IMAGE CODING WITH PARAMETER-ASSISTANT INPAINTING

Zhiwei Xiong[1*], *Xiaoyan Sun*[2], *Feng Wu*[2] *and Shipeng Li*[2]

[1]Dept. of Electronic Science and Technology, Univ. of Science and Technology of China, Hefei, China
[2]Internet Media Group, Microsoft Research Asia, Beijing, China

ABSTRACT

This paper carves out an image compression approach that integrates our parameter-assistant inpainting (PAI) technique to exploit the visual redundancy inherent in color-gradation image regions. In our scheme, an input image is first classified at block level according to the degree of edge content as well as chromatic variation in each block. An exemplar selection approach is then adopted to skip a majority of the gradation blocks during encoding. Only their positions and certain parameters extracted for condensed description are encoded along with the reserved blocks. At the decoder side, the skipped regions are recovered through image inpainting, relying on both the delivered parameters and reserved regions. Experimental results show that our proposed method outperforms baseline JPEG at color-gradation regions by nearly 80% bits-saving, at similar visual quality levels.

Index Terms— image compression, image inpainting, color-gradation, parameter-assistant

1. INTRODUCTION

Over the past two decades, great improvements continue to be made in image and video compression fields. 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. Perceptual quality, however, is largely ignored during conventional algorithm design. In addition, current developments also demonstrate that even small improvements are commonly accomplished at the expense of multiplying encoding complexity.

Meanwhile, attempts have also been made to develop new compression techniques by utilizing features within images to achieve high coding efficiency [1]. Moreover, recent vision-related technologies have shown remarkable progress in hallucinating pictures with good perceptual quality, among which image inpainting is a very promising approach to be utilized in image compression, aiming at visual perceptual quality instead of pixel-wise fidelity.

Image inpainting, also know as image completion, was first introduced into digital image processing by Bertalmio et al. in [2], as a process of restoring missing data in a designated region of an image in a visually plausible way. Subsequently, several mathematical models, including total variation (TV) model [3], curvature driven diffusions (CCD) model [4] and Mumford-Shah model [5], along with important applications of inpainting have been investigated and presented. Current results of image inpainting [6]-[8] illustrate that it can naturally recover homogenous regions as well as certain kind of structural regions. Furthermore, compression schemes have also been reported in literature, which employ image inpainting techniques to improve the visual quality in a straight-forward fashion [8].

Current inpainting methods, as mentioned above, are feasible for image restoration of small plots. However, when applied to images with large-scale unknown color-gradation regions in which the hue or lightness changes smoothly, simplex inpainting can hardly deduce the gradation patterns from the known image content, so that it is unable to preserve the fidelity and smoothness of the gradation during recovering. Thus, in our scheme, some representative information extracted from gradation areas is coded and delivered instead of the pixel values inside these regions. Due to the transmitted assistant information, the reliability of the inpainting at decoder is greatly enhanced, whereas high coding efficiency can also be acquired.

The rest of this paper is organized as follows. In Sec. 2 we propose a mathematical model of PAI based on Taylor's theorem and the detailed algorithm is also discussed. Sec. 3 describes our compression scheme. Experimental results are presented in Sec. 4, and Sec. 5 concludes the paper.

2. MATHEMATICAL ANALYSIS OF PAI

In this section we give a detailed mathematical analysis of our PAI model which is aiming at image compression. As shown in Fig. 1, suppose $f(x, y)$ is a C^n function defined on a 2-D domain Ω and B is a subset of Ω bounded by L. Values on B are unknown and to be inpainted. Given some prior information such as all nth ($n \geq 1$) partial derivatives at a point $z_0 = (x_0, y_0)$ which belongs to L, Taylor's theorem can be applied here to estimate a local $f(x, y)$ for recovery by

$$f_r(x,y) = f(x_0,y_0) + \sum_{q=1}^{n} \frac{1}{q!}[(x-x_0)\frac{\partial}{\partial x} + (y-y_0)\frac{\partial}{\partial y}]^q f(x_0,y_0) \quad (1)$$

where $(x,y) \in U(z_0,\delta)$, the region within a radius of δ from z_0, as shown by the dashed circle in Fig. 1.

