Learning View Invariant Semantic Segmentation for UAV Video Sequences*

Abhay Venkatesh † Glenn Fung‡ Vikas Singh§

Abstract

There are several excellent image datasets with pixel-level annotations available in the computer vision community to enable semantic segmentation of scenes, motivated by applications such as autonomous driving. Examples of such datasets include Cityscapes [5] or Vistas [9]. However, data is scarce for training computer vision models for unmanned aerial vehicles (UAVs), also known as drones. We propose a framework to compensate for this lack of training data and still obtain generalizable models for segmentation of images/videos acquired by drones. We start with street view annotations, i.e., pixel-labeled images captured at the street-view level – either provided in a dataset or generated by running an existing "street-view" semantic segmentation model. Then, we consider images at varying poses or elevation angles captured by a drone. By leveraging good segmentations of the street-view data, we train parameters of a "helper" network that learns to nominally change the internal feature representations of a segmentation neural network to yield good segmentations for viewing angles other than street-view pose, acquired from a drone. We show promising preliminary results on a synthetic dataset obtained from the Unreal engine.

1 Introduction

Despite the availability of a wide variety of "street perspective" datasets in computer vision dealing with pixel-level annotations of objects and scene elements encountered, e.g., by an ego-centric camera or an autonomous vehicle, the availability of similar datasets in fully annotated form for unmanned aerial vehicles remains limited at best. In part due to this gap in the suite of resources available to the community at large, automatic scene understanding based on image/video data acquired from UAVs has been relatively unexplored. The technical difficulty here is that taking deep neural network models trained on street perspective data and applying it to UAV video sequences yields unsatisfactory results at best, and unusable results at worst – this is partly due to potentially significant differences in perspective or pose of the scene. Therefore, the internal representations of a convolutional neural network (CNN) that will work well on data acquired at street level may not necessarily transfer well to image understanding tasks defined on aerially acquired images. The goal of this paper is to investigate mechanisms by which the abundance of pixel-level annotated data for the street-perspective setting can be used in conjunction with large amounts of unlabeled UAV acquired data in a way that yields good segmentations, independent of the perspective of the camera.

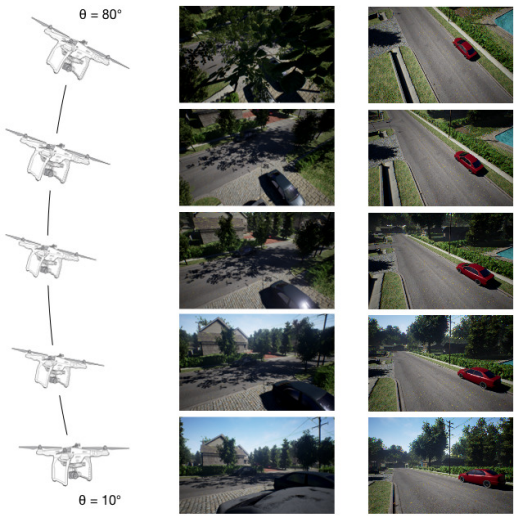

Figure 1: Data collected from Unreal Engine [1] to emulate video from a UAV.

In Figure 1, we show a sequence of images that a drone might record while flying around. For demonstration, we only show five images with perspective changes between them. While the figure shows significant changes between successive frames, in practice, the sequence captures perspective changes that are much finer, i.e., the angle denoting the change in perspective is smaller. Notice that between any *successive* pair of image frames in this type of sequence, we only see a small change in the scene. However, between the frame at $\theta = 10^o$ and $\theta = 80^o$, the change in perspective is significant. Even if a model for performing semantic

*Partially supported by funding from a collaboration between American Family Insurance and the University of Wisconsin Madison

†University of Wisconsin-Madison
‡American Family Insurance
§University of Wisconsin-Madison

segmentation works well at street view, it is unlikely to expect that the internal representations derived in a CNN for segmentation for that perspective will automatically yield good segmentations for $\theta = 80^o$. In fact, our experiments suggest that the segmentation results are often quite poor (see Figure 2).

