Correspondence

Block-Based Image Compression With Parameter-Assistant Inpainting

Zhiwei Xiong, Xiaoyan Sun, *Member, IEEE*, and Feng Wu, *Senior Member, IEEE*

***Abstract*—This correspondence presents an image compression approach that integrates our proposed parameter-assistant inpainting (PAI) to exploit visual redundancy in color images. In this scheme, we study different distributions of image regions and represent them with a model class. Based on that, an input image at the encoder side is divided into featured and non-featured regions at block level. The featured blocks fitting the predefined model class are coded by a few parameters, whereas the non-featured blocks are coded traditionally. At the decoder side, the featured regions are restored through PAI relying on both delivered parameters and surrounding information. Experimental results show that our method outperforms JPEG in featured regions by an average bit-rate saving of 76% at similar perceptual quality levels.**

***Index Terms*—Assistant parameter, image compression, image inpainting, model class, perceptual quality.**

I. Introduction

Over the past two decades, image and video compression has seen rapid development in both academia and industry. Current state-of-the-art JPEG2000 and MPEG-4 AVC/H.264 are two examples that greatly outperform previous generations in terms of coding efficiency. However, these transform-based schemes that uniformly compress images consider little of the feature variations in image regions. In addition, perceptual quality is seldom considered as a key factor when these schemes are designed and developed, although some visual tests are performed.

Attempts have been made to develop new compression techniques by utilizing different features within images to achieve high coding performance [1]. Especially, recent vision-related technologies have shown remarkable progress in hallucinating pictures with good perceptual quality. Among them, image inpainting is a very promising approach to be utilized in image compression, aiming at perceptual quality instead of pixel-wise fidelity. Inpainting was first introduced into digital image processing by Bertalmio *et al.* [2], as a process of restoring missing data in a designated region of an image in a visually plausible way. Subsequently, several mathematical models as well as various applications of inpainting have been presented [3]–[5]. Current results of image inpainting [6], [7] demonstrate that it can naturally recover homogenous regions as well as certain kinds of structural regions.

One straightforward way to employ inpainting in compression is to directly drop some regions in encoding but fill them up in decoding by inpainting, as proposed in [8]. However, we notice that it is hard for current inpainting schemes to restore large dropped regions, since the correlations between available pixels and the interior pixels of dropped regions are quite weak due to long distance. Moreover, current inpainting techniques implicitly employ a predefined image prior (model) to derive missing information from known surroundings, whereas different image regions may have different distributions. If the imposed prior does not match the property of dropped regions, the restoration could result in severe visual artifacts.

Facing these problems, edge-based inpainting techniques are proposed and applied into image compression [9], [10], where, instead of completely dropping all information, certain edge information is extracted from those dropped regions and transmitted in a compressed fashion to the decoder to guide the inpainting process. Due to the edge-based inpainting, more edge regions can be dropped and recovered well to achieve a higher compression ratio at similar perceptual quality levels. However, as inpainting is involved in image compression, it is desirable to drop as many regions as possible to save coding bits while still maintaining good visual quality.

In this correspondence, we propose a generic method of parameter-assistant inpainting (PAI) for the purpose of compression. We study different distributions of image regions and fit them with a model class rather than a single prior. In other words, we intend to view an image as a class of inhomogenous distributions. Moreover, we consider the distribution parameters as a kind of unpredictable information. Therefore, *assistant parameters*, usually a sparse representation of the distribution parameters, are introduced as a supplement for more reliable inpainting when large image regions are dropped.

We further design an image compression system that integrates PAI. According to the proposed model class, an input image at the encoder side is analyzed at block level and then divided into featured (including gradated and structural) and non-featured blocks. For the majority of featured blocks, assistant parameters instead of pixel values are coded and transmitted. A few featured blocks (selected as exemplars) as well as the non-featured blocks, are coded using traditional DCT-based methods. At the decoder side, dropped regions are recovered by PAI, relying on both delivered parameters and reserved regions. Due to the assistant parameters, the reliability of inpainting is greatly enhanced. Since large regions can be dropped during encoding, the proposed scheme achieves a higher coding performance when compared with traditional image compression methods at similar perceptual quality levels.

The rest of this correspondence is organized as follows. Section II gives a generic description of PAI for compression. Its adaptations for two kinds of image regions are further discussed in Section III. Section IV presents the framework and some details of our compression system. In Section V, experimental results are shown in terms of perceptual quality and bit-rate saving, compared with baseline JPEG and H.264-Intra. Concluding remarks are offered in Section VI.

II. Parameter-Assistant Inpainting for Compression

As shown in Fig. 1, let Ω denote a complete image domain, I is a subset of Ω consisting of unknown pixels, and Γ is the boundary of I.

Manuscript received May 17, 2008; revised January 19, 2010. First published March 08, 2010; current version published May 14, 2010. A preliminary version of this correspondence was printed in the International Conference on Image Processing, Vol. 2, pp. 369–372, 2007. This work was done during Z. Xiong's internship at Microsoft Research Asia. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Srdjan Stankovic.

Z. Xiong was with Microsoft Research Asia, Beijing 100081, China. He is now with the University of Science and Technology of China, Hefei 230027, China (e-mail: xzw@mail.ustc.edu.cn).

X. Sun and F. Wu are with Microsoft Research Asia, Beijing 100081, China (e-mail: xysun@microsoft.com; fengwu@microsoft.com).

Color versions of one or more of the figures in this correspondence are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2010.2044960

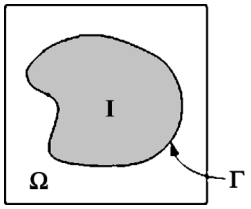

Fig. 1. Image inpainting problem.

