Shape-Adaptive Image Compression Using Lossy Shape Coding, SA-Prediction, and SA-Deblocking

Li-Ang Chen [1], Jian-Jiun Ding [2], and Yih-Cherng Lee [3]
Graduate Institute of Communication Engineering,
National Taiwan University, Taipei, Taiwan
E-mail: r03942036@ntu.edu.tw [1], jjding@ntu.edu.tw [2], mailappserver@gmail.com [3], Tel: +886-2-33669652

***Abstract*— As the annoying blocking or ghost artifacts tend to appear in the conventional compression approaches either in the JPEG or JPEG2000 standards at low bitrate, the concept of the object-oriented image compression is proposed. This kind of methods is able to retain the image structural boundaries and therefore has relatively good visual qualities even in high compression ratios. In this paper, we propose a shape-adaptive image compression scheme employing an efficient lossy contour compression algorithm to encode the region information, which is usually the main overhead data in such systems. In addition, the prediction and deblocking techniques commonly used in novel compression approaches are also applied with the proposed shape-adaptive versions. Simulation results suggest that the proposed compression system is able to provide compressed images with much better visual qualities and more reasonable degradation forms compared to other prevailing methods.**

I. INTRODUCTION

Efficiencies of compression techniques become more important due to the increasing amount of multimedia contents like images and videos on social network nowadays. The JPEG [1] and JPEG2000 [2] are classic and still commonly used image compression standards. The JPEG employs block partitioning and discrete cosine transform (DCT) while the JPEG2000 apply the discrete wavelet transform (DWT). However, in low bitrate scenarios, there will be blocking artifacts in the JPEG and ghost and ringing artifacts in the JPEG2000. These degradations are annoying and unpleasant to human visual system (HVS). Even the recent H.264/AVC intra-coding [5][6] and the advanced image coding (AIC) [7] suffer from the blocking artifacts as well since the whole systems are also block-based.

As a result, the concept of the shape-/object-oriented image compression was proposed. The target image is segmented into regions with different attributes before the compression. Then, each region are processed separately to utilize image regional properties and maintain visual qualities. In the MPEG-4 video coding standard [3][4], the object or region-of-interest (ROI) region can be encoded separately from the background or non-ROI region. To process arbitrary shape image patch, the shape-adaptive DCT (SA-DCT) was proposed in [8]. In [11], Ding et al. propose a pseudo shape-adaptive approach by partitioning the image regions into triangles and trapezoids. The graph-based image compression method in [12] view the segmentation results as a graph and apply Fourier graph transform (FGT) to replace the orthogonal transforms like DCT or integer transform in the H.264/AVC standard.

(a) (b) (c) (d)

Fig.1 (a) The original regions (20598 bits using the chain code). (b) The approximated regions (4926 bits). (c) The compressed image under 2 bpp. (d) The compressed image under 1 bpp

Generally, besides the image segmentation in the first place, the shape-adaptive compression scheme can be divided into two phases: region/shape coding and region-wise compression. In the proposed method, the following key techniques are used.

Lossy shape coding: instead of record the segmentation results without any loss, we suggest to encode these boundary information with tolerable error to eliminate unnecessary noises using an efficient contour compression algorithm. In addition, a refinement technique to mend the potential overlapping and empty pixels is proposed. Fig. 1 (a) and (b) show the approximated and encoded results.

Shape-adaptive image prediction: we modify the image prediction scheme that is commonly employed in novel image compression system to be shape-adaptive usage. The proposed version can provide better prediction due to region-wise processing as depicted in Fig. 6.

