Learned Distributed Image Compression with Multi-Scale Patch Matching in Feature Domain

Yujun Huang[1,3], Bin Chen[2,3,4*], Shiyu Qin[2], Jiawei Li[5], Yaowei Wang[3], Tao Dai[6], Shu-Tao Xia[1,3]

[1] Tsinghua Shenzhen International Graduate School, Tsinghua University
[2] Harbin Institute of Technology, Shenzhen
[3] Research Center of Artificial Intelligence, Peng Cheng Laboratory
[4] Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies
[5] HUAWEI Machine Co., Ltd. DongGuan
[6] Shenzhen University
huangyj20@mails.tsinghua.edu.cn, chenbin2021@hit.edu.cn, 190110427@stu.hit.edu.cn, li-jw15@tsinghua.org.cn,
wangyw@pcl.ac.cn, daitao.edu@gmail.com, xiast@sz.tsinghua.edu.cn

Abstract

Beyond achieving higher compression efficiency over classical image compression codecs, deep image compression is expected to be improved with additional side information, e.g., another image from a different perspective of the same scene. To better utilize the side information under the distributed compression scenario, the existing method only implements patch matching at the image domain to solve the parallax problem caused by the difference in viewing points. However, the patch matching at the image domain is not robust to the variance of scale, shape, and illumination caused by the different viewing angles, and can not make full use of the rich texture information of the side information image. To resolve this issue, we propose **M**ulti-**S**cale **F**eature **D**omain **P**atch **M**atching (MSFDPM) to fully utilizes side information at the decoder of the distributed image compression model. Specifically, MSFDPM consists of a side information feature extractor, a multi-scale feature domain patch matching module, and a multi-scale feature fusion network. Furthermore, we reuse inter-patch correlation from the shallow layer to accelerate the patch matching of the deep layer. Finally, we find that our patch matching in a multi-scale feature domain further improves compression rate by about 20% compared with the patch matching method at image domain.

Introduction

Distributed image compression is less studied than the commonly known single image compression, as the inherent correlation among images taken from different viewpoints is hard to capture. With the development of the stereo camera, camera array, self-driving system, and multiple UAV/monitor camera systems during the past decade, high-volume multi-view images become the primary data source in large-scale digital communication. For example, it has been reported that self-driving cars can capture 1G of data per second (Mearian 2013). Since multi-view images are usually captured from different angles of the same scene at the same time, there exist overlapping fields with high correlation among these images. In order to improve the compression efficiency for such distributed compression scenario, the inherent correlation must be utilized to remove redundancy in data. This inspires the development of distributed compression with side information, where the sensors encode their data, independently, and the decoders recover the data with the help of side information from another source only available at the decoder. The Distributed Source Coding (DSC) theorem (Slepian and Wolf 1973; Cover 1975; Wyner and Ziv 1976) reveals that the same compression rate can be asymptotically achieved by using side information (SI) only

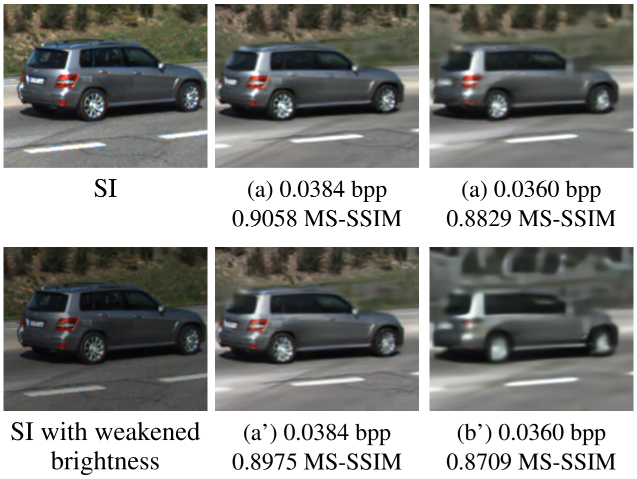

SI
(a) 0.0384 bpp
0.9058 MS-SSIM
(a) 0.0360 bpp
0.8829 MS-SSIM
SI with weakened brightness
(a') 0.0384 bpp
0.8975 MS-SSIM
(b') 0.0360 bpp
0.8709 MS-SSIM

Figure 1: Visual comparisons of decoded main images with different compression methods when the brightness of the side information (SI) image is normal or weakened. (a) and (b) are examples of path matching in the multi-scale feature domain (proposed) and patch matching in the image domain, respectively, when the brightness is normal. (a') and (b') are the corresponding examples when the brightness of side information image is 60% of the original image. The visual perception of (a) is similar to (a'). However, (b') loses some color and texture details compared to (b).

[*] Corresponding author.

at the decoding end as when available at both encoder and decoder. However, there is still a performance gap between these two coding schemes in practice.

Recently, with the success of deep learning in various computer vision tasks (Liao et al. 2022; Chai et al. 2022), deep image compression (Ballé, Laparra, and Simoncelli 2017; Ballé et al. 2018; Mentzer et al. 2018) has made great improvement over classical codecs like JPEG. In a deep image compression model, a variational autoencoder is adopted as a basic architecture for end-to-end optimized encoding and decoding. For a given input image, the multiple nonlinear layers extract its latent feature representation. Then the entropy of the quantized feature layer is estimated by an entropy model. During decoding, the quantized feature is input into a symmetric decoder to reconstruct the image. By optimizing the rate-distortion (R-D) loss over the large-scale training set, deep image compression achieves state-of-the-art compression ratios.

