Compressing Sensor Data for Remote Assistance of Autonomous Vehicles using Deep Generative Models

Daniel Bogdoll[1*] **Johannes Jestram**[2*] **Jonas Rauch**[2*]

Christin Scheib[2*] **Moritz Wittig**[2*] **J. Marius Zöllner**[1]

Abstract

In the foreseeable future, autonomous vehicles will require human assistance in situations they can not resolve on their own. In such scenarios, remote assistance from a human can provide the required input for the vehicle to continue its operation. Typical sensors used in autonomous vehicles include camera and lidar sensors. Due to the massive volume of sensor data that must be sent in real-time, highly efficient data compression is elementary to prevent an overload of network infrastructure. Sensor data compression using deep generative neural networks has been shown to outperform traditional compression approaches for both image and lidar data, regarding compression rate as well as reconstruction quality. However, there is a lack of research about the performance of generative-neural-network-based compression algorithms for remote assistance. In order to gain insights into the feasibility of deep generative models for usage in remote assistance, we evaluate state-of-the-art algorithms regarding their applicability and identify potential weaknesses. Further, we implement an online pipeline for processing sensor data and demonstrate its performance for remote assistance using the CARLA simulator.

1 Introduction

Motivation. Both SAE International (SAE) level 4 and 5 autonomous vehicles (AV) [12] operate without relying on a human driver. However, in such complex environments as road traffic, AVs will not be able to operate without failures in the foreseeable future. Remote assistance [9] allows remote operators to resolve situations, where AVs face problems they cannot resolve on their own by granting a remote operator access to the vehicle's state and sensor data [17]. Depending on the specific situation and the implementation of the remote assistance, the remote operators can either provide the vehicle information that enable it to continue the operation, or take over manual control.

Thus, to not overload network infrastructures, the volume of data that is sent must be drastically reduced. To achieve this, efficient compression of sensor data is elementary. Recent research on deep generative neural networks has achieved impressive results in image and lidar scan compression [10; 1; 32; 2; 11]. Deep generative models are an attractive choice as they work with diverse sensor data and can achieve higher compression rates while maintaining a better reconstruction quality than JPEG and MPEG [2]. Further, such models also work for lidar scans and outperform tree-based and JPEG-based approaches [47; 10; 48]. Lastly, deep generative models allow combining encoders and decoders of different depths with each other, making them well suited for applications with differing hardware capabilities on the AV and the remote operator side [45].

[1]FZI Research Center for Information Technology, Germany; {bogdoll, zoellner}@fzi.de

[2]Karlsruhe Institute of Technology, Germany; {johannes.jestram, jonas.rauch, christin.scheib, moritz.wittig}@student.kit.edu

[*]These authors contributed equally

Machine Learning for Autonomous Driving Workshop at the 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Virtual.

Gap to related work. Existing research focuses on compressing either image or lidar data [10; 47; 48; 1; 32; 2; 11]. However, there is a lack of studies regarding the choice of compression approaches for the remote assistance of AVs. Multiple approaches exist that compare generative models with traditional codecs [29; 14]. Instead of comparing just one generative approach to traditional engineered algorithms for image compression, we provide a comprehensive comparison in the field of autonomous driving between multiple generative models and traditional algorithms.

As point cloud and image data are complementary, a single compression approach for processing both sensor modalities in the application of remote assistance is beneficial but has not been demonstrated yet in the literature. Further, to the best of our knowledge, there is no study that empirically evaluates the reduction in volume of data that can be achieved by employing compressing a vehicle's sensor data using deep generative models.

Contributions. Therefore, our main motivation is to identify suitable approaches for compressing sensor data for remote assistance of autonomous vehicles. We thoroughly evaluate two state-of-the-art image compression approaches regarding their fit for remote assistance, using different datasets from the autonomous driving domain.

Based on several error metrics for reconstruction, we identify scenarios where generative compression performs either very good, or very bad. Additionally, we evaluate the reconstruction quality by performing object detection on the original as well as the reconstructed images.

We demonstrate a distributed online-pipeline for processing simulated image and lidar data, and evaluate its performance for our use case. Our chosen architecture allows camera and lidar data to be processed by the same pipeline, rather than having to perform individual compression passes.

We study the performance of the online pipeline with regards to its real-time capabilities, and highlight critical processing steps that affect performance. By implementing the online processing pipeline in the Robot Operating System (ROS) [40], we show the applicability of our approach on our test vehicle [53], that operates on the same framework.

Paper outline. The rest of this paper is structured as follows. Section 2 introduces generative-model-based approaches for image and lidar compression and applications of compression pipelines. Section 3 explains our model choice and implementation of the online pipeline. Finally, we evaluate the image and lidar compression approaches in Section 4, before concluding our work in Section 5.

2 Related Work

2.1 Approaches for Image Compression

Traditional, lossy image compression algorithms, such as JPEG [50], JPEG2000 [43] or HEVC [41] based BPG [4], are popular and commonly known. Nevertheless, neural networks have also been used for image compression for more than 20 years [26]. In 2006 Hinton and Salakhutdinov [25] introduced a deep autoencoder to convert high-dimensional data to low-dimensional codes. One key problem of the standard autoencoder is that it generates a fixed-length code for images with the same resolution, independent of the complexity. Toderici et al. [46] introduced a variable-rate image compression framework where LSTM models are used for both the encoder and the decoder. The results showed remarkable improvements compared to JPEG on the similarity metric SSIM. In recent years Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN) have reached a lot of attention due to their success in the field of image compression. GANs produce high quality, perceptual reconstructions, but with the danger of mode collapse. The reconstructed images by VAEs are often more blurry and not necessarily as visually appealing to humans, but due to their setup mode-collapse is not an issue.