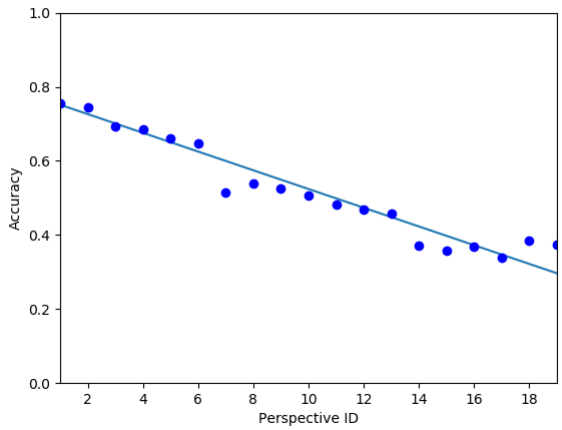

Figure 2: Deteriorating accuracy of a segmentation model trained only on street view.

Relevance to the insurance domain. In the last few years, there is a surge of interest in the deployment of Unmanned Aerial Vehicles (or drones) for several applications in the insurance industry. Collection of data pertaining to to-be-insured properties a key requirement in assessing risk in the workflow. This information is used to assess, among other things, premium costs and property damage for a claim. Therefore, using drones to gather information (images, video, and other property data), a process traditionally performed by humans during costly and time-consuming inspections could significantly impact the cost and the efficiency of the overall process. But automated analysis of drone images remains challenging due to the acquisition and manual annotation of the data. At the very least, drone images must be monitored due to privacy concerns. What makes this task particularly difficult is that large-scale data are difficult to collect because the FAA forbids flying drones over uninvolved individuals who have not provided consent. While a nominal amount of such data collection is possible under controlled circumstances, even apart from the acquisition issues, manual annotation of a large number of video frames poses challenges. Mitigating these challenges due to data acquisition and annotation is one target application of the ideas described in this work.

