8th INTERNATIONAL WORKSHOP ON ON-BOARD PAYLOAD DATA COMPRESSION (OBPDC 2022)

28-30 SEPTEMBER 2022

RAW DATA COMPRESSION FOR SYNTHETIC APERTURE RADAR USING DEEP LEARNING

Georgios Pilikos[1], Mario Azcueta[1], Roberto Camarero[1], Nicolas Floury[1]

[1] ***European Space Agency (ESA)***
European Space Research and Technology Centre (ESTEC)
Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands
Email: Georgios.Pilikos@esa.int

As the demand grows for higher resolutions, wider swaths, higher revisit times and multiple-frequency acquisitions, the amount of acquired data for Synthetic Aperture Radar (SAR) will increase substantially. This data increase is heavily linked to the operational costs of future SAR missions (cost of data downlinking to ground stations), making efficient on-board data compression a crucial component in the data processing chain. Performance of current on-board SAR data compression algorithms could be improved by utilizing more correlations in the data using transforms. Recently, deep learning has been proposed to learn analysis and synthesis transforms for lossy optical image compression. Nevertheless, SAR raw data compression requires further considerations given the complex nature of the data. More importantly, while compression is performed on the raw data, the products of greater interest are computed by image focusing, exploiting the amplitude and phase content. Thus, algorithms should attain a high compression ratio in the raw data domain, while simultaneously retaining high quality in the image domain, preserving the amplitude and phase. In this work, we take these considerations into account and propose a lossy SAR raw data compression algorithm based on a convolutional encoder-decoder architecture, with vector quantization in the bottleneck. Two strategies of our proposed encoder-decoder architecture are examined, illustrating their corresponding rate performance and taking into account quality metrics based on the SAR impulse response function. Experiments comparing different strategies are demonstrated, as well as a comparison with Flexible Dynamic Block Adaptive Quantization (FDBAQ).

INTRODUCTION

Synthetic Aperture Radar (SAR) satellites have been growing in popularity in recent years. One main reason for their increased deployment is due to their ability to image through clouds and during day/night. Throughout the course of their operation, institutional missions and commercial SAR constellations acquire and downlink a vast amount of data. Before downlinking, on-board data compression is usually performed, e.g. using Block Adaptive Quantization (BAQ) [1] and Flexible Dynamic Block Adaptive Quantization (FDBAQ) [2]. These techniques rely on a set of quantizers to reduce the bit-rate, however this could be improved by exploiting more data correlations. Analysis and synthesis transforms (e.g. Wavelets Transform) are able to transform the data into a domain, potentially, more suitable for compression [3].

Nevertheless, designing hand-crafted transforms and performing quantization separately is not optimal. Recently, deep learning has been proposed for end-to-end image compression, to simultaneously learn transforms, entropy models and quantization [4], [5]. Building on this, reduced-complexity variational autoencoders were applied to on-board lossy optical image compression [6]. Another approach for lossy optical image compression learns to perform vector quantization using the Vector Quantized Variational Autoencoder (VQVAE) [7], [8]. This results in the transmission of codebook indices as compact bit-arrays, with a fixed size, depending on the codebook and the bottleneck dimensions.

Deep learning algorithms used for SAR data require further considerations [9]. On-board SAR raw data are not the final product aimed at regular users but often the products of greater interest are computed on-ground [10], by focusing raw data and applying radiometric corrections to get calibrated images. Moreover, the data are complex which increases the difficulty of using convolutional neural networks [11], while at the same time, amplitude and phase needs to be preserved. Recent work has tackled SAR image compression using deep learning [12], but for on-ground image processing. In this paper, we tackle the problem of raw data compression keeping in mind that successful SAR data compression algorithms should attain a high compression ratio in the raw data domain, while simultaneously retaining high quality in the Single Look Complex (SLC) image domain. To this end, we develop two different strategies for SAR raw data compression. The first utilizes the VQVAE that compresses raw data and de-compresses them directly. The second is an extension of the VQVAE which compresses raw data and de-compress them to SLC image. To achieve this, we implement a differentiable Range-Doppler Algorithm (RDA) layer, trained end-to-end together with convolutional filters and vector quantization. This enables gradients to pass through the RDA layer, optimizing the model parameters with respect to a SLC image directly. We refer to this as VQVAE-RDA, and note that this strategy uses raw data as input and is optimized for SLC images, therefore raw data at the intermediate output (before the RDA layer) might not be directly usable.

The rest of the paper is organized as follows: next we briefly introduce the SAR imaging steps used during the RDA. Then, we describe traditional lossy data compression and provide further details on our methodology. Afterwards, we describe three datasets used during experiments and demonstrate preliminary results on each. Finally, conclusions are provided with a discussion on potential improvements.

