our experiment with the freely flying projectile suggests that the readout circuitry limits are more strict when most events come from a small number of columns, rather than from rows. Omissions of event data appear already at event rates below 6 million events per second. Furthermore, we believe that the estimated global event rate limit of 22 million events per second was not exceeded in any of our experiments.

The Basler camera is only limited by the motion blur when the scene lighting is strong. Weaker scene lighting demands larger exposure time which reduces the maximum motion speed with reliable marker detection.

Our experimental results suggest that the ATIS camera can detect markers moving $4.5\times$ faster than the Basler camera with weaker lighting. We note, however, that this difference may not be entirely caused by the event-based nature of the ATIS camera, as the pixel photodiode area of the ATIS sensor is at least two times larger than the same area in the Basler camera.

The ATIS performance drop due to weaker lighting is likely caused by a larger share of noise events, i.e. a lower signal to noise ratio. Noise events are caused by light shot noise. Shot noise more likely causes a sufficiently large apparent change in relative contrast when the light intensity is lower. Marker reconstruction from events works best with the highest tested ATIS sensitivity setting. This suggests that the amount of informative events gained from increased sensitivity outweighs the increased number of events triggered by shot noise.

The trailing events of positive polarity in Figure 19 are a mystery to us. They suggest that there may be an asymmetry in the pixel response latency to positive and negative high-speed contrast changes. Furthermore, the nature of the latency to positive changes would be stochastic. We asked the ATIS camera manufacturer Prophesee about this and about the event readout ommissions in December 2019 but did not receive an answer to the question before the 22nd of January 2020.

7. Conclusions

We constrained the scope of this work to understanding the limits of event cameras when sensing fast movement. Our experimental event camera data were generated by a single ATIS sensor.

A strongly lit square textured object of size 34×34 pixels with appearance similar to ArUco markers can be reliably recognised in images reconstructed from ATIS events when the object moves at most at 24 kpx/s image velocity. A global shutter camera with exposure time 59 μs can recognise the same object moving at $6\times$ higher image velocity. We believe the ATIS is limited by its event readout circuitry in this case. However, we think that this issue may be mitigated in future generations of event cameras.

When a dynamic scene generates the majority of events in a small number of pixel rows (e.g. seven), the event readout is more reliable. As a result, the position of a tiny object as fast as 220 kpx/s can be tracked at 100 kHz sampling rate by the ATIS.

The decrease in ATIS detection performance from 24 kpx/s to 13 kpx/s due to weaker lighting was likely caused by the decreased signal-to-noise ratio. Photon shot noise triggers relative contrast change events more likely when the light intensity is lower.

With the exception of tracking the position of a tiny object translating almost horizontally, we conclude that we have not found an application where the event camera is significantly better than the ordinary frame-based camera.

Acknowledgements

This work was supported by the European Regional Development Fund under project Robotics for Industry 4.0 (reg. no. $CZ.02.1.01/0.0/0.0/15\_003/0000470$).

We thank Libor Wagner for manufacturing and mounting the plastic disk for the experiment with rotating ArUco markers.

References

[1] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 X 128 120dB 30mW Asynchronous Vision Sensor that Responds to Relative Intensity Change," in *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*. IEEE, 2006. 1, 3

[2] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," *CoRR*, vol. abs/1904.08405, 2019. 2

[3] D. Scaramuzza, G. Gallego, and K. Daniilidis. (2019) Second International Workshop on Event-based Vision and Smart Cameras. [Online]. Available: http://rpg.ifi.uzh.ch/CVPR19_event_vision_workshop.html 2

[4] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "Events-to-Video: Bringing Modern Computer Vision to Event Cameras," *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019. 2, 3

[5] D. Falanga, S. Kim, and D. Scaramuzza, "How Fast Is Too Fast? The Role of Perception Latency in High-Speed Sense and Avoid," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1884–1891, 2019. 2

[6] J. Barrios-Avilés, T. Iakymchuk, J. Samaniego, L. Medus, and A. Rosado-Muñoz, "Movement Detection with Event-Based Cameras: Comparison with Frame-Based Cameras in Robot Object Tracking Using Powerlink Communication," *Electronics*, vol. 7, no. 11, p. 304, nov 2018. 2

[7] S. Garrido-Jurado, R. Muoz-Salinas, F. Madrid-Cuevas, and M. Marn-Jimnez, "Automatic Generation and Detection of Highly Reliable Fiducial Markers under Occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280 – 2292, 2014. 2

[8] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "High Speed and High Dynamic Range Video with an Event Camera," *arXiv e-prints*, 2019. 2

[9] C. Posch, D. Matolin, and R. Wohlgenannt, "A QVGA 143 dB Dynamic Range Frame-free PWM Image Sensor with Lossless Pixel-level Video Compression and Time-Domain CDS," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 259–275, Jan. 2011. 3

[10] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240 X 180 130 dB 3 μs Latency Global Shutter Spatiotemporal Vision Sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, oct 2014. 3

25^{th} Computer Vision Winter Workshop
Domen Tabernik, Alan Lukežič, Klemen Grm (eds.)
Rogaška Slatina, Slovenia, February 3–5, 2020

movie2trailer: Unsupervised trailer generation using Anomaly detection

Orest Rehusevych

The Machine Learning Lab of
Ukrainian Catholic University, ELEKS Ltd.
Lviv, Ukraine
rehusevych@ucu.edu.ua

Taras Firman

Ukrainian Catholic University, ELEKS Ltd.
Lviv, Ukraine
firman@ucu.edu.ua

Abstract. *In this work, we present movie2trailer - a novel unsupervised approach for automatic movie trailer generation. To our knowledge, it is the first-ever application of anomaly detection to such a creative and challenging part of the trailer creation process as a shot selection. One of the main advantages of our approach over the competitors is that it does not require any prior knowledge and extracts all needed information directly from the input movie. By leveraging the recent advancements in video and audio analysis, we produce high-quality movie trailers in equal or less time than professional movie editors. The proposed approach reaches state-of-the-art in terms of visual attractiveness and closeness to the "real" trailer. Moreover, it exposes new horizons for researching anomaly detection applications in the movie industry. The trailers, that were used in evaluation stage are available at the following link - https://bit.ly/2GbOj4R.*

1. Introduction

With the massive expansion of online video-sharing websites such as YouTube, Vimeo, and others, movie promotion through advertisements becomes much more widespread than earlier. In contrast to previous decades, nowadays, trailers became the most crucial part of the movie promotion campaign. Since the trailer creation requires a lot of human efforts and creative decisions considering the selection of scenes, montages, special effects, teams of professional movie editors have to go through the entire film multiple times to select each potential candidate for the best moment. This process can take between 10 days to 2 years to complete [27]. On the high-cost movies, there can be up to six different trailer creation companies involved in this process. During working on the creation of a movie trailer, the editor makes multiple alternative versions of the trailer, the best to be chosen by the target group of specialists afterward. According to [27], there can be created up to 200 variants of the trailer for the target movie. These facts reveal what a significant role a trailer plays in movie success and how much resources it takes to produce a great trailer.

All these factors were the main stimulus for us to make a research on the problem of automatic trailer generation and raise its possibilities to an entirely new level. In our opinion, the area of automatic trailer generation has not been explored enough, and many people underestimate the capabilities of AI advancements over the recent years and how they could be utilized to create high-quality trailers similar to the real one. We strongly believe that AI, to some point, can simulate the expertise and creativity of professional movie editors and reduce huge costs and time consumption.

2. Related works

In this section, we present a short overview of all main approaches for movie trailer generation. The literature divides these methods into two main groups: fully-automated methods and those with human assistance. Until the advent of advanced methods, video summarization techniques, such as Clustering-based Video Summarization [9] and Attention-based Video Summarization [19], were applied to the problem of automatic movie trailer generation. Because of this fact, all the approaches, which

focus on movie trailer generation, were using video summarization techniques as competitors in the evaluation stage. Similarly to them, as an addition, we compare our approach with Muvee[1] - commercial video summarization software.

2.1. Video2Trailer (V2T)

Vid2Trailer (V2T) [12] is a content-based movie trailer generation method. In this paper, the authors set two main requirements for trailers properties to be pleased: they must include specific symbols, such as the title logo sequence shot or/and the main theme music, and they should be visually and audibly attractive to the viewers. As is stated, the algorithm satisfies both of them. The complete pipeline consists of three main stages: symbol extraction, impressive components extraction, and reconstruction. According to the authors, at the time of the publication in 2010, V2T was more appropriate to trailer generation than conventional movie summarization techniques.

2.2. Point Process-Based Visual Attractiveness Model (PPBVAM)

In [29], the authors propose an automatic trailer generation approach, which mainly focused visual attractiveness. Based on common observation, authors assume that during attractive scenes, viewers mostly look at the same area of the screen and, on the other side, lost their focus when boring scenes appear. Consequently, they propose a surrogate measure of visual attractiveness based on viewers' eye-movement, named fixation variance, which is further used as a metric for shots selection. To sum it up, in this paper, authors propose the novel metric for visual attractiveness named fixation variance and learn an attractiveness dynamics model for movie trailers by applying self-correcting point process methodology [13, 22]. The authors mention that their approach outperforms all the previous automatic trailer generation methods and reaches SOTA in terms of both efficiency and quality.

