Low-Latency Neural Stereo Streaming
Supplementary Material

Qiqi Hou Farzad Farhadzadeh Amir Said Guillaume Sautiere Hoang Le*
Qualcomm AI Research†
{qhou, ffarhadz, asaid, gsautie, hoanle}@qti.qualcomm.com

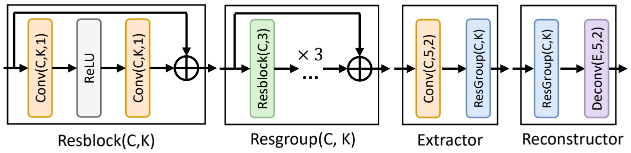

Figure 1. Feature extractor and the image reconstructor.

1. Model Configurations

In this section, we provide more details on the configuration for each module of our network.

Feature extractor and image reconstructor. Figure 1 shows the architecture of the feature extractor and the image reconstructor. Following FVC [5] and LSVC [2], the feature extractor extracts the features from the RGB input images, which are subsequently compressed within the network. The extraction process involves downsampling the image using strided convolutions, followed by a stack of residual convolution blocks. The image reconstructor synthesizes the final RGB images from the reconstructed features. It shares a similar architecture to the feature extractor, consisting of one residual convolution block and strided transposed convolutional layers. The weights of the feature extractor and image reconstructor are shared between the left and right branches of the network. Both are including 3 ResGroup modules with a channel number of $C = 64$, a kernel size of 3, and a stride of 1.

Motion estimation and motion compensation. Figure 2 illustrates the network architectures for the motion estimation and motion compensation modules. Following FVC [5], our motion estimation module contains two convolutional layers. This module takes the features $\mathbf{F}_t$ and $\hat{\mathbf{F}}_{t-1}$ as inputs and estimates the offset vectors $\mathbf{M}_t$ between them. These offset vectors are then quantized, encoded, and sent to a decoder side where they will be reconstructed to $\hat{\mathbf{M}}_t$. The coding process is performed with the commonly used hyperprior-based network [1].

The motion compensation module aims to generate the current feature maps by warping $\hat{\mathbf{F}}_{t-1}$ using $\hat{\mathbf{M}}_t$ through a deformable convolution. Our network uses two convolutions as well as a skip connection to fuse the warped features with the previous feature $\hat{\mathbf{F}}_{t-1}$. We set the group number as 8 in the deformable convolutional layer for the motion compensation. Motion estimation module and motion compensation module contain two convolutional layers with a channel number of $C = 64$, a kernel size of 3, and a stride of 1. We use ReLU as our activation function.

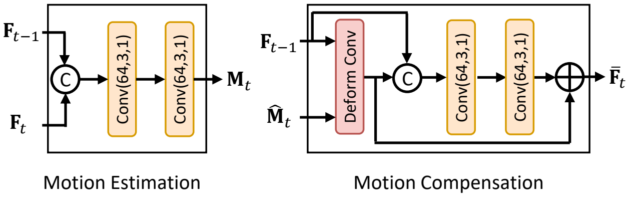

Figure 2. The structure of the motion estimation module and the motion compensation module.

Parallel Motion Autoencoder. Figure 4 in the main paper shows the architecture of the parallel motion autoencoder. We set the number of channels $C = 64$ for the parallel motion autoencoder. For its parallel HyperCodec, the number of channels is set to $C = 128$. The ResGroup layers utilize a kernel size of 3 and a stride of 1. The conv layers, primarily designed for feature downsampling, incorporate a kernel size of 5 along with a stride of 2. Conversely, the deconv layers, aiming to upsample the feature, employ a kernel size of 5 and a stride of 2 as well. ReLU is adopted as our activation function.

Parallel Context Autoencoder. The context autoencoder also uses the architecture shown in Figure 4 of the main paper. Different from the parallel motion autoencoder, we set the number of channels as $C = 128$ in the parallel context autoencoder. The channel number of its parallel HyperCodec is set to $C = 128$. To match the channel number after concatenation layers, we use one convolutional layer with a channel number of 128, a kernel size of 3, and a stride of 1. We use ReLU for the activation function.

