IMAGE COMPRESSION VIA DENSE DESCRIPTORS ASSISTED SYNTHESIS

Yuan Yuan[*], *Amin Zheng*[*], *Haitao Yang*[†] *and Oscar C. AU*[*]

[*]The Hong Kong University of Science and Technology
[†]Huawei Technologies Co., Ltd

ABSTRACT

In this paper, we propose a novel image compression approach towards visual quality rather than pixel fidelity. We intentionally remove several blocks at the encoder and reconstruct them at the decoder to get bits reduction. The removal blocks are wisely and adaptively selected based on blocks clustering, patch similarity and removal priority. A well-suited similarity measurement is defined to capture the common pattern between patches as well as tell their substitutability based on boundary consistency. To assist the removal blocks reconstruction at the decoder, we extract some dense descriptors as the side information to the decoder. Encouraging experimental results show that our compression scheme achieves up to 20.26% bits reduction with a comparable visual quality compared to the most recent standard High Efficiency Video Coding (HEVC).

Index Terms— Image compression, dense descriptors, similarity measurement, image reconstruction

1. INTRODUCTION

With the explosive growth of the Internet and social websites over the past a few years, ubiquitous images and videos available online have bring enormous pressures as well as challenges to the traditional compression system. Nowadays, many efforts have been made to explore new directions [1] of compression system towards visual quality rather than pixel fidelity which considered by the existing compression standards such as JPEG [2] and HEVC [3]. With the development of computer vision, machine learning and cloud computing, several methods and techniques such as segmentation [4], colorization [5], sparsity [6] and saliency detection [7] as well as the active learning and semi-supervised learning are utilized to achieve the bit reduction while maintaining the visual quality of images or videos at the same time.

Recently, there is emerging a new direction to solve the image compression problem with the image inpainting technique [8] [9]. The basic idea is to intentionally remove several blocks at the encoder, while recovering them at the decoder by image inpainting. Since the source image is available at the encoder, the inpainting based image compression is actually different from the traditional image inpainting problem at two aspects: 1) the removal blocks to restore at the decoder can be actively selected at the encoder; 2) various features of the removal blocks can be extracted as the side information to guide the inpainting at the decoder. However, it remains difficult issues how to select the removal blocks and which features are capable of capturing the characteristics of the blocks.

Our previous work [6] selects the removal blocks by minimizing the reconstruction error with the preserved blocks, where the cost function is more objective with pixel fidelity than with the visual quality. Dong Liu *et al.* [10] preserves the blocks locating at the ends and intersections of edges, while removing the blocks far away from edges. They extract the location of edges as the side information to the decoder. Besides the edges, other features such as histograms, sketches and epitomes derived from source image can also assist inpainting as well. But these features only provide sparse descriptors of image blocks and may also not be accurate themselves. Considering that similar patches are highly correlated and contain a lot of high order statistics, we directly utilize the similar patch as dense descriptors to depict and help to restore the removal blocks.

In this paper, we propose a novel image compression approach by intentionally removing several blocks at the encoder and reconstructing them at the decoder. The removal blocks are wisely and adaptively selected based on both patch similarity and feature analysis. To guide the removal blocks reconstruction at the decoder, some side information are extracted at the encoder. The experiments demonstrate that our approach successfully reduces the bit rate up to 20.26% while maintains a comparable visual quality compared to HEVC.

The remainder of this paper is organized as follows. Section 2 is the overview of our proposed image compression method. Section 3 describes the strategies of removal blocks selection and side information extraction. Section 4 describes the image restoration method. Section 5 and 6 provides experimental results and conclusion respectively.

2. THE PROPOSED IMAGE CODING APPROACH

This section gives a brief overview of our proposed image compression algorithm. As illustrated in Fig. 1, the original image is divided into the removal blocks and the preserved blocks. In our approach, the preserved blocks are encoded

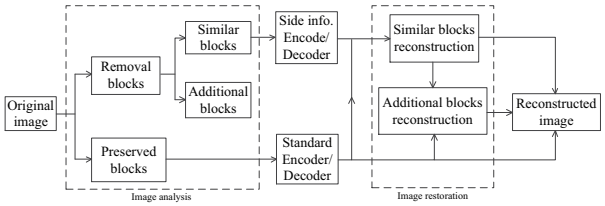

Fig. 1. The flowchat of proposed image compression scheme.

and decoded with the standard coding method, such as JPEG and HEVC. By analyzing the original image, we further category the removal blocks into two classes:similar blocks and additional blocks. For the similar blocks, we extract some dense descriptors as side information to the decoder. As the side information captures statistic characteristics of the similar blocks, it assists the restoration of them from the decoded preserved blocks. For the additional blocks, we just remove and infer them from the decoded preserved blocks and similar blocks at the decoder.