2.3. Human-AI joint trailer generation

Unlike the two automatic trailer generation algorithms mentioned above, IBM Research, in cooperation with 20th Century Fox, introduced the system for first-ever Human-AI trailer creation collaboration, described in [26]. The primary purpose of the system was to identify ten candidates among all movie scenes as the best moments. Further, the professional filmmaker would edit and arrange these moments to construct a comprehensive movie trailer. The system was designed to understand and encode patterns of emotions presented in horror movies. The following steps were performed: Audio Visual Segmentation, Audio Sentiment Analysis, Visual Sentiment Analysis, Scene Composition Analysis, Multimodal Scene Selection. The main system advantage is that it can significantly reduce the involvement of the film editor in the trailer creation process.

3. Approach

Based on our assumptions that by using anomaly detection we can reveal the nonstandard frames among others and that they are the ones that are regularly used in professional movie trailers, we have created a system for automatic trailer generation without any previous knowledge about the target movie. One of the main advantages of our approach is its flexibility in terms of visual appearance. By changing visual features, we can easily put accents on what a user wants to observe in the generated trailer. Figure 1 shows the high-level architecture of our approach.

3.1. Shot Boundary Detection

Shot boundary (transition) detection is one of the major research areas in video signal processing. The main problem it solves is the automated detection of changes between shots in the video. Even though cut detection appears to be an easy task for a human, it is still a non-trivial task for machines. Taking into account a vast number of different types of transitions during shot changes, the problem remains very challenging even nowadays. A lot of researches [14, 30, 1] studying a comparison of various shot boundary detection algorithms were made. Still, there is no silver bullet for detecting all types of transitions accurately. For our work, we decided to go with an open-source Python library for detecting scene changes in videos and automatically splitting the video into separate clips, named *PySceneDetect* [3]. It provides us with two different detection methods:

- Simple threshold-based fade in/out detection
- Advanced content-aware fast-cut detection

The second one appeared to be more appropriate for our problem. The content-aware scene detector finds areas where the difference between two subsequent

[1] https://www.muvee.com

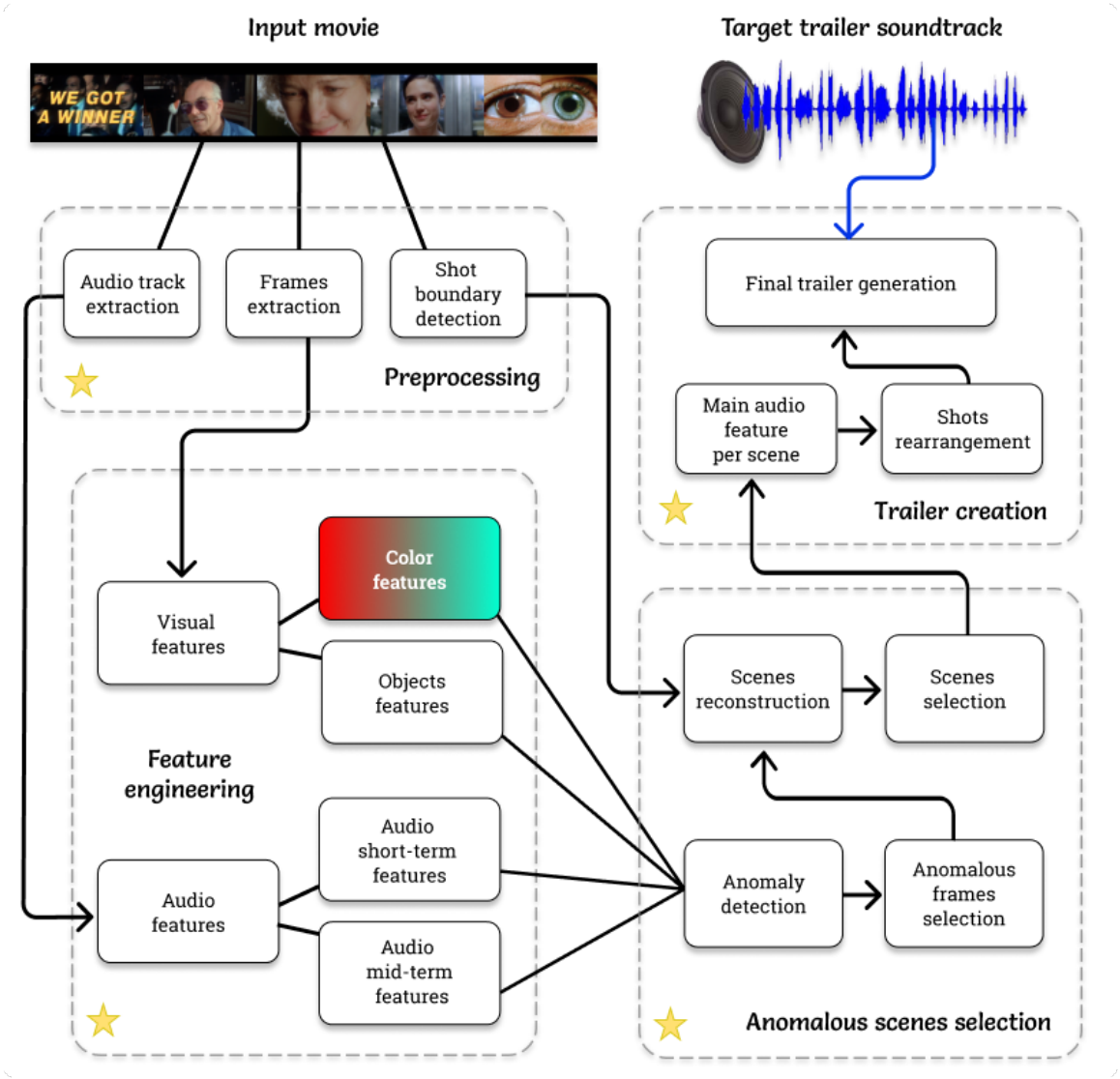


Figure 1: High-level architecture of movie2trailer.

frames exceeds the set threshold value. In contrast to the most traditional scene detection methods, the content-aware detector allows detecting cuts between the scenes, both containing similar content. With a fine-tuned threshold, this approach can detect even minor and sudden changes, such as jump cuts.

3.2. Feature engineering

Feature engineering without exaggeration can be named the most important part of the whole pipeline. This component directly influences the outcomes of all further steps and consequently changes the appearance of the final generated trailer. The selection choice of features leads to changes in what exactly a person wants to see in a trailer. For example, if we want to have a lot of scenes with explosions in our trailer, we need to add a custom feature, which is responsible for detecting explosions (can be done either with the video or audio feature). Table 1 shows all three types of features (visual, audio short-term, and audio mid-term) that was calculated for the given movie.

3.2.1 Visual features

Visual features were selected based on our understanding of what people usually expect to see in the trailer. They can be divided into two subgroups: color model features and object detection features. For color features, we chose the HSL color model,

Visual	Audio short-term	Audio mid-term
Delta hue	Zero Crossing Rate	Mean and standard deviation of all 34 audio short-term features
Delta saturation	Energy	
Delta lightness	Entropy of Energy	
Content value	Spectral Centroid	
Number of people	Spectral Spread	
Number of non-people objects	Spectral Entropy	
Total number of objects	Spectral Flux	
Area of detected people	Spectral Rolloff	
Area of detected non-people	MFCCs	
Total area of detected objects	Chroma Vector	
	Chroma Deviation	

Table 1: The chosen visual, audio short-term and audio mid-term features.

where H corresponds to hue, S - saturation, L - lightness. These properties represent a color spectrum in different forms, which we consider an essential visual aspect of human perception. Additionally, we include the *content value* parameter (mean between Hue, Saturation, and Lightness) to this group of features, as it takes the most significant role in our shot boundary detection process. Hence we are inclined to believe that content value provides information responsible for shot change detection. All the other visual features can be attributed to another (object detection) group. The creation of these features was achieved by leveraging the capabilities of Faster R-CNN [23], pretrained on MS COCO dataset [16]. As a result, we were able to distinguish 80 classes of the most common objects, such as a person, different vehicles, various animals, and everyday things in their natural context. From the extracted information about objects on the frame, we construct six features which can be split into quantity and area groups. The first one was taken because of the hypothesis that frames with many people correspond to scenes with lots of action which keeps viewers' attention on the screen. Another group was formed under the assumption that close-up shots are attractive to view.

3.2.2 Audio short-term and mid-term features

In the majority of the cases, the most salient audio parts are accompanied by outstanding visual scenes and vice versa. Therefore, audio features are not less important than the visual ones. In our algorithm we have used a set of audio features previously introduced in [6]. All audio features were retrieved by exploiting the potential of the open-source library for audio signal analysis named *pyAudioAnalysis* [6]. The main reason of this choice was that because of the significant coverage of sound signal properties, these features had been used in multiple audio analysis and processing techniques. Before the feature extraction step, an audio signal is usually cut into nonoverlapping windows (frames). For the short-term feature sequences, we have used a frame size of 50 msecs of an audio signal and a 1-second window size for the mid-term, correspondingly. As a result of feature extraction, we get a sequence of 34-dimensional and 68-dimensional feature vectors for short-term and mid-term audio signals, respectively. Mid-term features accumulate statistics over the short-term features for a more extended time period to catch more general changes in the audio signal. The statistics include the mean and variance over each short-term feature sequence. To sum it up, we have gathered together all the essential properties of the audio signal for both time and frequency domains that could be further utilized for multiple purposes: from detecting speech among other sounds, to determining the saliency of different parts of the audio.