SYNTHETIC APERTURE RADAR (SAR) IMAGING

In SAR imaging, a platform is equipped with a transmitter/receiver that emits pulses (chirps) to an area of interest and then records the echoes as it moves. The dimension along the track of the platform is referred to as azimuth and across-track as range. Each received pulse is digitised into what is called an azimuth line, consisting of range samples. The travel times taken by the electromagnetic waves to propagate to the imaging region and back to the platform are utilized in the process of focusing the raw data. To improve computational performance, fast methods have been proposed that take advantage of fast Fourier transform implementations and perform operations in the frequency domain [13].

More specifically, the Range-Doppler Algorithm (RDA) is widely used [13] and will be considered in this work. It involves three main steps: range compression, range cell migration correction and azimuth compression [14]. Range compression convolves the raw data with a matched filter (based on the emitted chirp), range cell migration correction aligns the echo responses into constant-range lines and azimuth compression convolves the processed data with another matched filter (based on the imaging geometry). In this context, SAR raw data compression involves the efficient encoding of the unfocused data using a low bit-rate. Then, the compressed representation should be transmitted to the ground and decoded accurately. When the RDA or another focusing method [13] is used, the resulting image should be of high quality, retaining the original range and azimuth resolutions, preserving the magnitude and phase of the data.

LOSSY DATA COMPRESSION AND PROPOSED METHODOLOGY

Traditionally, lossy data compression schemes involve three steps: data decorrelation, quantization and entropy coding (e.g. Huffman or arithmetic coding). These are usually optimized independently and changes to parts of the compression chain can impact overall performance. For SAR data compression, BAQ-based algorithms such as FDBAQ [2] perform quantization on blocks of samples. The quantizers and consequently the bit-rate varies depending on the block statistics, with entropy coding following quantization. However, no data transformation is performed, which could help improve compression further. In addition, often, higher level products are of greater interest (e.g. SLC images) and the optimization of data compression algorithms could be designed for improved image quality. In this work, we introduce a lossy SAR raw data compression framework, where all components of the compression chain are optimized jointly.

Two strategies are followed. The first one uses only raw data, both at the input of the compression and the output of the de-compression. Fig. 1 (a) illustrates a simplified diagram of the Vector Quantized Variational Autoencoder (VQVAE) [8] where analysis and synthesis transforms are learned using convolutional layers. Vector Quantization (VQ) is then performed, by learning a codebook to be shared by both the compression and the de-compression process. As the raw data are acquired, the learned analysis transform and vector quantization are performed, collectively referred to as the encoder. At the output of the encoder, the corresponding indices to the codewords within the learned codebook are to be transmitted to the ground for further processing. Entropy coding is not yet applied but planned for future work. The decoder uses these indices to recover the intermediate vectors and a learned synthesis transform is applied to them, to recover the raw data. Optimization at the bottleneck is performed based on the straight-through operator, where during the backwards pass, the gradients are copied from the decoder input to the encoder output [8]. A loss term is also added in order to learn the codebook as described in [8].

The second strategy proposed, called VQVAE-RDA, is illustrated in Fig. 1 (b). Raw data are passed through the encoder similarly as the first strategy. However, the decoder is different. At the output of the learned synthesis transform, a differentiable Range-Doppler Algorithm (RDA) layer is developed. All steps of the RDA are implemented as custom layers, enabling gradients to pass through them during the backwards pass. To achieve this, the adjoint RDA program has