Ballé et al. [2] introduced a VAE for image compression with an hierarchical prior to improve the entropy model. The data compressed by the prior can be used as side information for the compression of the latent representation with the entropy model. Thereby, the entropy model can adjust to different complexities of images. Based on this, Minnen et al. [33] extended the hyperprior to a mean and scale Gaussian distribution alongside an autoregressive component that predicts latents from their causal context to get a more accurate entropy model. Cheng et al. [11] further improved the model using discretized Gaussian Mixture Likelihoods to parameterize the distributions of latent codes.

GANs [21] have led to impressive results for learning intractable distributions with generative models. Santurkar et al. [36] show how GANs can be used for lossy image compression by adding an autoencoder to a GAN architecture.

Adversarial losses are also used in the field of image compression in the rate-distortion objective [31; 5]. Agustsson et al. [1] show impressive, subjective results in their user study with extremely high compression rates. Mentzer et al. [32] improve the distortion quality by the introduction of the hyperprior approach on the latents of Ballé et al. [2]. The inherent danger of mode collapse for GANs is tackled by the additional distortion loss, which optimizes the reconstructions on the pixel level and thereby penalizes a mode collapse.

2.2 Point Cloud Compression

Several directions for compressing 3D data, including lidar measurements, have been explored. Ochotta and Saupe [35] propose an approach that decomposes dense 3D surfaces into smaller patches represented as elevation maps and applies a shape-adaptive wavelet encoder on those patches. For dense point clouds, tree-based compression approaches have been proposed [37]. Golla and Klein [20] achieve real-time compression of point clouds by grouping points into larger voxels and applying compression on each voxel separately. They compute height-, color-, and occupancy-maps and compress those maps using different standard compression approaches, such as JPEG. However, lidar-generated point clouds tend to be sparsely populated. Therefore, the aforementioned approaches are not well-suited for our use case.

Tu et al. [47] suggest representing raw lidar scans as 2D matrices. They propose utilizing the knowledge that most lidars generate scans by rotating an array of lasers by exactly one revolution. This means that there is an inherent order to the array raw sensor data that the lidar produces, although it might differ between manufacturers. Utilizing this order allows the 2D matrix representation to be created relatively cheaply. To this 2D representation of the raw lidar data Tu et al. subsequently apply compression approaches, such as JPEG and MPEG [47], an RNN with residual blocks [48], and U-Net for optical flow interpolation between reference frames [49]. However, the RNN-based approach performs compression by sending data through the encoder as well as the decoder, making it not suitable for a distributed use case such as remote assistance.

Another recent line of work explores point cloud compression using deep generative models, such as VAEs and GANs. After transformation of raw lidar scans into 2D grids, represented as matrices [47], the encoder of a CNN-based VAE or GAN can compress the matrix analogous to an image. Conditional generation, i.e., reconstruction of a compressed scan, as well as compression rate have been shown to perform well with a VAE using such an ordered 2D representation [10].

2.3 Generative Data Compression Applications

There exist several publications on generative models for image compression. Regarding comparative studies Löhdefink et al. [29] implement the GAN approach of [1] and compare it to JPEG2000 with similarity metrics. Dash et al. [14] improve the GAN architecture and compare the results to traditional compression algorithms. Siam et al. [39] evaluate different approaches for image compression in the autonomous driving domain on semantically segmented images.

Outside of the autonomous driving community, end-to-end image compression based on generative models has been applied successfully to various other applications, such as facial images for surveillance [24] and general video compression [30; 22].

3 Approach

Our work is divided into two parts, an offline and an online segment. In the offline part we select two state-of-the-art generative models for image and lidar compression to evaluate the compression and quality of the reconstructions and compare them to traditional approaches. We implement these approaches in an online AV simulation to evaluate their real-world potential. Code and supplementary material is available at `https://github.com/daniel-bogdoll/deep_generative_models`.

Offline Image Compression. For remote assistance of AVs efficient data compression with a high quality reconstruction is necessary. Therefore, our objective is to evaluate models based on their

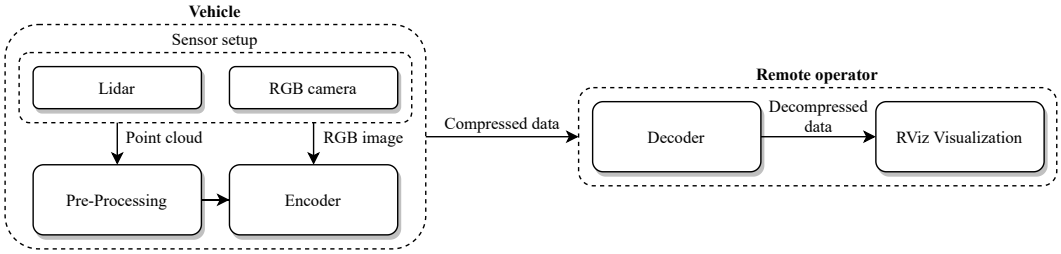

Figure 1: Overview of the online processing pipeline. The pre-processed lidar data passes through the same encoder and decoder architecture as RGB data.

rate-distortion-perception [6] properties. We include the perception category, since the receiving remote operator is human, and assess it based on a domain-specific object-detection metric.

To meet these mentioned requirements we select the VAE model of Ballé et al. [2] and the GAN model by Mentzer et al. [32] as they achieved state-of-the-art results in the rate-distortion for image compression. Both models take a 256×256 pixels resize of the original image as an input and have been validated on various datasets and metrics.