3.3. Anomalous scenes selection

Anomalous scenes selection is a long process containing multiple steps: anomaly detectors selection, retrieval of anomaly frames for each type of fea-

tures, choice of abnormal visual frames, audio short-term and mid-term frames, merging them together taking into an account the difference in duration of each feature type frame, constructing final set of video frames, scenes reconstruction, threshold-based anomalous scenes selection. Figure 2 shows the complete pipeline of this scene selection approach.

3.3.1 Anomaly detectors selection

Having extracted frame-level visual, short-term, and mid-term audio features for the entire movie, now we are ready to use them in the process of anomalous scenes selection. As a first step, we need to determine what anomaly detectors to use. Based on our experiments, we have concluded, that by applying multiple types of detectors, the result would be much more credible than by using a single one, because of the very different underlying logic between all of them, the various types of data that generated features were based on and possibly very different scale of features. For that reason, we have chosen 8 anomaly detection algorithms covering most of these cases, which could be divided into 4 groups (2 detectors per each group):

- **Linear models: MCD** (Minimum Covariance Determinant) [24], **OCSVM** (One-Class SVM) [20].
- **Proximity-based models: LOF** (Local Outlier Factor) [2], **HBOS** (Histogram-based Outlier Score) [7].
- **Ensembles: IsoForest** (Isolation Forest) [17], **Feature Bagging** [11].
- **Neural networks: AE** (fully-connected AutoEncoder) [10], **MO-GAAL** (Multiple-Objective Generative Adversarial Active Learning) [18].

3.3.2 Anomalous frames selection

With the selected anomaly detectors, we run them separately on each type of the features: visual, audio short-term, and audio mid-term. Each of these types includes its own set of features with diverse frame duration. Since each of the detectors has its pros and cons, we have introduced a voting system to determine the most appropriate frames of each feature type. The frame is considered suitable if at least five of eight detectors have chosen it as anomalous. Having selected frames of each feature type, we reduce audio short-term and mid-term frames to their corresponding visual frames taking into consideration the duration periods of each feature group frame. After that, We obtain a set of anomalous final video frames by taking an intersection between all groups of frames. These frames serves as the basis to identify trailer-worthy scenes from an input movie.

3.3.3 Scenes selection and reconstruction

With the already defined final set of visual frames and information about each scene start and end timestamps, we are ready to reconstruct scenes. The primary constraints for scenes selection are that scenes should have the maximum percentage of anomalous frames and their total duration should be not less than the length of the accompanying soundtrack. Through the visual examination of selected scenes, we have determined that the scenes with the highest number of abnormal frames are the most valid candidates for making the trailer.

3.4. Shots rearrangement

Shots reordering is a beneficial step because it can additionally improve the overall human perception of the viewed video by maximizing the attractiveness with some particular order of shots. By conducting multiple experiments, we have tested a hypothesis that lots of percussive timbres (claps, snares, drums) accompany fast shots with lots of action. Furthermore, we had an assumption that there are some audio features, that should be responsible for detecting percussive sounds. Based on the idea, described in [8], we have found out that by using zero-crossing rate, we could be able to detect such type of sounds quickly and accurately. With our experience watching numerous trailers, we have concluded that in most trailers, the accompanied music increases its intensity through the entire video. To validate that idea, we have calculated the zero-crossing rate vector for each scene and tried different flows with sorting by mean, median, max value of this feature. After that, we have visually examined each of the generated trailers and compared them with the trailer, where scenes are ordered as in the original movie. Since the visual appearance of the arranged by audio feature trailers was visually worse than the one ordered by chronology, we consequently stuck to the latter option.

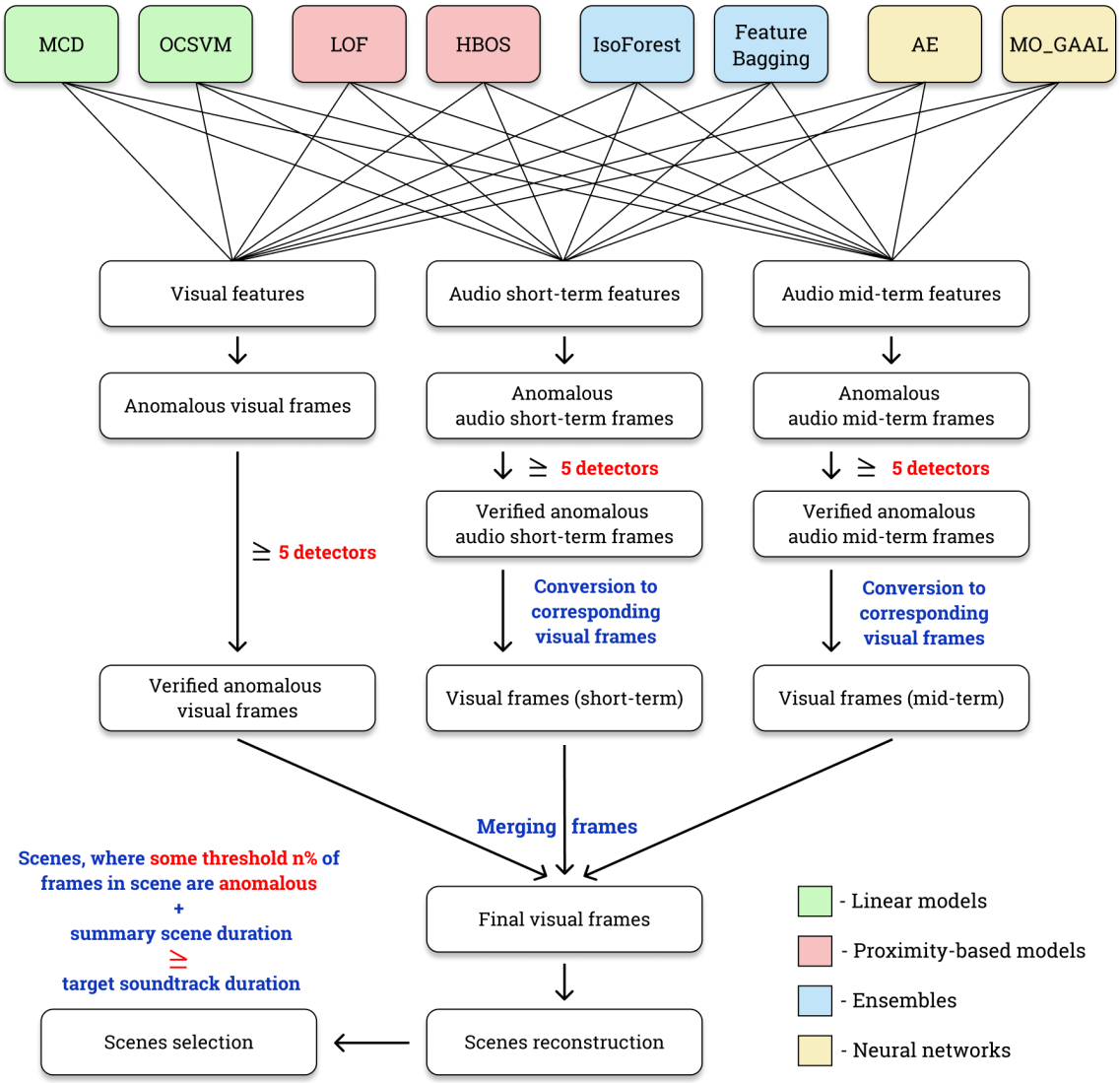


Figure 2: The detailed pipeline of anomalous scenes selection.

4. Evaluation and results

In this section, we evaluate our method **movie2trailer** against all the leading opponents for the automatic trailer generation problem:

- **V2T** [12] - Trailer generation method;
- **Muvee** - Commercial software for video summarization;
- **PPBVAM** (Point Process-Based Visual Attractiveness Model) [29] - SOTA for automatic trailer generation;
- **RT** - The original official real trailers;
- **RTwS** - The same real trailers without speech information

4.1. Qualitative results

For the objective evaluation, we have taken a series of measures to avoid assessment bias. None of the volunteers has seen any of the generated trailers previously. None of the volunteers knew the order of the methods while observing trailers. All final generated trailers were downscaled to the resolution of other trailers (480x360) produced by our competitors' approaches, and all the speech pieces were replaced with the original soundtrack. Similarly to our predecessors, on the input, we give the entire movie without cutting any parts from it to remove spoilers. With the steps above, we can be confident that all the approaches are on an equal footing and would

be evaluated without any bias. Similarly to [12] and [29], we have invited 23 volunteers with different movie tastes and preferences to evaluate the visual appearance of each testing trailer created with different approaches by answering on the following three questions:

- **Appropriateness:** "How similar this trailer looks to an actual trailer?"
- **Attractiveness:** "How attractive is this trailer?"
- **Interest:** "How likely you are going to watch the original movie after watching this trailer?"