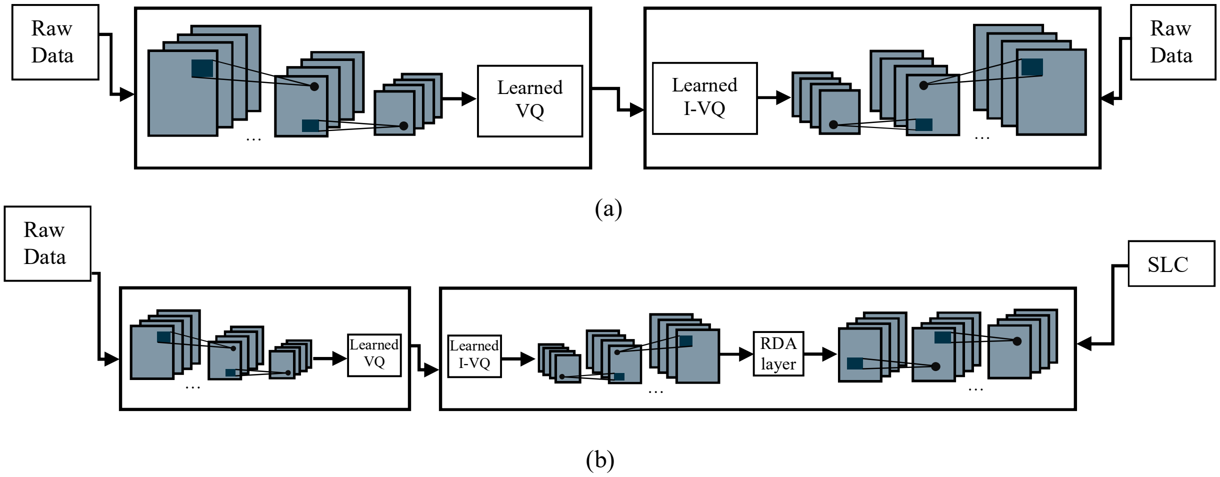


Fig. 1 (a) VQVAE compression, aiming to compress and de-compress raw data with minimal mean squared error. (b) VQVAE-RDA compression uses raw data at the input and SLC images at the output, optimized for minimal mean squared error for SLC images. Intermediate raw data might not be directly usable in (b). One filter per layer is shown for illustration purposes. VQ and I-VQ refer to Vector Quantization and its inverse, i.e. the recovery of vectors, respectively.

been implemented as well. This includes Fourier Transforms and interpolation. After the RDA layer, an image enhancement network comprising of convolutional layers is implemented. Given that the loss function minimizes the mean squared error between original and recovered SLC images, the intermediate raw data, before the RDA layer might not be usable and should not be used in isolation. Optimizing for both raw data and SLC images could improve results.

There are various choices that could affect the achievable bit-rate, the quality of recovery and the inference time. For example, the number of layers, channels, striding, codebook size, activation functions and normalization. Furthermore, the learning rate, the use of skip connections in intermediate layers and the optimization routine need to be studied further on their effects in training. We empirically chose the following to obtain the first preliminary results with future architecture search required. Figure 1 (a), (b) gives a simplified representation, for illustration purposes only, with:

1. ***VQVAE:*** *Encoder:* 5 convolutional layers, 16 channels each (except last layer with 128), codebook size of 512 codewords, and striding of one or two depending on layer to control amount of down-sampling. After every layer, normalization and ReLU is used except the last layer. *Decoder:* same as encoder but in reversed order.

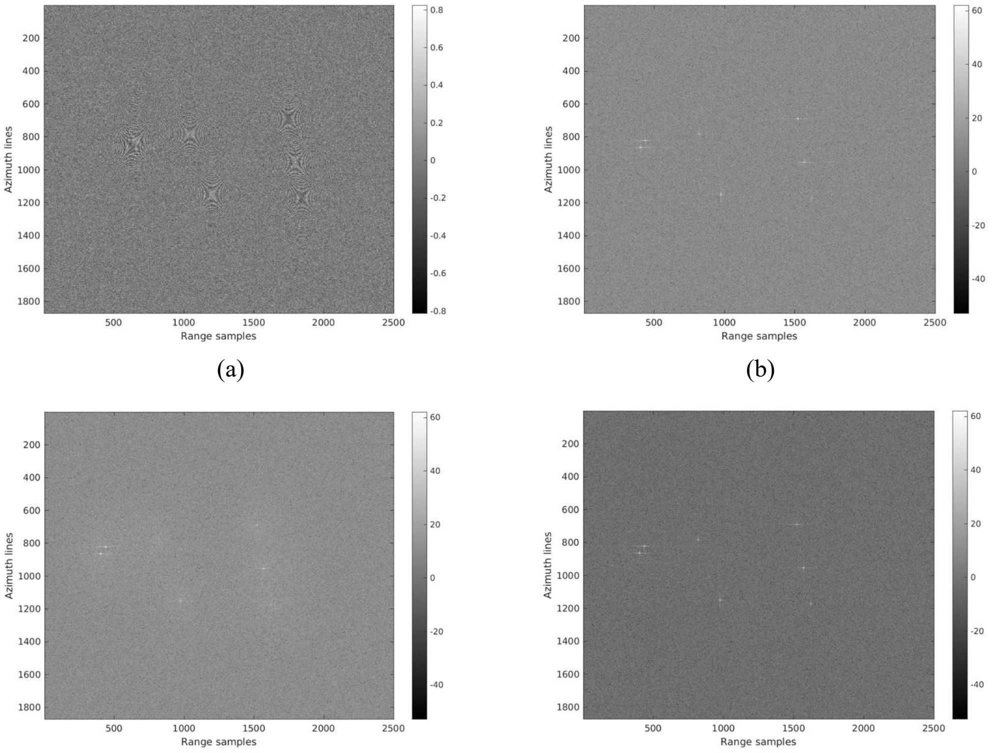
2. ***VQVAE-RDA:*** *Encoder:* 5 convolutional layers, 8 channels each (except last layer with 128), codebook size of 512 codewords, striding of one or two depending on layer to control amount of down-sampling. *Decoder*: same as encoder but in reversed order. Then a differentiable RDA layer is used followed by an image enhancement network, with 8 convolutional layers and 8 channels each. No down-sampling is performed within this network. After every layer, normalization and ReLU is used except the last layer.