In recent years, some learning-based distributed image coding methods have been proposed (Diao, Ding, and Tarokh 2020; Ayzik and Avidan 2020; Mital et al. 2022). Ayzik et al. (Ayzik and Avidan 2020) proposed to align the side information image with the main image by patch matching in the image domain. However, operating only in the image domain can not make full use of the abundant texture details in side information, and is not robust to the variants of illumination and scale caused by different viewing points. As shown in Fig. 1, the gain of the side information in the image domain achieves a slight improvement when the brightness of the side image is weakened. By contrast, our proposed multi-scale feature domain patch matching method still makes good use of the correlative texture information within the side information image. This significantly demonstrates the robustness and efficiency of the multi-scale feature domain patch matching for side information utilization.

We introduce a **M**ulti-**S**cale **F**eature **D**omain **P**atch **M**atching (MSFDPM) method to better utilize the side information image at the decoder. Similarity matching in the multi-scale feature domain can fully explore the correlative texture information in the side information image and is robust to variants of illumination and scale. Meanwhile, the complementary scale features can effectively recover the decoded main image from fine-grained texture to coarse-grained structures simultaneously. Specifically, MSFDPM consists of three modules: (1) A feature extractor network to extract multi-scale side information features from side information image; (2) Patch matching module to obtain aligned side information features, in which the inter-patch correlation of the largest scale feature layer is reused to later feature layers to accelerate the decoding process; (3) Feature fusion network to concatenate the decoded multi-scale features of the main image with the side information image to obtain high-quality reconstruction.

To summarize, we make the following contributions.

- We propose a new paradigm of multi-scale feature domain patch matching method for deep distributed image compression.
- We validate that the patch matching in the multi-scale feature domain is more robust to variants of illumination and scale than patch matching in the image domain.
- We provide comprehensive experiments to show that our method achieves the state-of-the-art performance.

Related Work

Deep Image Compression Deep image compression has attracted more and more attention in recent years. Thanks to the powerful representation capability of autoencoder, flexible entropy model, and end-to-end optimization architecture, deep image compression methods have outperformed reigning image compression standards. Ballé *et al.* first proposed an end-to-end autoencoder structure based on rate-distortion optimization for image compression (Ballé, Laparra, and Simoncelli 2017). Ballé *et al.* further proposed a hyperprior model to capture the spatial correlation of latent feature map and transfer the captured information to the decoder as side information (Ballé et al. 2018). Minnen *et al.* proposed an autoregressive prior and a hierarchical prior to obtain the correlation between neighboring components in the latent feature map (Minnen, Ballé, and Toderici 2018). Cheng *et al.* proposed to use the discretized mixture Gaussian model to exactly model the distribution of latent feature map and introduced the attention modules into the autoencoder (Cheng et al. 2020). Xie *et al.* proposed to use an Invertible Encoding Network instead of the autoencoder structure to mitigate information loss (Xie, Cheng, and Chen 2021). Kim *et al.* proposed a novel entropy model to focus on long-range correlations in hidden layer features through an attention mechanism (Kim, Heo, and Lee 2022).

Learned Distributed Source Coding Recently, some papers use deep learning to implement distributed source coding. Ding *et al.* proposed to use recurrent autoencoder for distributed image coding, but without using side information images when decoding main image (Diao, Ding, and Tarokh 2020). It only uses decoder model to learn the correlation. Ayzik *et al.* proposed that the side information image could be aligned with the main image by patch matching method when there is a large overlapping field between them, and the aligned side information image could be used to obtain better main image reconstruction (Ayzik and Avidan 2020). However, the information available in image domain is limited and not robust enough. Mital *et al.* proposed to extract the common information from the side information image through a network and input the common information and the latent feature map of main image into a decoder network to obtain an enhanced reconstruction (Mital et al. 2022). However, the proposed method does not align the side information image with main image, which will lead to the degraded reconstruction quality when the parallax is large.

The Proposed Method

The main idea of our proposed MSFDPM decoder is to transmit both high-level semantic information and low-level texture information to the main image based on multi-scale feature domain patch matching, so that useful information can be screened out for reconstruction. Specifically, our method first uses a feature extractor to obtain multi-scale side information features, aligns the features with the main

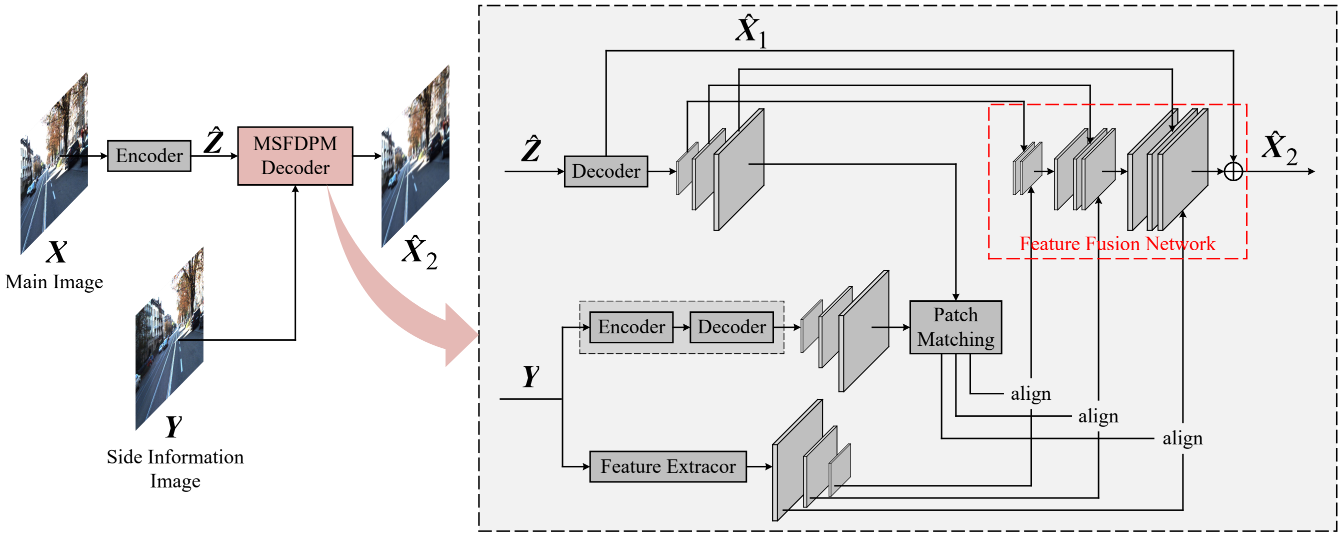


Figure 2: The proposed MSFDPM architecture for side information based decoding of deep image compression

features by patch matching, and then fuses the aligned side information features with main features to get a better reconstruction. The proposed method is shown in Fig. 2.