3. IMAGE ANALYSIS AND BLOCKS REMOVAL

In this section, we analyze the original image to extract the side information and obtain the removal regions, i.e., the similar blocks and the additional blocks respectively. In typical inpainting scenarios, the restoration of unknown region is usually an ill-posed problem because information in missing region is totally unknown, especially for the unknown region containing rich structures. Taking Fig. 2 as an example, the area to be restored consists of two homogenous regions divided by an edge denoted by the solid curve. The dashed curve is the inferred edge with diffusion based inpainting, which is quite different from the actual one. Besides, the human eyes are more sensitive to the structure difference than the texture difference. Therefore, we prefer to keep the structure of the image by considering its difficulty to recover and importance for human visual system.

3.1. Feature based image blocks classification

At the encoder side, we firstly classify the original image blocks into two categories: structure blocks and texture blocks. The structure block usually contains features like edges and corners. An edge depicts the boundary of two objects. A corner can be defined as the intersection of two edges. A corner can also be defined as a point for which there are two dominant and different edge directions in a local neighborhood of the point. A block contains edge and corner also includes the transition of two or more objects. Here, we apply the canny edge detection [11] and harris corner detection [12] algorithm to the Y, U, V components of the original image separately to extract the structure features. The corresponding binary maps are denoted as E_Y, E_U, E_V and C_Y, C_U, C_V. For each block B_i, we calculate the numbers of edge pixels $N_e(B_i)$

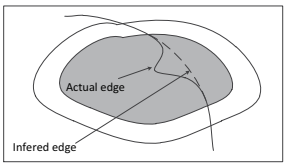

Fig. 2. The edge inferred by inpainting.

and corner pixels $N_c(B_i)$ in it, which are

$$N_e(B_i) = \#\{p|p \in B_i, E_Y(p) = 1||E_U(p) = 1||E_V(p) = 1\}, \quad (1)$$

$$N_c(B_i) = \#\{p|p \in B_i, C_Y(p) = 1||C_U(p) = 1||C_V(p) = 1\}. \quad (2)$$

If the number of edge pixels or corner pixels is greater than a threshold, then the block is a structure block. The label of each block is determined by the discriminant equation:

$$label(B_i) = \begin{cases} structure, & \text{if } N_e(B_i) \geq t_1 \text{ or } N_c(B_i) \geq t_2 \\ texture, & \text{otherwise} \end{cases} \quad (3)$$

3.2. Similar block and patch pairs selection

Although we wish to preserve the structure regions after we divide the image into structure and texture two parts, there are some redundancies that we can further exploit in the structure regions. As the existing of spatial redundancy in the image, it is possible for several blocks highly similar to some patches. In order to efficiently exploit the spatial redundancy, we can remove these blocks while maintaining their corresponding similar patches to assist the restoration. At the decoder, we wish to recover the removal regions such that it looks just like the original one and is consistent with the preserved regions at the boundary. For selecting the removal blocks, a well-suited similarity measurement is necessary to capture the common pattern between patches as well as tell their substitutability given the same neighbors or boundary.

We incorporate color information consistency, pattern consistency, boundary consistency into our algorithm. In this paper, all the blocks and patches are with the same size, which is always square and fixed. Instead of using SSD, we define the distance between block B_p and patch Ψ_q as follows:

$$d(B_i, \Psi_j) = \lambda_1 D_c(B_i, \Psi_j) + \lambda_2 D_p(B_i, \Psi_j) + \lambda_3 D_b(B_i, \Psi_j), \quad (4)$$

where λ_1, λ_2, λ_3 are positive weighting factors and $\lambda_1 + \lambda_2 + \lambda_3 = 1$. $D_c(B_i, \Psi_j)$ is actually the SSD which reflects the difference of color information,

$$D_c(B_i, \Psi_j) = \sum_{\mathbf{x} \in \Omega} |f_{B_i}(\mathbf{x}) - f_{\Psi_j}(\mathbf{x})|^2, \quad (5)$$

where Ω is a square window of the block size in the image, and $f_{B_i}(\mathbf{x})$ and $f_{\Psi_j}(\mathbf{x})$ is the intensity value of the pixel in the region B_i and Ψ_j at the corresponding location of $\mathbf{x} \in \Omega$.