EXPERIMENTS

In order to test our proposed lossy SAR raw data compression framework, we used three different datasets. These include data from (1) a custom implementation of a point target SAR raw data simulator, (2) a Sentinel-1 instrument simulator and (3) Sentinel-1 data from the Copernicus Open Access hub (https://scihub.copernicus.eu/).

Custom point target SAR raw data simulator

We implemented a custom SAR raw data simulator of point targets and homogeneous clutter. The chirp is represented with a complex exponential and the two-way travel time between the platform and targets is calculated. A phase shift on the chirp is applied based on the travel-time, and the antenna gain is modelled with a sinc function, varying with the measurement angle. Circular complex Gaussian noise is used to simulate the background clutter. Hundreds of scenes are



(a)

(b)

(c)

(d)

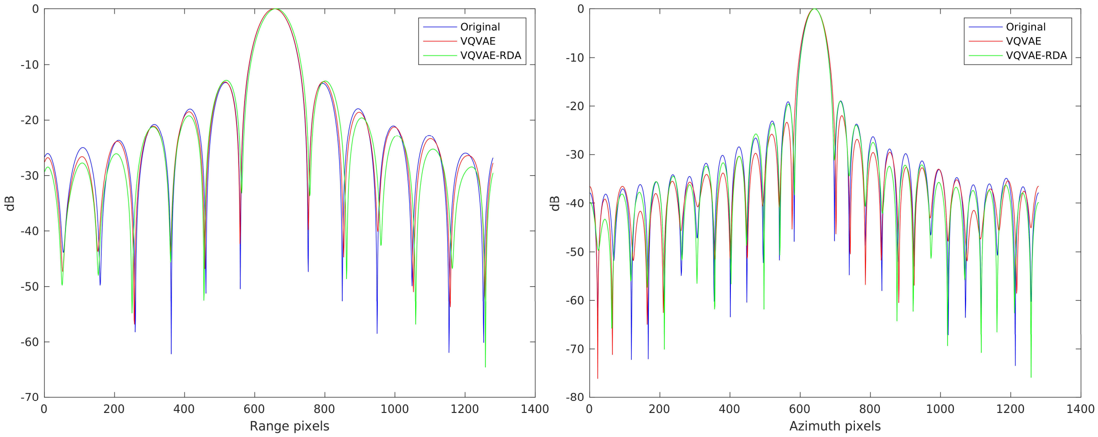
Fig. 2 (a) Real part of original raw data, (b) SLC image from original raw data, (c) SLC image formed using recovered data from VQVAE, (b) SLC image formed from VQVAE-RDA.

simulated by randomly varying the number and location of point targets. An example of the real part of raw data and SLC image is included in Fig. 2 (a) and (b) respectively. The parameters used can be seen in Table 1, however these do not correspond to any actual Sentinel-1 acquisition mode. The pulse duration, sampling frequency and height were chosen for simplicity and ease of experiments.

Table 1. Parameters used in the raw data simulation

Azimuth lines	1871
Range samples	2500
Centre frequency	5.405 GHz
Chirp bandwidth	50 MHz
Chirp pulse duration	3μs
Range sampling frequency	150 MHz
Pulse Repetition Frequency	1871
Antenna length	12 m
Platform height	690 km

Two hundred and fifty data volumes (1871, 2500, 2) were used during training. We separated each complex sample into real and imaginary and used two input channels to the networks. Raw data were normalized between -1 and 1 by dividing with the maximum absolute value. Figure 2 (c) shows an example of a SLC image formed using recovered data by VQVAE. Figure 2 (d) shows the SLC image from VQVAE-RDA. The VQVAE blurs the area surrounding the point targets, as opposed to a sharper SLC using VQVAE-RDA. However, there is an image bias, which could be due to the normalization during pre-processing and within each layer. Further investigation on the effects of data normalization is required to improve overall image quality.