For each question, a volunteer should give an integer score of how much he/she agree on the particular statement on the Likert scale [15]: from 1 (the lowest) to the 7 (the highest). Figure 3 shows the overall results for all 3 testing movies: *"The Wolverine (2013)", "The Hobbit: The Desolation of Smaug (2013)", "300: Rise of an Empire (2014)"*. Authors of the **PPBVAM** provided trailers [2] generated with main competitors' approaches for abovementioned movies. We were limited to use only these three movies since the reproduction of some parts of competitors' algorithms is infeasible. The results of the poll show that our method is superior to **V2T**, **Muvee** and **PPBVAM** in all three questions, indicating that our approach to shot selection using anomaly detection is reasonable, and can provide us with such types of shots that satisfy our subjective feelings and perception.

We believe that **RTwS** and **RT** were usually preferred more by volunteers, because all trailers generated using automatic trailer generation methods were deprived of speeches, subtitles, and special effects of montages. Since the information that these factors provide to improving visual attractiveness, we should also supplement our system with these information sources in the future.

4.2. Quantitative results

To the best of our knowledge, the only publicly available method for video aesthetics assessment - Semi-automatic Video Assessment System [21]. This framework incorporates diverse set of visual features that are closely related to aesthetics: Luminance, Optical Flow, Colourfulness and lots of other. All these features are used by SVM [4] to determine the level of aesthetics and interestingness of

[2] https://vimeo.com/user25206850/videos

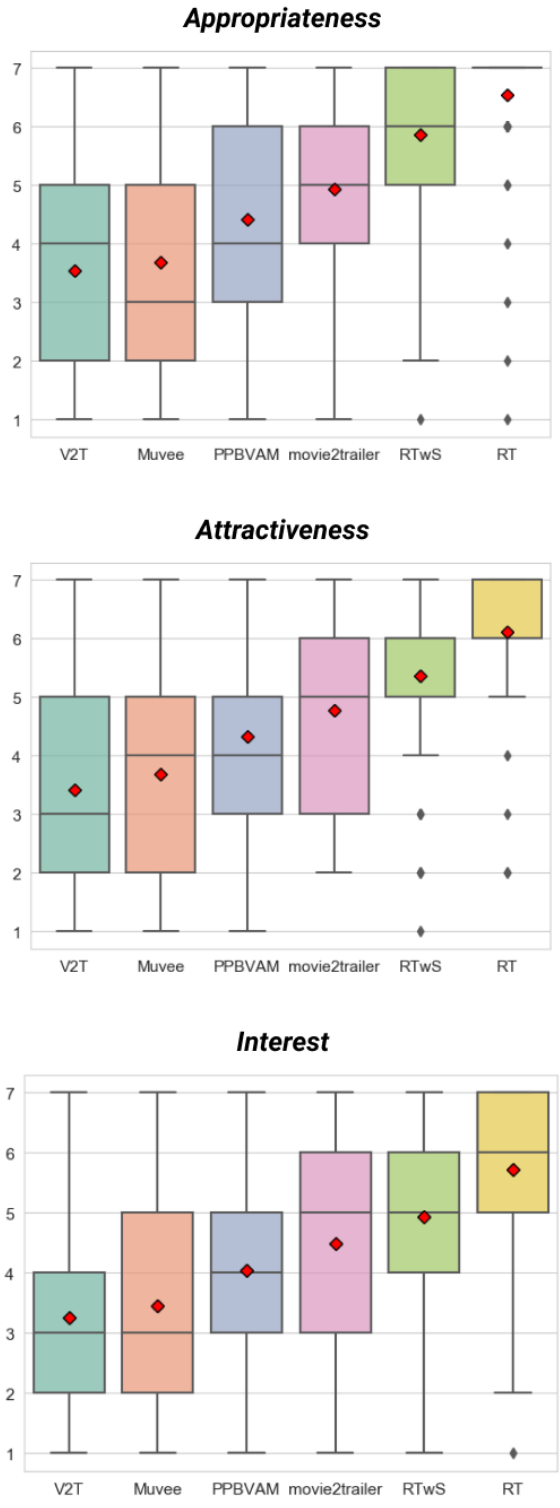


Figure 3: The box plots of scores for various methods on three questions considering Appropriateness, Attractiveness and Interest. The dark lines inside boxes are medians and red diamonds are means. Dark points outside of the whiskers are outliers.

metric	V2T	Muvee	PPBVAM	**ours**	RTwS	RT
mean	4.39	4.31	4.00	**4.73**	4.71	**4.76**
std	0.42	0.33	0.36	0.60	0.65	0.62

Table 2: Quantitative statistics of the NIMA scores.

the target video. To train that method, CERTH-ITI-VAQ700 dataset [28] was used. In view of its large size, the authors decided to use only 1 second of each video, and as a result, their method does not work well on longer videos. During our evaluation, it gave aesthetics score 0 for all trailers, including real and generated ones.

We propose a new approach for video aesthetics evaluation based on evaluating the aesthetics of each video frame separately:

1. Extract all the frames f_i from the video.
2. Compute aesthetics score s_i for each frame f_i.
3. Compute metrics (mean, standard deviation) based on the obtained aesthetics scores s_i.

As a candidate for image aesthetics scoring function, we tested NIMA (Neural Image Assessment) [5] and Will People Like Your Image? [25].

The results obtained using NIMA aesthetics scores (from 1 to 10) (Figure 4 and Table 2) shows that our approach works at the level of **RT** (real trailer).

We have also applied the same approach for evaluation using another image aesthetics assessment algorithm [25]. We have not included the results of this scoring method in this section, because in all cases, it evaluated real trailers worse than the generated ones.

The quantitative results obtained by the aesthetics scoring systems shows that our method outperforms all existing automatic movie trailer generation approaches and is at the level of the real trailers.

5. Conclusion

In this paper, we have presented an unsupervised trailer generation method, named *movie2trailer*. Our approach automatically creates high-quality trailers by identifying anomalous frames relying on the selected set of visual and audio features. A series of quantitative and qualitative experiments show that *movie2trailer* outperforms all the previous automatic trailer generation methods in terms of visual attractiveness and similarity to the "real" trailer and thus is more appropriate to trailer generation than previous techniques. We demonstrated the tremendous potential of the intelligent multidomain analysis system in applying to such a profoundly creative task as creating a movie trailer. This research study opens doors for further investigations of the anomaly detection applications in the movie industry.

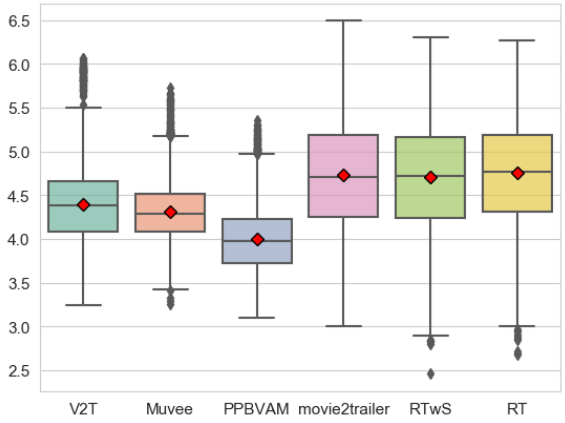


Figure 4: Quantitative comparison of movie trailer approaches based on NIMA aesthetics metric [5].

Acknowledgements

The authors were supported by the ELEKS Ltd. and Ukrainian Catholic University. Special thanks to Oles Dobosevych for his help with the preparation of this publication and providing computational resourses, without which this project hasn't been possible.

References

[1] S. H. Abdulhussain, A. R. Ramli, M. I. Saripan, B. M. Mahmmod, S. A. R. Al-Haddad, and W. A. Jassim. Methods and challenges in shot boundary detection: A review. *Entropy*, 20(4):214, 2018. 2

[2] M. M. Breunig, H. Kriegel, R. T. Ng, and J. Sander. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA.*, pages 93–104, 2000. 5

[3] B. Castellano. Video scene cut detection and analysis tool. https://github.com/Breakthrough/PySceneDetect, 2018. 2

[4] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. 7

[5] H. T. Esfandarani and P. Milanfar. NIMA: neural image assessment. *IEEE Trans. Image Processing*, 27(8):3998–4011, 2018. 8

[6] T. Giannakopoulos. pyaudioanalysis: An open-source python library for audio signal analysis. *PloS one*, 10(12), 2015. 4

[7] M. Goldstein and A. Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track*, pages 59–63, 2012. 5

[8] F. Gouyon, F. Pachet, O. Delerue, et al. On the use of zero-crossing rate for an application of classification of percussive sounds. In *Proceedings of the COST G-6 conference on Digital Audio Effects (DAFX-00), Verona, Italy*, 2000. 5

[9] A. G. Hauptmann, M. G. Christel, W. Lin, B. Maher, J. Yang, R. V. Baron, and G. Xiang. Clever clustering vs. simple speed-up for summarizing rushes. In *Proceedings of the 1st ACM Workshop on Video Summarization, TVS 2007, Augsburg, Bavaria, Germany, September 28, 2007*, pages 20–24, 2007. 1

[10] G. E. Hinton and R. S. Zemel. Autoencoders, minimum description length and helmholtz free energy. In *Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]*, pages 3–10, 1993. 5

[11] T. K. Ho. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8):832–844, 1998. 5