$D_p(B_i, \Psi_j)$ is the difference between the gradients of patches which reflects the pattern consistency.

$$D_p(B_i, \Psi_j) = \sum_{\mathbf{x}\in\Omega} | \bigtriangledown f_{B_i}(\mathbf{x}) - \bigtriangledown f_{\Psi_j}(\mathbf{x})|^2, \quad (6)$$

where $\bigtriangledown \cdot = [\frac{\partial \cdot}{\partial x}, \frac{\partial \cdot}{\partial y}]^T$ is the gradient operator. $D_b(B_i, \Psi_j)$ is the distance to measure the continuity of the isophotes at the boundaries. We replace B_i by Ψ_j and calculate the average changes of the Laplacian estimator along the tangent direction [13] at the interior boundary, as shown in (7).

$$D_b(B_i, \Psi_j) = \sum_{\mathbf{x}\in\partial\_\Omega} |\frac{\bigtriangledown(\triangle f(\mathbf{x}))}{|\bigtriangledown(\triangle f(\mathbf{x}))|} \cdot \frac{\bigtriangledown^{\perp} f(\mathbf{x})}{|\bigtriangledown^{\perp} f(\mathbf{x})|}| \times |\bigtriangledown f(\mathbf{x})| \quad (7)$$

where ∂_Ω is the interior boundary of Ω. $\frac{\bigtriangledown(\triangle f(\mathbf{x}))}{|\bigtriangledown(\triangle f(\mathbf{x}))|}$ stands for the normalized gradient of the Laplacian estimator, which depicts the changes of the Laplacian estimator, and $\frac{\bigtriangledown^{\perp} f(\mathbf{x})}{|\bigtriangledown^{\perp} f(\mathbf{x})|}$ is the normalized vector along the tangent direction. The inner product measures the change of the Laplactian estimator along the tangent direction. $|\bigtriangledown f(\mathbf{x})|$ is an edge adaptive weight to give a larger penalty for the intensive edge and a smaller penalty for the weak edge. If the edge is intensive, the changes along the tangent direction should be small. If the edge is weak, the changes along the tangent direction is allowed to be large. In other words, the boundary of B_i and Ψ_j should be consistency especially at the intensive edges. The similarity measurement is thus defined as follows:

$$s(B_i, \Psi_j) = e^{\frac{-d(B_i,\Psi_j)}{2\sigma_1^2}}. \quad (8)$$

For each block B_i in the structure regions I_s, we search for its most similar patch Ψ_{i^*} in the structure region without intersection with the block itself as $\Psi_{i^*} = \arg\max_{\Psi_j} s(B_i, \Psi_j)$, such that $\Psi_j \subset I_s \; and \; B_i \cap \Psi_j = \emptyset$. We select the block-patch pairs with similarity $s(B_i, \Psi_{i^*})$ greater than a threshold Th. The block in a pair is called a similar block and we extract the displacement vector of its corresponding most similar patch as the side information. Similarly, we can also exploit the redundancies in texture regions by finding similar blocks and their corresponding patches.

3.3. Additional blocks selection

Besides the redundancy among similar blocks, it still exists lots of redundancy among the remaining texture blocks which means we can further remove some texture blocks. We define a removal priority term P for each remaining block. The block with highest removal priority will be selected as an additional block. We consider two factors into the removal priority term in (9): the existing status of the block neighbours, the texture complexity of the block.

$$P(B_i) = (\sum_{B_k\in N_4(B_i)} map(B_k) + c) \cdot \exp(-\frac{var(B_i)}{2\sigma_2^2}) \quad (9)$$

where $N_4(B_i)$ is the four neighborhood of block B_i, and $map(B_k)$ depicts the weight that the existing status of a block providing to its neighbouring block's removal priority, which is defined as (10) for the different status of a block. If a neighbouring block of B_i is missing, then the removal priority of B_i become lower for preventing the removal region merge into a large hole, and vice versa. c is a bias to make sure the removal priority positive.

$$map(B_k) = \begin{cases} 1 & , if\ B_k\ is\ a\ preserved\ block \\ -3 & , if\ B_k\ is\ an\ additional\ block \\ s(B_k, \Psi_{k^*}) & , if\ B_k\ is\ a\ similar\ block \\ 0 & , otherwise \end{cases} \quad (10)$$

We utilize variance to measure the texture complexity of a block, as the second term in (9). Iteratively select the block with highest removal priority term and update the priority scores for the remaining blocks until a pre-set additional removal rate r is given. We call all the selected blocks as the additional removal blocks. They will be skipped at the encoder. Besides the similar blocks and additional blocks, the remaining blocks are preserved and encoded by standard coding method.