Multi-Scale Features of Main Image and Side Information Image

Before patch matching, we need the model to obtain multi-scale features of the main image and side information image. First, the main image $\boldsymbol{X}$ is fed into the encoder of the single-image compression model to obtain latent feature map $\hat{\boldsymbol{Z}}$. Because single-image decoding is an up-sampling process, the decoder could take the latent feature map $\hat{\boldsymbol{Z}}$ and output multi-scale decoded main features $\left\{\boldsymbol{F}_{\hat{\boldsymbol{X}}}^{h} \in \mathbb{R}^{\frac{H}{2^h} \times \frac{W}{2^h} \times C} \middle| h = 1, 2, 3, 4\right\}$ and the first stage decoded main image $\hat{\boldsymbol{X}}_1 \in \mathbb{R}^{H \times W \times C}$, where H and W are the height and width of the main image. We use smaller superscripts to represent larger scale features, which facilitates the following description of patch matching. With the same single-image autoencoder, the side information image $\mathbf{Y}$ can be used to produce multi-scale decoded side information features $\left\{\boldsymbol{F}_{\hat{\boldsymbol{Y}}}^{h} \in \mathbb{R}^{\frac{H}{2^h} \times \frac{W}{2^h} \times C} \middle| h = 1, 2, 3, 4\right\}$. Because the decoded main features and the decoded side information features are produced by the same autoencoder, they have a similar property, which will improve matching accuracy. However, the decoded side information features lose part of the side information because of the quantization noise, lossless side information features are needed to improve the main image reconstruction. A trainable feature extractor is proposed to take the side information image and produce the multi-scale lossless side information features $\left\{\boldsymbol{F}_{\boldsymbol{Y}}^{h} \in \mathbb{R}^{\frac{H}{2^h} \times \frac{W}{2^h} \times C} \middle| h = 1, 2, 3, 4\right\}$. Our feature extractor uses the encoder structure in (Cheng et al. 2020).

Multi-Scale Feature Domain Patch Matching

Because of the disparity problem caused by different viewing points, the side information image needs to be aligned with the main image to effectively use the side information. In addition, compared to the image domain, which would be more affected by variance of illumination and color caused by different viewing points, the feature domain will lay more emphasis on the texture and structure to be migrated. Moreover, because it is time-consuming to calculate the inter-patch correlation on all feature layers, we choose to calculate only this correlation on the first feature layer, which can be reused by other layers. For the convenience of description, we define the set of small patches sampled by a $B \times B$ window with stride s on a $H \times W \times C$ feature map T as:

$$S^{T,B,s} = \{p_{i,j}^{T,B,s} \mid i = 0, \cdots, I, j = 0, \cdots, J\} \quad (1)$$
$$\text{where } I \triangleq \left\lfloor \frac{W-B}{s} \right\rfloor, J \triangleq \left\lfloor \frac{H-B}{s} \right\rfloor$$

According to this definition, we can obtain the patch set of the first decoded main feature layer $S^{\boldsymbol{F}_{\hat{\mathbf{X}}}^{1},B,B}$, the first decoded side information feature layer $S^{\boldsymbol{F}_{\hat{\mathbf{Y}}}^{1},B,1}$, and multi-scale lossless side information $\left\{S^{\boldsymbol{F}_{\boldsymbol{Y}}^{h},\frac{B}{2^{h-1}},1} \middle| h = 1, 2, 3, 4\right\}$. It is worth noting that the patches of the decoded main feature are not overlapping and cover all the regions of the feature map, while the patch sampling of the side information feature is fine-grained because we need to match the most similar patch on the side information feature map. We use the pearson correlation coefficient to measure the inter-patch correlation between patches of the first decoded main feature layer and the first decoded side information feature layer:

$$r_{(i,j),(k,l)} = \Pr(p_{i,j}^{\boldsymbol{F}_{\hat{\mathbf{X}}}^{1},B,B}, p_{k,l}^{\boldsymbol{F}_{\hat{\mathbf{Y}}}^{1},B,1}) * m_{(i,j),(k,l)}, \quad (2)$$

where $m_{(i,j),(k,l)}$ is the Gaussian mask proposed in (Ayzik and Avidan 2020), which is the prior to choose adjacent

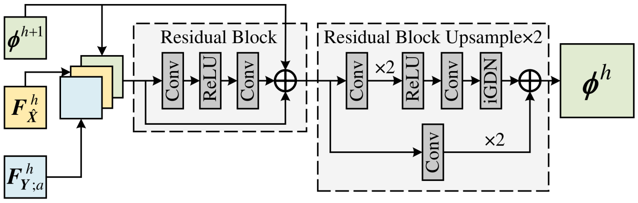


Figure 3: One iteration of feature fusion network.

patches with higher probability. The computationally expensive inner product operation in the pearson correlation can be efficiently calculated by a series of convolutions, where the convolution kernel is each patch of the decoded main feature, with the decoded side information feature as input:

$$p_{i,j}^{\boldsymbol{F}_{\hat{\boldsymbol{X}}}^1,B,B} * \boldsymbol{F}_{\hat{\boldsymbol{Y}}}^1, \tag{3}$$

where $*$ denotes the convolution operation. Based on the pearson correlation coefficient, we can find the location of the side information feature patch that are most similar to each patch in the main feature:

$$(k^*_{(i,j)}, l^*_{(i,j)}) = \underset{(k,l)}{\arg\max}\, r_{(i,j),(k,l)}. \tag{4}$$