[12] G. Irie, T. Satou, A. Kojima, T. Yamasaki, and K. Aizawa. Automatic trailer generation. In *Proceedings of the 18th International Conference on Multimedia 2010, Firenze, Italy, October 25-29, 2010*, pages 839–842, 2010. 2, 6, 7

[13] V. Isham and M. Westcott. A self-correcting point process. *Stochastic Processes and their Applications*, 8:335–347, 1979. 2

[14] R. Lienhart. Comparison of automatic shot boundary detection algorithms. In *Storage and Retrieval for Image and Video Databases VII, San Jose, CA, USA, January 26-29, 1999*, pages 290–301, 1999. 2

[15] R. Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932. 7

[16] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, pages 740–755, 2014. 4

[17] F. T. Liu, K. M. Ting, and Z. Zhou. Isolation forest. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pages 413–422, 2008. 5

[18] Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, and X. He. Generative adversarial active learning for unsupervised outlier detection. *CoRR*, abs/1809.10816, 2018. 5

[19] Y. Ma, L. Lu, H. Zhang, and M. Li. A user attention model for video summarization. In *Proceedings of the 10th ACM International Conference on Multimedia 2002, Juan les Pins, France, December 1-6, 2002.*, pages 533–542, 2002. 1

[20] L. M. Manevitz and M. Yousef. One-class svms for document classification. *Journal of Machine Learning Research*, 2:139–154, 2001. 5

[21] P. Martins and N. Correia. Semi-automatic video assessment system. In *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing, CBMI 2017, Florence, Italy, June 19-21, 2017*, pages 33:1–33:7, 2017. 7

[22] Y. Ogata and D. Vere-Jones. Inference for earthquake models: A self-correcting model. *Stochastic Processes and their Applications*, 17:337–347, 1984. 2

[23] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015. 4

[24] P. J. Rousseeuw and K. van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999. 5

[25] K. Schwarz, P. Wieschollek, and H. P. A. Lensch. Will people like your image? learning the aesthetic space. In *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*, pages 2048–2057, 2018. 8

[26] J. R. Smith, D. Joshi, B. Huet, W. H. Hsu, and J. Cota. Harnessing A.I. for augmenting creativity: Application to movie trailer creation. In *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017*, pages 1799–1808, 2017. 2

[27] C. Snyder. How movie trailers are made - business insider. https://www.businessinsider.com/how-movie-trailers-are-made-2018-7, 2018. 1

[28] C. Tzelepis, E. Mavridaki, V. Mezaris, and I. Patras. Video aesthetic quality assessment using kernel support vector machine with isotropic gaussian sample uncertainty (KSVM-IGSU). In *2016 IEEE International Conference on Image Processing, ICIP 2016,*

Phoenix, AZ, USA, September 25-28, 2016, pages 2410–2414, 2016. 8

[29] H. Xu, Y. Zhen, and H. Zha. Trailer generation via a point process-based visual attractiveness model. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 2198–2204, 2015. 2, 6, 7

[30] J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang. A formal study of shot boundary detection. *IEEE Trans. Circuits Syst. Video Techn.*, 17(2):168–186, 2007. 2

25^{th} Computer Vision Winter Workshop
Domen Tabernik, Alan Lukežič, Klemen Grm (eds.)
Rogaška Slatina, Slovenia, February 3–5, 2020

Segmentation and Recovery of Superquadric Models using Convolutional Neural Networks

Jaka Šircelj[1,2], Tim Oblak[2], Klemen Grm[1], Uroš Petković[1],
Aleš Jaklič[2], Peter Peer[2], Vitomir Štruc[1] and Franc Solina[2]
[1] Faculty of Electrical Engineering, UL, Tržaška 25, Ljubljana, Slovenia
[2] Faculty of Computer and Information Science, UL, Večna pot 113, Ljubljana, Slovenia
jaka.sircelj@fe.uni-lj.si

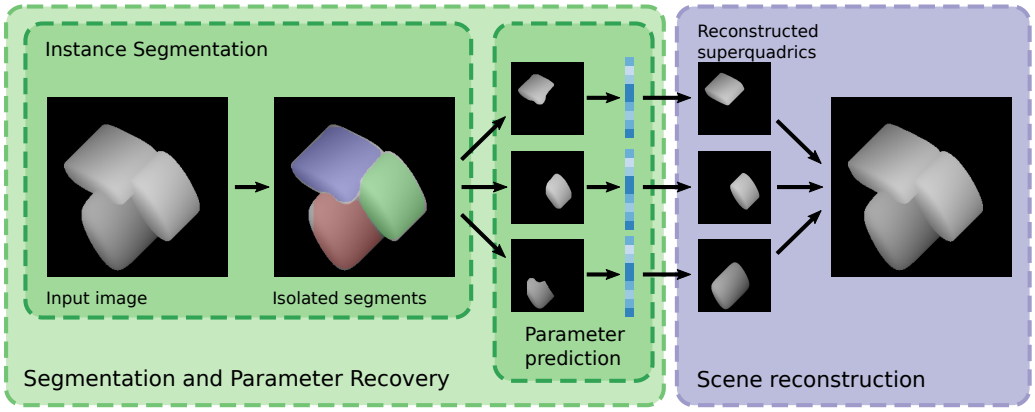


Figure 1: We study the problem of segmenting and recovering superquadric models from depth scenes. Our approach uses instance segmentation with Mask-RCNNs followed by superquadric-parameter estimation from incomplete data with a standard CNN (left part of the figure). Using the recovered superquadric models we are able to efficiently reconstruct the original depth scene (right part of the figure).

Abstract. *In this paper we address the problem of representing 3D visual data with parameterized volumetric shape primitives. Specifically, we present a (two-stage) approach built around convolutional neural networks (CNNs) capable of segmenting complex depth scenes into the simpler geometric structures that can be represented with superquadric models. In the first stage, our approach uses a Mask-RCNN model to identify superquadric-like structures in depth scenes and then fits superquadric models to the segmented structures using a specially designed CNN regressor. Using our approach we are able to describe complex structures with a small number of interpretable parameters. We evaluated the proposed approach on synthetic as well as real-world depth data and show that our solution does not only result in competitive performance in comparison to the state-of-the-art, but is able to decompose scenes into a number of superquadric models at a fraction of the time required by competing approaches. We make all data and models used in the paper available from* https://lmi.fe.uni-lj.si/en/research/resources/sq-seg.

1. Introduction

Representing three-dimensional visual data in terms of parameterized shape primitives represents a longstanding goal in computer vision. The interest in this problem is fueled by the vast number of applications that rely on concise descriptions of the physical 3D space in various sectors ranging from autonomous driving and robotics to space exploration, medical imaging and beyond [13, 21, 14].

Past research in this area has looked at different models that could act as volumetric shape primitives, such as generalized cylinders [28] or cuboids [27, 17, 11], but superquadrics established themselves as one of the most suitable choices for this task [1, 26, 10,

25, 18, 20] due to their ability to represent a wide variety of 3D shapes, such as ellipsoids, cylinders, parallelopipeds and various shapes in between. Formally, superquadrics are defined by an implicit 3D closed surface equation, i.e.:

$$\left(\left(\frac{x-x_0}{a_1}\right)^{\frac{2}{\epsilon_2}}+\left(\frac{y-y_0}{a_2}\right)^{\frac{2}{\epsilon_2}}\right)^{\frac{\epsilon_2}{\epsilon_1}}+\left(\frac{z-z_0}{a_3}\right)^{\frac{2}{\epsilon_1}}=1 \quad (1)$$

where a_1, a_2, a_3 define the bounding box size of the superquadric, ε_1 and ε_2 define it's shape and $(x_0, y_0, z_0)^\intercal$ represent the center of the superquadric in a reference coordinate system [10]. Existing techniques for recovering superquadric models typically involve costly iterative parameter-estimation procedures that further increase in complexity if more than a single superquadric needs to be fitted to a scene [12, 10]. With complex scene geometries, superquadric recovery must necessarily be combined with segmentation techniques capable of partitioning the scene into simpler superquadric-like structures. This, however, puts a considerable computational burden on the fitting procedure as state-of-the-art techniques for recovery-and-segmentation of multiple superquadric models are typically extremely resource demanding.

With recent advances in computer vision and more importantly deep learning, it is possible to design solutions for simultaneous segmentation and recovery of superquadrics that are much more efficient than existing solutions. In this paper, we, therefore, revisit the problem of representing complex depth scenes with multiple superquadrics and develop an efficient solution for this task around convolutional neural networks (CNNs). Specifically, we assume that small superquadric-like structures in range images can be modeled as instances of a specific class of objects, and, therefore, train a Mask-RCNN [7] model to segment the scene, as illustrated in Fig. 1. The results of this instance segmentation are then used as input to a second CNN that recovers superquadric parameters for each of the identified superquadric-like objects. Because the identified superquadric-like objects may be partially occluded, we account for this fact during training and learn the parameters of the second CNN in a robust manner. We evaluate the performance of our approach on simulated, but also real-world range images. We achieve segmentation and recovery results comparable to the state-of-the-art, but achieve a considerable speed-up, which makes the developed solution suitable for a much wider range of applications. We note that in this paper we approach a constrained superquadric recovery problem, where we assume that the depth scene can be approximated by a number of unrotated superquadric models.