It can be observed that the restoration error increases rapidly with δ. Since our compression system prefers large

* *This work has been done during Z. Xiong's internship at MSRA.

removed area, propagating from the boundaries using Taylor's theorem cannot guarantee enough precision for large region recovery, especially when color-gradation is contained. (The situation is almost the same when Green's second formula is utilized for smooth inpainting in [3]) Therefore, more information, that is assistant information, is required to enhance the accuracy of inpainting.

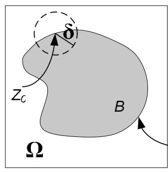


Fig. 1: Inpainting via Taylor's theorem

2.1. Assistant parameter extraction

Inspired by (1), one solution for large gradation region inpainting is that we can take 1 to nth derivatives ($n \geq 1$) at certain points uniformly distributed in the removed area as reserved parameters. Then an iterative inpainting process can be piecewise optimal to minish the estimation errors induced by the long "propagation" distance.

However, for a real discrete image function $u[x, y]$, in which pixel values are integer, gradations actually comprise a series of step-like segments. In this case, it is difficult to tell whether there is a gradation and the exact gradation pattern in a local area only by derivatives at one point. Thus we propose a block-wise gradient estimation to calculate parameters representing the average gradient in a gradation block for more reliable inpainting.

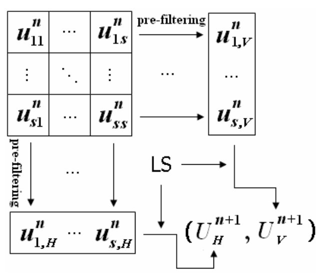


Fig. 2: Block gradient estimation

As shown in Fig. 2, $u_{xy}^n (1 \leq x, y \leq s)$ denotes one of the nth partial derivatives of $u[x, y]$ in an $s \times s$ block, which can be iteratively calculated from the original image. A pre-filtering is first carried out to achieve the vertical gradation array V that consists of the mean value of each row, and the horizontal gradation array H for that of each column. Then we use the LS (Least-Square) method to approximate the gradation pattern presented by them. For instance, given the vertical array $V = (u_{1,V}^n, \cdots, u_{s,V}^n)$, a linear function $u_V^n = a_V x + b$ can be adopted to fit the points where

$$a_V = \sum_{i=1}^{i=s} (i - \frac{s+1}{2})(u_{i,V}^n - \overline{u_V^n}) \Big/ \sum_{i=1}^{i=s} (i - \frac{s+1}{2})^2 \qquad (2)$$

Here $\overline{u_V^n}$ represents the average value of V. Accordingly, a_V will be taken as the vertical-element of the $(n+1)^{th}$ block gradient U^{n+1} corresponding to u_{xy}^n and the horizontal-element a_H of U^{n+1} can be obtained the same way from H. The block gradients present a reliable gradation pattern in a local area, so they are selected as the assistant parameters for gradation region inpainting.

2.2. Parameter-assistant inpainting

As indicated above, the nearer the unknown pixel to the known ones, the better recovery can be obtained. Thus, our PAI completes an unknown region from known boundaries inward block by block, whereas the interpolation in one block is performed at pixel level to maintain the continuity of gradation. Specifically, blocks with the maximum known neighbors in their four-neighborhood will be recovered first by the following inpainting process.

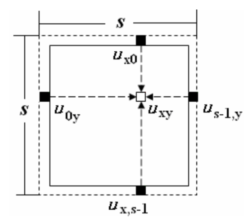


Fig. 3: Interpolation inside unknown block

As shown in Fig. 3, suppose only the border pixels are available in an $s \times s$ block. Then each unknown inner pixel $u_{xy} (1 \leq x, y \leq s-2)$ can be readily estimated from the four corresponding border pixels: u_{x0}, $u_{x,s\text{-}1}$, u_{0y}, and $u_{s\text{-}1,y}$ by

$$\begin{aligned} u_{xy} &= k_1 p(u_{x0}) + k_2 p(u_{x,s-1}) + k_3 p(u_{0y}) + k_4 p(u_{s-1,y}) \\ p(u_{ij}) &= u_{ij} + \sum_{q=1}^{n} \frac{1}{q!}[(x-i)\frac{\partial}{\partial x} + (y-j)\frac{\partial}{\partial y}]^q U \end{aligned} \qquad (3)$$

