Compressed Domain Multiframe Processing

Chengyu Wang[1] Jing Li[1] Saurabh Kumar[2] Seok-Jun Lee[1] Hamid R. Sheikh[1]
[1]Samsung Research America [2]Samsung Research India-Bangalore
{chengyu.wang, jing.li1, saurabh.k1, seokjun1.lee, hr.sheikh}@samsung.com

Abstract

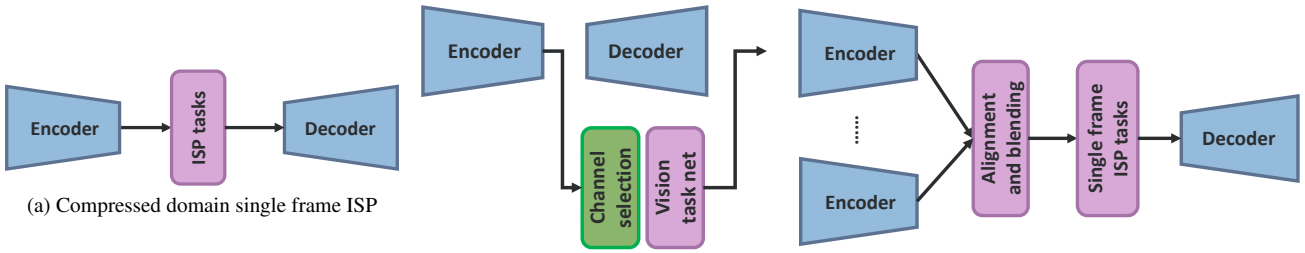
Multiframe processing has become an essential component in mobile devices to produce images with better qualities, such as reduced noise and improved dynamic range. However, processing multiple frames poses challenges in system memory and computation power, especially for high resolution images. In this work we present a compressed domain multiframe processing pipeline that operates in a compressed domain defined by an encoder-decoder and vector quantization. The encoder-decoder learns the features from the raw frame and produces the RGB image. Vector quantization is used to quantize the feature to achieve compression. In this compressed domain we show common multiframe processing functions, including demosaicing, denoising, image registration, image deghosting and HDR blending. Experiments on real mobile captures demonstrate the effectiveness of the proposed compressed domain multiframe processing pipeline. The proposed method achieves image quality similar to non-compression methods with less memory and computation requirement.

1. Introduction

Mobile cameras suffer from low image quality because of the small optical lenses and imaging sensor. To improve the image quality, multiframe processing (MFP) is commonly used. Instead of capturing one frame from the scene, the camera captures multiple frames and outputs a single image by blending these frames. MFP can produce images with better details, less noise and high dynamic range (HDR). Meanwhile, the imaging resolution has been increasing in recent years. Several smartphone cameras today can capture images with 200 megapixels. However, for high resolution imaging, capturing and processing multiple frames requires high system memory. For example, when a 200-megapixel image is considered, each raw frame takes 400MB to store assuming 16-bit processing, and a demosaiced image takes more than 1GB. In this case, processing multiple frames is extremely challenging.

Image compression techniques have been proposed to tackle the problem of high memory demand in storing images. Conventional methods, such as JPEG [38], BPG [6] and PGF [33], achieve high compression ratio and high reconstruction fidelity, but the compressed data cannot be directly used for image signal processing (ISP) tasks, meaning the decompression needs to be applied before any operations can be performed. In recent years, learning based compression methods have achieved superior compression ratio using techniques such as autoencoder, transformer, compressed sensing and vector quantization [2, 4, 7, 32, 34, 37], and the compressed data have been used for various image processing and computer vision tasks, including denoising, warping, object detection, semantic segmentation and object detection [23, 24, 43]. All these work, however, consider only single image processing, and the MFP in the compressed domain has been under-explored. In fact, moving the processing to the compressed domain not only reduces system memory but also reduces the computation burden, because the ISP tasks are applied to a reduced data volume. Motivated by this, we investigate processing multiple frames in the compressed domain to enable MFP for high-resolution imaging on mobile devices.

To build the MFP pipeline in the compressed domain, several features should be considered when designing such a domain. First, the compressed domain should be naturally suitable for ISP downstream tasks, and no minimal decompression is required before images can be processed. Second, similar to an ISP pipeline, the inputs to the compression algorithm should be raw frames with certain color filter array patterns, and the decompression algorithm outputs the RGB images. Third, image warping is essential in MFP because multiple frames need to be first registered before they can be blended, so the compressed domain should be warpable without introducing artifacts in the decompressed outputs. Fourth, to enable the HDR imaging, the compressed domain should have the capability to process images with extended dynamic range. Fifth, certain image analysis tasks in ISP, such as semantic analysis, do not need a full resolution image, so the compressed domain should be interpretable in a way, also called progressive encoding [43], that partial data can be used for image processing



(a) Compressed domain single frame ISP

(b) Vision tasks in the compressed domain

(c) Proposed multiframe compressed domain processing

Figure 1. Comparison between image processing pipeline with compression model. (a) Single image compressed domain processing pipeline applies ISP tasks on the compressed data [43]. (b) Computer vision tasks are applied on selected latent representation and output the results in the pixel domain [23]. (c) Proposed compressed domain multiframe processing pipeline.

tasks to reduce computation.