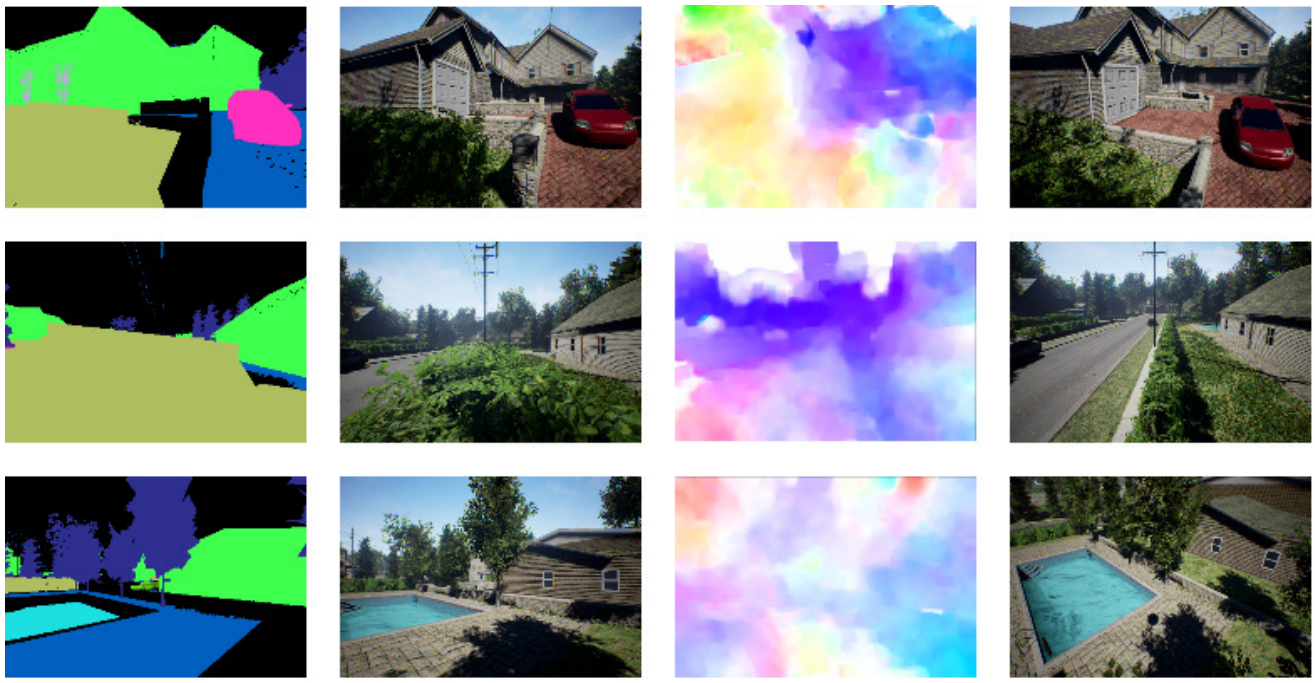

Figure 3: Optical flow captures the motion between the frames in our sequence of frames from street to top view. Using the optical flow, we can synthesize annotations for scenes that do not have segmentations.

Main Idea. Our main idea is based on the following observation. Assume that our segmentation engine works well for images acquired at perspective, $\theta = 10^o$. This means that for this pose angle, the internal representations of the image in the CNN are appropriate. Let us now change the pose angle slightly to $\theta = 15^o$. Perceptually, if the segmentation quality deteriorates, this is primarily because the internal representations are not ideal for this pose angle, although it is reasonable to expect that the ideal internal representations for $\theta = 15^o$ should still be "close" to those that work well for $\theta = 10^o$. This means that if the convolutional filters were slightly adapted, obtaining modified representations that work well for $\theta = 15^o$ may be possible. The difficulty is that we do *not* have ground-truth segmentations for $\theta = 15^o$. Fortunately, since the change between these two views is small, their changes can be calculated via an optical flow procedure [13] [6][11].

With this in hand, the ground truth segmentations for the image at $\theta = 10^o$ could simply be propagated to provide a good proxy for the ground truth segmentation at $\theta = 15^o$. The key issue is to design mechanisms or a recipe to adapt the convolutional layers to modify the native weights to be better suited for a given θ. We note that it makes sense to run a scale-invariant feature transform [8] to extract features, followed by calculating a perspective transformation, as opposed to calculating an optical flow. This is because in our training setting, only our camera's position is shifting and not the relative position of the objects. We did try to use this method. However, experimentally, we found that optical flow estimates were much more representative of the type of view transformations we are interested in.