Offline Point Cloud Compression. To use a VAE or GAN for lidar compression, we pre-process the data according to Caccia et al. [10]. The intuition behind the pre-processing approach is to sort and down-sample the lidar scans into a tensor representation, so that the resulting structure closely resembles the structure of an RGB image. We further base our compression of lidar scans on a 2D transformation and use a VAE for compression. We perform compression on diverse lidar data sources, i.e., KITTI [16], Waymo Open Perception Dataset [42], and CARLA [15].

Online Compression. To evaluate and compare the performance of the aforementioned approaches with regard to their applicability in remote assistance systems, we design an online sensor data compression pipeline. We use ROS to interconnect the different sub-systems. Furthermore, we employ the CARLA Simulator as our sensor data source. CARLA is an open-source simulator for development, training, and validation of autonomous urban driving systems [15]. A schematic overview of the system is given in Figure 1. Examples of the CARLA environment and the decoded camera image are shown in the Appendix, Figure 8.

The AV in the CARLA simulator is equipped with a set of sensors such as a lidar and camera. The system consists of an encoder node on the AV side and a decoder node on the remote operator side. In case of compressing lidar data, an additional pre-processing node on the vehicle side is necessary. The sensor data is sent to the respective encoder node to compress the data. Afterwards, the decoder receives the data via a network connection and decompresses it.

4 Evaluation

4.1 Image Compression

Training. Models of the methods selected in Section 3 were trained using the KITTI [16] dataset, which consists of 7,481 RGB training images and 7,518 RGB test images with a resolution of 1242×375 pixels. During training, fragments with a size of 256×256 pixels were randomly cut out of the training images and fed to the corresponding model. Using this input size makes a good compromise between quality and compression rate. The VAE was trained with a batch size of 32 and a learning rate of 0.0001 for 100 epochs to achieve good convergence of validation metric values. The library compressai [3] was used to perform training. Rate-distortion-loss, consisting of MSE as distortion loss and bit-rate as rate loss, was used as a combined loss function for the VAE. It is calculated as follows:

$$\mathcal{L}_V = \mathcal{D}_V + \lambda \mathcal{R}_V = \mathcal{L}_{MSE} + \lambda \mathcal{L}_{bpp}. \tag{1}$$

The GAN was trained for 8 epochs using a learning rate of 0.0001 and a batch size of 4. As distortion loss, learned perceptual image patch similarity (LPIPS) [52] was used additionally. Therefore, the

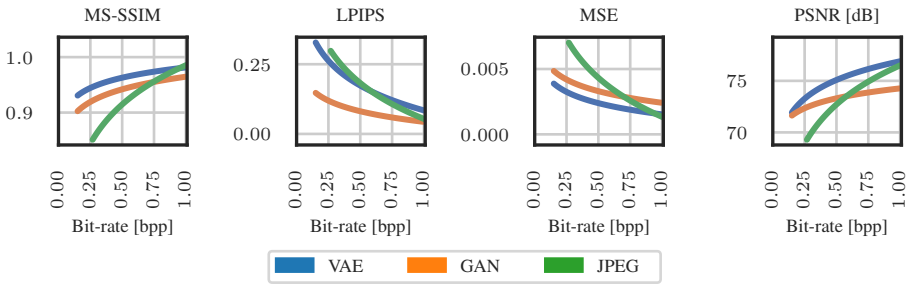

Figure 2: Comparison of VAE, GAN and JPEG2000 compression with the metrics MS-SSIM, LPIPS, MSE and PSNR. The performances are plotted over the bit-rate in bits per pixel.

loss function used is calculated by

$$\mathcal{L}_G = \mathcal{D}_G + \lambda \mathcal{R}_G = (k_m \mathcal{L}_{MSE} + k_p \mathcal{L}_{LPIPS}) + \lambda \mathcal{L}_{bpp} \tag{2}$$

where $k_m = 0.075 \cdot 2^{-5}$ and $k_p = 1$ are hyper-parameters chosen as in [32]. Several trainings with different values of $\lambda \in \{0.001, 0.0025, 0.05, 0.01, 0.05, 0.1\}$ were performed to achieve different quality levels in reconstruction and different compression rates. We executed all trainings on an NVIDIA GeForce GTX 1080 Ti.

Image Reconstruction Quality. We compare the methods at different quality levels and show what level of compression still provides sufficient quality for the use case of remote assistance. For the evaluation of the image reconstruction quality, the metrics MSE, LPIPS, peak signal to noise ratio (PSNR) and multiscale structural similarity index measure (MS-SSIM) [51] are taken into account. Both VAE and GAN image reconstruction results are evaluated and compared to JPEG2000, see Section 2.1. Figure 2 presents the metric values over bit-rates from 0 to 1.0 bits per pixel (bpp).

The JPEG compression delivers competitive results, but gets worse at lower bit-rates. This becomes especially clear at bit-rates below 0.3 bpp, where JPEG fails completely, since compression rates become unusable. However, we downsize every image to 256×256 pixels as done for VAE and GAN approaches, which negatively influences the reconstruction quality. For structural similarity, VAE and GAN yield similar trends. For MSE and PSNR, GAN and VAE approaches perform the same for lower bit-rates.
At higher bit-rates, the curves diverge. For LPIPS the GAN approach performs much better. Nevertheless, it should be noted here that the GAN was trained on LPIPS in addition to MSE in distribution loss which impacts the evaluation. To demonstrate the differences, multiple examples are presented.