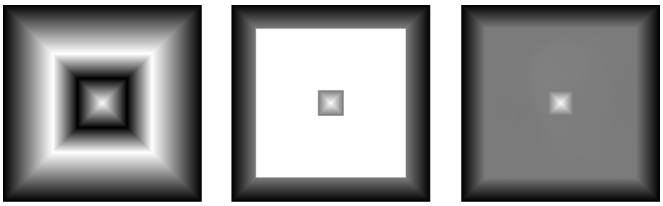

Fig. 2. Inpainting with a predefined prior. (a) original image, (b) incomplete image (white region dropped), and (c) PDE-based inpainting result [6].

Inpainting is the technique that aims to recover a whole image u on the entire domain Ω only from the partial observation $u_0|_{\Omega\setminus \mathrm{I}}$. To accomplish this task, various image priors are employed in the literature, most of which impose certain distribution consistency on the missing region and its known surroundings. For example, the well-known total variation (TV) inpainting based on bounded variation (BV) image prior [3] minimizes the following energy item:

$$E(u) = \int_{\Omega} |\nabla u|. \tag{2-1}$$

Inpainting based on a predefined prior gives good results when missing regions are small and consistent with the distribution assumption of the prior. However, an inpainting-based compression system requires as many regions as possible to be intentionally dropped for coding efficiency. Since the dropped region may have a different distribution from its known surroundings, the principle of minimizing a total variation may result in severe distortion. Fig. 2 shows a simple example in which inpainting with a PDE prior (a kind of TV inpainting) [6] fails to deduce the correct gradation pattern in the dropped region. Actually, we believe such inpainting problems are too difficult to be solved without any other assistance.

Therefore, to develop an effective inpainting method in the compression scenario, we suggest that image regions to be recovered are restricted to a class of parametric distributions rather than a fixed prior. We denote

$$\mathbf{M} = \left\{ P_k(u|\theta) : \theta \in R^{d_k}, k = 1, \ldots, K \right\} \tag{2-2}$$

where $\mathbf{M}$ is referred to as a *model class* [11], P_k represents a featured distribution with d_k-dimensional parameter θ, and K is the total number of adopted distributions.

In case of compression, when the distribution forms in $\mathbf{M}$ are given, the proper model k is first determined for each image region. Then, before a certain image region I is dropped at the encode side, an estimate of θ is involved as

$$\hat{\theta} = F(u|_{\mathrm{I}}, P_k). \tag{2-3}$$

F is an estimation function based on the image observations in the dropped region as well as the distribution it belongs to. With the help of this assistant parameter $\hat{\theta}$, a valid inpainting can be performed at the decode side by

$$\hat{u}|_{\mathrm{I}} = G(\hat{u}_0|_{\Omega\setminus \mathrm{I}}, P_k, \hat{\theta}) \tag{2-4}$$

where $\hat{u}$ is the restored image content on the dropped region, $\hat{u}_0$ is the known surrounding compressed traditionally, and the PAI function G is designed in accordance with F.

Generally, the performance of a compression system is evaluated by its rate-distortion (R-D) characteristic. For the PAI-based compression method, R-D on the dropped region can be depicted as

$$C = R(\hat{\theta}) + \lambda D(u|_{\mathrm{I}}, \hat{u}|_{\mathrm{I}}) \tag{2-5}$$

where C is regarded as the cost of compression, λ is a weighting factor, the rate R depends on the coding of assistant parameters and the distortion D is evaluated with certain perceptual quality metrics between original and restored image regions.

It can be observed that our PAI-based compression is compliant with traditional schemes in terms of R-D optimization and can be readily integrated with other blocked-based methods, e.g., JPEG and H.264-Intra. The whole compression system is designed to minimize the total cost $\mathbf{C}_{\mathrm{M}}$ for the entire image

$$\mathbf{C}_{\mathrm{M}} = \sum_{k=0}^{K} C_k|_{\mathrm{I}_k} = \sum_{k=0}^{K} \left(R_k|_{\mathrm{I}_k} + \lambda D_k|_{\mathrm{I}_k} \right) \tag{2-6}$$

where C_k, R_k, and D_k $(k = 1, \ldots, K)$ are the cost, rate and distortion of PAI for dropped regions I_k in accordance with each featured distribution P_k in $\mathbf{M}$, and C_0, R_0 and D_0 are the corresponding items for the reserved regions I_0 coded traditionally.

Due to the diversity of nature images, PAI with a model class can be more efficient in representing image regions with different distributions compared with traditional methods that uniformly compress the entire image. The total cost in the latter case will be

$$\mathbf{C}_{\mathrm{S}} = \sum_{k=0}^{K} C_0|_{\mathrm{I}_k} = \sum_{k=0}^{K} \left(R_0|_{\mathrm{I}_k} + \lambda D_0|_{\mathrm{I}_k} \right) \geq \mathbf{C}_{\mathrm{M}}. \tag{2-7}$$

The previous condition always holds as long as we set

$$C_k = \min \left(R_k|_{\mathrm{I}_k} + \lambda D_k|_{\mathrm{I}_k}, R_0|_{\mathrm{I}_k} + \lambda D_0|_{\mathrm{I}_k} \right), \quad k = 1, \ldots, K. \tag{2-8}$$

III. PAI for Different Image Regions

In this section, we present the PAI adaptations for different image regions. Two featured distributions, gradation P_G and structure P_S, are taken for detailed illustration. In the following discussion, we have $\mathbf{M} = \{P_G, P_S\}$.