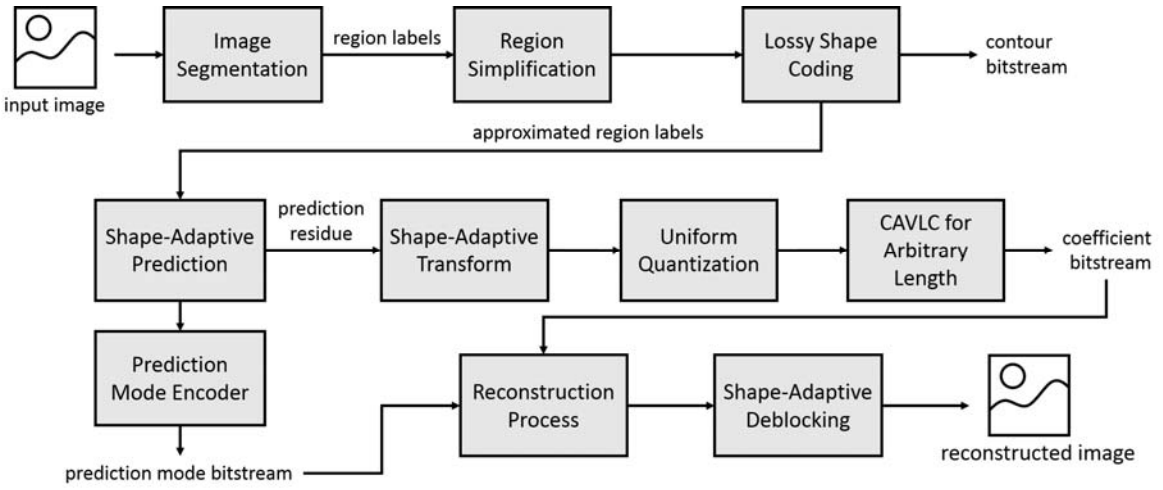


Fig.2 The proposed shape-adaptive compression framework.

Usage of the PO-SA-DCT: although the pseudo-orthonormal SA-DCT (PO-SA-DCT) are regarded as an imperfect transform for its possibility to suffer from the mean weighting defect, we find it more efficient than other commonly used forms of the SA-DCTs while processing the prediction residues.

Shape-adaptive deblocking: in order to reduce the discontinuity caused by block-partitioning in high compression ratios, we propose a shape-adaptive deblocking process based on the in-loop deblocking filter. This technique allows the compressed image remain smooth region-wisely in low bitrates. Fig. 1 (c) and (d) provide the compressed images from the proposed method in 2 and 1 bpp (bit per pixel) cases.

This paper is organized as follows. In section II, some related works of shape-adaptive compression methods are discussed. The proposed method and techniques are describe in section III. Simulation results and performance analysis are provided in section IV. Finally, the conclusion comes in section V.

II. Related Work

Fig. 2 shows the proposed framework of the shape-adaptive image compression system. Among these phases, the main step of large varieties is the shape-adaptive transform coding for arbitrarily shaped patches or blocks.

Between irregular and square blocks, Ding et al. in [11] proposes a DCT expansion method based on trapezoidal and triangular blocks. For a given triangle or trapezoid block, we are able to compose a corresponding rectangle by mirroring against its hypotenuse. As a result, the orthogonal DCT bases $B_{p,q}$ can be calculated efficiently from the DCT bases of the rectangle as in (2). Equation (1) is the DCT bases for rectangles of the size M by N. The obtained orthogonal bases for trapezoidal and triangular blocks are then used to calculate the corresponding DCT coefficients.

$$C_{p,q}(m,n) = k_p h_q \cos\left(\frac{\pi p(m+1/2)}{M}\right)\cos\left(\frac{\pi q(n+1/2)}{M}\right) \quad (1)$$

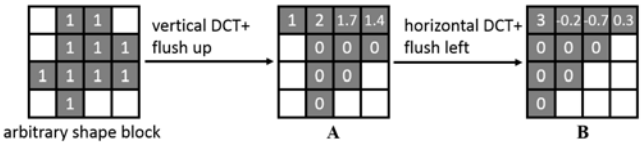


Fig. 3 An illustration of the SA-DCT process and the mean weighting defect.

$$B_{p,q}(m,n) = \{\sqrt{2}C_{p,q}(m,n) \mid p = 0,1,...M-1,\ q = 0,1,...N-1,\ p+q \text{ is even/odd}\} \quad (2)$$

where p and q are the indices of the DCT bases, m and n denote the indices inside each basis, and M and N are the height and width of the rectangle.