$$k_1 = \frac{1}{2}(1 - \frac{y}{s-1}),\ \ k_2 = \frac{1}{2} - k_1,\ \ k_3 = \frac{1}{2}(1 - \frac{x}{s-1}),\ \ k_4 = \frac{1}{2} - k_3$$

where $p(u_{ij})$ represents the prediction of u_{xy} generated by u_{ij} via Taylor's theorem using the block gradients U^q. We have observed that, for gradation regions, the first-order block gradient plays a dominant role in pixel value prediction. Thus in our system only the first-order block gradient item is considered. Then, (3) can be simplified to

$$u_{xy} = k_1 u_{x0} + k_2 u_{x,s-1} + k_3 u_{0y} + k_4 u_{s-1,y} \qquad (4)$$

However, not all of the four borders are available in most circumstances. As shown in Fig. 4, there are still other scenarios in which we need to obtain the borders first. In these cases, the assistant parameters play a key role in border restoration. Taking case (a) for example, only three borders are known and one item in (4) is missing. Then for each pixel in the unknown border (right border here), it can be predicted by

$$u_{s-1,y} = p(u_{0y}) = u_{0y} + (s-1)U_H, \quad 1 \leq y \leq s-2 \qquad (5)$$

where U_H represents the horizontal-element of the first-order block gradient. Accordingly, the interpolation inside block can be performed as

$$u_{xy} = k_1 u_{x0} + k_2 u_{x,s-1} + 0.5(u_{0y} + xU_H) \tag{6}$$

Similarly, when only two conjunct borders are available as denoted by (b), the unknown pixels are recovered by

$$u_{xy} = 0.5(u_{x0} + yU_V + u_{0y} + xU_H) \tag{7}$$

Here U_V is the vertical-element of the first-order block gradient. If only two parallel borders are available as shown in (c), u_{1y} and $u_{s-2,y}$ are estimated in place of u_{0y} and $u_{s-1,y}$.

$$u_{1y} = 2k_1 p(u_{10}, u_{1y}) + 2k_2 p(u_{1,s-1}, u_{1y}), \quad 1 \le y \le s-2 \tag{8}$$

So can we get $u_{s-2,y}$. Then the restoration is performed by

$$\begin{aligned} u_{xy} &= k_1 u_{x0} + k_2 u_{x,s-1} + 2k_3'(k_1 u_{10} + k_2 u_{1,s-1}) \\ &\quad + 2k_4'(k_1 u_{s-2,0} + k_2 u_{s-2,s-1}) \\ k_3' &= \frac{1}{2} - \frac{x-1}{2(s-3)}, \quad k_4' = \frac{x-1}{2(s-3)} \end{aligned} \tag{9}$$

Moreover, for the case (d) that only one border is available, each unknown pixel is filled in by

$$u_{xy} = 0.5u_{x0} + k_3' u_{10} + k_4' u_{s-2,0} + yU_V \tag{10}$$

Among the above four cases, (b) is the most frequently appearing one because inpainting always starts from the corners according to the priority determination. Therefore, assistant parameters are used adequately.

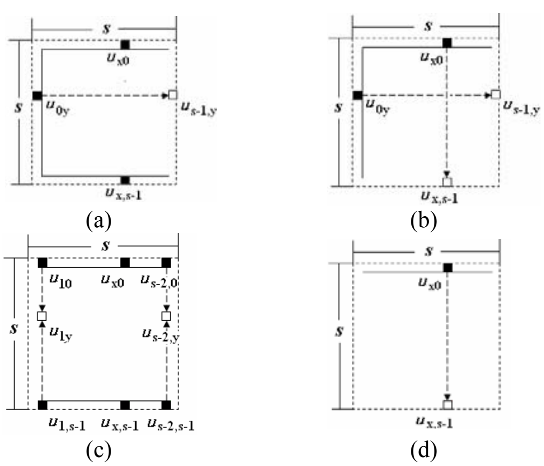


Fig. 4: Border restoration

3. PAI-BASED COMPRESSION SCHEME

The framework of PAI-based compression scheme is shown in Fig. 5. At the encoder side (a), an original image is first classified into different type of blocks as gradation and non-gradation, according to the edge content and chromatic variation of each block with a uniform size $s \times s$ (s = 16 in the current system). In specific, blocks that have edge inside (determined by edge detection similar to [9]) will be treated as non-gradation ones; for any other blocks, the chromatic variation is calculated by

$$\text{var} = \sum_{\forall x}\sum_{\forall y}[(R_{xy} - \overline{R})^2 + (G_{xy} - \overline{G})^2 + (B_{xy} - \overline{B})^2] \tag{11}$$

where $\overline{R}$, $\overline{G}$ and $\overline{B}$ are the average value of corresponding chrominance components in the block. Then, blocks with variation less than a threshold will be categorized into gradation type while others are of non-gradation blocks.