A. Region Classification

To best represent the target region for inpainting, an effective region classifier is required. A generic solution is to study the distribution of each image region and find the best match among the candidates in the predefined model class. For simplicity, in our scheme, an entire image is divided into three categories: gradated, structural, and non-featured, at non-overlapping block level of size $S \times S$.

The classification is performed based on edge content and color variance in each block. First, simple edge detection is carried out and blocks

with edge pixels inside are judged into the structural category. Then, color variance V is calculated in the remaining blocks

$$V = \sum_{y=1}^{S}\sum_{x=1}^{S}\left[(R_{xy}-\overline{R})^2 + (G_{xy}-\overline{G})^2 + (B_{xy}-\overline{B})^2\right]$$
$$\overline{R} = \frac{1}{S^2}\sum_{y=1}^{S}\sum_{x=1}^{S} R_{xy}$$
$$\overline{G} = \frac{1}{S^2}\sum_{y=1}^{S}\sum_{x=1}^{S} G_{xy},$$
$$\overline{B} = \frac{1}{S^2}\sum_{y=1}^{S}\sum_{x=1}^{S} B_{xy} \tag{3-1}$$

where R, G, B represent the color components, $\overline{R}, \overline{G}, \overline{B}$ are their mean values in a block, and x, y are the pixel indices. Blocks with a color variance less than a threshold V_T are treated as gradated. Finally, the remaining blocks are considered to be non-featured.

B. Gradated Region

For gradated image regions where the hue or lightness changes smoothly, the most important feature is the gradation pattern shaped by the inside gradients. The parametric distribution on a gradated region is thus given as

$$P_G(u|\mathbf{g}) : \mathbf{g} = \nabla u = (g_x, g_y) \in R^2. \tag{3-2}$$

The gradient $\mathbf{g}$ in discrete form can be denoted in a piecewise smooth form

$$\mathbf{g} = \sum_{l=1}^{L}(g_{lx}, g_{ly})\delta(\Omega_l), \quad \Omega_1 \cup \cdots \cup \Omega_L = \Omega$$
$$\delta(\Omega_l) = \begin{cases} 1, & (x,y) \in \Omega_l \\ 0, & \text{else} \end{cases} \quad l = 1, \ldots, L \tag{3-3}$$

where $(\Omega_1, \ldots, \Omega_L)$ is a partition of the entire image domain Ω and L is the number of subareas. Notice that the estimation of the gradation pattern is closely related to the partition method. We may estimate $\mathbf{g}$ at pixel level in pursuit of accuracy. However, it is not favorable in compression as the parameters need to be coded at pixel level as well. Since the gradation pattern always exists steadily and varies gently (otherwise edge or texture will appear), a coarse level description should be adequate. Also, pixel values in digital images are all integers, so a gradation pattern actually comprises a series of step-like segments. It is difficult to tell the exact gradation pattern only from gradients at individual pixels. Considering the above reasons, we estimate a *block gradient* as the assistant parameter to represent the gradation pattern.

The block gradient estimation is illustrated in Fig. 3. Here $U_{xy} (1 \le x, y \le S)$ denotes the image observations in a block. First, prefiltering is carried out to achieve the vertical gradation array $\mathbf{V}$ and the horizontal gradation array $\mathbf{H}$ that consists of the mean value of each row and column, respectively

$$\mathbf{V} = (U_{1,\mathbf{V}}, \cdots, U_{S,\mathbf{V}}), \quad U_{i,\mathbf{V}} = \frac{1}{S}\sum_{x=1}^{S} U_{x,i},$$
$$\mathbf{H} = (U_{1,\mathbf{H}}, \cdots, U_{S,\mathbf{H}}), \quad U_{i,\mathbf{H}} = \frac{1}{S}\sum_{y=1}^{S} U_{i,y}, \quad i = 1, \ldots, S. \tag{3-4}$$

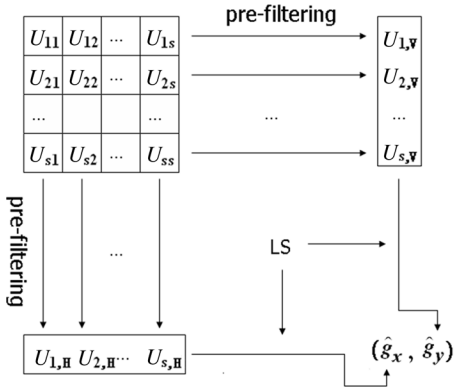

Fig. 3. Block gradient estimation.

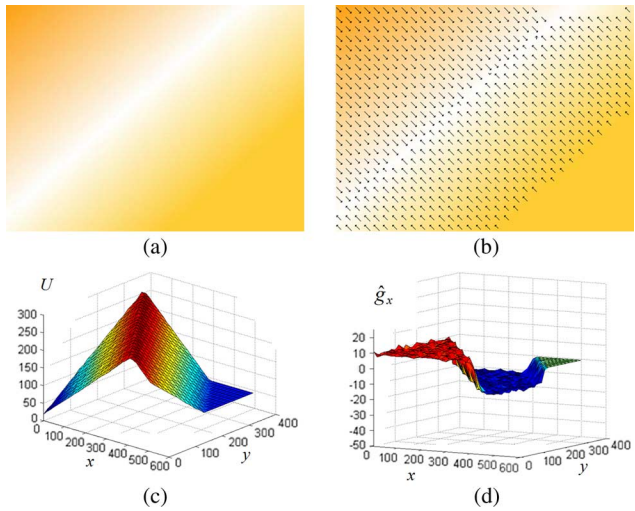

Fig. 4. Block gradients in a gradated image region. (a) Gradated image region; (b) block gradients; (c) observation in image space; (d) observation in gradient space.