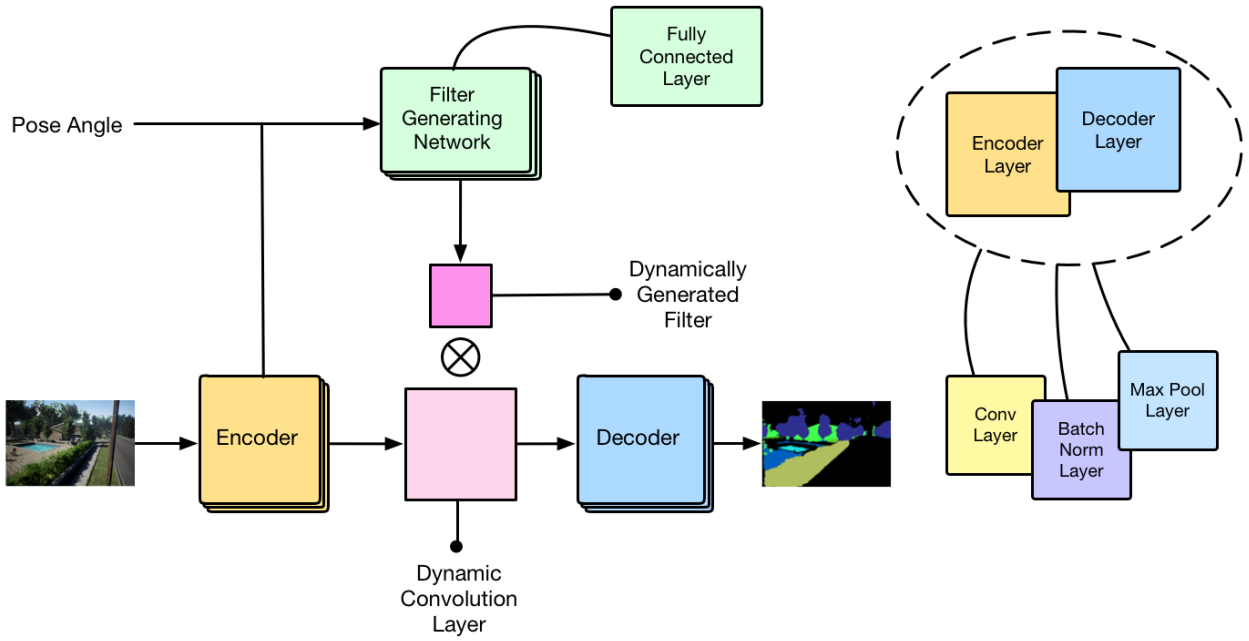

Figure 4: Overview of our architecture. We encode features from an image, and then we use the encoded features and the image's pose angle to generate a filter. The filter is convolved over the encoded features and then decoded to produce a segmentation.

The main **contribution** of this work is to demonstrate an efficient procedure to accomplish the above goal and show its efficacy in a dataset where the ground-truth segmentation for all perspective views are indeed available. We remark that recently a number of groups have also shown the value of using synthesized or simulated data from game engines for various problems in computer vision [12].

2 Problem Specification and Algorithm

Consider a convolutional neural network (CNN) Ψ which is designed for a segmentation task. We denote the encoded features as $\mathbf{x} \in \mathcal{X}$: these are derived from the input image based on the internal weights of the network. We aim to learn a ϕ such that given a $\theta \in \Theta$,

$$\phi : \mathcal{X} \times \Theta \to \hat{\mathcal{X}}, (\mathbf{x}, \theta) \to (\hat{\mathbf{x}}) \tag{2.1}$$

such that our primary segmentation engine Ψ can produce a good segmentation output $s \in \mathcal{S}$ on our desired image from $\mathcal{I}_\theta$ using $\hat{\mathbf{x}}$ instead of $\mathbf{x}$

$$\Psi : \mathcal{I} \to \mathcal{S} \tag{2.2}$$

We assume that θ is either provided by sensors on the acquisition device or can be estimated from the image.

2.1 Architecture of the learning model We describe our training procedure and neural network architecture (see Figure 4) [1]. We first train a segmentation engine on street view assuming that either pixel level annotations are available or the network is pre-trained on the Cityscapes dataset and works well for street-view images. We choose a deep encoder-decoder network inspired by the *SegNet* [2] model based on its performance on our data. The encoder and decoder modules are a number of sandwiched convolutional, batch normalization [7] and max pooling layers described in that paper. We will refer to this construction as our *primary* segmentation engine Ψ.

