Variable Rate Image Compression Method with Dead-zone Quantizer

Jing Zhou[1], Akira Nakagawa[2], Keizo Kato[2], Sihan Wen[1], Kimihiko Kazui[2], Zhiming Tan[1]

[1]Fujitsu R&D Center Co. Ltd., [2]Fujitsu Laboratories Ltd.

[1]{zhoujing, wensihan, zhmtan}@cn.fujitsu.com,[2]{anaka, kato.keizo, kazui.kimihiko}@fujitsu.com

Abstract

Deep learning based image compression methods have achieved superior performance compared with transform based conventional codec. With end-to-end Rate-Distortion Optimization (RDO) in the codec, compression model is optimized with Lagrange multiplier λ. For conventional codec, signal is decorrelated with orthonormal transformation, and uniform quantizer is introduced. We propose a variable rate image compression method with dead-zone quantizer. Firstly, the autoencoder network is trained with RaDOGAGA [6] framework, which can make the latents isometric to the metric space, such as SSIM and MSE. Then the conventional dead-zone quantization method with arbitrary step size is used in the common trained network to provide the flexible rate control. With dead-zone quantizer, the experimental results show that our method performs comparably with independently optimized models within a wide range of bitrate.

1. Introduction

Image compression is a kind of traditional and well-studied technique. With the key challenge of rate and distortion tradeoff, traditional codecs, such as JPEG [14] and JPEG2000 [10], usually break the pipeline into 3 modules: transformation, quantization, and entropy codec. Joint optimization over rate and distortion has longly been considered as an intractable problem. For transform coding [5], components are optimized separately, then they are fit together manually. Taking JPEG for example, it uses 8x8 block in Discrete Cosine Transform (DCT), then adapts run-length encoding to exploit the sparsity pattern of extracted frequency coefficients. Quantization is applied on coefficients to realize different compression level.

With the development of deep learning technology, more and more methods have been proposed to realize end-to-end rate distortion tradeoff [1, 2, 7–9, 16, 17]. Autoencoder has been proved to reduce dimensionality effectively by introducing an $'information\ bottleneck'$ that forces the network to find and exploit redundancies. Variational AutoEncoder (VAE) was evolved to model the underlying data distribution explicitly with sacrifice of distortion. To some extent the autoencoder-based compression methods [2,12] can be seen as VAEs, while the entropy model for rate estimation corresponds to the prior on the latents.

In these RDO methods, the Lagrange multiplier λ is introduced to modulate tradeoff between rate and distortion as $R + \lambda \cdot D$. A common way for rate adaptation is training multiple models with different λ, which is very tedious. To deal with such limitation, many methods have been proposed [3, 4, 12, 13]. Toderici et al. [13] proposed a RNN based progressive encoding and decoding scheme to generate target quality from low to high level, which requires large hardware storage and high performance in application. Choi et al. [3] combines the Lagrange multiplier and quantization bin size to realize rate control. But the range of quantizer is constrained within a small range to reduce performance degradation. To cover a broad rate range, multiple models trained with different λ are used.

Taking traditional codec into consideration, if neural network based image compression can be trained to realize orthonormal transformation, such as Karhunen-Loéve transform (KLT) and DCT, we can introduce the conventional quantization method to the neural network based orthonormal encoder. Inspired from this, we propose a variable rate image compression method with dead-zone quantizer [15]. RaDOGAGA [6] can realize an orthonormal latent space with minimal entropy, such as KLT and DCT. Because of orthonormality (orthogonal and uniform scaling) and minimal entropy, the common network can be used for an arbitrary quantizer. Firstly, the autoencoder network is trained with RaDOGAGA with metrics such as Mean Square Error (MSE), and Multi-Scale Structural SIMilarity (MS-SSIM), et al. Latents isometric to these metric spaces are derived. In inference, a uniform quantizer is utilized to obtain variable rates with different step sizes as traditional codec with