In [12], Fracastoro et al. suggest a graph-based compression approach. After the segmentation, instead of further partitioning each region into blocks, they view pixels in a region L_i as a connected graph $G(L, E)$ with pixels as vertices, pixel connectivity as edges, and color differences as edge weights. By performing the eigen value decomposition on the Laplacian matrix of the graph (3), the graph Fourier transform can be committed by (4). As a result, the rate control is achieved by quantizing or discarding the Fourier coefficients.

$$\mathbf{L} = \mathbf{U}^T \mathbf{\Lambda} \mathbf{U} \quad (3)$$

$$\text{GFT}\begin{cases}\text{forward}: \mathbf{F} = \mathbf{U}\mathbf{f} \\ \text{inverse}: \mathbf{f} = \mathbf{U}^{\mathrm{T}}\mathbf{F}\end{cases} \quad (4)$$

where $\mathbf{L}$ is the Laplacian matrix of the graph $G(L, E)$, $\mathbf{U}$ is a matrix whose rows are eigenvectors of $\mathbf{L}$, $\mathbf{\Lambda}$ is a diagonal matrix with eigenvalues of $\mathbf{L}$ as its diagonal elements, $\mathbf{f}$ and $\mathbf{F}$ are the corresponding pixel values and transformed coefficients, respectively.

The SA-DCT [8] is a commonly employed transform coding approach for shape-adaptive compressions. As long as we obtain an arbitrary shape block, the SA-DCT is perform by vertical DCT followed by horizontal DCT on each column and row as shown in Fig. 3, where grey pixels indicate the

opaque pixels. The transformed coefficients are flushed toward the top-left corner, where the lowest frequency locates. Besides, (5) and (6) denote the L-point DCT.

$$DCT_L(u,v) = c_0 \times \cos\left(u\left(v+\frac{1}{2}\right)\times\frac{\pi}{L}\right) \quad (5)$$

$$y = \frac{2}{L \times S_L} \times DCT_L \times x \quad (6)$$

where u, v=0, 1,…, L-1; x and y are the one-dimensional signal and the transformed coefficients respectively; DCT_L is the pre-calculated transform matrix; $c_0 = \sqrt{1/2}$ if u=0 and $c_0 = 1$ otherwise; S_L is the scaling factor.

According to the selection of the scaling factor S_L, different defects may occur to the reconstructed patches. In [9] and [10] the mean weighting defect and the noise weighting defect are addressed. If S_L is not inverse proportional to the length L, it will result in unnecessary AC coefficients due to the different energies of the DCT matrices of different lengths as depicted in Fig. 3. Therefore, S_L =4/L is used in [8]. This type of the SA-DCT is referred to nonorthonormal SA-DCT (NO-SA-DCT). Since the concept of the NO-SA-DCT is to scale the DCTs of different lengths to the same length, the quantization noises may be amplified by the cause of this act, which is also noted as the noise weighting defect. Alternatively, $S_L=\sqrt{2/L}$ can be used to solve the noise weighting defect and referred as pseudo-orthonormal SA-DCT (PO-SA-DCT). However, if the mean value of the block is not equal to zero, the mean weighting defect will still exist. In [9], a solution to this trade-off problem is proposed as ΔDC-SA-DCT, using the ΔDC correction to prevent both systematical defects simultaneously.

III. Proposed Method

A. Polynomial Contour Approximation

In our proposed method, the shape coding is based on the polynomial curve approximation algorithm in [16]. The algorithm starts with finding dominant points using weighted curvature values in (8) on each shape contour. Therefore, the whole contour is divided by these dominant points into several curve segments.

$$c_{linear}(n,\tau) = 1 - \frac{distance(n+\tau, n-\tau)}{distance(n, n+\tau) + distance(n, n-\tau)} \quad (7)$$

$$c_{weighted}(n) = \sum_{\tau=1}^{D} w(\tau) c_{linear}(n,\tau) \quad (8)$$

For each segment, the Lagrange polynomial approximation is applied in (9). In [16], Ding et al. suggest that using the third order polynomials is the most proper choice. In addition, since every segment is projected to fixed x and y axis first, it is proved in [16] that we only need to record two coefficients of the estimated polynomial:

$$y_c[\mathrm{n}] \approx \mu[n] = \sum_{k=0}^{3} h_k B_k[n], \quad (9)$$

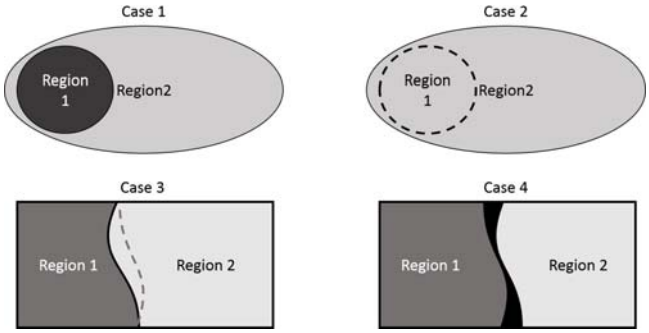


Fig. 4 The region overlapping or emptiness problems.