First, Figure 3 displays a standard street scene from the KITTI dataset. It shows the original image, the JPEG compression and the reconstructions made by VAE and GAN approaches. In order to compare the results at the same compression rates, a similar target bit-rate was chosen for all compressions leading to equally large compressed data.

JPEG achieves the worst compression quality at these bit-rates. Details are hard to discern and the image looks fragmented. With the VAE approach, the reconstruction looks better, but more washed out. The GAN provides the best reconstruction, also showing details in higher sharpness. This can be seen with the tree in the background and with the gap between the right car's trunk and its side parts.

Further, two out-of-domain corner case situations, as described in [8], were used as input for the KITTI trained models to test the stability of the approaches. The first case is a night scene from the Nighttime Driving dataset [13] and the second case is a scene from the Fishyscapes benchmark [7] that has an unconventional object positioned in a street scene. Figure 10 in the appendix presents the results. Note that the target bit-rate is set lower than the one in Figure 3 to point out the methods' performances with lower bit-rates. While both generative approaches produce good results, poor performance can be seen with JPEG compression. At low bit-rates, colors may not be represented correctly, and there are also large fragments in the image. In contrast, the GAN approach still gives very good results, even for out-of-domain scenes for which it has not been trained specifically. Based on this, JPEG2000 was dismissed from further evaluation since GAN and VAE provide high-quality results with lower bit-rates than JPEG2000 can handle.

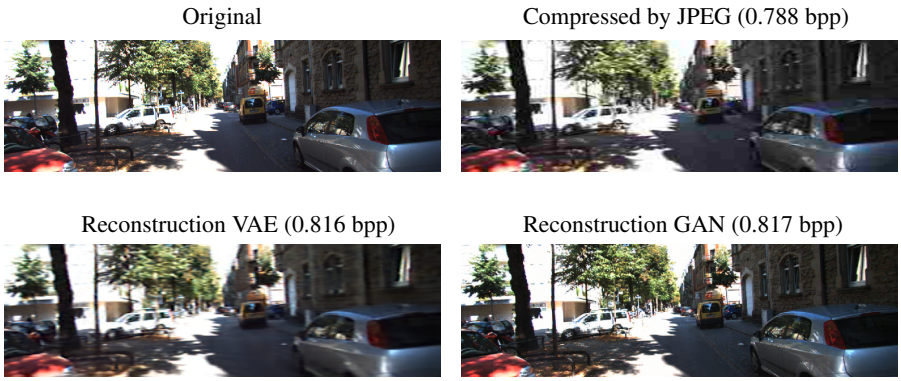

Figure 3: KITTI street scene with reconstruction comparisons. Target bit-rate is approx. $0.8\,bpp$.

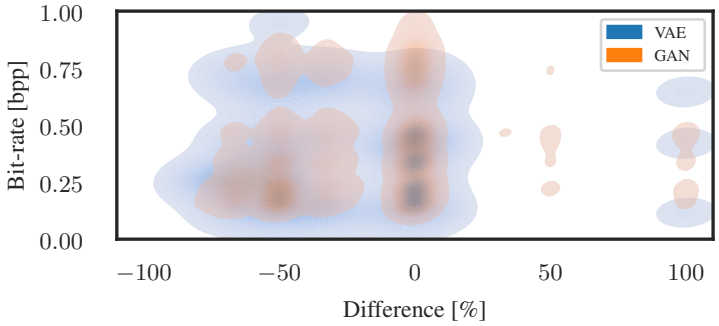

Figure 4: Relative error of the number of detected cars in the original image and the reconstructed image by VAE respectively GAN. The distribution is given over the bit-rate in bpp. Scott's rule [44] was used for smoothing with a scaling factor of 0.6. The darker the color in the graph, the higher the density of the values.

Object Detection Performance in Reconstructions. While the metric evaluation performed in Section 4.1 assesses the overall quality of reconstructions, the following evaluation refers to the remote operator's requirement, which is to understand the scene. Therefore, the extent to which object recognition is possible on the reconstructed image is investigated. For this purpose, an SSD [28] model with ResNet50 [23] as feature extractor trained on COCO [27] was used. Object detection for the class car was performed on both the original and the reconstructed images. For evaluation, the number of cars detected in the original and the reconstructed image were counted, followed by a comparison using the following formula for the relative error:

$$\textit{Relative error} = \frac{n_{recon} - n_{orig}}{n_{orig}} \times 100 \tag{3}$$

where n_{recon} is the number of cars detected in the reconstructed image and n_{orig} in the original image. Figure 4 shows this deviation in percent, i.e., -100% deviation means that out of n cars in the original image, none were detected in the reconstructed image. Images with $n_{orig} = 0$ were not considered. The minimum confidence score for the object detection network was chosen as 0.7. It is noticeable that across all quality levels, the deviations in the negative range are large, especially for VAE. Also, there is a large number of images where the GAN approach has almost no errors. Likewise, there are some images reconstructed by both VAE or GAN for which the object detection algorithm detects more objects as in the original counterparts, with the maximum difference being 100%. Overall, images reconstructed with the GAN allow for better object recognition of cars.