(a) (b)

Fig. 3 (a) Range IRF for original (res: 2.62 m, PSLR: -13.29 dB), VQVAE (res: 2.65 m, PSLR: -13.14 dB) and VQVAE-RDA (res: 2.65: m, PSLR: -12.89 dB). (b) Azimuth IRF for original (res: 5.95m, PSLR = -18.94 dB), VQVAE (res: 6.21 m, PSLR: -21.95 dB) and VQVAE-RDA (res: 5.83 m, PSLR: -19.02 dB).

Thus, for the remainder of the experiment, we will concentrate on the point targets. To analyse the image quality of the SLC images, we performed Impulse Response Function (IRF) analysis using two metrics. Range and azimuth IRFs are extracted by taking the corresponding profile across the location of the target. An example of a range and azimuth IRF for a target from an original SLC image, a target from a SLC image using recovered data with VQVAE and a target from a recovered SLC using VQVAE-RDA can be seen in Figures 3 (a) and 3 (b) respectively. For each point target, we extract its range and azimuth IRF, perform interpolation and estimate:

1. *Spatial resolution*: - 3dB IRF main lobe width,
2. *Peak side-lobe ratio (PSLR):* ratio of intensity of largest peak outside main lobe to main lobe.

These metrics are estimated over 50 SLC images, with their average listed in Table 2. Both the VQVAE and the VQVAE-RDA obtain comparable results based on the IRF metrics used. The bit-rate and inference time are also listed in Table 2, calculated over 50 raw data volumes. The achievable bit-rate is calculated using:

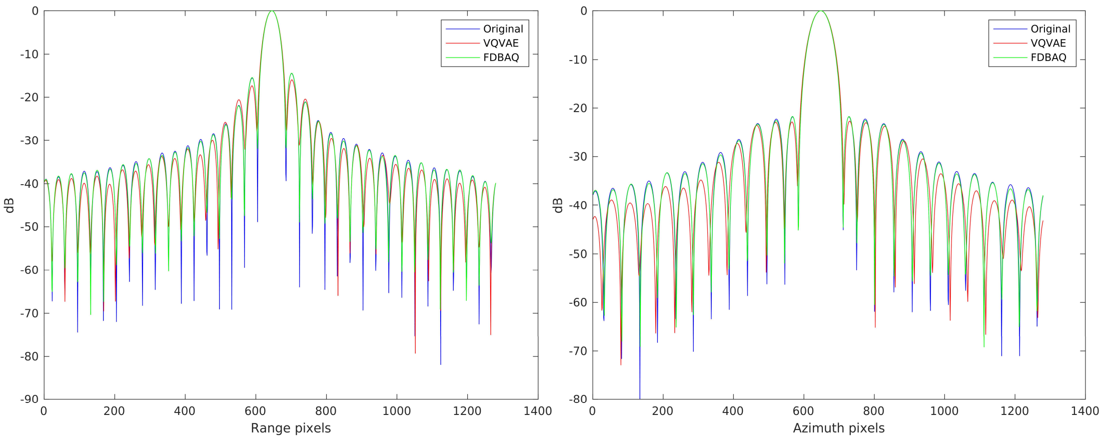
Bits/sample = (first dimension of indices*second dimension of indices*9)/(azimuth lines*range samples*2),

where 9 bits per index are used (with codebook size of 512), 2 is for the real and imaginary parts of raw data. Indices are assumed to be written to file as fixed-length bit-arrays, i.e. a bit-stream is not generated but assumed. For both strategies, the input dimensions are (1871, 2500, 2) and the indices' dimensions are (935, 1249), resulting in a bits/sample = 1.12. This is a design choice to enable effective network training while still being at a lower bit-rate than FDBAQ.

Inference time was calculated on an NVIDIA Tesla T4. VQVAE's encoder is slower since it uses 16 channels per layer as opposed to VQVAE-RDA's encoder which uses 8 channels per layer (except last layer in both cases). The VQVAE-RDA's decoder is slower because it performs the RDA layer and then an image enhancement network with 8 layers and 8 channels. Given the small differences in the IRF performance, the increased decoding time and that the intermediate raw data might not always be directly usable with VQVAE-RDA, we continue the experiments using VQVAE.

Table 2. Impulse Response Function (IRF) metrics for different configurations on custom point target SAR data simulator (azimuth lines = 1871, range samples = 2500).