In this paper, we present a compressed domain incorporating these features. The compressed domain is defined by an encoder-decoder and a vector quantization (VQ) module. The encoder-decoder adopts the U-Net [30] structure to learn the multiscale features from the input raw frame, and the VQ is then used to quantize and compress the features. To enable warping in the compressed domain, we introduce augmentation during the training so that the decoder can process warped features, and for HDR imaging, we propose to use multiple encoders, each corresponding to an individual exposure value (EV), and a single decoder. In this compressed domain, we develop an MFP pipeline that achieves demosaicing, denoising, registration, deghosting and HDR blending. This demonstrates how the compressed data can be utilized in different ways for ISP tasks.

The major contributions of this paper can be summarized as follows:

- We propose an image compression method with an encoder-decoder and a VQ module. The encoder-decoder learns the multi-scale features from the input raw image that is suitable for downstream ISP tasks, and the VQ compressed the features, achieving 3.3 times compression.
- We develop an MFP pipeline in the compressed domain that achieves demosaicing, denoising, registration, deghosting and HDR blending.
- Experiments with data captured from real mobile devices demonstrate the effectiveness of the proposed MFP pipeline.

2. Related work

2.1. Compression methods

Image compression aims to reduce the image data volume by removing the redundant and unrelated information with little or without degradation in the quality of the image. Conventional lossy compression methods typically contain three stages: transform, quantization and entropy coding. A linear transformation, such as DCT, first decorrelates the image into a set of coefficients in the transformed domain. Then the quantization method maps the continuous coefficients to several discrete values. Two types of quantization methods are usually considered: scalar quantization and vector quantization. Scalar quantization maps each coefficient to a discrete value, and vector quantization maps a vector or a block of coefficients with an index of a vector in a codebook [13]. In comparison, vector quantization achieves higher compression ratio [20]. Entropy coding further losslessly compresses the indexes, approaching the lower bound established by Shannon's source coding theorem. In the past decades, conventional methods, such as JPEG [38], BPG [6] and PGF [33], have achieved satisfactory Rate-Distortion (RD) performance.

Learned image compression are now state-of-the-art compression methods in research. Ballé *et* . [4] pioneered this by proposing the first CNN-based image compression model. Thereafter, various CNN-based methods have been proposed. Among these methods, variational auto-encoder (VAE) has been the most popular one [5, 8, 10, 18, 22]. To improve the entropy model, Minnen *et al.* [27] proposed a local context model, and Guo *et al.* [14] proposed a casual context model. Other than CNN, some work used recurrent neural networks for image compression [17, 35, 40]. Generative models have also been used for this task. Torfason *et al.* [36] presented joint image compression and classification with Generative Adversarial Networks (GAN). Agustsson *et al.* [3] provided the first study of using GAN for image compression, and Mentzer *et al.* [26] further studied each components in GAN. Recently diffusion-based compression has proposed. Hoogeboom *et al.* [15] and Ghouse *et al.* [12] proposed to enrich details using diffusion for images compressed with autoencoder. Relic *et al.* [29] proposed to use foundation diffusion models to remove quantization error. Yang *et al.* [42] proposed to use conditional diffusion models for end-to-end image compression. With the development of vision transformers, transformer-based image compression methods have been proposed. Zhu *et*

al. [44] proposed to use swin transformer for image compression. Qian *et al.* [28] proposed a transformer-based entropy model. Koyuncu *et al.* [21] proposed a transformer-based context model. Liu *et al.* [25] proposed image compression with mixed transformer-CNN architectures.

2.2. Compressed domain processing

Deep learning has achieved superior performance in various image processing tasks [1, 7, 11, 16]. However, compressed domain processing has received less attention. Xu *et al.* [41] used a gate module to select DCT features for image classification and segmentation. Shen *et al.* [31] proposed a DCT-mask to improve the instance segmentation task. As for learned compression method, Torfason *et al.* [36] proposed image classification and segmentation in the compressed domain. The compressed latent representation was directly sent to a computer vision task network. To reduce the bitrate in this scheme, Liu *et al.* [23] proposed to use only selective channels with high information entropy for the vision tasks, and this was further improved with a learned method for channel selection [24]. Wang *et al.* [39] also introduced a feature selection method using Gaussian approximation. Codevilla *et al.* [9] proposed to jointly train the compression model and the vision tasks to obtain a better deep representation that was suitable for other vision tasks. Ji *et al.* [19] proposed a vision transformer in the compressed domain for image classification. These work differ compressed domain ISP in two aspects. First, these work focus on high-level vision tasks, but the compressed domain ISP focuses on low-level image processing and enhancement. Second, the vision tasks directly estimate pixel domain results based on the compressed domain feature, while the compressed domain ISP needs to run tasks entirely in the compressed domain, which means any operation should be performed directly on the latent features.