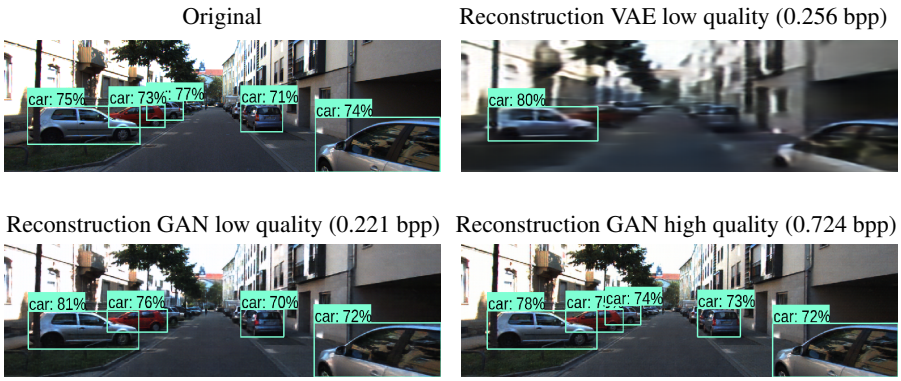

Figure 5: Object detection example performed in the original image and several reconstructions made by GAN and VAE with differing bit-rates.

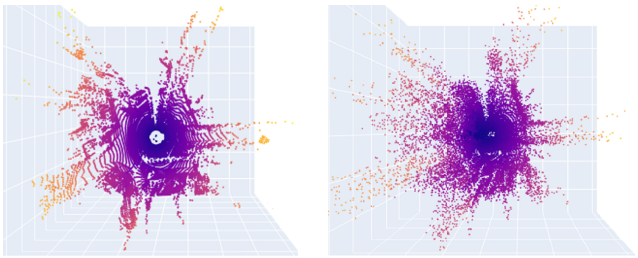

Figure 6: VAE point cloud reconstruction. Left: Original, transformed point cloud. Right: Reconstruction with a bit-rate of 1.83 bpp. The scenery is viewed from above.

Figure 5 shows an example of the object detection performance. The VAE compression with low quality performs inadequately with the pre-trained object detection model. However, at the same bit-rate the GAN compression results in a better object detection, improving with a higher bit-rate.

Considering that objects in the distance might be of less importance, even lower bit-rates in the range of 0.2 to 0.3 bpp are possible with the GAN. At such low rates, the VAE compression performs insufficient. Therefore, using a GAN trained with low λ values is preferable and achieves high-quality reconstructions with a high compression.

4.2 Point Cloud Compression

Early performance tests, which are further elaborated in Section 4.3, have shown the significantly higher transmission rates of VAEs. Therefore, we focus on VAEs for point cloud compression. The combined inference time for both camera and lidar GAN processing would be too high for the targeted real-world application. The VAE approach introduced in Section 3 was trained and tested on KITTI lidar data. The same training parameters as in Section 4.1 were used. As input the VAE receives the preprocessed point clouds with a shape of 512×64, since a general resizing of the point clouds as it is done for images is not possible as the channels contain distances and not color value information. Due to the different information type, the bit-rate increases as well.

Despite using the same VAE architecture as used for image compression, the results in compressing and reconstructing point clouds are surprisingly good. We measure the point-wise reconstruction quality with the standard euclidean distance metric. For a target bit-rate of 1.8 bpp, the mean euclidean distance is between $0.3\,\mathrm{m}$ and $0.5\,\mathrm{m}$, while points close to the center have a smaller distance to their counterparts than points located further away.

Figure 6 shows that the typical lidar rings are no longer reconstructed, and that there is increased noise. This can also be seen in the PSNR value, which is about 48.0 at a target bit-rate of 1.8 bpp.

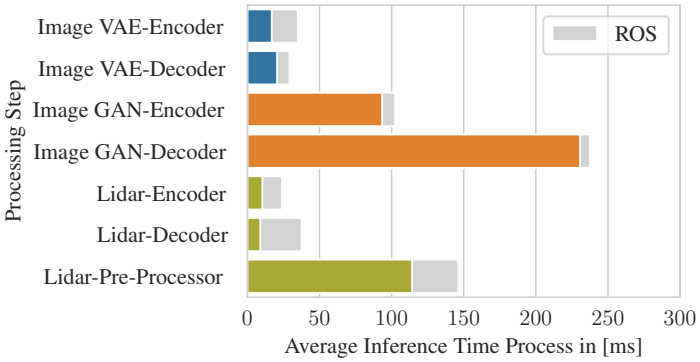

Figure 7: Average inference time for the different ROS nodes.

Still, general distance trends can be identified and certain deflections can be recognized. Therefore, this approach generates early, but promising results.

4.3 Online Pipeline

In a remote assistance system the transmission of the sensor data is most important for taking appropriate action. As the sensor data must be encoded and decoded to increase data throughput, the transmission of the sensor data is prone to latency. Therefore, fast processing times are necessary. We test the compression approaches introduced in Section 3 regarding their processing times and throughput on an NVIDIA GeForce GTX 1080 Ti to investigate their suitability for the use case of remote assistance. We measure the processing time of the nodes over a time span of 300 seconds. The processing time gives information about the throughput of the pre-processor, encoder and decoder. To understand the overhead introduced by ROS we also measure the mere inference time for only encoding and decoding of the images and point clouds. The results are summarized in Figure 7.

For image compression the VAE requires significantly less processing time than the GAN, leading to a throughput of about 28 FPS. The ROS encoder has a mean processing time of 34.9 ms of which 16.8 ms is the mere inference time for the VAE compression. The ROS decoding node requires 29.1 ms while the mere decompression takes 20.5 ms. The resulting system latency adds up to 57.1 ms due to the processing times of encoder and decoder in addition to the network latency. The generation of the ROS message from the latent representation and its publishing is more time consuming than the image compression itself. The decoder must first reconstruct the latent representation from the transmitted ROS message which is less time consuming.