4. IMAGE RECONSTRUCTION

4.1. Similar blocks reconstruction with refinement

At the decoder, after we decode the preserved blocks and the side information, which are the displacement vectors indicating the location of the corresponding most similar patches for the similar blocks. A straightforward method to reconstruct the removed similar blocks is to directly copy the pixels from their corresponding patches. Because of the absence of the motion compensated residue, the visible blocking artifacts will exist in the reconstructed image. To get a consistent visual quality, we refine the reconstruction of the removed similar blocks using partial differential equation and formulate the problem as an optimization problem.

$$\begin{aligned} \min_{f(\mathbf{x})} \quad & \textstyle\sum_{\mathbf{x}\in\Omega} | \bigtriangledown f(\mathbf{x}) - \bigtriangledown g(\mathbf{x})|^2 \\ \text{s.t.} \quad & \forall \mathbf{x} \in \partial\Omega, f(\mathbf{x}) = f^*(\mathbf{x}) \end{aligned} \quad (11)$$

where $\bigtriangledown g(\mathbf{x})$ is the gradient of the corresponding most similar patch whose location is indicated by the side information, $f^*(\mathbf{x})$ is the known decoded pixel values. The cost function means that we wish the gradients of the unknown region similar to its corresponding most similar patch. The constraint forces the pixel values at the boundary of the removal similar block the same with its neighboring known pixel values. We solve this problem with Euler-Lagrange equation.

4.2. Additional blocks restoration with global inpainting

We use a global inpainting [14] method to reconstruct the additional removal blocks. The unknown regions are updated

Table 1. Bit-saving compared to HEVC intra coding (QP=22)

Test Sequence	*Overhead*		*Bit-rate(bpp)*		*Bits saving*	
	Side info.	*Blocks ind.*	*Proposed*	*HEVC*	*Raw*	*Pure*
PeopleOnStreet	2.29%	0.45%	0.7875	0.8671	11.91%	**9.17%**
Kimono	5.53%	0.92%	0.3414	0.4282	26.71%	**20.26%**
BasketballDrive	5.49%	0.80%	0.4214	0.4941	21.00%	**14.71%**
BasketballDrill	1.26%	0.38%	0.8673	1.0216	16.74%	**15.10%**

pixel by pixel. Each pixel in the unknown regions is contained by several patches. For each patch, we choose a candidate value from its most similar patch at the corresponding current unknown pixel location. Then the current unknown pixel is updated by the weighted average of the several candidate values. The pixel values are iteratively updated until they converge or a pre-set iterative time reaches.

5. EXPERIMENTAL RESULTS

The proposed method can be applied to any block-based video compression scheme. In this paper, we implement the proposed algorithm in HEVC reference software HM12.0 [15]. The side information and removal blocks indicators are encoded into the bitstream using arithmetic coding. The performance of several standard video test sequences with different resolution are evaluated using intra main configuration. Specifically, we set the size of largest coding unit(LCU) to 16×16. Four quantization parameters are tested:22, 27, 32, 37. Three threshold Th: 0.3, 0.5, 0.7 and three additional removal rate r: 0.05, 0.1, 0.2 are tested for each sequence. In all the following experiments, the parameters t_1, t_2, λ_1, λ_2, λ_3, σ_1, σ_2 and c are set to 20, 1, 0.6, 0.2, 0.2, 2, 4 and 13 respectively.

Table. 1 lists the bits reduction results of the first frame of four test sequences with $QP = 22$, $Th = 0.3$ and $r = 0.05$, compared to HM12.0. It shows that the proposed method saves considerable bits for all test images which is up to 20.26%. Fig. 3 shows the visual quality of the same images and same parameter settings as that in Table. 1. The left column is the incomplete images with region removal, where the removal regions are indicated by the green mask; the middle column are the decoded images by the proposed method; and the right column are the decoded images by HEVC intra coding. It is shown that the decoded images by the proposed method and that by HEVC intra coding have comparable visual quality. Combining the bits reduction results in Table. 1 and the visual quality results in Fig. 3, it is concluded that the proposed method achieves great bits reduction with a comparable visual quality compared to HEVC.