Then we can get the first aligned side information feature layer $\boldsymbol{F}_{\boldsymbol{Y};a}^1$ by putting the most similar lossless side information feature patch into the corresponding position in the main feature:

$$p_{i,j}^{\boldsymbol{F}_{\boldsymbol{Y};a}^1,B,B} = p_{k^*_{(i,j)},l^*_{(i,j)}}^{\boldsymbol{F}_{\boldsymbol{Y}}^1,B,1}. \tag{5}$$

Reusing First Feature Layer Inter-Patch Correlation We reduce the computational complexity and GPU memory footprint of patch matching by reusing the above-mentioned first feature layer inter-patch correlation. First, we correspond the patches of the second to the fourth feature layer to the patches of the same position in the first feature layer. A toy example is shown in Appendix A. We find that for the decoded main features the position subscripts of patches corresponding to the same position in different feature layers are the same. For the decoded side information features, the position subscript of the patch in the previous layer is twice that of the corresponding patch in the later layer. So we can use the correlation between the corresponding patches of the first feature layer to represent the inter-patch correlation of the following layer:

$$R(p_{i,j}^{\boldsymbol{F}_{\hat{\boldsymbol{X}}}^h,\frac{B}{2^{h-1}},\frac{B}{2^{h-1}}}, p_{k,l}^{\boldsymbol{F}_{\hat{\boldsymbol{Y}}}^h,\frac{B}{2^{h-1}},1}) = r_{(i,j),(2^{h-1}k,2^{h-1}l)}, \quad h = 2,3,4. \tag{6}$$

Then we can get the second to the fourth aligned side information feature layers $\{\boldsymbol{F}_{\boldsymbol{Y};a}^2, \boldsymbol{F}_{\boldsymbol{Y};a}^3, \boldsymbol{F}_{\boldsymbol{Y};a}^4\}$ by using the method similar to Eq. 5.

Feature Fusion Network

Feature fusion network is designed to iteratively fuse aligned side information features and decoded main features from small scale to large scale, as illustrated in the red box in Fig. 2. The effectiveness of feature fusion at multiple scales is demonstrated in ablation experiments. Specifically, in one iteration of the feature fusion network, the aligned side information feature and the decoded main feature at scale h, as well as the output feature of the previous iteration (if any) are concatenated and input into two residual blocks to obtain the fused feature at scale h:

$$\begin{aligned} \boldsymbol{\phi}^4 &= \mathrm{Res}_2^4\left(\mathrm{Res}_1^4\left(\boldsymbol{F}_{\hat{\boldsymbol{X}}}^4, \boldsymbol{F}_{\boldsymbol{Y};a}^4\right)\right), \\ \boldsymbol{\phi}^h &= \mathrm{Res}_2^h\left(\mathrm{Res}_1^h\left(\boldsymbol{F}_{\hat{\boldsymbol{X}}}^h, \boldsymbol{F}_{\boldsymbol{Y};a}^h, \boldsymbol{\phi}^{h+1}\right) + \boldsymbol{\phi}^{h+1}\right), \\ & \qquad h = 1,2,3. \end{aligned} \tag{7}$$

The architectures of the residual blocks refer to the decoder of single-image compression model (Cheng et al. 2020). Fig. 3 illustrates in detail one iteration of the feature fusion network. Finally, the second stage decoded main image is obtained by adding the latest output of the feature fusion network to the first stage decoded main image:

$$\hat{\boldsymbol{X}}_2 = \boldsymbol{\phi}^1 + \hat{\boldsymbol{X}}_1 \tag{8}$$

Loss Function

Lossy compression is a joint optimization problem of compression rate and distortion. For the proposed model, the loss function consists of the entropy of the latent feature map $\hat{\boldsymbol{Z}}$, the distortion of the first stage decoded main image and the second stage decoded main image:

$$\mathcal{L} = H(\hat{\boldsymbol{Z}}) + \lambda((1-\alpha)d(\boldsymbol{X}, \hat{\boldsymbol{X}}_1) + \alpha d(\boldsymbol{X}, \hat{\boldsymbol{X}}_2))), \tag{9}$$

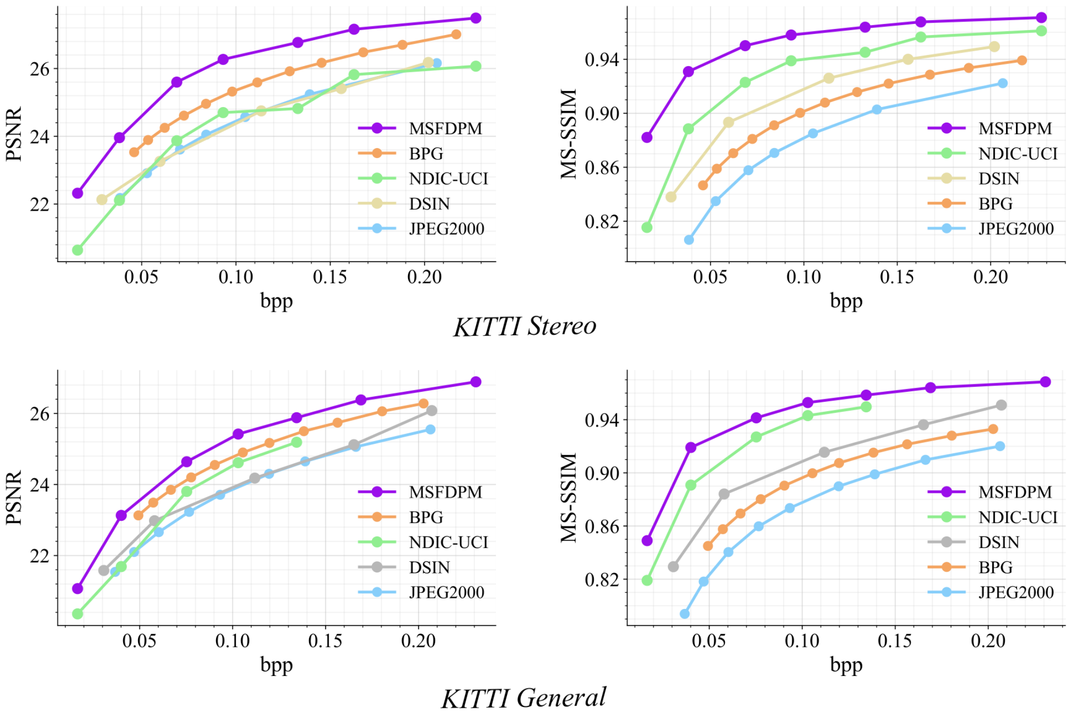
where λ is the weight that controls trade-off between the compression rate and the distortions. α is the weight that controls the trade-off between the two distortions.