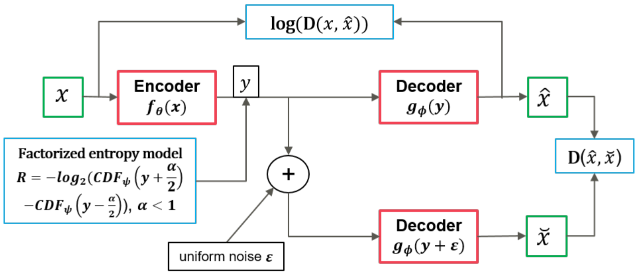

log(D(x, $\hat{x}$))
x
Encoder
$f_\theta(x)$
y
Decoder
$g_\phi(y)$
$\hat{x}$
Factorized entropy model
$R = -log_2(CDF_\psi\left(y+\frac{\alpha}{2}\right)$
$-CDF_\psi\left(y-\frac{\alpha}{2}\right)), \alpha < 1$
+
D($\hat{x}$, $\breve{x}$)
uniform noise ε
Decoder
$g_\phi(y+\varepsilon)$
$\breve{x}$

Figure 1. The RaDOGAGA framework for training

fixed encoder/decoder. To further improve its performance, the dead-zone quantizer is introduced. According to our experimental results, our method can achieve comparable R-D performance with models optimized separately both in PSNR and MS-SSIM.

2. Variable rate image compression

2.1. Training framework: RaDOGAGA

As for RaDOGAGA, which is a rate-distortion optimization guided autoencoder, it proves that deep autoencoder can achieve orthonormal transform with RDO. Just like common compression framework, it mainly contains encoder $f_\theta(\boldsymbol{x})$, decoder $g_\phi(y)$ and factorized entropy model Cumulative Density Function $CDF_\psi(y)$ with parameters θ, ϕ and ψ. For the factorized entropy model as described in [2], bounds of latents y should be finite and known ahead. The bounds for latents are also trained, which contains not only the maximum and minimum ranges, but also median value as a kind of quantization offset. The probability can be calculated as follows from $CDF_\psi(y)$, where α<1.

$$P_\psi(y) = CDF_\psi(y + \frac{\alpha}{2}) - CDF_\psi(y - \frac{\alpha}{2}) \qquad (1)$$

Compared with the mainstream $R + \lambda \cdot D$ loss, the difference can be illustrated from two aspects. The first is that two inputs exist for the decoder $g_\phi(y)$. One is the extracted latents y, and the corresponding output is $\hat{\boldsymbol{x}}$. The other is noised latents $y + \epsilon$, where ϵ is uniform noise with variance σ, and the corresponding output is $\breve{\boldsymbol{x}}$. The second aspect is loss function shown in Eq. 2.

$$L = -log_2(P_\psi(y)) + \lambda_1 \cdot h(D(\boldsymbol{x}, \hat{\boldsymbol{x}})) + \lambda_2 \cdot D(\hat{\boldsymbol{x}}, \breve{\boldsymbol{x}}) \quad (2)$$

The first distortion $D(\boldsymbol{x}, \hat{\boldsymbol{x}})$ is for reconstruction, and the second $D(\hat{\boldsymbol{x}}, \breve{\boldsymbol{x}})$ aims to influence the scaling, i.e., Jacobian matrix. According to Rolíneck et al. [11], $D(x, \breve{x}) \simeq D(\boldsymbol{x}, \hat{\boldsymbol{x}}) + D(\hat{\boldsymbol{x}}, \breve{\boldsymbol{x}})$. Here we use $h(d) = log(d)$, which can encourage better reconstruction and orthogonality. $D(\cdot)$ is an arbitrary metric, such as MSE, SSIM, or Binary Cross Entropy (BCE), etc.