$$B_k[n] = \prod_{\substack{j=0 \\ j\neq k}}^{3} \left(x_c[n] - \alpha_j\right) \Bigg/ \prod_{\substack{j=0 \\ j\neq k}}^{3} \left(\alpha_k - \alpha_j\right), \ \alpha_j = \frac{d_{ab}}{3} j \quad (10)$$

where h_k is the obtained coefficient.

The solution for h_1 and h_2 are described in (11). For full derivations and subsequent dominant point and coefficient encoding, reader can refer to [16].

$$\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} m_{1,1} & m_{1,2} \\ m_{1,2} & m_{2,2} \end{bmatrix}^{-1} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \quad (11)$$

where $m_{k,\eta}[n] = \sum_{n=1}^{T-1} B_k[n] B_\eta[n]$ and $s_k = \sum_{n=1}^{T-1} B_k[n] \beta[n]$.

B. Pre-Processing

As the overall framework of the proposed algorithm has been provided in Fig. 2, we can view the flow as two main part: pre-processing on image regions and shape-adaptive compression algorithm. In the pre-processing step, the image segmentation is performed by using the algorithm in [20].

Since the division usually consists of fractional small regions, we first merge these small regions with their neighbors. Then, a simplification step using morphological closing and opening are applied to each region to eliminate some disturbances on region boundaries. Subsequently, our method makes use of the efficient contour compression algorithm in [16] to approximate and encode the simplified segmentation results.

After the approximation, a refinement procedure is applied to deal with region overlapping or emptiness in Fig. 7 caused by the approximation. To solve the problem that larger regions lay over smaller inside regions, we reorder the coding sequence by their areas. Therefore, the latter decoded regions are able to directly cover on the previously decoded regions. On the other hand, some pixels may receive multiple or idle labels due to the lossy compression. In such cases, we use voting mechanism from its neighboring pixels.

A pair of examples of the segmentation and approximated outputs are provided in Fig. 1(a) and (b). We can observe that despite some differences, the structural boundaries in the image are still remain approximately the same or even without the noises. As a result, the simplified and approximated region information is forwarded to the main shape-adaptive compression procedure.

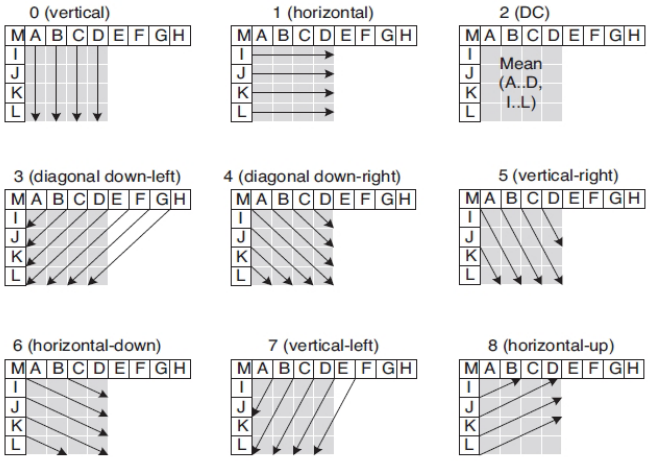


Fig. 5 The nine prediction modes used in the H.264 intra coding and the AIC [6].

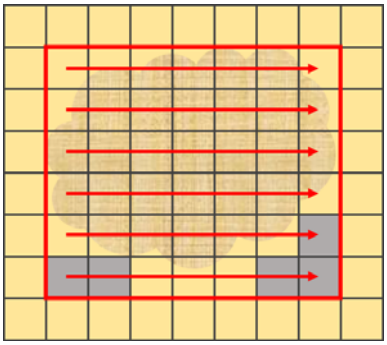


Fig. 6 The scanning procedure in the shape-adaptive prediction. Each square represents a 8×8 block. Blocks outside the red box are the marginal paddings; blocks with grey color will not be processed since no regional pixels are contained.