Recently, Zhang *et al.* [43] proposed a compressed domain ISP that demonstrated compressed domain denoising. They also demonstrated that affine registration matrix could be estimated from the compressed domain, but it didn't apply warping to the latent features. So far there has been no work on multiframe processing in the compressed domain. The comparison between different methods are summarized in Figure 1.

3. Method

In this section we describe how to design the compressed domain and develop the MFP pipeline in this domain.

3.1. Compression Model

As discussed in previous sections, the compressed domain should be suitable for downstream MFP tasks and provide the interpretability for high level tasks. Here we define the compression space with an encoder-decoder and a vector quantization (VQ) module. This process is illustrated in Figure 2. The input to the encoder is the raw image data, and the output is the corresponding RGB image. Different from [43] where the encoded image is represented by a single feature matrix, the proposed method adopts the U-Net structure that produces multiscale features, and the VQ maps the continuous feature into discrete embeddings to achieve compression. The advantages of this compression scheme are two-fold. First, the multiscale features are naturally suitable for progressive encoding. The features close to the bottleneck contain more low-frequency structural information of the image, and the features with larger spatial dimension contain more high-frequency details. This can be ensured by adding loss constraint during the training period. Second, this structure allows us to compress features with various bit-depth based on their importance. Bottleneck features are quantized with less bits (more compression), and the features with more detail information are quantized with more bits (less compression). The details of the network dimensions and the codebook size are summarized in Table 1. With this setup, 3.3 times compression compared to the input, or 9.9 times compression compared to the RGB image, can be achieved, assuming 12-bit raw input. Note that here we do not consider entropy coding, which will further increase the compression ratio. One may question that de-quantization is still needed to process the images. The benefits of such a compression method is better illustrated with the system memory analysis. As shown in Figure 3, traditional image processing stores the full frame image in DRAM (dynamic random access memory), and the full frame is used by the processing unit. With the proposed method, only the bitstream is stored in DRAM, and we adopt tile-based processing, so only a small set of coefficients are dequantized for ISP tasks, further reducing the required computing memory.

Features	Index map	Size of codebook/bits
256×256×4	256×256	2048/11
128×128×8	128×128	2048/11
64×64×16	64×64	2048/11
32×32×32	32×32	1024/10
16×16×64	16×16	1024/10
8×8×128	8×8	1024/10
4×4×512	4×4	1024/10

Table 1. The dimensions of the proposed compression model, assuming a 256×256 input.

Although this compression method suffices for single frame processing, extra considerations are necessary for MFP functions in the compressed domain. For handheld captures, MFP needs to warp the input frames to a target frame before any analysis and blending can be per-

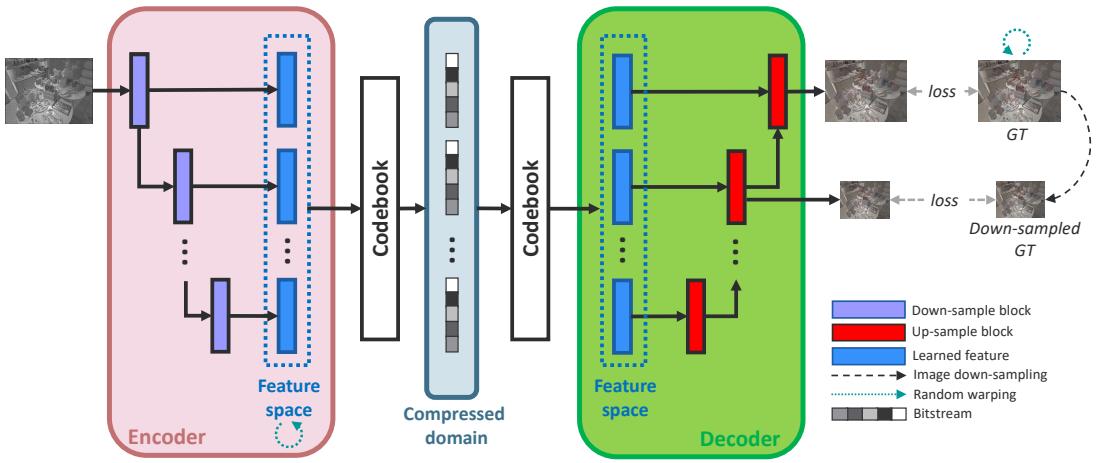
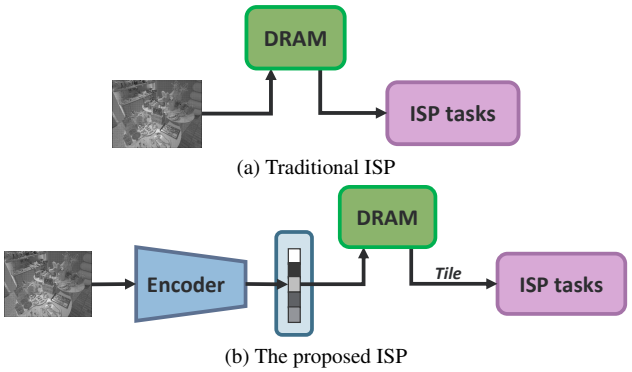


Figure 2. The proposed compression model. The compressed domain is defined by an encoder-decoder and a VQ module. Random warping is added during training to increase the warpability of the compressed domain, and downsampled GT is used to train subset of the decoder to achieve compressive encoding.