Experiments and Results

Experimental Setup

Datasets We conduct experiments on two datasets: *KITTI Stereo* and *KITTI General* proposed in (Ayzik and Avidan 2020). *KITTI Stereo* contains 1578 training pairs and 790 test pairs, which are paired stereo images from the KITTI Stereo 2012 (Geiger, Lenz, and Urtasun 2012) and KITTI Stereo 2015(Menze and Geiger 2015) dataset. *KITTI General* has 174936 training pairs and 3609 test pairs. The pairs of data are different not only in viewing points but also in time steps.

Evaluation Metrics We use bits per pixel (bpp) to measure the compression ratio. Both peak signal-to-noise ratio (PSNR) and multi-scale structural similarity (MS-SSIM) (Wang, Simoncelli, and Bovik 2003) are used to measure image quality/signal distortion. Moreover, we use BD-Rate, negative numbers to indicate the percentage of average bit savings at the same image quality across the rate-distortion curve compared with some chosen baselines. Specifically, BD-Rate$_{\mathrm{P}}$ and BD-Rate$_{\mathrm{M}}$ indicate the performance gain under the same PSNR and MS-SSIM, respectively.



KITTI Stereo

KITTI General

Figure 4: The rate-distortion performance comparison of different methods.

Baselines We compare the MSFDPM with 4 baselines. Two classical image compression baselines: JPEG 2000(Rabbani 2002) and BPG (Bellard 2014). And two learned distributed image compression baselines: DSIN (Ayzik and Avidan 2020) and NDIC-UCI (Mital et al. 2022). DSIN and NDIC-UCI are briefly introduced in related work.

Implementation Details The proposed MSFDPM is implemented with PyTorch (Paszke et al. 2019) and the experiments are conducted on four Intel(R) Xeon(R) E5-2698 v4 CPUs and eight NVIDIA Tesla V100 GPUs. We first train a single-image compression baseline (Cheng et al. 2020). Then we train the full model with cropped 320×960 image pair, where the parameters of the autoencoder are initialized with the pretrained baseline. The number of epochs on *KITTI Stereo* is 10 and the number of epochs on *KITTI General* is 1. We used a batch size of 1 and the Adam optimizer (Kingma and Ba 2015) with $1 \cdot 10^{-4}$ learning rate. Other hyper-parameters are listed as follows: (**i**) The number of features, $C = 128$. (**ii**) The patch size, $B = 16$. (**iii**) The weight for rate-distortion trade-off, $\lambda \in \{0.005, 0.01, 0.02, 0.035, 0.05, 0.07, 0.1\}$. (**iv**) The weight for two stages of distortions, α is equal to 0 when training the autoencoder baseline and 1 when training the full model.