Then we use the least square (LS) fitting to approximate the block gradient as

$$\hat{g}_y = \sum_{i=1}^{S}\left(i - \frac{S+1}{2}\right)(U_{i,\mathbf{V}} - \overline{U}_{\mathbf{V}}) \Big/ \sum_{i=1}^{S}\left(i - \frac{S+1}{2}\right)^2,$$
$$\hat{g}_x = \sum_{i=1}^{S}\left(i - \frac{S+1}{2}\right)(U_{i,\mathbf{H}} - \overline{U}_{\mathbf{H}}) \Big/ \sum_{i=1}^{S}\left(i - \frac{S+1}{2}\right)^2 \tag{3-5}$$

where $\overline{U}_{\mathbf{V}}$ and $\overline{U}_{\mathbf{H}}$ are the mean values of $\mathbf{V}$ and $\mathbf{H}$.

Fig. 4 illustrates the block gradients in a gradated image region. For the input image region (a), the block gradients (take the blue color component for example) calculated via (3-4) and (3-5) are shown in (b) by directional arrows. Observations explored in the image and gradient spaces are presented in (c) and (d). The spatial correlation in the gradient space, as can be seen, is greatly enhanced (assembled to three levels) compared with that in the image space. Since strong correlation is favorable to compression, it is promising to describe the gradated regions more efficiently by block gradients.

Given the gradient information, our PAI completes a dropped gradated region from known boundaries inwards block by block, whereas the inpainting within a block is performed at pixel level to maintain the continuity of gradation. Specifically, blocks with maximum known neighbors in their four-neighborhood are recovered first.

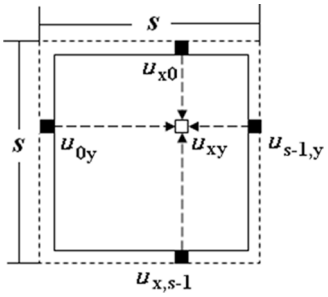

Fig. 5. PAI in a gradated block.

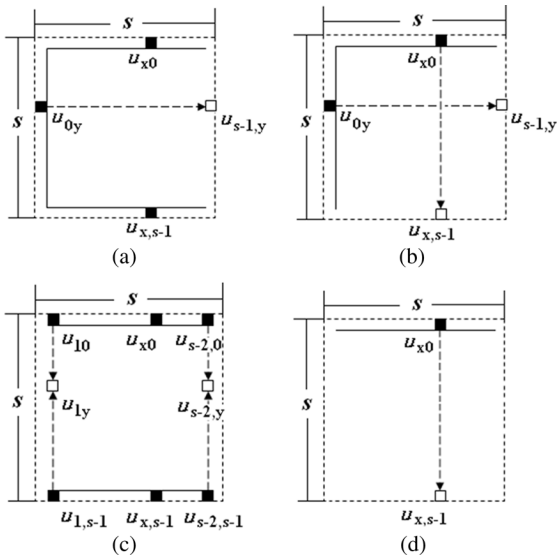

Fig. 6. Border restoration.

As depicted in Fig. 5, suppose only the border pixels are available in an $S \times S$ block. Each unknown inner pixel $u_{xy}(1 \le x, y \le S-2)$ is estimated from four corresponding border pixels as

$$\begin{aligned} u_{xy} &= k_1 p(u_{x0}) + k_2 p(u_{x,s-1}) + k_3 p(u_{0y}) + k_4 p(u_{s-1,y}) \\ p(u_{ij}) &= u_{ij} + (x-i)\hat{g}_x + (y-j)\hat{g}_y \\ k_1 &= \frac{1}{2}\left(1 - \frac{y}{s-1}\right) \\ k_2 &= \frac{1}{2} - k_1 \\ k_3 &= \frac{1}{2}\left(1 - \frac{x}{s-1}\right) \\ k_4 &= \frac{1}{2} - k_3 \end{aligned} \tag{3-6}$$

where $p(u_{ij})$ represents the prediction generated by border pixel u_{ij} with the block gradient $(\hat{g}_x, \hat{g}_y)$. However, not all of the four borders are available in most cases. Fig. 6 illustrates the scenarios in which border restoration is required first. Take case (a) for example, only three borders are known and one item in (3-6) is missing. Pixels in the missing border are then predicted by

$$u_{s-1,y} = p(u_{0y}) = u_{0y} + (s-1)\hat{g}_x. \tag{3-7}$$

In the other three cases, the missing borders are restored similarly.

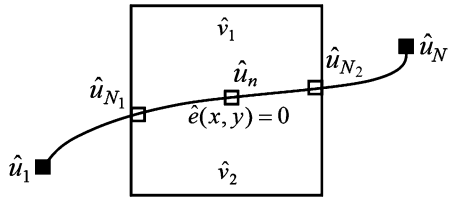

Fig. 7. PAI in a structural block.