(a) Traditional ISP

(b) The proposed ISP

Figure 3. Comparing system memory between traditional ISP and the proposed compressed domain ISP. The proposed method reduces the required DRAM of the system by reducing the data volume and using tile-based processing.

formed. As shown in Section 4.2.1, image warping in the compressed domain can result in artifacts in the output image. To avoid this, augmentation is applied when training the compression model. During the training period, random warping is applied to the features in the compressed domain, and the same warping is also applied to the label image. This increases the warpability of the compressed space.

In high dynamic scenes, multiple frames with varying exposures are captured. In order to analyze and blend these frames, the encoder needs to map them into the same compressed domain. A single encoder may be trained to achieve this functionality, but the different noise levels and the extended dynamic range will reduce the capability of the encoder-decoder and hurt the quality of the output images. Here we propose to use an individual encoder for each exposure and a single decoder to produce the blended HDR image. To ensure that these encoders map the frames into the same space, all the encoders and the decoder are jointly trained with respect to the losses in both pixel domain and feature domain. The loss function is defined as

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{x_0,y_0}[\mathcal{L}_{pixel}(D(E_0(x_0)), y_0)] \\ &+\mathbb{E}_{x_-,y_-}[\mathcal{L}_{pixel}(D(E_-(x_-)), y_-)] \\ &+\lambda\mathbb{E}_{x_0}[\mathcal{L}_{feature}(E_0(x_0), E_-(x_0))],\end{aligned} \tag{1}$$

where E_0 and E_- represent the encoders for the normal exposure frame and the short exposure frame respectively, and D represents the decoder. (x_0, y_0) and (x_-, y_-) represent the raw input and the corresponding RGB output pair for normal exposure and short exposure cases. Here we assume one short exposure frame, but this can be extended to more frames. Notice that a loss is defined in the compressed domain, and we find this stabilizes the training and forces the encoders to map the images into the same space. However, a small λ should be used to avoid hurting the capability of encoders.

3.2. Compressed domain MFP pipeline

Common MFP functions include demosaicing, denoising, image registration, deghosting and HDR blending. Here we explain how to achieve these functionalities to build the compressed domain MFP pipeline.

3.2.1. Demosaicing and denoising

The compressed domain is defined by training an encoder-decoder to map a raw image to an RGB image, which implicitly achieves demosaicing. Similarly, image denoising can be included by adding noise to the training data. Alternatively, a separate denoiser can be trained in the com-

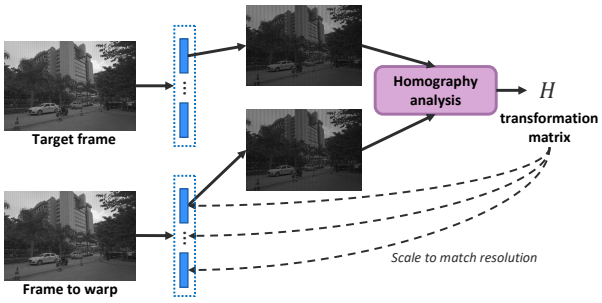


Figure 4. The proposed registration module. One of the feature map from the largest-scale feature resembles a gray-scale image, so registration warping matrix can be estimated from this feature map.

pressed domain [43]. For MFP, denoising can also be achieved by averaging multiple frames, and this still applies in the compressed domain.

3.2.2. Image registration

To analyze and process multiple frames, all the frames need to be aligned to a target frame. This compensates for the camera motion, and sometimes objects dynamics, during the capture time. In normal camera ISP, this can be achieved with image registration and warping using homography or other warping methods. In the compressed domain, we can also calculate the warping matrix for the image features and warp the features accordingly. Figure 4 visualizes one feature map from the largest-scale feature, and it is obvious that this feature map resembles a low-resolution gray-scale version of the image. Thus we can estimate the warping matrix from this feature map of the multiple frames. The warping matrix is then scaled to match the spatial resolution of the multi-scale features and applied to warp the features. This process is illustrated in Figure 4.

3.2.3. Image deghosting

To blend registered frames, an ISP algorithm needs to analyze the motion in the scene and blend only the pixels that match across frames. Here we use a network to predict the blending map in the compressed domain, and the network structure is shown in Figure 5. We used only the first three feature matrices to predict the motion map. As discussed in the previous section, feature maps with larger spatial dimension contain more detailed information that can better capture finer motions in the image. The blend map is then scaled to match the spatial resolution of the multi-scale features and used to blend the features.

3.2.4. HDR blending

To obtain an image with higher dynamic range, HDR blending blends images with different EVs so that details in the saturated region in the normal exposure frame can be recovered from short frames. This is typically done by substituting the saturated pixels in the normal exposure frame with short frames. Similar to image registration in the compressed domain, HDR blending in the compressed domain also takes advantage of the observation that the feature maps assemble low resolution images. Based on this, a threshold can be used to obtain a binary saturation map indicating the saturated region, and the map is then used to blend the features. This process is illustrated in Figure 6.