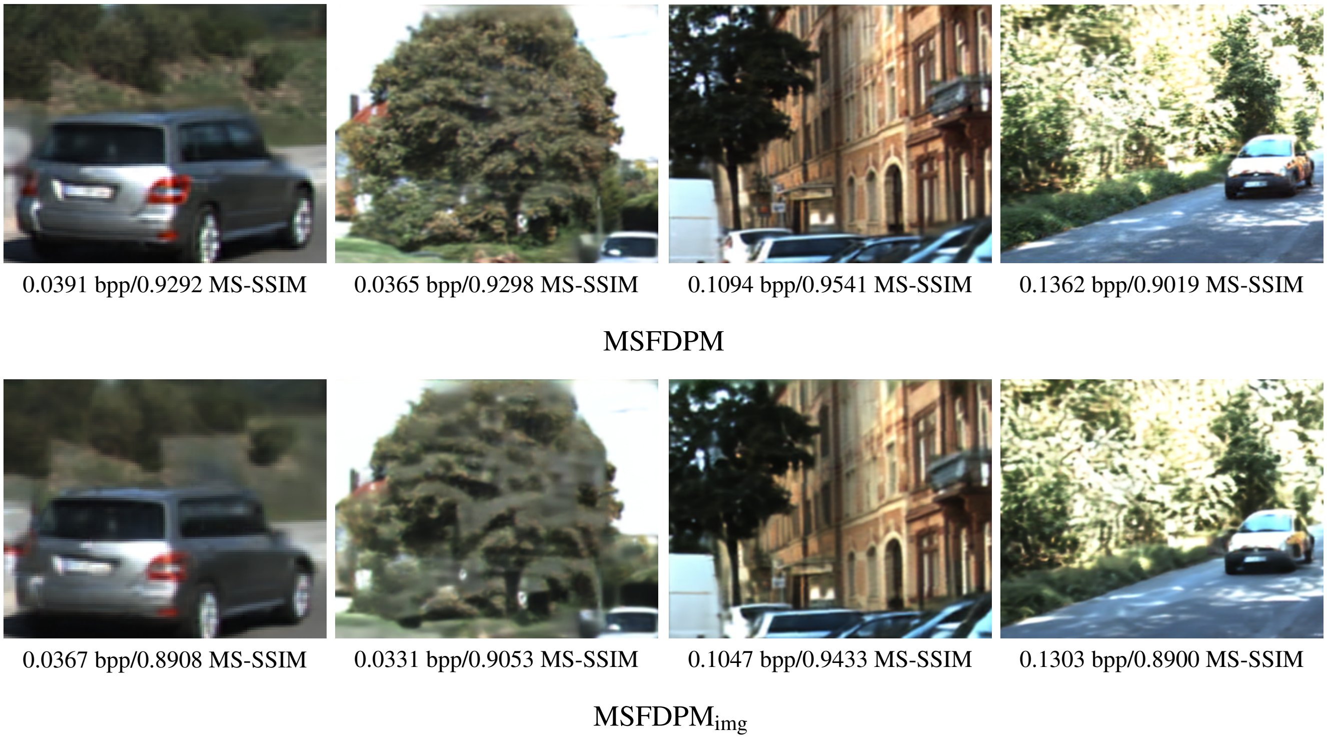
Results and Analysis

Results We report the rate-distortion results in Fig. 4, which shows that our MSFDPM outperforms all comparison baselines across different compression ratios. The increase in compression is significant. For example, in the KITTI Stereo dataset, when the MS-SSIM is 0.90, the bpp of MSFDPM is about 0.025, that of NDIC-UCI is about 0.048, and that of DSIN is about 0.07, that of BPG is about 0.098, and that of JPEG 2000 is about 0.134. Compared with NDIC-UCI, DSIN, BPG, and JPEG 2000, MSFDPM has a compression rate increase of 1.92 times, 2.80 times and 3.92 times, 5.36 times, respectively.

Variant	*KITTI Stereo*		*KITTI General*	
	BD-Rate$_\mathrm{P}$	BD-Rate$_\mathrm{M}$	BD-Rate$_\mathrm{P}$	BD-Rate$_\mathrm{M}$
MSFDPM	**-49.26%**	**-51.11%**	**-40.37%**	**-40.89%**
MSFDPM$_\mathrm{img}$	-27.81%	-24.54%	-18.72%	-13.53%
MSFDPM$_\mathrm{f1}$	-41.29%	-42.02%	-31.15%	-29.92%
MSFDPM$_\mathrm{f2}$	-45.99%	-49.13%	-38.77%	-38.50%
MSFDPM$_\mathrm{f3}$	-25.43%	-33.25%	-32.72%	-35.12%
MSFDPM$_\mathrm{f4}$	-24.80%	-33.22%	-24.41%	-27.89%
MSFDPM$_\mathrm{w/o\ r}$	-47.01%	-49.89%	-40.06%	-40.41%

Table 1: The BD-Rate results of different MSFDPM variants. Baseline is single-image compression model (Cheng et al. 2020).

Ablation Experiments To explore the effect of multi-scale feature patch matching and inter-patch correlation reusing, we design 3 variants of MSFDPM: (**i**) MSFDPM$_\mathrm{img}$ replaces multi-scale feature domain patch matching with image domain patch matching. The aligned side information image and the first stage decoded main image are input into a network containing four residual blocks to obtain fine reconstruction. (**ii**) MSFDPM$_{fh}$, $h = 1, 2, 3, 4$, only fuses the h-th aligned side information feature layer in the feature fu-



0.0391 bpp/0.9292 MS-SSIM 0.0365 bpp/0.9298 MS-SSIM 0.1094 bpp/0.9541 MS-SSIM 0.1362 bpp/0.9019 MS-SSIM

MSFDPM

0.0367 bpp/0.8908 MS-SSIM 0.0331 bpp/0.9053 MS-SSIM 0.1047 bpp/0.9433 MS-SSIM 0.1303 bpp/0.8900 MS-SSIM

$\text{MSFDPM}_{\text{img}}$

Figure 5: Visual comparisons of MSFDPM and $\text{MSFDPM}_{\text{img}}$.

sion network. (**iii**) $\text{MSFDPM}_{\text{w/o r}}$ performs block matching at each feature layer without reusing the inter-patch correlation of the first feature layer.

- **Multi-Scale Feature Domain Patch Matching v.s. Image Domain Patch Matching.** MSFDPM outperforms $\text{MSFDPM}_{\text{img}}$ by $\text{BD-Rate}_{\text{P}}$ of -21.45% and $\text{BD-Rate}_{\text{M}}$ of -26.57% on *KITTI Stereo* and $\text{BD-Rate}_{\text{P}}$ of -31.65% and $\text{BD-Rate}_{\text{M}}$ of -27.36% on *KITTI General*. This implies that the multi-scale feature domain can extract more texture or structure information to help obtain higher quality reconstruction compared to the image domain. Fig. 5 presents some visual comparisons of MSFDPM and $\text{MSFDPM}_{\text{img}}$. Benefiting from the multi-scale feature patch matching, MSFDPM can provide finer texture structure and richer color details than $\text{MSFDPM}_{\text{img}}$.
- **Multi-Scale Feature Fusion v.s. Single-Scale Feature Fusion.** MSFDPM outperforms $\text{MSFDPM}_{\text{f1}}$, $\text{MSFDPM}_{\text{f2}}$, $\text{MSFDPM}_{\text{f3}}$ and $\text{MSFDPM}_{\text{f4}}$ on both $\text{BD-Rate}_{\text{P}}$ and $\text{BD-Rate}_{\text{M}}$. This indicates that multi-scale feature fusion can improve the quality of reconstruction by using complementary fine-grained texture information to coarse-grained structure information. Specifically, MSFDPM outperforms $\text{MSFDPM}_{\text{f1}}$, $\text{MSFDPM}_{\text{f2}}$, $\text{MSFDPM}_{\text{f3}}$, $\text{MSFDPM}_{\text{f4}}$ by an average $\text{BD-Rate}_{\text{P}}$ of -8.59%, -2.43%, -15.74%, -20.21% and an average $\text{BD-Rate}_{\text{M}}$ of -10.03%, -2.18%, -11.82%, -15.45%. The BD-rate performance of $\text{MSFDPM}_{\text{f2}}$ is the best among all the single-scale feature fusion models. This may be because the second feature layer has a better trade-off between high-frequency detailed texture and low-frequency structural information. The performance of $\text{MSFDPM}_{\text{f3}}$ and $\text{MSFDPM}_{\text{f4}}$ is worse than that of $\text{MSFDPM}_{\text{f1}}$ and $\text{MSFDPM}_{\text{f2}}$, which indicates that texture information is more useful to improve image quality than structure information.