Our main contributions in this paper are:

- We present a novel solution for segmentation and recovery of multiple (unrotated) superquadric models from range images built around CNNs and evaluate it in experiments with simulated and real-world depth data.
- We show that existing Mask-RCNNs may be used for identifying superquadric-like structures in range images in an efficient manner.
- We demonstrate that superquadrics can be recovered from partial depth data using a simple CNN-based regressor and the parameter estimation errors are comparable to the error produced by state-of-the-art techniques used for this task.

2. Related work

Existing techniques to scene segmentation with superquadrics can in general be divided in one of two groups: *i)* techniques that approach the problem by segmenting the scene and recovering superquadrics at the same time (*segment-and-fit*), and *ii)* techniques that first segment the scene and then fit superquadric models to the segmented parts (*segment-then-fit*). In this section we briefly review both groups of techniques with the goal of providing the necessary context for our work. For a more comprehensive coverage of the subject, the reader is referred to [10].

Segment-and-fit. Techniques from this group typically combine the segmentation and superquadric recovery stages and often rely on superquadric models to guide the segmentation [5, 12, 10, 9]. Due to the fact that segmentation is performed with the final scene representations (i.e., the superquadric) methods from this group are considered highly robust. However, on the down side, they often also induce a considerable computational burden on the segmentation procedure. Recently, a CNN-solution [20] that falls into this group was proposed, but unlike the approach presented in this paper, was limited to segmentation of predefined classes of objects.

Segment-then-fit. Techniques from this group follow a two-stage procedure, where the data is first segmented up front and independently of superquadric recovery [10]. Thus, the entire procedure is broken down into two independent parts. Examples of techniques from this group include [6, 22, 2,

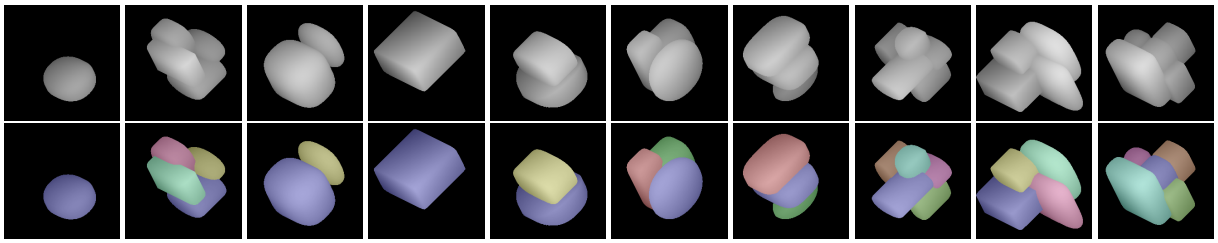


Figure 2: Example images from the generated dataset. The top row shows examples of the rendered images with different numbers of superquadric in the scene. The lower row shows examples of the corresponding segmentation masks. The figure is best viewed in color.

23]. The solution described in this work also follow the segment-then-fit paradigm, but as we show in the experimental section result in competitive performance compared to a state-of-the-art approach from the segment-and-fit group that is in general considered to be more robust.

3. Dataset

In order to train our instance segmentation and parameter estimation models, we require a large dataset of depth scenes with appropriate ground truth labels. Since no such datasets are publicly available, we generate our own and make it publicly available for the research community. In this section we present the dataset creation procedure and discuss the characteristics of the generated data.

3.1. Prerequisites

In this work we follow the methodology of Oblak *et al.* [18] and focus on unrotated superquadric models. Thus, we only try to recover the 8 open parameters from Eq. (1) for each superquadric model and omit rotations, which introduce ambiguities in the superquadric-recovery process [18]. The main goal of this work is to extend the superquadric recovery method from [18] to depth scenes with complex geometry that need to be represented with multiple superquadrics. Consequently, we fix the rotation of the objects in our dataset and render them in an axonometric projection that ensures that three sides of the objects are always visible in the rendered images.

3.2. Dataset creation

We synthesize our dataset by rendering range images with multiple superquadrics in the scene. To construct the range images we create a custom rendering tool that accepts multiple superquadric parameter sequences. The renderer then constructs the range image of a scene by finding the surface points of the superquadrics and choosing the closest point to the viewport, if there are overlapping superquadrics in the line of sight. The scene is constrained inside a $256 \times 256 \times 256$ grid, where the first two dimensions represent the width and height of the resulting image, while the last dimension represents the depth. The scene is then mapped to the zero depth plane, resulting in a 256×256 range image, where its pixel indexes i, j correspond to the x, y coordinates in the 3D scene, while the pixel intensity relates to the z depth in the scene.

To generate a dataset with representative superquadric objects, we uniformly sample the superquadric parameters similarly to [18]. However, uniformly sampling the position and size of superquadrics independently from their neighbors causes dramatic overlaps and intersections in the scene, which hides a large number of objects. We solve this by constraining the allowed intersection-over-union volume between pairs of superquadrics in each scene, where the volume is approximated using the superquadrics bounding-box. Following this requirement we first sample the number of superquadrics in the scene from the discrete uniform distribution $\mathcal{U}(1, 5)$. Then, for each scene, we iteratively sample superquadric parameters. If the new superquadric intersects with the superquadrics already in the scene, we discard it and sample again. This procedure continues until there are as many superquadrics on the scene as determined in the initial sampling step. Each superquadric has its size parameters sampled from a continuous uniform distribution $\mathcal{U}(25, 76)$ and the shape parameters from $\mathcal{U}(0.01, 1)$ limiting the appearance of the rendered models to convex shapes, which are also more representative of the real world. We sample the x_0 and y_0 center coordinates from $\mathcal{U}(88, 169)$ while the z_0 coordinate

Table 1: Dataset summary.

#Superquadrics	1	2	3	4	5	Any
#Train Images	15882	16108	15930	15983	16097	80000
#Validation Images	3989	3944	4020	3948	4099	20000
#Test Images	3949	4023	3996	4059	3973	20000

is sampled from a tighter region $\mathcal{U}(100, 150)$. This is done to constrain the vertical overlap between the superquadrics in the scene.

Along with the range image we also render a ground truth segmentation mask image of the scene, by coloring the different visible parts of the superquadrics with a different shade of gray. This ground truth information is used for training and evaluating the segmentation model.

3.3. Dataset totals

The complete dataset contains 120000 rendered scenes and corresponding segmentation masks. We also store range images of individual superquadrics in each scene in the dataset along with their parameters. For the experiments we split the dataset into three disjoint parts: for training, validation and testing. We use the training set to learn the parameters of our models, the validation set to observe over-fitting issues during training and the test for the final performance evaluation. A few illustrative examples from the generated dataset together with the corresponding segmentation masks are shown in Fig. 2 and a high-level summary of the dataset and experimental setting is given in Table 1.

4. Superquadric recovery methodology

In this section we now present our approach to segmentation and recovery of multiple superquadrics using CNN models.

4.1. Segmentation

As our range images contain multiple objects of the same class (i.e., superquadric-like objects), we resort to instance segmentation to identify parts of the range images belonging to structures that can be represented with superquadrics. One of the most popular models for instance segmentation is Mask R-CNN [7], which operates in a two-stage fashion. In the first stage, it uses a region proposal network (RPN) that finds candidate regions in the image. In the second stage, the final predictions are made. Here, three model heads are used: one for detection (two-class classification: object present or not), one for regression of the bounding boxes, and one for prediction of the binary segmentation mask.

In our implementation, we use a ResNet-101 [8] backbone as the feature extractor along with a feature pyramid network (FPN) that makes it possible to exploit multiple scales of the feature maps. These features get fed trough a region proposal network which predicts object scores and their bounding boxes at each feature position. The predictions are then filtered by a non-maximum suppression algorithm, which removes overlapping bounding boxes.

The RPN bounding boxes and the FPN features get combined using the RoIAlign operator and fed into the three network heads to obtain the final class (object present or not), bounding box, and binary mask for each region proposal. Here the classification scores are used for the elimination of any background instances. For more information on Mask R-CNNs, the reader is referred to [4, 3, 24, 15, 7].

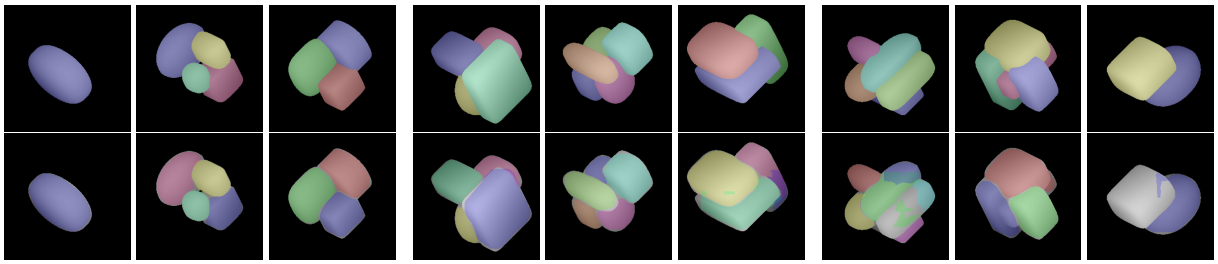
Table 2: Architecture of the CNN regressor used for superquadric parameter estimation.