In the GAN-based image compression pipeline the processing times of the encoder and decoder vary significantly. While the ROS encoder has a mean processing time of 102.5 ms, the complete pipeline can only process 4 FPS, as the decoder requires about 237.7 ms per frame. This results in a latency of about 340.2 ms. Without the ROS overhead the inference time for the image compression is about 93.4 ms for the encoder and 230.8 ms for the decoder. Comparing this to the VAE the relative overhead introduced by ROS is significantly smaller. This can be traced back to a time-consuming process for the generation of the customized ROS message, representing the VAE's latent space.

For point cloud compression the pre-processing node in its current implementation is a bottleneck. It determines the throughput of 6 scans per second of the complete pipeline. The mean processing time for the pre-processing node is 146.5 ms while the encoder and decoder node reach processing times of 23.9 ms and 37.7 ms respectively. Investigating the mere inference time without ROS overhead, the discrepancy between the lidar pre-processing and its compression itself is significant as well. Pre-processing already takes 114.5 ms per scan while compression only takes 10.4 ms and decompression 8.9 ms.

While the GAN and the point cloud compression reach impressive compression results, their processing times are too high to be currently applied in a remote assistance system. Georg et al. [18] consider frame rates of about 30 FPS sufficient for remote driving in low speed scenarios while for

high speed scenarios a frame rate of 55 FPS is needed. For remote driving latency values above 300 ms make the safe operation of the AV almost impossible [34]. While the latency requirements for remote assistance [38] are less stringent compared to remote driving [19], an operator still requires real-time knowledge of the state of the vehicle's environment to make decisions.

5 Conclusion and Outlook

To enable remote assistance for AVs, large amounts of data must be transmitted from the vehicle to a remote operator. In order to not overload communication infrastructure, improved data compression is necessary. For this purpose, we implemented and evaluated selected state-of-the-art generative approaches for image and lidar compression, and compared them with traditional compression methods. The approaches were trained and tested with the KITTI dataset, embedded into a ROS framework and simulated within the CARLA environment. We have shown that, taking into account rate-distortion-perception-latency requirements, a real-world application can currently only be satisfied by VAEs, while GANs show the more promising results regarding reconstruction quality.

Offline Rate-Distortion-Perception Analysis. Regarding our offline rate-distortion-perception analysis, the GAN approach yields the best results for image compression and provides good reconstruction results even at very low bit-rates of 0.2 to 0.3 bpp. The reconstructions were evaluated with multiple metrics. Moreover, even at these bit-rates, it is still possible to perform object detection with a sufficient detection rate, demonstrating the perception quality of the data for remote assistance.

As we used insights gained by preliminary results of the latency evaluation during the process, we only applied a VAE for the lidar data. The reconstructions showed a mean euclidean distance error between 0.3 m and 0.5 m and especially a loss of the characteristics of lidar point clouds. Since the similarities between the point clouds are already much lower than in the image domain, we did not perform a perception analysis.

Online Pipeline with Latency Analysis. Regarding the online latency evaluation, a complete end-to-end pipeline for image and lidar compression was implemented in ROS. We evaluated computing times, clearly showing that for image compression the GAN approach is significantly slower than the VAE and cannot be used for remote assistance in terms of processing speed. Thus, VAE-based compression shows a promising trade-off between processing time and reconstruction quality. For lidar compression, the bottleneck is the conversion of the point cloud to the tensor format.

Outlook. To fully represent the domain of autonomous driving, radar data could be considered in the future. Further, the perception evaluation in the image domain could be extended to other classes such as trucks, bicycles or pedestrians. While we did perform training for image compression on a portion of the much larger WAYMO Perception Open Dataset with promising results, a complete training and evaluation on such a data set is desirable. Finally, generative video compression is a promising research area for future works.

Regarding the lidar compression, future work needs to re-evaluate the suitability of the utilized compression approach. While results such as shown in [48] are desirable, it must be investigated if an adapted version of their concept can be designed for the remote assistance use case.

Regarding the latency, further work will be done to optimize the inference time of the models by optimizing their structure or using acceleration methods. Since the architecture allows for processing in a single pass, a combination of camera and lidar data can be considered. Likewise, we suggest using optimized conversion methods for point clouds and ROS messages. Here, a good compromise between image quality and pipeline speed must be found. Finally, the implemented ROS framework will be deployed on our test vehicle for a real-world demonstration.

6 Acknowledgment

This work results partly from the KIGLIS project supported by the German Federal Ministry of Education and Research (BMBF), grant number 16KIS1231.

References

[1] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 221–231. IEEE, 2019.

[2] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[3] Jean Bégaint, Fabien Racapé, Simon Feltman, and Akshay Pushparaja. CompressAI: a PyTorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020.

[4] Fabrice Bellard. BPG Image format, 2018. URL `https://bellard.org/bpg/`. Accessed: 12.09.2021.

[5] Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 675–685. PMLR, 2019.

[6] Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 675–685. PMLR, 2019.

[7] H. Blum, PE. Sarlin, J. Nieto, and et al. The fishyscapes benchmark: Measuring blind spots in semantic segmentation. In *Int J Comput Vis 129*, page 3119–3135. Springer International Publishing, 2021.

[8] Daniel Bogdoll, Jasmin Breitenstein, Florian Heidecker, Maarten Bieshaar, Bernhard Sick, Tim Fingscheidt, and Marius Zöllner. Description of Corner Cases in Automated Driving: Goals and Challenges. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 1023–1028, 2021.