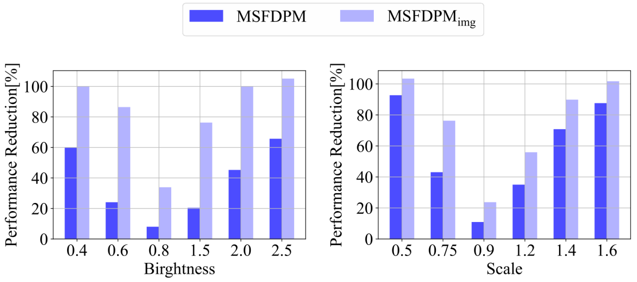


Figure 6: The performance reduction of MS-SSIM of MSFDPM and MSFDPMimg as side information image brightness and scale change ($\lambda = 0.035$).

Robustness Experiments In order to explore the robustness of the multi-scale feature domain and image domain to illumination and scale change, we adjuest the brightness and scale of the side information image during inference. We set the brightness and scale of the original side information image are 1, and proportionally change the brightness and scale of the original image. To evaluate the performance reduction

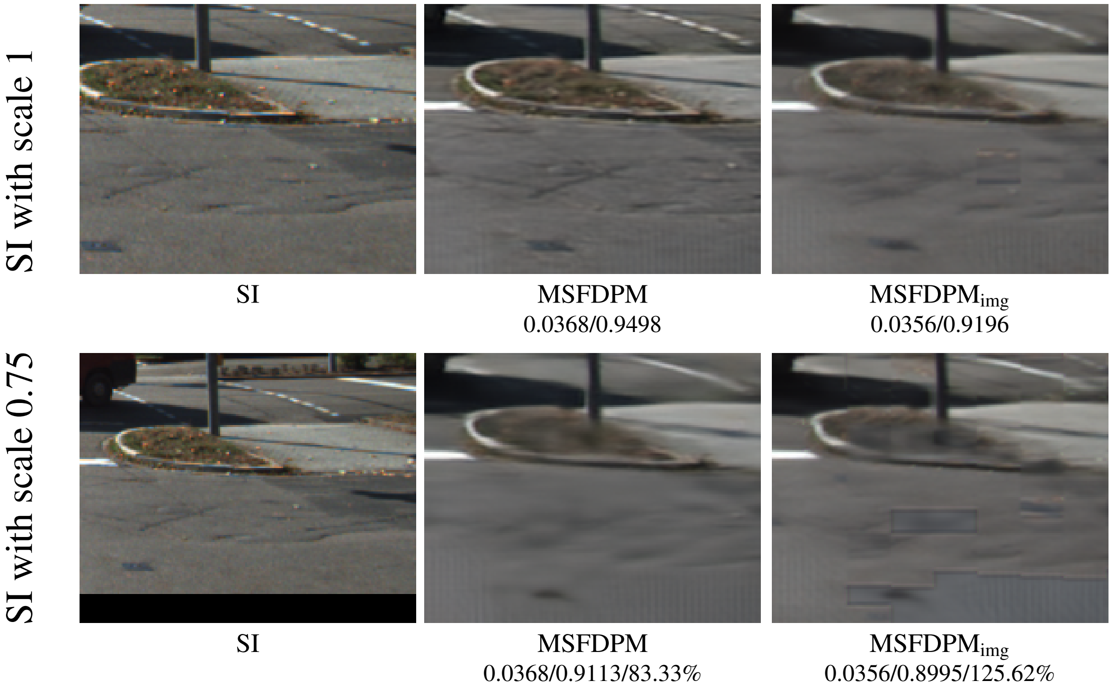


Figure 7: Visual comparisons of MSFDPM and MSFDPM$_{\text{img}}$ with scale 1 and 0.75. Evaluation Metric is denoted by bpp/MS-SSIM/PR.

(PR), we define it as

$$\text{PR} = 1 - \frac{\text{performance improvement after change}}{\text{performance improvement before change}} \quad (10)$$

As shown in Fig. 6, when the brightness is 0.8, the numerical MS-SSIM improvement of the model using side information is 0.015 compared with the model without using side information. When the brightness is 1, the numerical MS-SSIM improvement of the model using side information is 0.02. Then the performance reduction at 0.8 brightness is $1 - \frac{0.015}{0.02} = 25\%$.

MSFDPM outperforms MSFDPM$_{\text{img}}$ by an average of performance reduction of -46.39% and -18.45% with brightness change and scale change, respectively. Besides, it can be seen that the performance reduction of MSFDPM is consistently lower than that of MSFDPM$_{\text{img}}$ across different brightness and scales. Therefore, we can achieve the conclusion that the multi-scale feature domain is more robust to the variants of illumination and scale than the image domain. Besides, the performance reduction of the multi-scale feature domain is much lower during brightness change. For example, when the brightness is 0.75, the performance reduction of MSFDPM$_{\text{img}}$ is 86.44% and the performance reduction of MSFDPM is only 24.09%. This demonstrates that the feature domain can mitigate the brightness variation and emphasize texture and structure.