	Bit-rate [bits/sample]	**Mean Resolution (Rg\Az) [metres]**		**Mean PSLR (Rg\Az) [dB]**		**Mean compression time [seconds]**	**Mean de-compression time [seconds]**
Original	-	2.62	5.91	-13.32	-19.30	-	-
VQVAE	1.12	2.67	6.28	-13.39	**-22.48**	0.29	0.28
VQVAE-RDA	1.12	2.67	**5.83**	**-13.60**	-18.99	0.21	1.71



(a) (b)

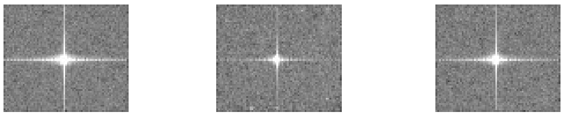
Fig. 4 (a) Original range IRF from the Sentinel-1 simulator (res: 1.59 m, PSLR: -14.54 dB), VQVAE (res: 1.64 m, PSLR: -16.00 dB) and FDBAQ (res: 1.59 m, PSLR: -14.49 dB). (b) Original azimuth IRF from the Sentinel-1 simulator (res: 6.01 m, PSLR: -21.77 dB), VQVAE (res: 6.25 m, PSLR: -22.67 dB) and FDBAQ (res: 6.01 m, PSLR: -21.71 dB).

Sentinel-1 instrument simulator

As a second step during the development of this work, after having narrowed down the most promising alternatives with the aid of the custom SAR raw data simulator, we switched to using a different SAR data simulator which was gently facilitated by the Sentinel-1 payload team at ESTEC. This software realistically emulates the complete Sentinel-1 SAR instrument reception chain, enabling us to test the developed techniques on a more realistic setup. For our specific experiment, Stripmap mode was simulated with point targets laid out randomly across two datasets, one dataset was used for training and the other for testing. Each dataset has 16659 azimuth lines with 25742 range samples per azimuth line. Given the image bias and normalization issues observed in the previous experiment, we opted to use entire azimuth lines for testing the VQVAE instead of splitting lines into range blocks. Different range blocks could be recovered with different image biases, which is undesirable. FDBAQ was implemented as nominally for Sentinel-1, but the image focusing was achieved using a custom RDA implementation.

We used one uncompressed raw data volume for training and one for testing the recovery obtained by VQVAE and the FDBAQ. Fig. 4 (a) and (b) show the range and azimuth IRFs of a point target from the original uncompressed raw data, the one obtained by recovered data from VQVAE and the one obtained by FDBAQ. The FDBAQ obtains better resolution but the VQVAE obtains better PSLR, although the differences are very small. Fig. 5 (a) illustrates this point target obtained by using the original, uncompressed raw data. Fig. 5 (b) shows the corresponding point target obtained by using recovered data from VQVAE, and Fig. 5 (c) illustrates the point target from recovered data using FDBAQ. We can see that both methods are able to locate and recover the point target, with side-lobes being supressed more by VQVAE.

The VQVAE's achievable bit-rate is estimated as in the previous sub-section, with a different bottleneck size but same dimensions' ratio as before. Further comparisons are required to understand the advantages, disadvantages and suitability of each method. However, it is worth noting that the VQVAE uses 1.12 bits/sample whereas FDBAQ uses a variable bit rate, for this azimuth line being 2.77 bits/sample. This is approximately 60% reduction in data usage when using VQVAE, without significant deterioration of the IRFs.



(a) (b) (c)

Fig. 5 (a) Original point target, (b) Point target using VQVAE (1.12 bits/sample), (c) Point target using FDBAQ (bit-rate across point target's azimuth line: 2.77 bits/sample).

FDBAQ-processed raw data from Sentinel-1 acquisition

In this experiment, we utilized data acquired by Sentinel-1, available from the Copernicus Open Access Hub (Copernicus Service Information 2022). The data are already FDBAQ-processed (quantized, de-quantized), altering the original data entropy. Nevertheless, the goal here was to test the VQVAE on more realistic scenes, albeit already processed. For this, we chose the Stripmap-6 Mode of Sentinel-1 and obtained data from Sao Paulo, Brazil, which includes a variety of scenes of interest (sea, ships, land, mountains, city).

Four large data volumes were used, each having 29934 azimuth lines with 19950 range samples per line. Three raw data volumes were used for training and one for testing. During testing, we experimented with and without splitting the azimuth lines into range blocks. Here, we include results with input sizes of (2000, 2500, 2), i.e. the algorithm processes 2000 azimuth lines with 2500 range samples each at a time. As before, the achievable bit-rate (with the assumption that indices are written to fixed-length bit-arrays) is estimated as in the previous sub-sections with a different bottleneck size but same dimensions' ratio. Without splitting azimuth lines into range blocks, image quality could be improved but the inference is more demanding with respect to memory and time (individual range blocks could be processed more efficiently). The choice of range block sizes is not trivial and its effect on image quality and computational resources needs to be studied. This is not considered, with future work being essential on ideal data pre-processing and light-weight architectures.