C. *Color Model*

Different from most compression approaches that convert the original RGB color image into the YC_bC_r color space. In our method, we employ the YC_gC_o color model [15] (12). Besides being a lossless color conversion with respect to the RGB color model, the YC_gC_o also proved to have better decorrelation ability compared to the YC_bC_r.

$$\begin{bmatrix} Y \\ C_g \\ C_o \end{bmatrix} = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ -1/4 & 1/2 & -1/4 \\ 1/2 & 0 & -1/2 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (12)$$

D. *Shape-Adaptive Prediction*

The image prediction step is applied before the transform coding in the H.264 intra coding and the AIC. It is based on the concept that, in natural images, some textures may occur periodically, causing lots of redundancies. Therefore, if we are able to generate a prediction on current block using causal processed blocks, only the differential residual data need to be sent to subsequent procedures. This scheme can help decorrelate the input image in the first place. The proposed shape-adaptive image prediction is extended from the intra

(a) (b)

Fig. 7 The prediction image of the Y channel under 1 bpp scenario. (a) H.264 intra prediction. (b) Proposed shape-adaptive prediction.

TABLE I
COMPARISON OF PREDICTION MODE ENCODING METHODS

	Exp-Golomb Code	Fixed Length Coding	Adaptive Arithmetic Coding
Lena	6295	4710	**4648**
Bamboo	5277	3946	**3856**
Penguin	3681	2810	**2694**

4×4 block prediction in the H.264/AVC standard, which provides nine prediction modes with respect to nine different directions as shown in Fig. 5.

In our shape-adaptive prediction, the arbitrarily shaped regions are first partitioned into 8×8 blocks. Secondly, we pad additional marginal blocks to the region borders and fill the non-region pixels with the regional mean value. Then, the prediction process that uses causal block information to estimate the current block in conventional block-based systems is able to be committed on the blocks inside the red box in Fig. 6. The only difference will be the error measurement while selecting the best prediction. For an incomplete block having a portion of regional and non-regional pixels, we only count the errors of the regional pixels. In [13], Liu and Ngan also suggest a shape-adaptive prediction scheme by assigning 128 to non-regional pixels to perform normal prediction process on regional pixels. However, in our consideration, the padding values should vary with region properties instead of fixed values. Fig. 11 provide a comparison of the proposed shape-adaptive prediction and the H.264 intra prediction of the Y channel in low bitrate scenario. Obviously, the shape-adaptive version is able to give better estimation.

For encoding the prediction modes and the regional means, we use the adaptive arithmetic coding instead of exponential Golomb code or fixed length coding that employed in other prediction scheme. The coding steps are as follows. If the current block share the same prediction mode with the previous one, we encode a symbol '0' to indicate the same mode. Otherwise, a symbol '1' and its mode are encoded using arithmetic coder. Table I shows the encoding results of different approaches.

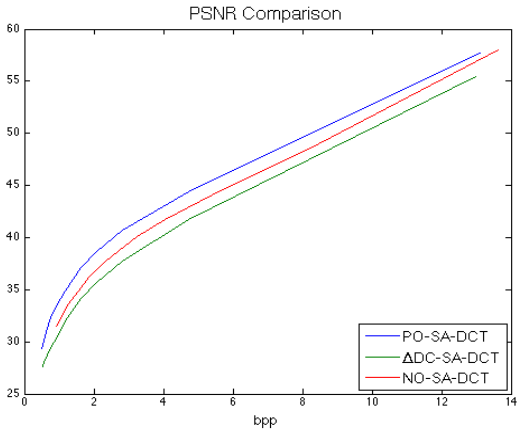
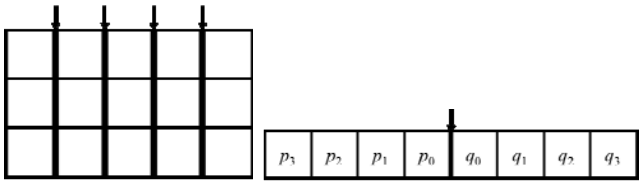


Fig. 8 Comparisons of different SA-DCTs by the PSNR.



(a) (b)

Fig. 9 The vertical block boundaries. (a) The block view. (b) The pixel view.