C. Structural Region

For structural image regions that have edges inside, the key features are the *edge location* and *edge variation* (i.e., variation across the edge). Here we model an edge as a Gaussian blurred step function

$$I(r) = ((I_1 - I_2)s(r) + I_2) * \frac{1}{2\pi\sigma^2} e^{-r^2/2\sigma^2}. \tag{3-8}$$

$I(r)$ denotes the pixel intensity whose distance to the nearest edge pixel is r. I_1 and I_2 are two intensities a certain distance away from the two sides of the edge. $s(r)$ is a step function and σ is the scale of the Gaussian blur kernel. The parametric distribution on a structural region is given as

$$P_S\left(u | e(x,y) = 0, v(e)\right) : (x,y) \in \Omega \tag{3-9}$$

where $e(x,y) = 0$ denotes the edge location and $v(e)$ the edge variation. The edge location is obtained by edge detection during region classification at pixel level, whereas the edge variation is estimated at block level considering the consistency along an edge as well as the coding efficiency. We calculate the edge variation in a structural block as

$$\hat{v}(\hat{e}) = \hat{v}_1\left(u|_{\hat{e}(x,y)<0}\right) + \hat{v}_2\left(u|_{\hat{e}(x,y)>0}\right) \tag{3-10}$$

where $\hat{e}(x,y) = 0$ is the detected edge location and $\hat{v}_1, \hat{v}_2$ are the variances of pixel values on the two sides of the edge (not including the edge itself), which can be obtained from the image observations in the block similarly to (3-1). Fig. 7 gives a simple illustration of the edge variation estimation.

The recovery of dropped structural regions is similar to that in [10] by edge propagation as well as texture synthesis. First, edge pixels in an unknown block are generated from pixels on the same edge piece in the known surroundings using linear interpolation

$$\begin{aligned} \hat{u}_n|_{\hat{e}(x,y)=0} &= \frac{\sum_{k=1}^{N} \lambda_{nk} u_k}{\sum_{k=1}^{N} \lambda_{nk}} \\ \lambda_{nk} &= \begin{cases} 0, & N_1 \le k \le N_2 \\ (n-k)^{-2}, & \text{else} \end{cases} \end{aligned} \tag{3-11}$$

where $\hat{u}_n (N_1 \le n \le N_2)$ are the restored edge pixels in the unknown block and N gives the total number of pixels on this edge piece, as also shown in Fig. 7. Then, in a designated *influence region* that includes unknown pixels within a short distance from the restored edge, edge propagation is performed through a pair matching algorithm. For each target pixel in the influence region, a structural candidate and a textural candidate are found from the surrounding available pixels. The one with smaller sum of squared difference (SSD) between its neighborhood and that of the target pixel is then chosen to fill in the target pixel. Finally, the remaining unknown pixels are restored by a patch-wise texture synthesis method.

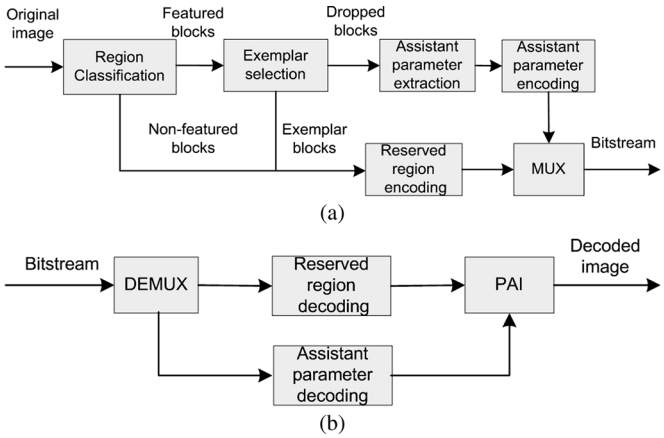

Fig. 8. Framework of PAI-based compression system. (a) Encoder; (b) decoder.

Our scheme is kind of different from [10] in that one more assistant parameter is introduced, i.e., the edge variation. It directs the inpainting process in the following two aspects.

1) Edge variation indicates the sharpness of an edge, e.g., an ideal step edge has zero variation according to (3-10). So blocks with small edge variations have a small width of influence region in edge propagation to avoid over-smoothing.
2) Edge variation also indicates the content complexity in a block. Since the content complexity of an image patch is restricted to its size, blocks with large edge variations require a large patch size to enhance the accuracy of the SSD-based similarity measurement during texture synthesis.

IV. PAI-Based Compression System

Based on the proposed PAI method, we present an image compression system aiming at high coding performance in terms of the perceptual quality. The framework of our system is shown in Fig. 8. At the encoder side (a), an original image is classified into featured and non-featured blocks. A small portion of featured blocks are reserved as exemplars while assistant parameters are extracted from the other dropped blocks. Then all the reserved blocks as well as the assistant parameters are encoded and transmitted to the decoder. The decoder side of our compression system is depicted in (b). After the reserved regions and assistant parameters are decoded, the entire image is automatically restored by PAI. In the latter part of this section, we will supply some implementation details of exemplar selection, parameter extraction, coding and adjusting during inpainting.

For all featured blocks, exemplar selection is processed to decide which ones will be dropped during encoding. The reason we keep a few featured blocks as exemplars is that inpainting needs necessary boundary constraints. Furthermore, when some featured blocks adjacent to non-featured ones are dropped, the inpainting process may "leak" inhomogeneous contents into featured regions and thus lead to artifacts. In view of these considerations, we select the exemplars based on the following two principles:

1) gradated block is determined as exemplar if structural or non-featured blocks exist in its eight-neighborhood;
2) structural block is determined as exemplar if its edge variation is higher than a threshold.

After exemplar selection, the other featured blocks are dropped during encoding, while some parameters are extracted from them to assist inpainting. Here we only want to point out a detail that the block gradient is calculated in an overlapped $(S + 2) \times (S + 2)$ block since, actually, the border pixels used in inpainting cannot belong to the unknown $S \times S$ block. Similarly, the recorded edge locations are not only the ones inside the dropped block, but also those on the same edge piece inside the neighboring reserved blocks.