[9] Daniel Bogdoll, Stefan Orf, Lars Töttel, and J. Marius Zöllner. Taxonomy and survey on remote human input systems for driving automation systems. *arXiv preprint arXiv:2109.08599*, 2021.

[10] Lucas Caccia, Herke van Hoof, Aaron Courville, and Joelle Pineau. Deep generative modeling of lidar data. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5034–5040, 2019.

[11] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 7936–7945. Computer Vision Foundation / IEEE, 2020.

[12] On-Road Automated Driving (ORAD) committee. *SAE-J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, 2021.

[13] Dengxin Dai and Luc Van Gool. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3819–3824, 2018.

[14] Shubham Dash, Giridharan Kumaravelu, Vijayakrishna Naganoor, Suraj Kiran Raman, Aditya Ramesh, and Honglak Lee. Compressnet: Generative compression at extremely low bitrates. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*, pages 2314–2322. IEEE, 2020.

[15] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[16] Andreas Geiger, Philip Lenz, and Raquel 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, 2012.

[17] Jean-Michael Georg and Frank Diermeyer. An adaptable and immersive real time interface for resolving system limitations of automated vehicles with teleoperation. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 2659–2664, 2019.

[18] Jean-Michael Georg, Johannes Feiler, Simon Hoffmann, and Frank Diermeyer. Sensor and actuator latency during teleoperation of automated vehicles. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 760–766, 2020.

[19] S. Gnatzig, F. Schuller, and M. Lienkamp. Human-machine interaction as key technology for driverless driving - a trajectory-based shared autonomy control approach. In *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pages 913–918, 2012.

[20] Tim Golla and Reinhard Klein. Real-time point cloud compression. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5087–5092, 2015.

[21] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014.

[22] Amirhossein Habibian, Ties van Rozendaal, Jakub M. Tomczak, and Taco Cohen. Video compression with rate-distortion autoencoders. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 7032–7041. IEEE, 2019.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[24] Tianyu He and Zhibo Chen. End-to-end facial image compression with integrated semantic distortion metric. In *IEEE Visual Communications and Image Processing, VCIP 2018, Taichung, Taiwan, December 9-12, 2018*, pages 1–4. IEEE, 2018.

[25] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[26] J. Jiang. Image compression with neural networks - A survey. *Signal Process. Image Commun.*, 14(9):737–760, 1999.

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, 2014.

[28] Wei et al. Liu. SSD: Single Shot MultiBox Detector. In *Computer Vision – ECCV 2016*, pages 21–37, Cham, 2016. Springer International Publishing.

[29] Jonas Löhdefink, Andreas Bär, Nico M. Schmidt, Fabian Hüger, Peter Schlicht, and Tim Fingscheidt. GAN- vs. JPEG2000 image compression for distributed automotive perception: Higher peak SNR does not mean better semantic segmentation. *CoRR*, abs/1902.04311, 2019.

[30] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. DVC: an end-to-end deep video compression framework. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 11006–11015. Computer Vision Foundation / IEEE, 2019.

[31] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4394–4402. Computer Vision Foundation / IEEE Computer Society, 2018.

[32] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in Neural Information Processing Systems*, 33, 2020.

[33] David Minnen, Johannes Ballé, and George Toderici. Joint autoregressive and hierarchical priors for learned image compression. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10794–10803, 2018.

[34] Stefan Neumeier, Ermias Andargie Walelgne, Vaibhav Bajpai, Jörg Ott, and Christian Facchi. Measuring the feasibility of teleoperated driving in mobile networks. In *2019 Network Traffic Measurement and Analysis Conference (TMA)*, pages 113–120, 2019.

[35] Tilo Ochotta and Dietmar Saupe. Image-Based Surface Compression. *Computer Graphics Forum*, 27(6):1647–1663, 2008.

[36] Shibani Santurkar, David M. Budden, and Nir Shavit. Generative compression. In *2018 Picture Coding Symposium, PCS 2018, San Francisco, CA, USA, June 24-27, 2018*, pages 258–262. IEEE, 2018.

[37] Ruwen Schnabel and Reinhard Klein. Octree-based point-cloud compression. In *Proceedings of the 3rd Eurographics / IEEE VGTC Conference on Point-Based Graphics*, SPBG'06, page 111–121. Eurographics Association, 2006.

[38] T.B. Sheridan. Space teleoperation through time delay: review and prognosis. *IEEE Transactions on Robotics and Automation*, 9(5):592–606, 1993.

[39] Mennatullah Siam, Mostafa Gamal, Moemen Abdel-Razek, Senthil Kumar Yogamani, Martin Jägersand, and Hong Zhang. A comparative study of real-time semantic segmentation for autonomous driving. In *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 587–597. Computer Vision Foundation / IEEE Computer Society, 2018.

[40] Stanford Artificial Intelligence Laboratory et al. Robotic operating system noetic ninjemys, 2021. URL `https://www.ros.org`. Accessed: 21.09.2021.

[41] Gary J. Sullivan, Jens-Rainer Ohm, Woojin Han, and Thomas Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.*, 22(12): 1649–1668, 2012.

[42] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020.

[43] David S. Taubmann and Michael W. Marcellin. *JPEG2000 Image Compression Fundamentals, Standards and Practice Michael W. Marcellin*. Springer US, 2002. ISBN 978-1-4615-0799-4.

[44] George R. Terrell and David W. Scott. Variable Kernel Density Estimation. *The Annals of Statistics*, 20(3):1236 – 1265, 1992.

[45] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *CoRR*, abs/1703.00395, 2017.