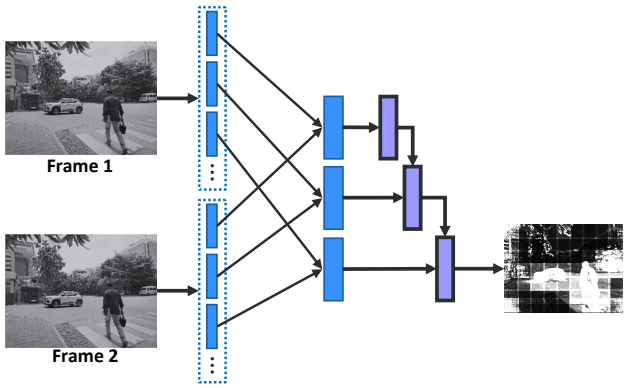


Figure 5. The proposed deghosting network. Only the first three largest-scale features that contain more details about the image are used for motion map estimation.

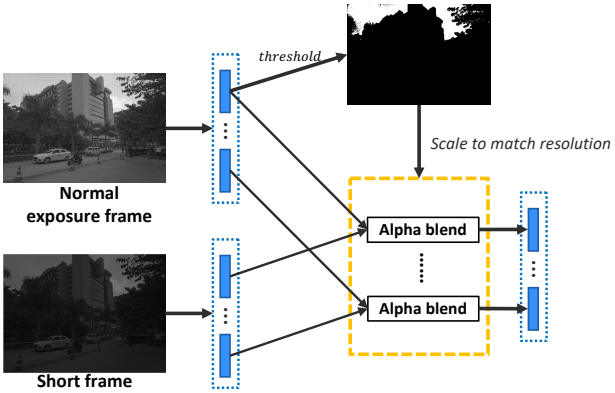


Figure 6. The proposed HDR blending module. The saturation map can be estimated from one of the image feature maps by thresholding it. The map is then scaled to match the spatial resolution of the feature for HDR blending.

3.2.5. Full pipeline

The proposed MFP pipeline is illustrated in Figure 7. Here we consider the HDR imaging with two raw frames. After the features of the two frames are computed by the two encoders, the features are first registered to the reference normal exposure frame. Then both the saturation map and the deghosting blend map are estimated from the features and used to blend the two frames.

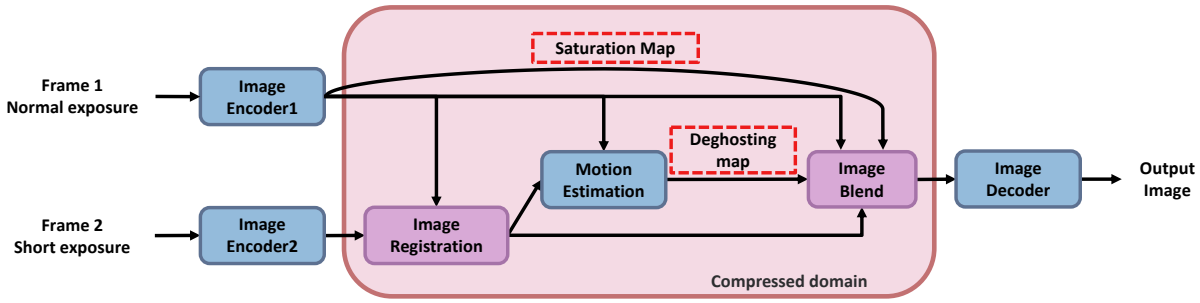


Figure 7. The proposed compressed domain MFP pipeline.

4. Experiments

4.1. Implementation Details

We conducted the experiments using a commercial smart phone camera. For each image, a normal exposure frame and a short exposure frame were captured. The image resolution was 6120×8160. For tile-based processing, the resolution of each tile was 1024×1024. Overlapping between adjacent tiles were applied to remove boundary artifacts. For training the networks, the deghosting blend maps and the groundtruth RGB images were obtained from the built-in ISP. For the compression model, SSIM and L1 loss were used in the pixel domain, and L2 loss was used in the compressed domain. For inference, the built-in tone mapping was applied for visualization.

4.2. Results

4.2.1. Compression model

First we evaluated the compression model. We captured 26 images for testing and evaluated the reconstruction quality. In average, the compression model achieved 49.31dB in PSNR and 0.9962 in SSIM. We also show visual results in Figure 8, where results with and without VQ are both included for comparison. In the first sample, a detail-rich patch is zoomed in. The compression model achieves visually lossless quality, and no visual artifact is observed. In the second example, a flat region is selected, where quantization artifacts, such as contour, tend to exist. The proposed method, however, shows no such artifacts.

To demonstrate the effectiveness of warping augmentation during training, Figure 9 shows the reconstructed RGB images after the image has been warped in the compressed domain. When the model was not trained with augmentation, warping artifacts and detail loss could be observed in the output image. To show that the loss in the feature domain is necessary in training the HDR compression model, Figure 10 shows the output HDR images. When the feature domain loss was not applied, artifacts showed up at the boundary of the saturated region and the boundary of the tile.