E. Shape-Adaptive Transform

In section II, we have discussed about the SA-DCT and its different forms. The NO-SA-DCT and PO-SA-DCT have tradeoff between the mean weighting and the noise weighting defects. On the other hand, the ΔDC-SA-DCT is said to be the optimal choice [10] to process "normal" image blocks. However, in our approach, since the image prediction is applied before the transform coding, the mean values in most blocks are zero or near zero. In such a case, the PO-SA-DCT is able to provide better performances. Fig. 8 shows the simulation results of our compression performance using different kinds of SA-DCTs.

F. Quantization and Entropy Coding

In order to achieve the rate control, the quantization step is required to quantify the transformed DCT coefficients. In our method, we follow the concept of the AIC which also deals with prediction residues and use uniform quantization steps. That is, no matter low or high frequencies the coefficients represent, we apply the same level of divisions (13). After the quantization, the zigzag scanning is applied to convert the coefficients into one-dimensional form.

$$C_q(i,j) = \frac{C(i,j)}{Q_{step}} \tag{13}$$

where C and C_q are the transformed and quantized DCT coefficient within a block, and Q_{step} is the quantization step.

As for encoding the resultant quantized coefficients, we employ the context-based adaptive variable-length coding (CAVLC) encoder from the H.264/AVC standard. In addition, some modifications are made to let the CAVLC encoder be able to encode arbitrary length inputs.

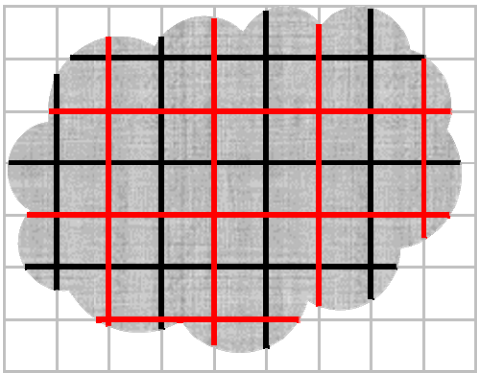


Fig. 10 The shape-adaptive deblocking in the block view. Red borders: macro blocks. Black borders: normal blocks.

Fig. 11 The deblocking results under 0.9 bpp. (a) The original compressed image. (b) The deblocked image.

G. Shape-Adaptive Deblocking

To reduce the blocking artifacts that appear in most compression systems using block partitioning, we further proposed a shape-adaptive deblocking technique based on the in-loop deblocking filter [14]. The debloking process will check every horizontal and vertical block boundaries and use smoothing filters to smoothen the potential blocking artifacts. Fig. 9 shows an example of the vertical block boundaries. Besides, thresholds related to the quantization levels are taken into consideration to prevent degradation on structural edges of the image that happen at the block boundaries. If the difference between the pixels at the boundary is too large it may be the true structural edge instead of additional block edges (14). In addition, depending on the boundary types, borders at macro 8×8 or 16×16 blocks or at normal 4×4 blocks, and the severity of the deformation, filters of different lengths as in (15) will be applied differently. The full decision tree about the filter usage can be found in [14].

$$\begin{cases} |p_0 - q_0| < \alpha \\ |p_1 - p_0| < \beta \\ |q_1 - q_0| < \beta \end{cases} \tag{14}$$

$$\begin{cases} 5-\text{tap on } p_0 \text{ and } q_0 : \{1,2,2,2,1\}/8 \\ 5-\text{tap on } p_2 \text{ and } q_2 : \{2,3,1,1,1\}/8 \\ 4-\text{tap} : \{1,1,1,1\}/4 \\ 3-\text{tap} : \{2,1,1\}/4 \end{cases} \tag{15}$$

where p_i and q_i are pixels from the adjacent blocks as depicted in Fig. 9 (b); α and β are quantization related thresholds.

In our shape-adaptive deblocking process, some features are added. First, as shown in Fig. 10, the smoothing filters are only applied on pixels within the current region even if they share the same block with the non-regional pixels. Second, instead of using the absolute differences in (14), we suggest to use relative difference ratios to check the thresholds (16).

$$\frac{|a-b|}{\max(a,b)} < threshold\ . \tag{16}$$

Third, if the difference of a block boundary is below the threshold but it is not large enough to be perceived, we may skip the smoothing and therefore speed up the deblocking procedure. An additional condition is set to check this scenario. Finally, since the uniform quantization are used in our approach, the adaptive thresholds α and β are designed as

$$\alpha = Q_{step} / c_1\ , \tag{17}$$

$$\beta = Q_{step} / c_2 \tag{18}$$

where c_1 and c_2 are two constant and $c_1 < c_2$.