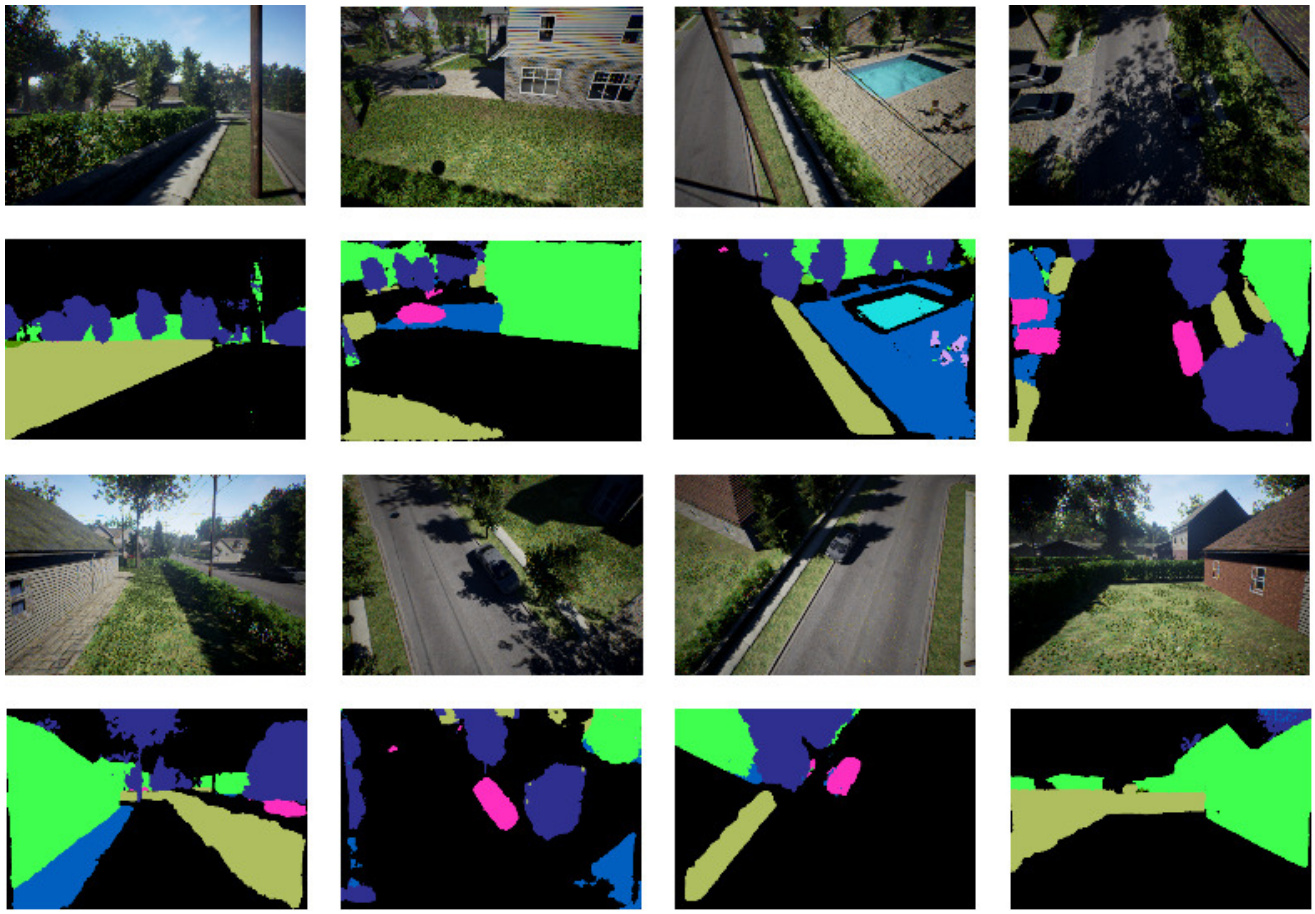

Figure 5: Semantic segmentation outputs on diverse perspectives and scenes. Our network is able to perform well on a variety of novel viewpoints even though annotated data was primarily available only for street-view.

Now, to model ϕ, our solution is roughly inspired by the Dynamic Filter Network [3]. We consider a list of 20 different poses corresponding to $\theta_i \in \Theta = \{\theta_0, ..., \theta_{19}\}$. We remark that this varying parameter need not necessarily be the pose angle, and may correspond to another covariate, such as depth. We use θ_i, and the encoded features $\mathbf{x}$ from our primary segmentation engine. For each $(\mathbf{x}, \theta)$, we seek to "generate" a filter that will be used to convolve over the encoded features $\mathbf{x}$ to produce the corresponding transformed features $\hat{\mathbf{x}}$. Our loss function will measure how well these transformed

[1] The implementation of the architecture is available at https://github.com/abhay-venkatesh/invariant-net

features, when passed through the decoder, yield a good segmentation output. The reader will notice that for all pose angles other than street-view, we do *not* have pixel-level annotations. Here, we exploit the continuity between successive image frames. Since the two successive images correspond to a small camera tilt, a dense optical flow [11] can provide information on pixel-level correspondences between the two images (see Figure 3). Then, the known segmentations can simply be propagated between successive frames yielding a good proxy for those perspectives where the segmentation is unavailable. This helps setup the loss function. We perform mini-batch training to train both the segmentation engine and the helper network.

2.2 Dataset Design. We now describe our dataset and the data collection procedure. To perform our experiments and obtain an accurate evaluation of the proposed procedure, we synthesize photo-realistic data from the Unreal Engine [1], utilizing the UnrealCV plug-in [10] (see Figure 1). [2] We decided to use this dataset to find a convenient way to obtain ground truth data for all perspective views – while the Unreal Engine enables us to generate ground truth segmentations for any perspective, we only use the ground truth segmentation from the street view in order to emulate the real-world scenario. In practice, such data will be collected via a drone. When applying our model to work on the real world, we would train our segmentation model on some dataset for street-view segmentation, and then use data from a drone on various pose angles to train our model to perform segmentation in a perspective invariant manner. To summarize, we collect a dataset with 20 perspectives, but to model the real-world setting, we assume that do not have ground truth semantic segmentations for all perspectives that a drone encounters. But using optical flow, we can train our neural network without assuming access to ground truth semantic segmentation for every view available in the wild.