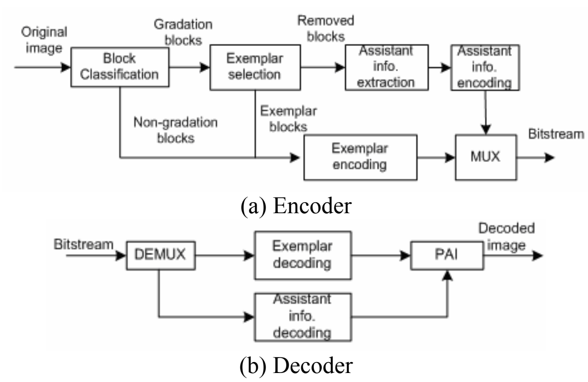


Fig. 5: Framework of PAI-based compression scheme

For all the gradation blocks, an exemplar selection is processed to decide which ones will be removed. In order to keep necessary boundary values and meanwhile to prevent structure and texture content from "leaking" into gradation regions in the inpainting process, we select some gradation blocks as exemplars based on the following two principles:

1) If edge pixels exist in a certain range (an $2s \times 2s$ rectangle in our experiments) centered with it;

2) Otherwise, there is at least one neighbor in its eight-neighborhood which is non-gradation and without edge;

Then this block is reserved as exemplars.

After that, the gradation blocks which have not been reserved will be skipped during encoding, but some assistant parameters will be extracted to represent them through the inpainting process, as discussed in Section 2.1. Here we only want to point out a detail that the gradients are calculated in an overlapped $(s+2) \times (s+2)$ area since, actually, the boundary values used in inpainting don't belong to the unknown $s \times s$ block.

For the reserved blocks including non-gradation ones and gradation exemplars, baseline JPEG is adopted to compress them. Also, we need a bitmap recording which blocks are removed. This map is encoded by JBIG. Besides, the assistant parameters are encoded in a carefully designed manner which includes quantization, intra prediction and entropy coding followed by run-length coding.

At the decoder side as shown in (b) of Fig. 5, reserved image regions and assistant parameters are both decoded. Then PAI begins to reconstruct the entire image with them. As introduced in Section 2.2, the proposed interpolation can well maintain the continuity of gradation in a local area. However, it is unavoidable that the inpainting error occurs and cumulates due to parameter quantization and iterative interpolation, especially when large-scale gradation regions are removed. Therefore, we have the parameters adjusted dynamically in such a simple way that the error, once detected, is averagely compensated along the inpainting "path" which can be predicted in advance. Thus it will not cumulate to cause visible gradation incontinuity.

4. EXPERIMENTAL RESULTS

We test our compression approach on a number of color images with different gradation patterns inside. Here, three examples are given. Two of them are from the USC-SIPI image database [11], and one component images is also used for testing.

Fig. 6 gives the result of image Gantry from [11]. Our inpainting method (c) successfully recovers the gradation regions denoted in blue in (b) (the exemplars are marked in red and the white arrows represent both the direction and magnitude of assistant parameters). Whereas inpainting without assistant information [10], as shown in (d), brings incontinuity in gradation regions. Compared with baseline JPEG (a), our method saves 79.7% bits at similar visual quality level in terms of gradation areas; in case the entire image is considered, our scheme still achieves 16.8% bit-saving. More comparison results can be found in Fig.7. It can be observed that our scheme achieves 31.3% and 26.3% bits reduction respectively at similar perceptual qualities.

5. CONCLUSION AND FUTURE WORK

The main problem we want to tackle in this paper is how to preserve the color-gradation patterns within images when inpainting is incorporated with image compression scheme. And our proposed PAI-based approach has shown that with careful selection of reserved exemplars as well as proper parameters extracted from removed gradation regions, the entire image can be satisfactorily reconstructed.