4.2.2. MFP functions

Image registration was evaluated by feeding the handheld captures into the pipeline, and we disabled the deghosting. The results are shown in Figure 11. When registration was not applied, the output image showed ghosting artifacts because of the hand motion, and the proposed registration module was able to detect and compensate for the hand motion.

Image deghosting was evaluated by feeding tripod captures with motion objects into the pipeline. The blended image and the motion map are shown in Figure 12. The results in the zoomed-in region shows that the proposed method can detect both large object motion (car) and small scene dynamics (tree) in the compressed domain.

The HDR blending module was evaluated with HDR scenes, and the results are shown in Figure 13. The Saturation map shows that the proposed method is able to accurately find the saturated region, even in the foliage area.

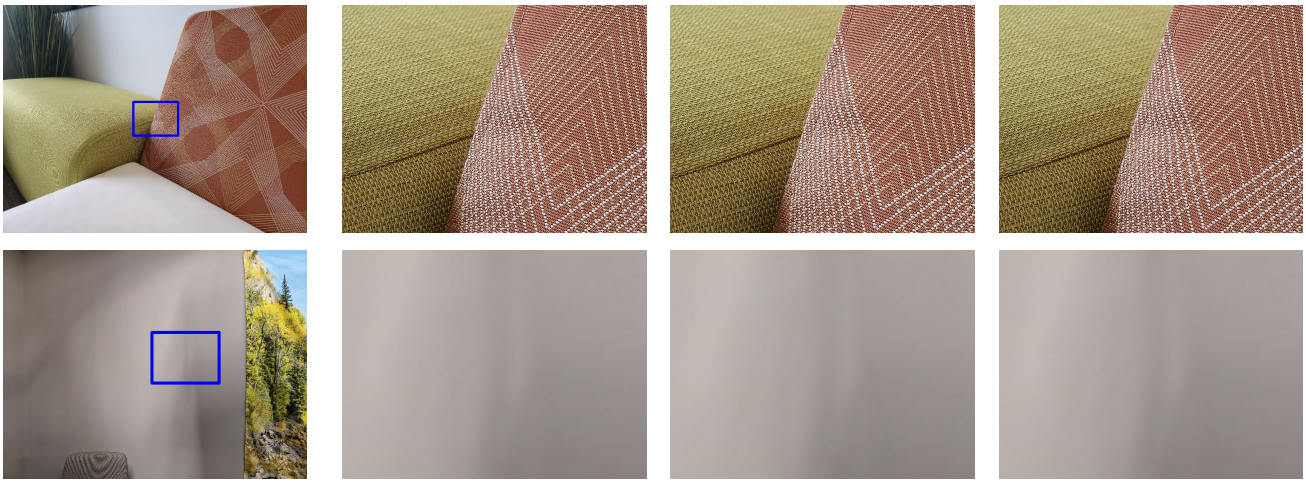
Lastly we tested the compressed domain denoising by averaging multiple frames, and the results are shown in Figure 14. By averaging more frames, we could recover more details.

4.3. Computation analysis

Compressed domain MFP not only reduces system memory required for data storage, it also reduces the computation. Table 2 compares the floating-point operation (FLOP) required for HDR blending in both the pixel domain and the compressed domain. For HDR blending, compressed domain processing saves computation by 50%.

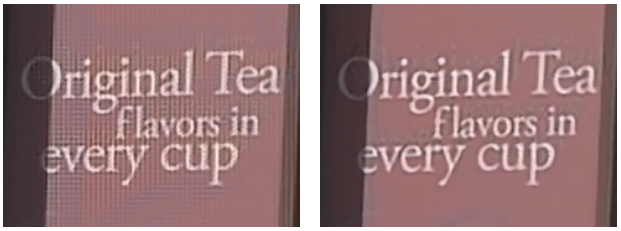
4.4. Limitations

The proposed method has its limitations. The success of image registration relies on accurate homography estimation, using features such as SIFT. However, in low light conditions or in smooth regions, feature matching may fail, resulting in inaccurate registration. When the scene is not well-lit, the deghosting module tend to treat noisy regions as dynamic regions, prohibiting the denoising through multiframe blending. That being said, the proposed method



(a) Full resolution image (b) Encoder-decoder (c) Encoder-decoder + VQ (d) Groundtruth

Figure 8. Example output images from the compression model. For each example, (b) the Encoder-decoder only, (c) the proposed method (Encoder-decoder + VQ), and (d) the groundtruth are shown for comparison.



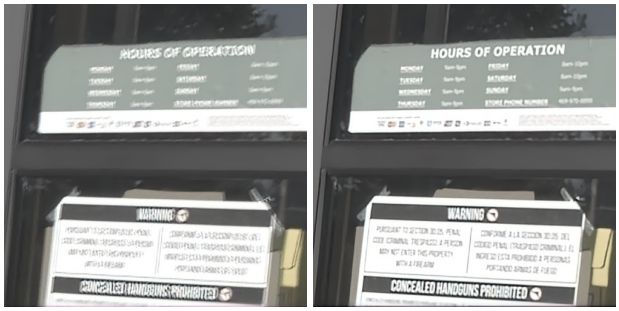
(a) Without augmentation (b) With augmentation

Figure 9. Comparison between augmentation free training and the proposed training method for image warping. (a) The compression model trained without warping augmentation. (b) The proposed training method.