Figures 6 (a) and (b) illustrate a portion of the Sao Paulo image, focused from FDBAQ-processed data and VQVAE-recovered data respectively. VQVAE uses as input the FDBAQ-processed data, compresses and de-compresses them using 1.12 bits/sample. Figures 7 (a) and (b) illustrate a second part of the same image, focused using FDBAQ-processed and VQVAE-recovered data respectively. The image bias is visible, and numerous details are lost. Nevertheless, this preliminary experiment on realistic scenes illustrates the potential of VQVAE for lossy SAR raw data compression.

CONCLUSIONS

An end-to-end, lossy SAR raw data compression framework was proposed in order to learn synthesis/analysis transforms and vector quantization simultaneously. Different strategies were followed, compressing raw data directly with VQVAE and compressing raw data to SLC images with a proposed variant referred to as VQVAE-RDA. Both configurations performed similarly with respect to an IRF analysis on a custom point target SAR raw data simulator. VQVAE-RDA was able to obtain sharper images, however intermediate raw data might not be directly usable. Adding an optimization term in the loss function, and optimizing for raw data and SLC images simultaneously could improve data recovery.

Entropy coding was not considered in this work. Learning an entropy model and performing entropy coding at the output of the encoder could further decrease the bit-rate, offering greater rate-distortion performance. More importantly, image biases were observed in the images formed using data recovered from VQVAE and VQVAE-RDA. Further studies on the effects of data normalization during training and testing with various range block sizes is essential. This could improve image quality and help reduce memory and inference time requirements, enabling efficient usage of the framework.

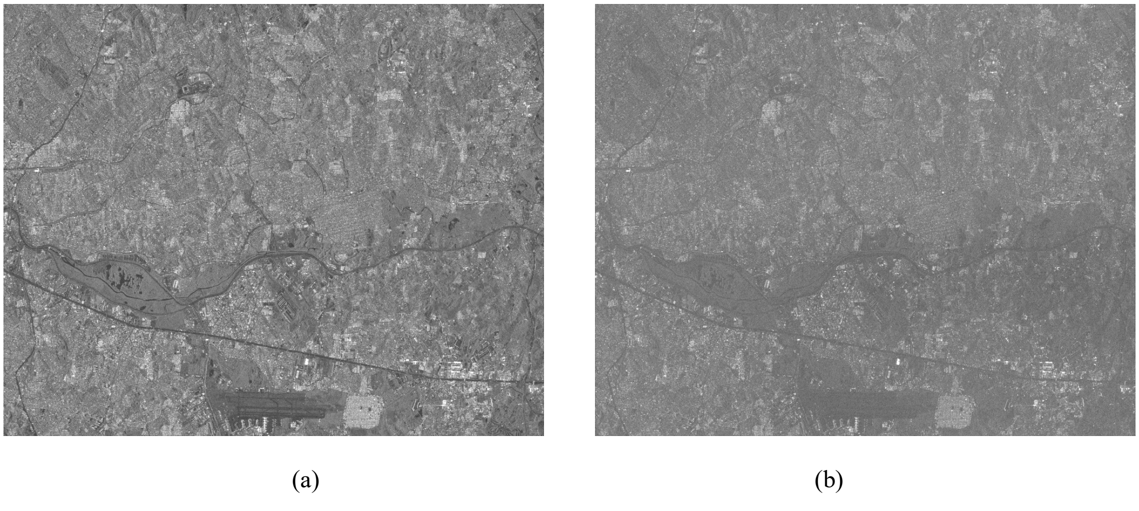
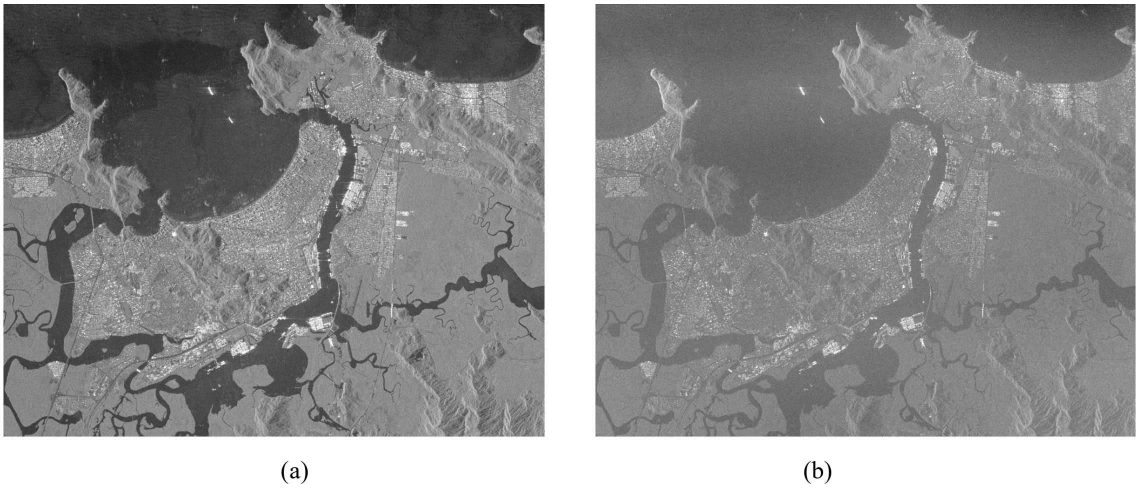