Fig. 7 shows the visualization results when the scale of the side information image changes. It can be seen that when the scale is 0.75, the example of MSFDPM$_{\text{img}}$ shows a clustering block effect while the example of MSFDPM does not. This may be because the change of scale causes the corresponding patches cannot be accurately matched in the image domain. Appendix B shows more visualization results.

Variant	Decoding	Guassian Masks	GPU Memory
MSFDPM	**0.5404s**	**22.86s**	**9908Mb**
MSFDPM$_{\text{w/o r}}$	0.5740s	25.30s	10791Mb

Table 2: Decoding speed, generation speed of Gaussian mask, and GPU memory usage of MSFDPM and MSFDPM$_{\text{w/o r}}$.

Efficiency Analysis To verify the efficiency of reusing the first feature layer inter-patch correlation, we compare the decoding speed, Gaussian mask generation speed, and GPU memory usage of MSFDPM and MSFDPM$_{\text{w/o r}}$, and the results are shown in Table 2. The decoding speed and Gaussian mask generation speed of MSFDPM are 6% and 11% faster than MSFDPM$_{\text{w/o r}}$. And MSFDPM has 8% less GPU memory usage than MSFDPM$_{\text{w/o r}}$. Besides, as shown in Table 1, the BD-rate of MSFDPM is slightly lower than that of MSFDPM$_{\text{w/o r}}$. This may be because the later decoded feature layers are closer to the quantization and are more likely inaccurately match patches due to the quantization noise.

Conclusions

In this paper, we propose multi-scale feature domain patch matching (MSFDPM) for distributed image compression. Compared with the previous patch matching in the image domain, MSFDPM is more robust to the variants of illumination and scale caused by different viewing points, and the feature domain contains richer texture information in the side information image. In addition, we reuse the first feature layer inter-patch correlation to improve the decoding efficiency. On the experimental side, we demonstrate the superiority of MSFDPM over state-of-the-art baseline, while being more robust to variants of illumination and scale.

Acknowledgments

This work was done when Yujun Huang was an internship at Harbin Institute of Technology, Shenzhen. This work is supported in part by the National Natural Science Foundation of China under grant 62171248, the PCNL KEY project (PCL2021A07), Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (2022B1212010005), Guangdong Basic and Applied Basic Research Foundation under grant 2021A1515110066, the GXWD 20220811172936001, the Shenzhen Science and Technology Innovation Commission (Research Center for Computer Network (Shenzhen) Ministry of Education), and Shenzhen Science and Technology Program under Grant JCYJ20220818101012025.

References

Ayzik, S.; and Avidan, S. 2020. Deep image compression using decoder side information. In *European Conference on Computer Vision*, 699–714. Springer.

Ballé, J.; Laparra, V.; and Simoncelli, E. 2017. End-to-end optimized image compression. 5th International Conference on Learning Representations, ICLR 2017 ; Conference date: 24-04-2017 Through 26-04-2017.

Ballé, J.; Minnen, D.; Singh, S.; Hwang, S. J.; and Johnston, N. 2018. Variational Image Compression with a Scale Hyperprior. In *6th Int. Conf. on Learning Representations (ICLR)*.

Bellard, F. 2014. BPG Image format. https://bellard.org/bpg/. Accessed: 2017-01-30.

Chai, H.; Yin, Z.; Ding, Y.; Liu, L.; Fang, B.; and Liao, Q. 2022. A Model-Agnostic Approach to Mitigate Gradient Interference for Multi-Task Learning. *IEEE Transactions on Cybernetics*, 1–14.

Cheng, Z.; Sun, H.; Takeuchi, M.; and Katto, J. 2020. Learned Image Compression With Discretized Gaussian Mixture Likelihoods and Attention Modules. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7936–7945.

Cover, T. 1975. A proof of the data compression theorem of Slepian and Wolf for ergodic sources (Corresp.). *IEEE Transactions on Information Theory*, 21(2): 226–228.

Diao, E.; Ding, J.; and Tarokh, V. 2020. Drasic: Distributed recurrent autoencoder for scalable image compression. In *2020 Data Compression Conference (DCC)*, 3–12. IEEE.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kim, J.-H.; Heo, B.; and Lee, J.-S. 2022. Joint Global and Local Hierarchical Priors for Learned Image Compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5992–6001.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.

Liao, Q.; Chai, H.; Han, H.; Zhang, X.; Wang, X.; Xia, W.; and Ding, Y. 2022. An Integrated Multi-Task Model for Fake News Detection. *IEEE Trans. Knowl. Data Eng.*, 34(11): 5154–5165.

Mearian, L. 2013. Self-driving cars could create 1GB of data a second. *Computerworld*, 23.

Mentzer, F.; Agustsson, E.; Tschannen, M.; Timofte, R.; and Van Gool, L. 2018. Conditional probability models for deep image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4394–4402.

Menze, M.; and Geiger, A. 2015. Object Scene Flow for Autonomous Vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Minnen, D.; Ballé, J.; and Toderici, G. D. 2018. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31.

Mital, N.; Özyılkan, E.; Garjani, A.; and Gündüz, D. 2022. Neural distributed image compression using common information. In *2022 Data Compression Conference (DCC)*, 182–191. IEEE.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, 32: 8026–8037.

Rabbani, M. 2002. JPEG2000: Image Compression Fundamentals, Standards and Practice. *Journal of Electronic Imaging*, 11(2): 286.

Slepian, D.; and Wolf, J. 1973. Noiseless coding of correlated information sources. *IEEE Transactions on information Theory*, 19(4): 471–480.

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

Wyner, A.; and Ziv, J. 1976. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on information Theory*, 22(1): 1–10.

Xie, Y.; Cheng, K. L.; and Chen, Q. 2021. Enhanced invertible encoding for learned image compression. In *Proceedings of the 29th ACM International Conference on Multimedia*, 162–170.