Further improvements of current scheme are still promising. Firstly, the assistant information as well as the selected exemplars can be described and compressed into bit stream in more compact fashion. Secondly, extraction of the distinctive features can be more flexible and adaptable. Besides gradation regions, textural regions should also be involved in our future work.

REFERENCES

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

[2] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, "Image inpainting". Proceedings of ACM SIGGRAPH 2000, pp. 417-424.

[3] T. F. Chan, J. H. Shen, "Mathematical models for local nontexture inpaintings". SIAM J. Applied Mathematics, vol. 62, no. 3, pp. 1019-1043. Feb. 2002.

[4] T. F. Chan, J. Shen, "Nontexture inpainting by curvature driven diffusions (CDD)". Journal Visual Comm. 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", European Journal on Applied Mathematics, vol. 13, pp. 353–370. 2002.

[6] M. Bertalmio, L. Vese, G. Sapiro, S. Osher, "Simultaneous structure and texture image inpainting". IEEE Trans. Image Processing, vol. 12, no. 8, pp. 882-889. Aug. 2003.

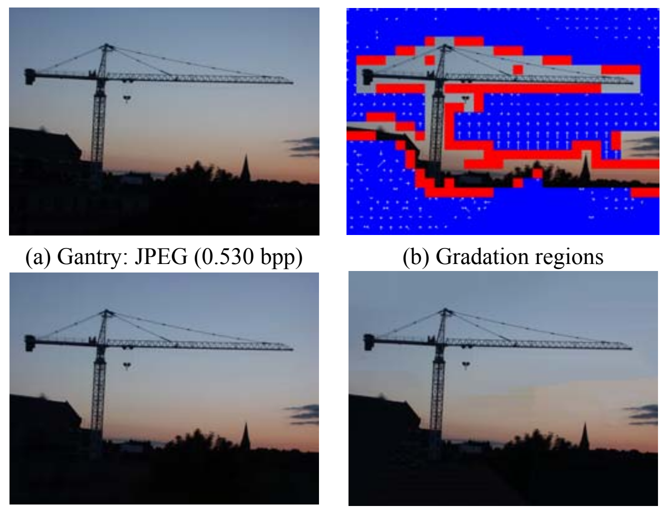
[7] H. Grossauer, "A combined PDE and texture synthesis approach to inpainting". Proceedings of ECCV 2004, pp. 214-224.

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

[9] C. A. Rothwell, J. L. Mundy, W. Hoffman, V. D. Nguyen, "Driving vision by topology". Proceedings of International Symposium on Computer Vision 1995, pp. 395-400.

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

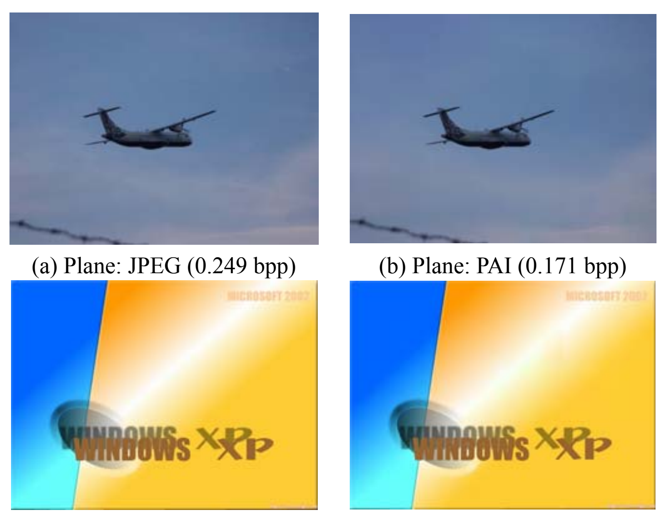
[11] http://sipi.usc.edu/services/database/



(a) Gantry: JPEG (0.530 bpp) (b) Gradation regions

(c) Gantry: PAI (0.441 bpp) (d) Simplex inpainting

Fig. 6: Test results of Gantry (512×384)



(a) Plane: JPEG (0.249 bpp) (b) Plane: PAI (0.171 bpp)

(c) Winxp: JPEG (0.357 bpp) (d) Winxp: PAI (0.263 bpp)

Fig. 7: Test results of Plane (512×384) and Winxp (1024×768)