3 Experimental Results

In 6., we show some representative results from our experiments. We trained our network on 60,000 images and we were able to achieve 60% mean IoU over our validation set across the range of 20 different perspectives. In our dataset, a majority (40,000) of the images were trained on pose angles that were street view or slightly higher than street view. 20,000 images were picked from non-street view pose angles and the semantic segmentation annotations were produced using an optical flow procedure from OpenCV [4]. Our results indicate that the helper network we trained was able to transform the features using the given pose angle in a way that the overall segmentation results remain good and almost invariant to the pose angle. Further, the relatively small amount of non-street view data indicates the ability of our architecture to quickly generalize to other views. This is ideal for the UAV case where the assumption is that we primarily will have street-view semantic segmentation data for training.

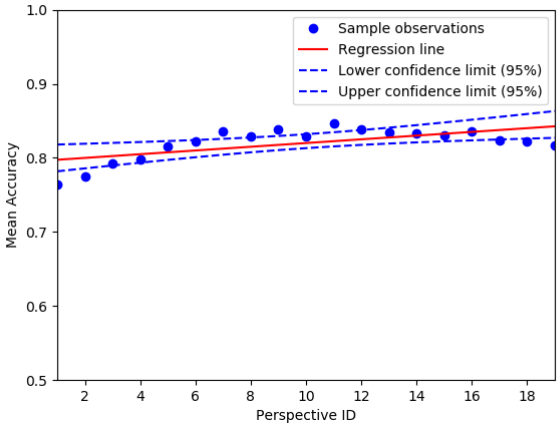

Figure 6: Our network is able to display robustness to perspective change. The figure displays mean accuracies tested over 2000 unique images. Perspective ID 1 refers to (near) street view, while perspective ID 19 refers to a pose angle close to a top view of the scene.

4 Discussion

In this paper, we demonstrated that learning a ϕ, i.e., a transformation recipe for pose-invariant semantic segmentation, based on given encoded features $\mathcal{X}$ and pose angle Θ is possible. Note that while the parameter Θ chosen here is a pose angle, it could easily be some other varying feature, e.g., depth. In general, we study the scenario where in the domain of interest we do not have well-annotated features or supervised data, but in a part of the dataset, supervised data is either available or an existing trained model can provide good results. Using such information, we can train a helper network to transform features that are appropriate for a part of the dataset in order to produce a good segmentation output on the full dataset. While these results are preliminary, future work will include a more extensive set of experiments to evaluate the performance of the model on actual UAV acquired data.

[2] Our codebase for extracting data from the Unreal engine is available at https://github.com/abhay-venkatesh/unreal-cv-data

Acknowledgments

We thank Luisa Polania Cabrera, Mona Jalal, Paul Coleman and Jiefeng Chen for contributing code or other help and discussion pertaining to this project.

References

[1] Unreal engine. *Epic Games*, 2018.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015.

[3] Bert De Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. *CoRR*, abs/1605.09673, 2016.

[4] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685, 2016.

[6] Berthold K.P Horn and Brian G Schunck. Determining optical flow. *Artificial Intelligence*, 17:185203, 1981.

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[8] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov 2004.

[9] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[10] Weichao Qiu, Fangwei Zhong, Yi Zhang, Siyuan Qiao, Zihao Xiao, Tae Soo Kim, and Yizhou Wang. Unrealcv: Virtual worlds for computer vision. pages 1221–1224, 10 2017.

[11] Sathya N. Ravi, Yunyang Xiong, Lopamudra Mukherjee, and Vikas Singh. Filter flow made practical: Massively parallel and lock-free. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[12] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. *CoRR*, abs/1608.02192, 2016.

[13] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate Optical Flow via Direct Cost Volume Processing. In *CVPR*, 2017.