Bidirectional Shift Module. Figure 4 in the main paper shows the architecture of the bidirectional shift module. We set the channel number in bidirectional shift modules as $C = 32$, $C_g = 32$, and $C_c = 12$. The group number of

* Corresponding author

† Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

Table 1. Complexity for each component of our network for input size 512 × 512.

Component	MACs(G)	FLOPs(G)	Params(M)
Feature extractor	59.2	118.7	0.23
Image reconstructor	42.4	84.8	0.23
Motion estimation	14.5	29.1	0.22
Motion compensation	25.4	50.8	0.40
Motion autoencoder	32.0	64.0	10.93
└ BiShiftMods	2.9	5.8	1.67
Context autoencoder	137.3	275.1	23.54
└ BiShiftMods	3.3	6.6	2.00

the GroupCor module is set as $G = 4$. Since the max disparity for the KITTI datasets [4, 7] is 192, we set the max shift distance $D = 192/2^{Scale-1}$, where $Scale$ indicates the downscale factor of the input feature maps with respect to the source image. Similar to the max disparity, we set the shift stride $S = max(1, 8/2^{Scale-1})$. The BiShiftMod component downsamples input features using `conv` layers with a kernel size of 5 and a stride of 2. Conversely, it upsamples the output features through `deconv` layers, also featuring a kernel size of 5 and a stride of 2. The remaining `conv` layers have a kernel size of 3 and a stride of 1. We use `Mish` [8] as the activation function.

2. Datasets Configurations

We utilized the official script provided by the authors of LSVC [2] to pre-process the CityScapes [3] and KITTI [4, 7] datasets, ensuring a fair comparison.

KITTI 2012 and 2015 datasets. We selected the “testing” subset for our experiments. In accordance with LSVC [2], we excluded videos containing fewer than 21 frames. Specifically, videos “000127” and “000182” from the KITTI 2012 dataset, and videos “000026” and “000167” from the KITTI 2015 dataset were removed. We then cropped the videos to a size of 1216 × 320 from the top left, employing `ffmpeg` and using the ”420p” pixel format.

CityScapes datasets. Following the approach of LSVC [2], we cropped 128 pixels from the left and 64 pixels from the bottom to eliminate artifacts resulting from rectification. Additionally, we removed 256 pixels from the bottom to exclude the ego-vehicle area. We applied `ffmpeg` with the ”420p” pixel format for cropping. After the cropping, we obtained videos with sizes of 1920 × 704.

3. Detailed Coding Complexity

In this study, we further examine the complexity of each component within our network. Table 1 presents the complexity in terms of MACs, FLOPs, and the number of parameters (Params). The majority of calculations are concentrated on the context autoencoder, as it is designed to capture detailed context information following motion compensation. Consequently, we set a larger channel number of 128. Additionally, we report the complexity of the BiShiftMods in both the motion autoencoder and the context autoencoder. Our BiShiftMod accounts for only a small fraction of the overall computational complexity.

4. Discussion

Figure 6 in the main paper shows that our methods achieve both smaller operational complexity and faster inference time compared to the competitive LSVC [2] method. Even when the difference in complexity is small given low resolution inputs, our method still reduces the inference time significantly compared to LSVC. This advantage is thanks to our novel architecture that is optimized for parallel processing. We note that this the inference time is still for benchmarking purpose and the current design is not aimed for deployment yet. In the future, it is possible to further optimize our network for practical deployment such as following the approach in the recent practical codec [6].

References

[1] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018. 1

[2] Zhenghao Chen, Guo Lu, Zhihao Hu, Shan Liu, Wei Jiang, and Dong Xu. LSVC: A Learning-Based stereo video compression framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6073–6082, 2022. 1, 2

[3] 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. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2

[4] 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. 2

[5] Zhihao Hu, Guo Lu, and Dong Xu. FVC: A new framework towards deep video compression in feature space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1502–1511, 2021. 1

[6] Hoang Le, Liang Zhang, Amir Said, Guillaume Sautiere, Yang Yang, Pranav Shrestha, Fei Yin, Reza Pourreza, and Auke Wiggers. MobileCodec: neural inter-frame video compression on mobile devices. In *Proceedings of the 13th ACM Multimedia Systems Conference*, pages 324–330, New York, NY, USA, 2022. Association for Computing Machinery. 2

[7] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3070. IEEE, 2015. 2

[8] Diganta Misra. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681*, 2019. 2