(a) HDR scene (b) Pixel domain loss (c) Proposed

Figure 10. Comparison between training with only the pixel domain loss and the proposed training method for HDR blending. (a) The HDR scene. (b) The compression model trained with only pixel domain loss. (c) The compression model trained with both pixel domain loss and feature domain loss.

can be further improved with adaptive tuning regarding the noise or the lighting conditions.



(a) Without registration (b) With registration

Figure 11. Results of compressed domain registration.

Operations	Pixel domain	Compressed domain
Sat map	512×512×3	256×256
Inverse sat map	512×512×1	256×256+128×128+ 64×64+32×32+16×16+ 8×8+4×4
Sat map × frame1	512×512×1	256×256×4+128×128×8+ 64×64×16+32×32×32+ 16×16×64+8×8×128+4×4×512
Inv map × frame2	512×512×3	same as above
Blending	512×512×3	same as above
Total FLOPs	**3407872**	**1725776**

Table 2. Comparing the required number of floating-point operations (FLOP) for HDR blending in both the pixel domain and the compressed domain. The saturation map is first obtained from the data, and then the inverse map, showing pixels not saturated, is computed. The two maps are used to blend (weighted sum) the two frames.

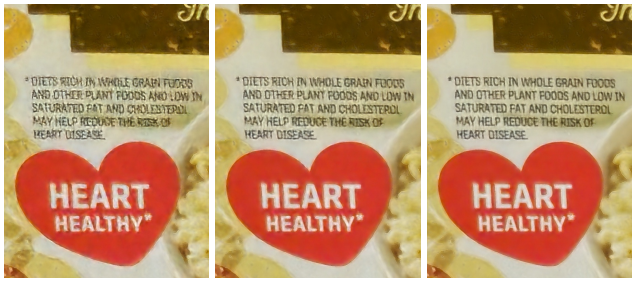
(a) Dynamic scene (b) Without deghosting (c) With deghosting

(d) Motion map (e) Without deghosting, zoomed-in view (f) With deghosting, zoomed-in view

Figure 12. Results of compressed domain deghosting. The deghosting model was able to detect both large object motion and small scene dynamics.

(a) LDR (b) HDR (c) Saturation map

Figure 13. Results of compressed domain HDR blending.



(a) Single frame (b) 2 frame blending (c) 4 frame blending

Figure 14. Results of compressed domain denoising by averaging multiple frames.

5. Conclusion

We presented a compressed domain MFP pipeline that processes and blends multiple frames in the learned compressed domain. This compressed domain achieved 3.3 times compression with respect to the input raw frame. In this compressed domain we presented MFP functions, including demosaicing, denoising, registration, deghosting and HDR blending, and we also showed that moving operations to the compressed domain reduced the computation burden. As part of future work, we plan to investigate more features in the compressed domain that enable more MFP functions. For example, how to apply non-linear operations in the compressed domain for tone-mapping, and how to add scalability to the compressed domain for resizing.

References

[1] Mahmoud Afifi and Michael S Brown. Deep white-balance editing. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 1397–1406, 2020. 3

[2] Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, and Luc Van Gool. Soft-to-hard vector quantization for end-to-end learned compression of images and neural networks. *arXiv preprint arXiv:1704.00648*, 3, 2017. 1

[3] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 221–231, 2019. 2

[4] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016. 1, 2

[5] 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. 2

[6] Fabrice Bellard. Bpg image format (2014). *URL http://bellard. org/bpg/.[Online, Accessed 2016-08-05]*, 1 (2), 2016. 1, 2

[7] David J Brady, Minghao Hu, Chengyu Wang, Xuefei Yan, Lu Fang, Yiwnheng Zhu, Yang Tan, Ming Cheng, and Zhan Ma. Smart cameras. *arXiv:2002.04705*, 2011. 1, 3

[8] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7939–7948, 2020. 2

[9] Felipe Codevilla, Jean Gabriel Simard, Ross Goroshin, and Chris Pal. Learned image compression for machine perception. *arXiv preprint arXiv:2111.02249*, 2021. 3

[10] Haisheng Fu, Feng Liang, Jianping Lin, Bing Li, Mohammad Akbari, Jie Liang, Guohe Zhang, Dong Liu, Chengjie Tu, and Jingning Han. Learned image compression with gaussian-laplacian-logistic mixture model and concatenated residual modules. *arXiv preprint arXiv:2107.06463*, 2021. 2

[11] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 3

[12] Noor Fathima Ghouse, Jens Petersen, Auke Wiggers, Tianlin Xu, and Guillaume Sautière. A residual diffusion model for high perceptual quality codec augmentation, 2023. 2