Table. 1 also lists the overhead consumption of the proposed method, i.e., the bits for side information and removal blocks index. We can see that the bits consumed by side information is much more than the removal blocks index. When Th and r are fixed for an image, the overhead are fixed. An-

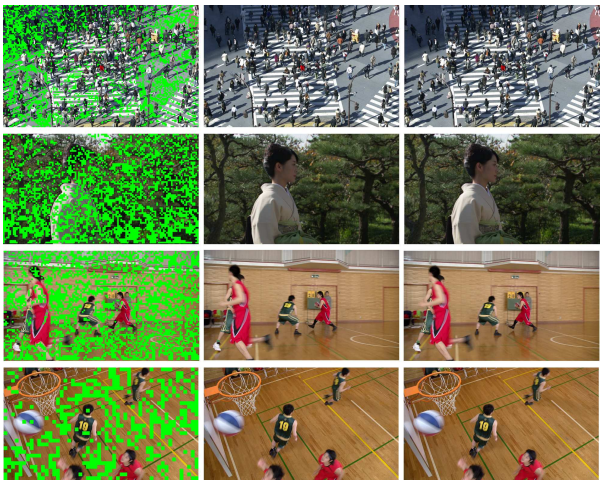

Fig. 3. Visual quality comparisons between the proposed scheme and HEVC. Image with removal regions(left), decoded images by the proposed method (middle) and by HEVC (right). From top to bottom: *PeopleOnStreet*, *Kimono*, *BasketballDrive*, *BasketballDrill*.

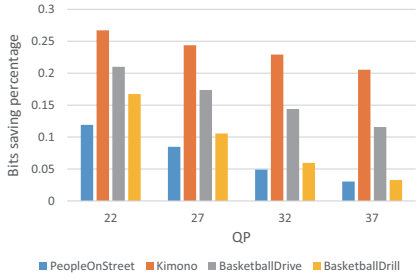

Fig. 4. Raw bits saving for different QP.

alyzing the experimental results of Fig. 4, we can see: (1) When QP increases, the pure bits reduction will decreases. Firstly, the residue consumes fewer bits in HEVC when QP increases, thus removing the residue will save fewer bits. Moreover, the percentage of the overhead bits increases as the total overhead bits stay the same with different QP. (2) If the image contains rich smooth regions, the bits reduction is smaller, such as *PeopleOnStreet*. Because smooth regions are efficiently compressed by HEVC. While for the image containing rich similar or repeated texture regions, the bits reduction is larger. Because these blocks are difficult to compress by HEVC but removed in the proposed encoder and well inferred at the decoder.

6. CONCLUSION

In this paper, an image compression scheme based on several blocks removal at the encoder and reconstruction at the decoder with side information is introduced. Experiments show that our proposed scheme achieves up to 20.26% bit rate reduction with a comparable visual quality compared to HEVC.

7. REFERENCES

[1] M.M. Reid, R.J. Millar, and N.D. Black, "Second-generation image coding: an overview," in *ACM Computing Surveys*, Mar. 1997, vol. 29, pp. 3–29.

[2] G.K. Wallace, "The jpeg still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.

[3] G.J. Sullivan, J. Ohm, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[4] M. Bosch, F. Zhu, and E. Delp, "Segmentation-based video compression using texture and motion models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1366–1377, 2011.

[5] C. Zhang and X. He, "Image compression by learning to minimize the total error," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 4, pp. 565–576, 2013.

[6] Y. Yuan, O. Au, A. Zheng, H. Yang, K. Tang, and W. Sun, "Image compression via sparse reconstruction," in *IEEE International Conference on Acoustic, Speech and Signal Processing*, 2014, pp. 2025–2029.

[7] C. Guo and L. Zhang, "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 185–198, 2010.

[8] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[9] N. Komodakis and G. Tziritas, "Image completion using efficient belief propagation via priority scheduling and dynamic pruning," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2649–2661, 2007.

[10] D. Liu, X.Y. Sun, F. Wu, S.P. Li, and Y.Q. Zhang, "Image compression with edge-based inpainting," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 10, pp. 1273–1287, 2007.

[11] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[12] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, 1988, pp. 147–151.

[13] Y. Chen, Y. Hu, O. Au, H. Li, and C. Chen, "Video error concealment using spatio-temporal boundary matching and partial differential equation," *IEEE Transactions on Multimedia*, vol. 10, no. 1, pp. 2–15, 2008.

[14] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 1–14, 2007.

[15] "HEVC Test Model," `https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/`.