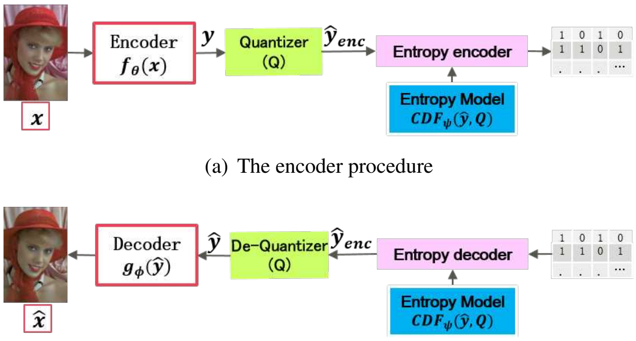
As shown in RaDOGAGA, the latent becomes orthonormal to the metric defined image space with minimal entropy such as KLT. Let $D(\boldsymbol{x}, \hat{\boldsymbol{x}})$ be a metric such as SSIM, MSE, and so on. Usually $D(\boldsymbol{x}, \boldsymbol{x} + \delta\boldsymbol{x})$ can be approximated by ${}^T\delta\boldsymbol{x}\boldsymbol{A_x}\delta\boldsymbol{x}$ where $\delta\boldsymbol{x}$ is an arbitrary micro displacement and $\boldsymbol{A_x}$ is a Riemannian metric tensor at $\boldsymbol{x}$. By using RaDOGAGA, each row vector of Jacobi matrix becomes orthonormal at any $\boldsymbol{x}$ in the inner product space with metric $\boldsymbol{A_x}$ as follows, where δ_{ij} denotes Kronecker's delta.

$${}^T(\frac{\partial \boldsymbol{x}}{\partial y_i})\boldsymbol{A_x}(\frac{\partial \boldsymbol{x}}{\partial y_j}) = \frac{1}{2\lambda_2\sigma^2}\delta_{ij} \qquad (3)$$

2.2. Compression framework with quantization

In this section, we'll explain a compression framework with an arbitrary step sized quantization by using the common encoder $f_\theta(x)$, decoder $g_\phi(y)$ and entropy model $CDF_\psi(y, Q)$. After training with RaDOGAGA, the latents are optimized to be orthonormal to metric defined inner product space. For example, if trained with MS-SSIM, the space of latent is isometric against Riemannian manifold whose metric is MS-SSIM. Thus rate control can be realized as conventional way.

Firstly, input signal is decorrelated by orthonormal transformation with the $Encoder$. Then decorrelated data y can be quantized uniformly with arbitrary Q, short for quantization step size. Then quantized symbols $\hat{y}_{enc}$ are entropy coded using the $Entropy\ Model$ after $Quantizer$. In the decoder procedure, the symbols are entropy decoded losslessly using the arithmetic coding with the estimated probability, dequantized, and fed to the $Decoder$ as shown in Figure 2(b). Thus, with arbitrary Q, flexible rate control can be realized easily.



(a) The encoder procedure

(b) The decoder procedure

Figure 2. The compression framework

2.3. Dead-zone quantizer

We employ a simple dead-zone quantizer widely used in image compression such as H.264 as shown in Figure 3. Based on the trained CDF_ψ, quantizer, and dead-zone offset, the probability of each latent in $\hat{y}$ is estimated in advance for each representative.

Compressed symbols $\hat{y}_{enc}$ for entropy codec is shown in Eq. 4 with dead-zone quantizer, where $sgn(\cdot)$ is the signum function. y is centered on the median value from trained CDF_ψ, while 0 is used for simplicity. The $offset$ can be from 0 to 0.5. If $offset = 0.5$, it means round quantization.

$$\hat{y}_{enc} = sgn(y) \cdot \lfloor \frac{|y|}{Q} + offset \rfloor \tag{4}$$

Then $\hat{y}$ can be obtained with $De\text{-}Quantizer(Q)$.

$$\hat{y} = \hat{y}_{enc} \cdot Q \tag{5}$$

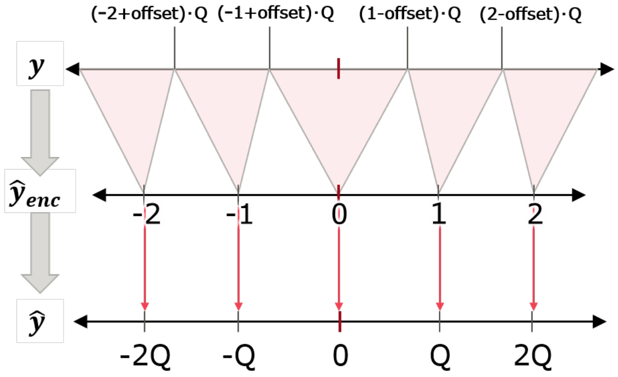


Figure 3. The dead zone quantizer.