[13] Robert Gray. Vector quantization. *IEEE Assp Magazine*, 1 (2):4–29, 1984. 2

[14] Zongyu Guo, Zhizheng Zhang, Runsen Feng, and Zhibo Chen. Causal contextual prediction for learned image compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(4):2329–2341, 2021. 2

[15] Emiel Hoogeboom, Eirikur Agustsson, Fabian Mentzer, Luca Versari, George Toderici, and Lucas Theis. High-fidelity image compression with score-based generative models, 2024. 2

[16] Litao Hu, Huaijin Chen, and Jan P Allebach. Joint multi-scale tone mapping and denoising for hdr image enhancement. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 729–738, 2022. 3

[17] Khawar Islam, L Minh Dang, Sujin Lee, and Hyeonjoon Moon. Image compression with recurrent neural network and generalized divisive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1875–1879, 2021. 2

[18] Sonain Jamil, Md Jalil Piran, MuhibUr Rahman, and Oh-Jin Kwon. Learning-driven lossy image compression: A comprehensive survey. *Engineering Applications of Artificial Intelligence*, 123:106361, 2023. 2

[19] Ruolei Ji and Lina J Karam. Compressed-domain vision transformer for image classification. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2024. 3

[20] Rajandeep Kaur and Pooja Choudhary. A review of image compression techniques. *Int. J. Comput. Appl*, 142(1):8–11, 2016. 2

[21] A Burakhan Koyuncu, Han Gao, Atanas Boev, Georgii Gaikov, Elena Alshina, and Eckehard Steinbach. Contextformer: A transformer with spatio-channel attention for context modeling in learned image compression. In *European conference on computer vision*, pages 447–463. Springer, 2022. 3

[22] Fangzheng Lin, Heming Sun, Jinming Liu, and Jiro Katto. Multistage spatial context models for learned image compression. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 2

[23] Jinming Liu, Heming Sun, and Jiro Katto. Learning in compressed domain for faster machine vision tasks. In *2021 International Conference on Visual Communications and Image Processing (VCIP)*, pages 01–05. IEEE, 2021. 1, 2, 3

[24] Jinming Liu, Heming Sun, and Jiro Katto. Improving multiple machine vision tasks in the compressed domain. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 331–337. IEEE, 2022. 1, 3

[25] Jinming Liu, Heming Sun, and Jiro Katto. Learned image compression with mixed transformer-cnn architectures. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14388–14397, 2023. 3

[26] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in neural information processing systems*, 33:11913–11924, 2020. 2

[27] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31, 2018. 2

[28] Yichen Qian, Ming Lin, Xiuyu Sun, Zhiyu Tan, and Rong Jin. Entroformer: A transformer-based entropy model for learned image compression. *arXiv preprint arXiv:2202.05492*, 2022. 3

[29] Lucas Relic, Roberto Azevedo, Markus Gross, and Christopher Schroers. Lossy image compression with foundation

diffusion models. In *European Conference on Computer Vision*, pages 303–319. Springer, 2024. 2

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 2

[31] Xing Shen, Jirui Yang, Chunbo Wei, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, Xiaoliang Cheng, and Kewei Liang. Dct-mask: Discrete cosine transform mask representation for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8720–8729, 2021. 3

[32] Wuzhen Shi, Feng Jiang, Shaohui Liu, and Debin Zhao. Image compressed sensing using convolutional neural network. *IEEE Transactions on Image Processing*, 29:375–388, 2019. 1

[33] Christoph Stamm. A new progressive file format for lossy and lossless image compression. In *Proceedings of the International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision, Plzen, Czech Republic*, pages 4–8, 2002. 1, 2

[34] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017. 1

[35] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5306–5314, 2017. 2

[36] Robert Torfason, Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Towards image understanding from deep compression without decoding. *arXiv preprint arXiv:1803.06131*, 2018. 2, 3

[37] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 1

[38] Gregory K Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991. 1, 2

[39] Zhenzhen Wang, Minghai Qin, and Yen-Kuang Chen. Learning from the cnn-based compressed domain. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3582–3590, 2022. 3

[40] Maurice Weber, Cedric Renggli, Helmut Grabner, and Ce Zhang. Observer dependent lossy image compression. In *Pattern Recognition: 42nd DAGM German Conference, DAGM GCPR 2020, Tübingen, Germany, September 28–October 1, 2020, Proceedings 42*, pages 130–144. Springer, 2021. 2

[41] Kai Xu, Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, and Fengbo Ren. Learning in the frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1740–1749, 2020. 3

[42] Ruihan Yang and Stephan Mandt. Lossy image compression with conditional diffusion models. *Advances in Neural Information Processing Systems*, 36:64971–64995, 2023. 2

[43] Molin Zhang, Soumendu Majee, Chengyu Wang, Seok-Jun Lee, and Hamid Sheikh. Codisp: Exploring compressed domain camera isp with rgb-guided encoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5878–5888, 2024. 1, 2, 3, 5

[44] Yinhao Zhu, Yang Yang, and Taco Cohen. Transformer-based transform coding. In *International conference on learning representations*, 2022. 3