At the encoder side, the contents to be coded are comprised of reserved regions and assistant parameters. The reserved regions, including non-featured blocks and featured exemplars, are compressed with baseline JPEG. In addition, the assistant information, including the positions of dropped blocks, the types of dropped blocks (gradated or structural) and edge locations, are coded as bitmaps with JBIG. The edge variations, placed in an array, are coded by an arithmetic encoder after uniform quantization. The block gradients are compressed by a scheme similar to differential pulse code modulation (DPCM) which includes quantization, intra prediction, and entropy coding followed by run-length coding. The detailed procedures are omitted due to space limitations.

At the decoder side, reserved regions and assistant parameters are both decoded. Then PAI begins to reconstruct the entire image with them. Although inpainting with block gradients can maintain the continuity of gradation patterns in the dropped regions, it is possible that inpainting errors (caused by parameter modeling, estimation and coding) occur and accumulate in the block-by-block restoration process, especially when large-scale gradated regions are removed. Therefore, we have the parameters adjusted dynamically in a simple way that the incontinuity, once detected, is averagely compensated along the inpainting "path" which can be predicted in advance. Thus it will not accumulate to cause visible artifacts.

V. Experimental Results

We test our compression scheme on a number of color images with different gradation patterns and edges inside. In these experiments, some parameters are empirically determined in terms of rate-distortion optimization. Here the distortion is evaluated in certain perceptual quality assessments. The block size is 16×16 and the threshold for color variance V_T is 2000 in region classification, where an edge detection algorithm similar to [12] is utilized. The preserved ratio of the structural exemplar is 0.3. The quantization value of edge variation is either 0 or 1; the corresponding width of influence region in edge propagation and the patch size in texture synthesis are 7, 3×3 and 11, 4×4, respectively. For different color components, the same set of edge locations and variations are utilized, whereas the block gradients are calculated and recorded separately.

Since our scheme adopts JPEG for the coding of reserved regions, its performance is mainly evaluated in comparison with baseline JPEG in terms of the perceptual quality and compression ratio. To demonstrate the potential of our scheme, some results compared with H.264-Intra are also presented. In Figs. 9 and 10, test images are taken from the UCID image database [13], the Kodak image library[1] and a synthetic image of Windows desktops.

In Fig. 9, the reserved image regions and some assistant parameters (edge location and variation here, block gradient can be referred to in Fig. 4) are shown in the first column. Compressed images with our scheme and baseline JPEG under the same bit-rate are given in the second and the third column. It can be observed (better in an enlarged view) that PAI gives much less artifacts than JPEG, especially in the sky of *Pole*, the wall of *Statue* and the sunglow of *Kodim20*. The artifacts are more severe in images in the last column resulting from exemplar-based inpainting [7], which further demonstrates the necessity and effectiveness of PAI. More results of comparisons between our scheme and baseline JPEG are given in Fig. 10, where we set the same quality parameter of JPEG coding in both methods. It means the reserved regions have the same quality in the two schemes, both subjectively and objectively. The dropped featured regions restored by PAI, as can be

[1][Online]. Available: http://r0k.us/graphics/kodak/

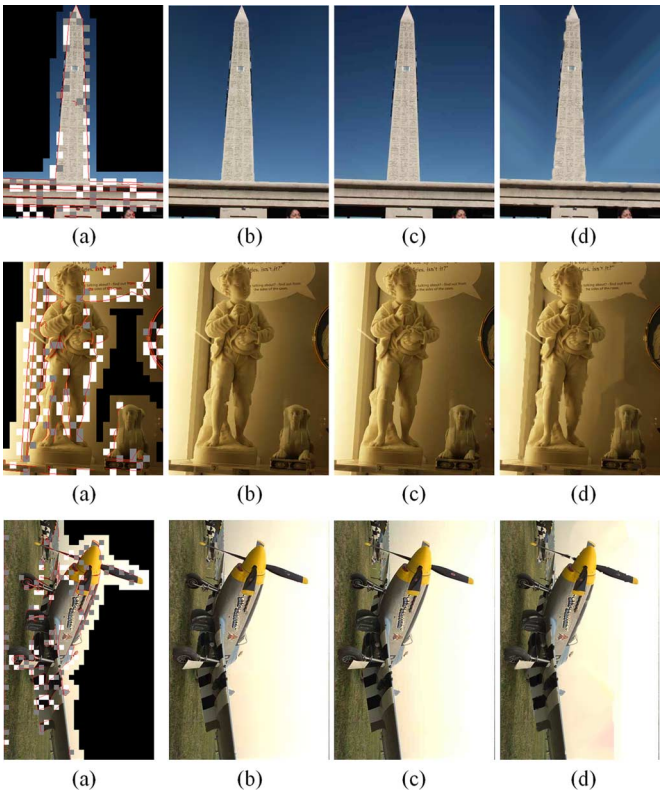

Fig. 9. Comparison results on test images *Pole* (384 × 512), *Statue* (384 × 512), and *Kodim20* (512 × 768). (a) reserved regions (black for dropped gradated blocks, white for dropped structural blocks with 0 edge variation, gray for those with 1 edge variation and red lines for recorded edge locations), (b) result of PAI, (c) result of JPEG, and (d) result of exemplar-based inpainting [7].