[46] George Toderici, Sean M. O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent neural networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[47] Chenxi Tu, Eijiro Takeuchi, Chiyomi Miyajima, and Kazuya Takeda. Compressing continuous point cloud data using image compression methods. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1712–1719, 2016.

[48] Chenxi Tu, Eijiro Takeuchi, Alexander Carballo, and Kazuya Takeda. Point Cloud Compression for 3D LiDAR Sensor using Recurrent Neural Network with Residual Blocks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3274–3280, Montreal, QC, Canada, 2019. IEEE.

[49] Chenxi Tu, Eijiro Takeuchi, Alexander Carballo, and Kazuya Takeda. Real-time streaming point cloud compression for 3d lidar sensor using u-net. *IEEE Access*, 7, 2019.

[50] Gregory K. Wallace. The JPEG still picture compression standard. *Commun. ACM*, 34(4): 30–44, 1991.

[51] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, volume 2, pages 1398–1402 Vol.2, 2003.

[52] Richard Yi Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings - 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 586–595. IEEE Computer Society, 2018.

[53] Marc René Zofka, Florian Kuhnt, Ralf Kohlhaas, Christoph Rist, Thomas Schamm, and J. Marius Zöllner. Data-driven simulation and parametrization of traffic scenarios for the development of advanced driver assistance systems. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 1422–1428, 2015.

A Appendix

Figure 8: CARLA environment (left) and VAE-based decompressed images in RViz (right).

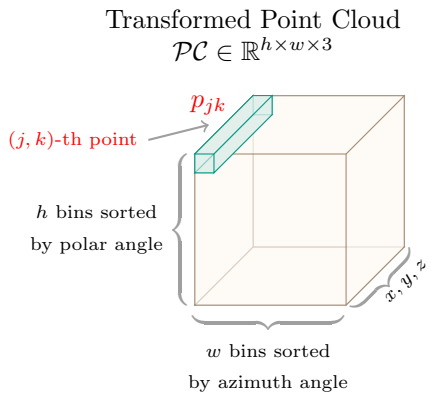

Figure 9: Sorted and binned 2d representation of the point cloud consisting of $h \times w$ bins. The (x, y, z)-coordinates of each point are calculated as the average of all points of the original point cloud that belong into that specific bin.

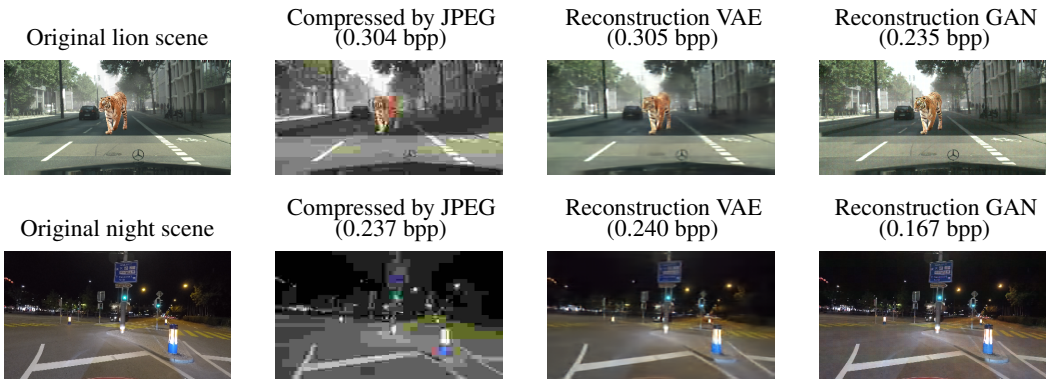

Figure 10: Out-of-domain testing of VAE trained on KITTI.

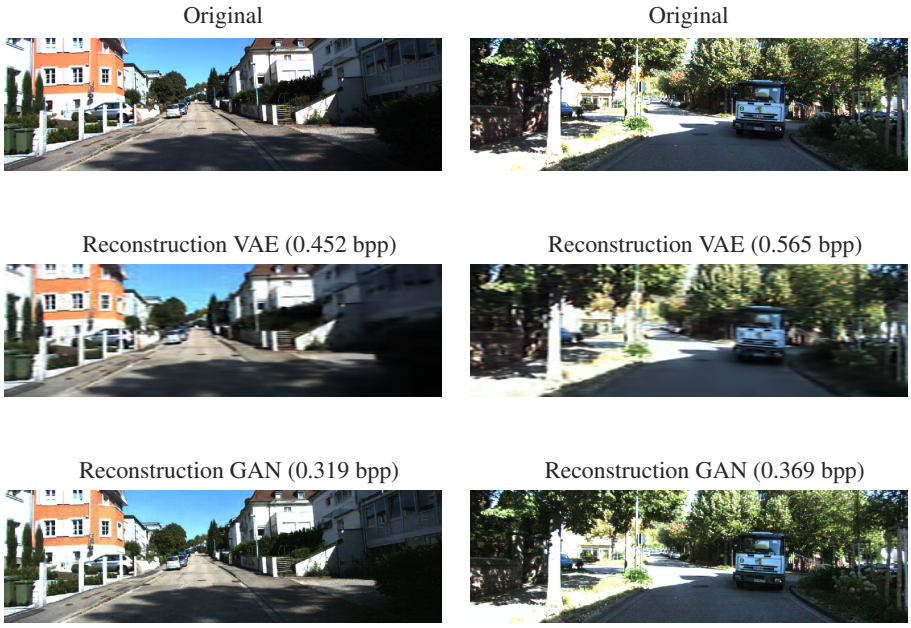

Figure 11: Examples of reconstructions with data from the KITTI dataset.