Figure 3 shows the upper y_{upper} and lower y_{lower} bound of y for each quantized representative value. A probability for a quantize symbol $\hat{y}$ is estimated with equations as follow, where $0 < \omega < 1$.

$$y_{upper} = (\hat{y}_{enc} + 0.5 + sgn(\hat{y}_{enc} + \omega) \times (0.5 - offset)) \cdot Q \tag{6}$$

$$y_{lower} = (\hat{y}_{enc} - 0.5 + sgn(\hat{y}_{enc} - \omega) \times (0.5 - offset)) \cdot Q \tag{7}$$

$$P_{\hat{y}_{enc}} = CDF(y_{upper}) - CDF(y_{lower}) \tag{8}$$

3. Experiments

3.1. Training method

The training dataset contains more than 6000 images. Some of them are from lossless dataset, such as dataset from Workshop and Challenge on Learned Image Compression (CLIC)[1] and DIV2K, a super-resolution dataset. Others are from flickr.com, where images are not in lossless format with 2×2 down sampling to reduce compression artefacts. In training, the network is fed with $256 \times 256 \times 3$ patches cropped from these full resolution images randomly with a minibatch size of 8. Each image is normalized by dividing by 255 for each RGB channel.

To compare the performance of our variable rate compression method with independently optimized method, we use the same network structure in [1], and models are trained with its official open source code[2]. We set the bottleneck number 128. If optimized for MSE, it's calculated after scaling the normalized images to 255.

3.2. Conventional model with independent training

For neural network based conventional method, such as [1], the R-D curve is pointed through models trained with different λ. Each point represents a compression level optimized with a specific R-D tradeoff λ independently. Such method can achieve ideal performance, but it requires more memory to store and more computing to train them. In other words, the training time is multiplied. To compare the RD curves both in PSNR and MS-SSIM, different λ and loss are used. For PSNR, MSE is distortion in training, and $\lambda \in \{0.001, 0.003, 0.005, 0.01, 0.02, 0.03, 0.1\}$. For MS-SSIM, it's the same training loss with $\lambda \in \{4, 8, 16, 32, 64, 96\}$.

3.3. Proposed model with arbitrary quantization

We train our model as RaDOGAGA framework explained in section 2.1 with MS-SSIM and MSE. Parameters are shown in Table 1. If optimized for MSE, it means two distortions in Eq. 2 are MSE in training. Once the training is completed, Q in a relative wide range can be used to tune the desired rate/distortion. What's more, we use the dead-zone quantizer in compression. Different quantization step sizes are applied to obtain different rates. For example, $Q \in \{0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.5, 3, 3.5, 4\}$ are used in models optimized for MS-SSIM.