#	Output size	Layer operation	#kernels, size, stride
1	128×128	Conv2D+BN+ReLU	$32,\ 7 \times 7,\ s2$
2	128×128	Conv2D+BN+ReLU	$32,\ 3 \times 3,\ s1$
3	128×128	Conv2D+BN+ReLU	$32,\ 3 \times 3,\ s1$
4	64×64	Conv2D+BN+ReLU	$32,\ 3 \times 3,\ s2$
5	64×64	Conv2D+BN+ReLU	$64,\ 3 \times 3,\ s1$
6	64×64	Conv2D+BN+ReLU	$64,\ 3 \times 3,\ s1$
7	32×32	Conv2D+BN+ReLU	$64,\ 3 \times 3,\ s2$
8	32×32	Conv2D+BN+ReLU	$128,\ 3 \times 3,\ s1$
9	32×32	Conv2D+BN+ReLU	$128,\ 3 \times 3,\ s1$
10	16×16	Conv2D+BN+ReLU	$128,\ 3 \times 3,\ s2$
11	16×16	Conv2D+BN+ReLU	$256,\ 3 \times 3,\ s1$
12	16×16	Conv2D+BN+ReLU	$256,\ 3 \times 3,\ s1$
13	8×8	Conv2D+BN+ReLU	$256,\ 3 \times 3,\ s2$
14	16384	Flatten	N/A
15	8	Dense	N/A

4.2. Parameter estimation

Once the scene is segmented and superquadric-like objects are identified in the input images, we feed the predictions into a CNN regressor for parameter estimation. We follow the work of [18] and use a regression model derived from the popular VGG architecture [19]. The model is designed as a 13 layer CNN with a fully-connected layer of size 8 on top. Each conv layer is followed by batch normalization and a ReLU activation, which reduces overfitting and allows the model to better generalize. The model is summarized in Table 2.

The input to the CNN regressor is a range im-



(a) 100% (b) 93.8% (c) 93.3% (d) 86.9% (e) 80% (f) 76.7% (g) 55.7% (h) 52.5% (i) 40%

Figure 3: Predicted segmentation masks from the Mask R-CNN model. The images are ordered in columns of three. Three good predictions (left), three average predictions (middle) and three bad predictions (right). In the first row we show range images with overlaid ground truth masks. The second row shows masks obtained with our segmentation model. Under the images we also report the mAP value for the segmentation. Most of the predictions are sufficient, even in the average subsection of the predictions. We observe that fine details are elusive to the model, such as disconnected masks (h) or narrow subparts of masks (e,f). Best viewed in color.

age containing a single superquadric-like instance and the output is a prediction of 8 parameters describing the size, shape and position, of the superquadric representing the input data, i.e., $\mathbf{y} = [a_1, a_2, a_3, \varepsilon_1, \varepsilon_2, x_0, y_0, z_0]$. Different from [18], the inputs to our model are not necessarily complete superquadrics, but automatically segmented range data, where parts of the object may be occluded due to overlap with other objects in the scene. Thus, we account for this in our training procedure and learn the parameters of our regressor by utilizing occluded data. As we show in the experimental section this allows us to quite efficiently estimate superquadric parameters even if part of the data is missing either due to occlusions or errors in the segmentation steps.

5. Experiments and results

5.1. Instance segmentation

The Mask R-CNN backbone is initialized with a ResNet-101 structure [8], pre-trained on the MS COCO dataset [16]. The training is split into two stages. In the first stage, we lock the training of the backbone and set the learning rate to 10^{-3}, with momentum of 0.9. In the second stage we unlock the backbone and fine-tune the network with a smaller learning rate of 10^{-4}. We present the standard mean average precision (mAP) scores of the instance segmentation in Table 3, as used in the COCO challenge. The model is trained on $80k$ training range-images of superquadric scenes, with a batch size of 2. We use an additional $20k$ images for validation and $20k$ images for testing. The model is trained on an NVIDIA GTX TITAN X GPU.

Table 3: Instance segmentation results. mAP_{50} and mAP_{75} denote scores computed at 50% and 75% IoU respectively, while mAP denotes the mean average precision averaged over IoU values from 50% up to 95%, taken at 5% steps.

mAP	mAP_{50}	mAP_{75}
85.57	97.33	95.95

In Table 3 we report the segmentation results using our Mask R-CNN model. We can see that average precision at Intersection-over-Union (IoU) thresholds 50% and 75% are higher than the averaged mAP over multiple IoU thresholds. This indicates that the model fail only at the highest intersections, segmenting the objects with good detail and precision.

In Figure 3 we present some examples of predicted masks for the training set. Most of the objects have been segmented with sufficient precision. On average, the model only misses smaller and highly occluded objects (Figures 3e and 3f). It also struggles with objects visually cut in half because of overlaps (Figure 3h). In these cases we either get multiple separate instance segments or the model fails to detect one of the parts completely. We suspect this might be caused by significant bounding box overlap between the foreground and background objects. The latter causing the former to get suppressed by the Mask R-CNN non maximum suppression algorithm.

5.2. Parameter prediction

We initialize the parameter prediction model with the weights from [18], as the same neural network architecture was used in that work. To train the pa-

Table 4: Parameter-prediction performance. The table shows MAE scores for each of the 8 superquadric parameters. The rows show results on different subsets of segmented range images test set, defined by the number of superquadrics the parent scene. The "All" row shows scores averaged over the entire set.

#sq	Dimensions [0-256]			Position [0-256]			Shape [0-1]	
	a_1	a_2	a_3	x_0	y_0	z_0	ϵ_1	ϵ_2
All	1.134	1.187	1.248	1.953	1.864	2.639	0.017	0.017
1	0.515	0.555	0.537	0.957	0.925	2.154	0.009	0.008
2	0.681	0.736	0.728	1.165	1.093	2.181	0.011	0.010
3	0.930	0.984	1.036	1.528	1.448	2.386	0.013	0.013
4	1.580	1.646	1.708	3.066	2.966	3.110	0.026	0.025
5	1.201	1.241	1.357	1.776	1.669	2.685	0.017	0.017

rameters of the model we use the ADAM minibatch stochastic gradient descent optimisation algorithm, which minimizes the MSE loss. We set the learning rate of the algorithm to 10^{-3} and keep the rate constant during training. As already indicated above, we use the segmentations produced by our Mask R-CNN model as the basis for the training to make the model robust to missing data. We only train on segmentations with an IoU higher than 50% compared to the ground truth masks. The model is trained for 63 epochs, with varying batch sizes constructed always from batches of 4 scene range images, giving us a maximum batch size of 20 segmented range images. We report performance for the CNN regressor in terms of the Mean Absolute Error (MAE) between the predicted and ground truth parameters. This measure was sufficient for our problem, since we predict superquadric parameters for superquadric visualizations, where the matching of parameters correlates with the 3D matching of the objects.

In Table 4 we present the MAE scores for each parameter on a test set of 20000 images. In addition to the MAE score for the entire test set, we also show separate MAE scores for scenes with different numbers of superquadrics. On average the model performs very well, predicting position and size in the order of one pixel accuracy compared to the $[0, 256]$ range of possible values. The shape parameters ε_0 and ε_1 also achieve about 0.017 mean absolute error which is also small compared to the $[0, 1]$ range of possible values. The model performs better in scenes with a smaller number of superquadrics since more superquadrics in the scene typically result in greater intersections and occlusions. Table 4 shows an almost monotonous increase in MAE as the number of superquadrics is increased, the only disparity is a larger error in scenes with 4 objects than in scenes with 5.

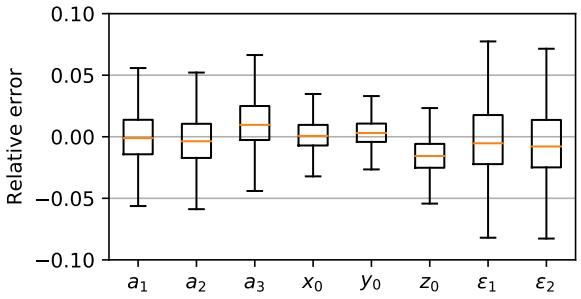


Figure 4: Box-and-whiskers plots of the relative error for each parameter.

In Figure 4 we show box-and-whiskers plots of the relative errors between ground truth and the predicted parameter values over the entire test set of segmented range images. We see that most of the error mass is close to the mean. The positional parameters are predicted with especially small variance in their errors. We also observe that the z axis size parameters are on average slightly overestimated. This seems to get compensated by an underestimation of the z axis position, thus aligning the top surface of the ground truth and the predicted superquadrics.