Fig. 6 (a) Portion of Sao Paulo image, focused from FDBAQ-processed data and (b) focused from VQVAE-recovered data (1.12 bits/sample). Contains modified Copernicus Sentinel Data [2022].



(a) (b)

Fig. 7 (a) Another portion of the Sao Paulo image, focused from FDBAQ-processed data and (b) focused from VQVAE-recovered data (1.12 bits/sample). Contains modified Copernicus Sentinel Data [2022].

Architecture choices with various layers, channels, codebook sizes and their effect on bit-rate, image quality and inference time need to be studied. Future investigations on the quality of the phase component of the image is also required, with additional performance indicators, as it is an important part of many downstream products. Experiments using data from different regions of the world, with various reflectivity scenarios are essential to understand the suitability of VQVAE. Currently, experiments were performed on isolated scenarios and greater generalizability has not been investigated. However, these preliminary results illustrate the potential of deep learning for lossy SAR raw data compression.

REFERENCES

[1] R. Kwok and W. T. K. Johnson, "Block adaptive quantization of Magellan SAR data," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 27, no. 4, pp. 375-383, July 1989, doi: 10.1109/36.29557.

[2] E. Attema et al., "Flexible Dynamic Block Adaptive Quantization for Sentinel-1 SAR Missions," in *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 4, pp. 766-770, Oct. 2010, doi: 10.1109/LGRS.2010.2047242.

[3] J. Ballé et al., "Nonlinear Transform Coding," *in IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 2, pp. 339-353, Feb. 2021, doi: 10.1109/JSTSP.2020.3034501.

[4] J. Ballé and V. Laparra and E. P. Simoncelli, “End-to-end optimized image compression” in *International Conference on Learning Representations (ICLR)*, 2017.

[5] J. Ballé and D. Minnen and S. Singh and S. J. Hwang and N. Johnston, “Variational image compression with a scale hyperprior”. in *International Conference on Learning Representations (ICLR)*, 2018.

[6] V. Alves de Oliveira et al Reduced-Complexity End-to-End Variational Autoencoder for on Board Satellite Image Compression. *Remote Sensing*. 2021; 13(3):447. https://doi.org/10.3390/rs13030447

[7] B. Beusen, X. Ivashkovych, and T. Van Achteren, “Image Compression using Vector-Quantized Auto-Encoders with semantically meaningful feature extraction” in *4S Symposium*, May 2022.

[8] A. Oord, K. Kavukcuoglu, and O. Vinyals. Neural discrete representation learning, *Advances on Neural Information Processing Systems (NIPS)*, 2017.

[9] X. X. Zhu et al., "Deep Learning Meets SAR: Concepts, models, pitfalls, and perspectives," in *IEEE Geoscience and Remote Sensing Magazine*, vol. 9, no. 4, pp. 143-172, Dec. 2021, doi: 10.1109/MGRS.2020.3046356.

[10] M. Martone, N. Gollin, P. Rizzoli and G. Krieger, "Performance-Optimized Quantization for SAR and InSAR Applications," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-22, 2022, Art no. 5229922.

[11] A. G. Mullissa, C. Persello and A. Stein, "PolSARNet: A Deep Fully Convolutional Network for Polarimetric SAR Image Classification," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 12, pp. 5300-5309, Dec. 2019, doi: 10.1109/JSTARS.2019.2956650.

[12] Q. Xu et al., "Synthetic Aperture Radar Image Compression Based on a Variational Autoencoder," in *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1-5, 2022, Art no. 4015905.

[13] R. Bamler, "A comparison of range-Doppler and wavenumber domain SAR focusing algorithms," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 30, no. 4, pp. 706-713, July 1992, doi: 10.1109/36.158864.

[14] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek and K. P. Papathanassiou, "A tutorial on synthetic aperture radar," in *IEEE Geoscience and Remote Sensing Magazine*, vol. 1, no. 1, pp. 6-43, March 2013.