Distortion	λ_1	λ_2	α
MSE	5	0.2	0.2
MS-SSIM	1	256	0.2

Table 1. Parameters for our model

3.4. Performance

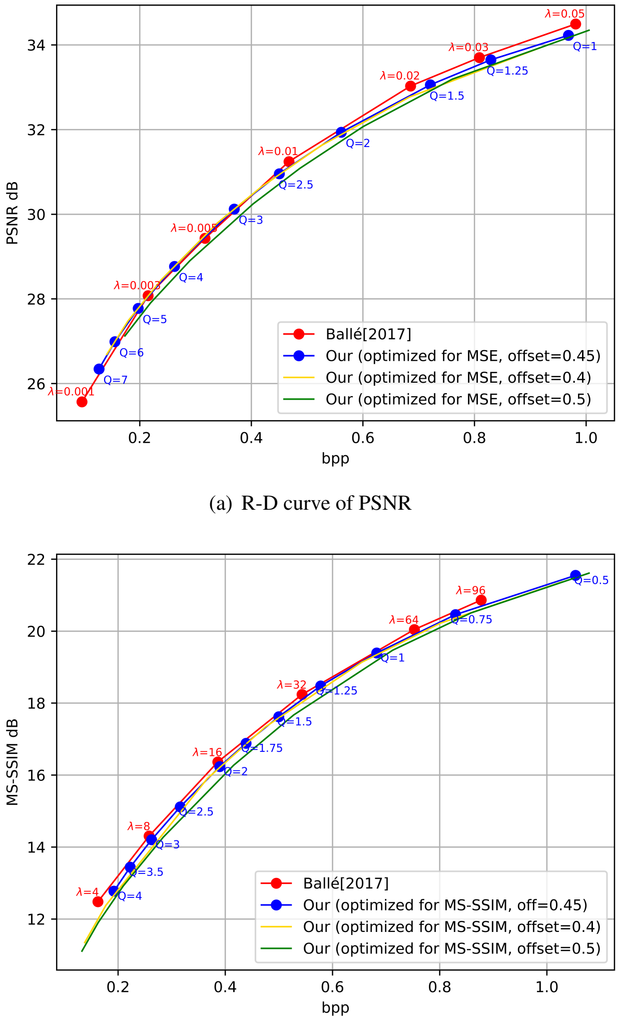
Evaluated on 24 Kodak images[3], the comparison results are shown in Figure 4. MS-SSIM scores are measured in dB: $MS\text{-}SSIM_{dB} = -10 \cdot log_{10}(1 - MS\text{-}SSIM)$. To check the influence of different offsets in dead-zone quantizer, $offset \in \{0.4, 0.45, 0.5\}$ are used. As we can see, our performance is comparative with independently optimized model both on PSNR and MS-SSIM with $offset = 0.45$. With small offset, quantization error increases for the centered value, which will result in aggravating distortion. So appropriate offset is beneficial to improve performance.

Figure 5 shows the reconstructed images through our variable rate image compression method to assess their visual quality. For $Q = 4$, we can find that our method can maintain the texture well, such as areas of girl's eyelash and

[1] https://www.compression.cc

[2] https://github.com/tensorflow/compression

[3] http://r0k.us/graphics/kodak



(a) R-D curve of PSNR

(b) R-D curve of MS-SSIM dB

Figure 4. PSNR 4(a) and MS-SSIM 4(b) comparison on Kodak dataset. The '$\lambda = 16$' around the red curve indicates model is optimized with $R + 16 \cdot D$. The blue text '$Q = 1$' means that quantization step is 1.

hair. However, the independently optimized model smooths small texture to realize better R-D tradeoff. If using larger quantization step, although the objective performance does not degrade, there is mosquito noise on the crisp edge of objects, which is a common phenomenon in classical codec compressed with DCT.

4. Conclusion

In this paper, we propose a variable rate image compression method based on a common autoencoder. We find that RaDOGAGA framework can realize the orthonormal latent space, which can be seen as the way of getting DCT coefficients. From this point, we introduce conventional quantizer, such as the one in JPEG/MPEG. According to our experimental results, our method can vary rate without objective metric degradation. We can deploy the trained model within a broad range of rate flexibly, which will bring great convenience in practical application. In this paper, we used simple one layer model. However, the concept of isometric latent space is generic, and will be widely applicable to more complex multi-layer compression model.

References

[1] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01706*, 2016.

[2] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

[3] Yoojin Choi, El-Khamy Mostafa, and Lee Jungwon. Variable rate deep image compression with a conditional autoencoder. pages 3146–3154, 2019.

[4] Yang Fei, Herranz Luis, Weijier Joost van de, Guitin Jos A.Iglesias, Lpez Antonio, and et al. Variable rate deep image compression with modulated autoencoder. In *IEEE Signal Processing Letters*, pages 27: 331–335, 2020.