Scenes with larger numbers of superquadrics are harder to segment, occasionally giving our parameter prediction model highly corrupted segmentation masks, that can either blend range information from multiple objects into one segmented range image or return smaller subsets of the actual masks. On such corrupt segmented range images our prediction model naturally performs much worse than on cleaner segmentations, resulting in a somewhat heavy-tailed error distribution. We show this in Figure 5 where we plot the error distribution for all parameter predictions and subsets over the number of superquadrics in the scene. We also show how our segmentation model performs on each subset by showing the distribution of IoU values for its pre-

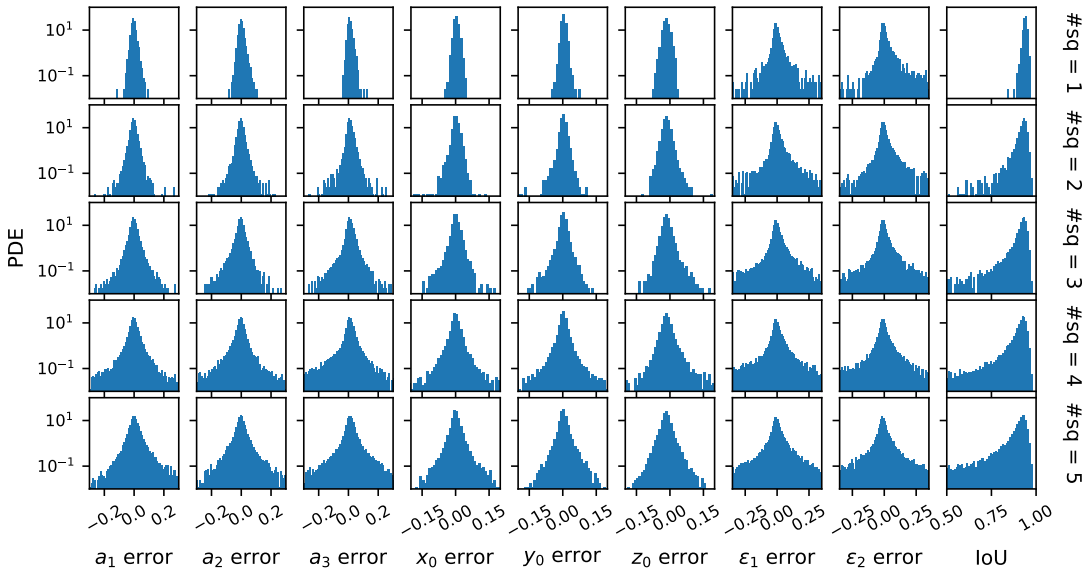


Figure 5: Our methods error distribution for each parameter. Each row shows results obtained from the 5 subsets scene images, each with a different number of superquadrics in its scenes. We also add the last column showing the IoU distribution of the predicted masks with Mask R-CNN.

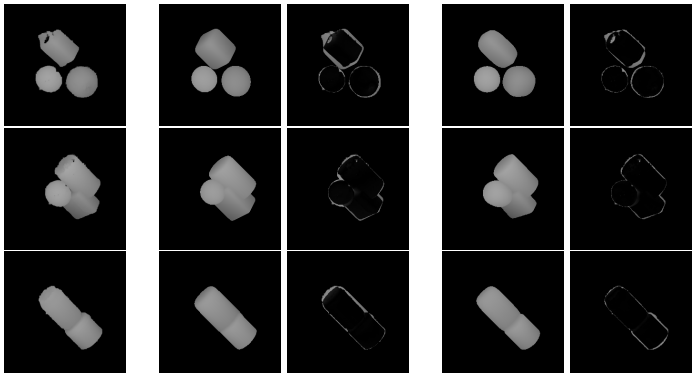


Figure 6: Qualitative comparison with the state-of-the-art: Input range images of (scanned) real-world objects (first column), Our reconstructions (second column), Absolute difference between the ground truth and our reconstruction (third column), Reconstructions by Leonardis et. al. [12, 10] (fourth column), Absolute difference between the ground truth and reconstruction by Leonardis et. al. [12, 10] (last column).

dicted segmentations. The distributions move away from a Gaussian shape quickly when more than one superquadric is present in the scene. The tails become larger when we increase the number of objects in the scene. As mentioned earlier, this can be explained by the inefficiency of the segmentation model, as the model also performs worse with greater numbers of objects in the scene - the IoU distribution becomes more and more skewed, with a heavier tail.

We also compare our approach to the state-of-the-art segmentation and superquadric recovery method from [12, 10] on range-images of real objects. For this experiment, we used range-image scans of real objects taken by Oblak et. al. for their work in [18]. We constructed range image scenes of multiple object by shifting the original images in pixel space and combining them using the max operator. The original range images, and their superquadric reconstructions using our approach and the state-of-the-art method from [12, 10] are shown in Figure 6. The iterative method from [12, 10] performs comparably to our solution, as we can see from the examples. Our method achieved 2.79 MAE calculated over all pixels differences from all pairs of ground truth and reconstructed images while [12, 10] scored 1.78. However, we note that the iterative algorithm of the original

method results in much higher processing times. Our method performs similarly in terms of reconstruction quality, but computes the segmentations and parameter predictions with a $100\times$ speed up over the state-of-the-art approach. Specifically, the iterative method converges in about 10 s on one image while our method needs 0.11 s on a GPU. While our methods advantage against [12, 10] is that we can parallelize its computations, it still performs faster on a single threaded CPU with about 5 s per image.

6. Conclusion

We have presented a CNN-based solution for segmentation and recovery of multiple superquadrics from range images. We have shown that the designed solution is able to efficiently decompose complex depth scenes into smaller parts that can be modelled by superquadric models. Our approach was shown to produce scene reconstruction on par with a state-of-the-art method from the literature, while ensuring a significant speed up in processing times. As part of our future work, we will extend the solution to account for rotated superquadrics as well.

Acknowledgements

This research was supported in parts by the ARRS (Slovenian Research Agency) Project J2-9228 "A neural network solution to segmentation and recovery of superquadric models from 3D image data", ARRS Research Program P2-0250 (B) "Metrology and Biometric Systems" and the ARRS Research Program P2-0214 (A) "Computer Vision".

References

[1] R. Bajcsy and F. Solina. Three dimensional object representation revisited. In *ICCV*, pages 231–240, 1987.

[2] F. P. Ferrie, J. Lagarde, and P. Whaite. Darboux frames, snakes, and super-quadrics: geometry from the bottom up. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(8):771–784, Aug 1993.

[3] R. Girshick. Fast R-CNN. In *ICCV*, Dec 2015.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, June 2014.

[5] A. Gupta and R. Bajcsy. Volumetric segmentation of range images of 3d objects using superquadric models. *CVGIP: Image Understanding*, 58(3):302 – 326, 1993.

[6] A. Gupta, G. Funka-Lea, and K. Wohn. Segmentation, Modeling And Classification Of The Compact Objects In A Pile. In D. P. Casasent, editor, *Intelligent Robots and Computer Vision VIII: Algorithms and Techniques*, volume 1192, pages 98 – 109. SPIE, 1990.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, pages 2980–2988, Oct 2017.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[9] T. Horikoshi and S. Suzuki. 3D parts decomposition from sparse range data using information criterion. In *CVPR*, pages 168–173, June 1993.

[10] A. Jaklič, A. Leonardis, and F. Solina. *Segmentation and recovery of superquadrics*. Kluwer, 2000.

[11] H. Jiang and J. Xiao. A linear approach to matching cuboids in rgbd images. In *CVPR*, pages 2171–2178, 2013.

[12] A. Leonardis, A. Jaklič, and F. Solina. Superquadrics for segmenting and modeling range data. *IEEE TPAMI*, 19(11):1289–1295, 1997.

[13] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE IV*, 2011.

[14] R. Li, X. Jia, J. H. Lewis, X. Gu, M. Folkerts, C. Men, and S. B. Jiang. Real-time volumetric image reconstruction and 3D tumor localization based on a single x-ray projection image for lung cancer radiotherapy. *Medical Physics*, 37(6Part1):2822–2826, 2010.

[15] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 936–944, July 2017.

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014.

[17] C. Niu, J. Li, and K. Xu. Im2Struct: Recovering 3D Shape Structure from a Single RGB Image. In *CVPR*, 2018.

[18] T. Oblak, K. Grm, A. Jaklič, P. Peer, V. Štruc, and F. Solina. Recovery of Superquadrics from Range Images using Deep Learning: A Preliminary Study. In *IWOBI*, pages 45–52. IEEE, 2019.

[19] O. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, volume 1, page 6, 2015.

[20] D. Paschalidou, A. O. Ulusoy, and A. Geiger. Superquadrics revisited: Learning 3D shape parsing beyond cuboids. In *CVPR*, pages 10344–10353, 2019.

[21] L. Pedersen. Science target assessment for Mars rover instrument deployment. In *IROS*, volume 1, Sep. 2002.

[22] A. P. Pentland. Automatic extraction of deformable part models. *International Journal of Computer Vision*, 4(2):107–126, 1990.

[23] N. Raja and A. Jain. Obtaining generic parts from range images using a multi-view representation. *CVGIP: Image Understanding*, 60(1):44 – 64, 1994.

[24] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[25] J. Slabanja, B. Meden, P. Peer, A. Jaklič, and F. Solina. Segmentation and reconstruction of 3D models from a point cloud with deep neural networks. In *ICTC*, 2018.

[26] F. Solina and R. Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE TPAMI*, 12(2):131–147, 1990.

[27] S. Tulsiani, H. Su, L. J. Guibas, A. A. Efros, and J. Malik. Learning shape abstractions by assembling volumetric primitives. In *CVPR*, pages 1466–1474, July 2017.

[28] Y. Zhou, K. Yin, H. Huang, H. Zhang, M. Gong, and D. Cohen-Or. Generalized cylinder decomposition. *ACM Trans. Graph.*, 34(6):171:1–171:14, Oct. 2015.