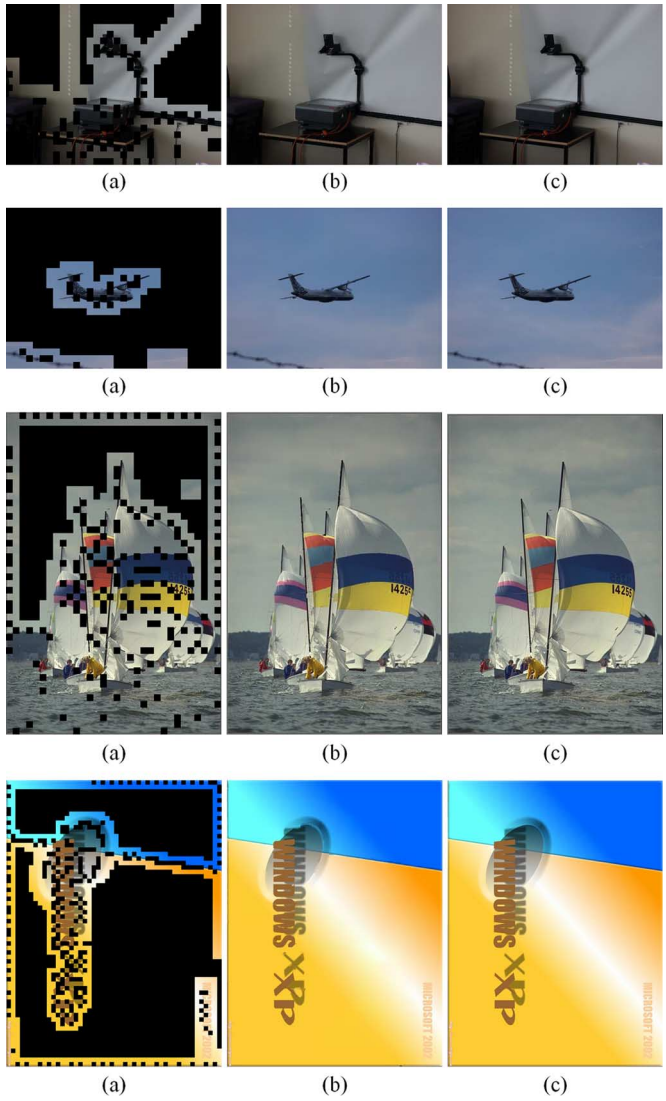

Fig. 10. Comparison results on test images *Projector* (512 × 384), *Plane* (512 × 384), *Kodim09* (512 × 768), and *Winxp* (768 × 1024). (a) reserved regions, (b) result of PAI, and (c) result of JPEG.

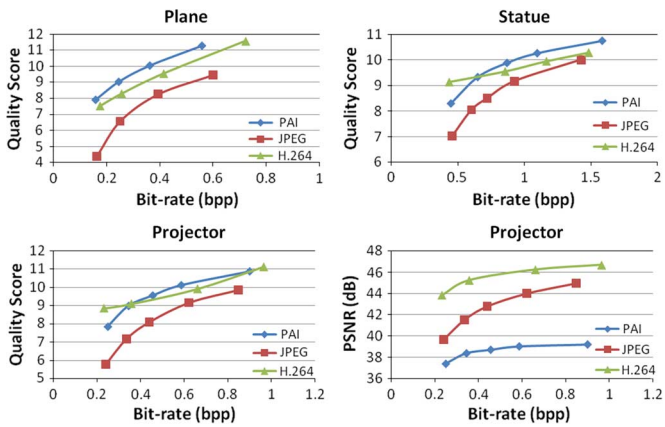

Fig. 11. Rate-distortion curves of PAI, JPEG and H.264-Intra.

seen, still have similar perceptual quality to those in baseline JPEG, whereas the coding bit-rates are dramatically saved.

As mentioned all through this correspondence, our PAI-based compression method mainly aims at rate-visual distortion optimization, instead of pixel-wise fidelity perfection, so as to further explore visual redundancy in color images. Therefore, we choose a perceptual quality assessment in [14] for presenting the rate-visual distortion curves in Fig. 11. Performance of three coding methods: PAI, JPEG, and H.264-Intra are provided. The quality score is independently calculated for RGB color components with the green component being taken for example. The perceptual quality assessment in [14] is designed to capture the blocking artifacts in the compressed image, which are fair for the above three block-based coding methods. Under this criterion, PAI greatly outperforms JPEG. Compared with H.264-Intra, PAI is still competitive, with a higher curve most of the time. In addition, a PSNR curve is shown in Fig. 11, giving the performance in pixel-wise fidelity. Despite lower objective quality for the proposed method, human observers seldom have corresponding perceptual experience. It reveals the fact that PSNR does not always provide an accurate and fair measure of visual quality for different image coding schemes.

The bit-rate savings of our scheme compared with JPEG at featured regions are given in Table I, where the quality parameter of JPEG coding is set to 75. Our method can save 76% of bits on average at similar perceptual quality levels in these featured regions.

The computational complexity of our scheme could be a little higher than that of the traditional methods. Suppose the input image size is N, the block size is M and the ratio of dropped regions is $r(0 \leq r \leq 1)$. At the encoder side, the additive complexity mainly comes from block classification (including edge detection and color variance calculation) with complexity of $O(N)$, and assistant parameter extraction (including block gradients and edge variations) with complexity $O(rN)$. At the decoder side, the complexity of PAI is approximately $O(rN^2/M)$.