[5] Vivek K. Goyal. Theoretical foundations of transform coding. pages 18(5):9, 21, 2001.

[6] Keizo Kato, Jing Zhou, Sasaki Tomotake, and Nakagawa Akira. Rate-distortion optimization guided autoencoder for isometric embedding in euclidean latent space. *arXiv preprint arXiv:1910.04329*, 2019.

[7] Jooyoung Lee, Seunghyum Cho, and Seung-Kwon Beack. Context-adaptive entropy model for end-to-end optimized image compression. *arXiv preprint arXiv:1809.10452*, 2018.

[8] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Rado Timofte, and Luc Van Gool. Conditional probability models for deep image compression. pages 4394–4402, 2018.

[9] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, pages 10771–10780, 2018.

[10] Majid Rabbani. Jpeg2000: Image compression fundamentals, standards and practice. pages 11(2):286, 2002, 2002.

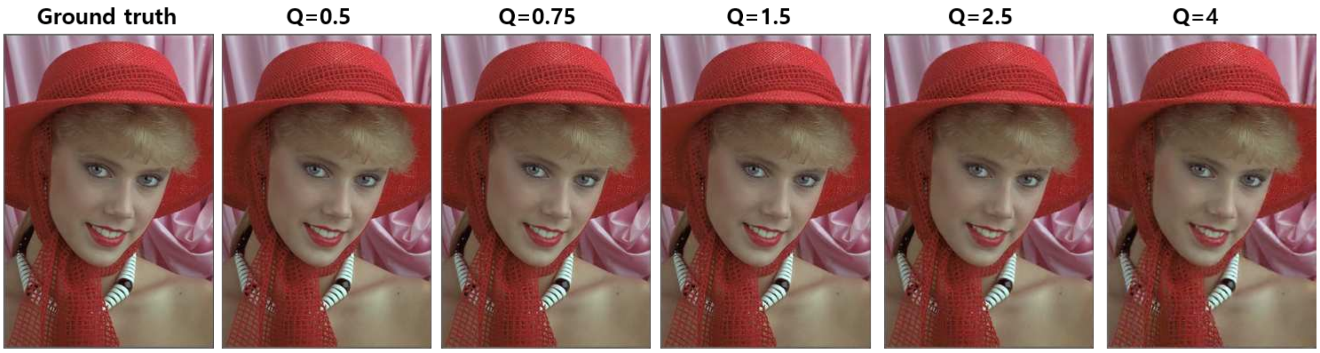
[11] Michal Rolínek, Zietlow Dominik, and Martius Georg. Variational autoencoders pursue pca directions (by accident). pages 12406–12415, 2019.

[12] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017.

[13] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. pages 5306–5314, 2017.

[14] Gregory K. Wallace. The jpeg still picture compression standard. pages 38.1: xviii–xxxiv, 1992.

[15] Thomas Wedi and Steffen Wittmann. Quantization offsets for video coding. pages Vol. 1:324–327, 2005.



Bits per pixel(BPP)	1.052	0.829	0.496	0.307	0.182
PSNR(dB)	31.0863	30.8652	30.0879	29.2659	28.2194
MS-SSIM	0.9928	0.9907	0.9815	0.9668	0.9292

Figure 5. Reconstructed images with different Q, model optimized with MS-SSIM and $offset = 0.45$

[16] Sihan Wen, Jing Zhou, Akira Nakagawa, Kazui Kimihiko, and Zhiming Tan. Variational autoencoder based image compression with pyramidal features and context entropy model. *In WORKSHOP AND CHALLENGE ON LEARNED IMAGE COMPRESSION*, 2019.

[17] Jing Zhou, Sihan Wen, Akira Nakagawa, Kazui Kimihiko, and Zhiming Tan. Multi-scale and context-adaptive entropy model for image compression. *arXiv preprint arXiv:1910.07844*, 2019.