TABLE I
BIT-RATE (BPP) OF PAI AND JPEG IN FEATURED REGIONS ($QP = 75$)

Image Name	JPEG	PAI	**Bits-saving**
Pole	0.307	0.047	**84.8%**
Statue	0.548	0.133	**75.8%**
Kodim20	0.366	0.080	**78.1%**
Projector	0.298	0.095	**68.2%**
Plane	0.168	0.056	**66.5%**
Kodim09	0.546	0.124	**77.2%**
Winxp	0.278	0.049	**82.4%**

VI. CONCLUSION

This correspondence presents an image compression system based on the proposed parameter-assistant image inpainting method to more deeply exploit visual redundancy inherent in color images. We have shown that with carefully selected dropped regions and appropriately extracted parameters from them, dropped regions can be satisfactorily restored using the proposed PAI algorithm. Accordingly, our compression scheme has a higher coding performance compared with traditional methods in terms of the perceptual quality.

Further improvements of the current scheme are still promising. First, assistant parameters may be described and compressed into the bit stream in a more condensed manner. Second, extraction of distinctive features could be more flexible and adaptable. Besides gradated and structural regions, other types, such as textural regions, could also be involved in our future work.

ACKNOWLEDGMENT

The authors would like to thank all the anonymous reviewers and the associate editor for their constructive comments and suggestions.

REFERENCES

[1] M. M. Reid, R. J. Millar, and N. D. Black, "Second-generation image coding: An overview," *ACM Comput. Surveys*, vol. 29, no. 1, pp. 3–29, Mar. 1997.

[2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. ACM SIGGRAPH*, 2000, pp. 417–424.

[3] T. F. Chan and J. H. Shen, "Mathematical models for local nontexture inpaintings," *SIAM J. Appl. Math.*, vol. 62, no. 3, pp. 1019–1043, Feb. 2002.

[4] T. F. Chan and J. Shen, "Nontexture inpainting by curvature driven diffusions (CDD)," *J. Visual Commun. Image Rep.*, vol. 12, no. 4, pp. 436–449, Dec. 2001.

[5] S. Esedoglu and J. Shen, "Digital inpainting based on the Mumford-Shah-Euler image model," *Eur. J. Appl. Math.*, vol. 13, pp. 353–370, 2002.

[6] D. Tschumperlé and R. Deriche, "Vector-valued image regularization with PDEs: A common framework for different applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 506–517, Apr. 2005.

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

[8] S. D. Rane, G. Sapiro, and M. Bertalmio, "Structure and texture filling-in of missing image blocks in wireless transmission and compression applications," *IEEE Trans. Image Process.*, vol. 12, no. 3, pp. 296–303, Mar. 2003.

[9] C. Wang, X. Sun, F. Wu, and H. Xiong, "Image compression with structure-aware inpainting," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2006, pp. 1816–1819.

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

[11] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2743–2760, Oct. 1998.

[12] C. A. Rothwell, J. L. Mundy, W. Hoffman, and V. D. Nguyen, "Driving vision by topology," in *Proc. Int. Symp. Computer Vision*, 1995, pp. 395–400.

[13] G. Schaefer and M. Stich, "UCID—An uncompressed colour image database," in *Proc. SPIE*, 2004, vol. 5307, pp. 472–480.

[14] Z. Wang, H. R. Sheikh, and A. C. Bovik, "No-reference perceptual quality assessment of JPEG compressed images," in *Proc. IEEE Int. Conf. Image Process.*, 2002, vol. 1, pp. 477–480.

A Completed Modeling of Local Binary Pattern Operator for Texture Classification

Zhenhua Guo, Lei Zhang, and David Zhang

***Abstract*—In this correspondence, a completed modeling of the local binary pattern (LBP) operator is proposed and an associated completed LBP (CLBP) scheme is developed for texture classification. A local region is represented by its center pixel and a local difference sign-magnitude transform (LDSMT). The center pixels represent the image gray level and they are converted into a binary code, namely CLBP-Center (CLBP_C), by global thresholding. LDSMT decomposes the image local differences into two complementary components: the signs and the magnitudes, and two operators, namely CLBP-Sign (CLBP_S) and CLBP-Magnitude (CLBP_M), are proposed to code them. The traditional LBP is equivalent to the CLBP_S part of CLBP, and we show that CLBP_S preserves more information of the local structure than CLBP_M, which explains why the simple LBP operator can extract the texture features reasonably well. By combining CLBP_S, CLBP_M, and CLBP_C features into joint or hybrid distributions, significant improvement can be made for rotation invariant texture classification.**

***Index Terms*—Local binary pattern (LBP), rotation invariance, texture classification.**

I. INTRODUCTION

Texture classification is an active research topic in computer vision and pattern recognition. Early texture classification methods focus on the statistical analysis of texture images. The representative ones include the co-occurrence matrix method [1] and the filtering based methods [2]. Kashyap and Khotanzad [3] were among the first researchers to study rotation-invariant texture classification by using a circular autoregressive model. In the early stage, many models were

Manuscript received December 07, 2009; revised January 26, 2010. First published March 08, 2010; current version published May 14, 2010. This work was supported in part by the CERG fund from the HKSAR Government, by the central fund from the Hong Kong Polytechnic University, by Key Laboratory of Network Oriented Intelligent Computation (Shenzhen), and by SZHK-innovation funds SG200810100003A (China). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mark Liao.

Z. Guo is with the Graduate School at Shenzhen, Tsinghua University, Shen Zhen, China (e-mail: cszguo@gmail.com).

L. Zhang and D. Zhang are with the Department of Computing, the Hong Kong Polytechnic University, Kowloon, Hong Kong, China (e-mail: cslzhang@comp.polyu.edu.hk; csdzhang@comp.polyu.edu.hk).

Color versions of one or more of the figures in this correspondence are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2010.2044957