In summary, the proposed algorithm employs the following techniques. First, instead of using YCbCr color model, the image is converted into YCgCo color space instead. Secondly, the shape-adaptive version of the nine mode prediction is proposed and used. In the transform coding, we discover that for processing prediction residues, the non-optimal PO-SA-DCT is able to give better efficiencies than other SA-DCT approaches. The quantization to the transformed coefficients is perform uniformly following the experiments results from ourselves and the AIC in processing prediction residues. Finally, we propose a shape-adaptive deblocking scheme, which is modified from the in-loop deblocking filter, to reduce the potential blocking artifacts at high compression ratios.

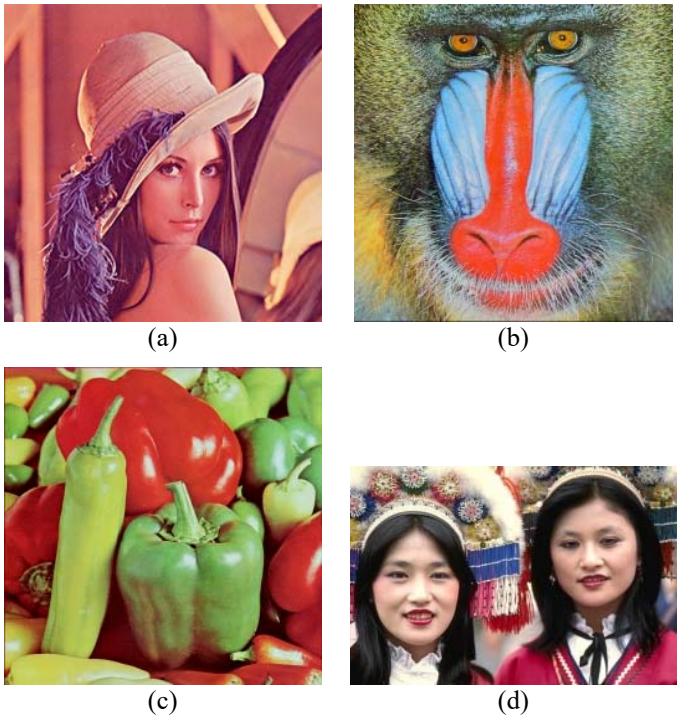
IV. Simulation Result

In this section, we present the compressed images of our algorithm compared to other commonly used approaches and standards. The visual comparison under various bitrates are provided in the first part. Then, we employ commonly used measurements, the peak signal-to-noise ratio (PSNR), and a recent proposed image quality assessment using gradient similarity [22] to measure the objective scores of compressed images.

A. Visual Comparison

Fig. 13-21 show the compression results of the four images depicted in Fig. 12. We would like to focus on the low bitrate scenarios to observe how the different forms of degradations affect the visual qualities. Generally, some additional artifacts start to unveil when the bitrate is below 1 bpp.

The results of the H.264 intra coding degrade the most due to the high quantization and prediction errors. Images from both the JPEG and the H.264 intra coding are suffer from blocking artifacts. In addition, the results of the JPEG tend to eliminate the high frequency component due to its



(a) (b) (c) (d)

Fig. 12 The test images in the following visual comparisons. (a) The "Lena" image. (b) The "baboon" image. (c) The "pepper" image. (d) The "girls" image.

quantization design. However, in the images having lots of details like the "baboon" image, the overall contrast and the small features are deformed. As for the results from the JPEG2000, although they seem to retain most of the details of the original image, there are also additional and noticeable ghost artifacts which are annoying and may mislead the computer analysis of image features.

On the other hand, the shape-adaptively compressed images from our method offer a different kind of degradation at high compression ratios. Thanks to the deblocking scheme and the region-wise processing, the structural edges are able to remain sharp and the main deformation in our method is the smoothing effect in texture areas. Therefore, despite some losses in details, the overall structures of the image are still distinguishable. In our opinion, the degradation form of our algorithm is more reasonable than other approaches.

B. Objective Measurement

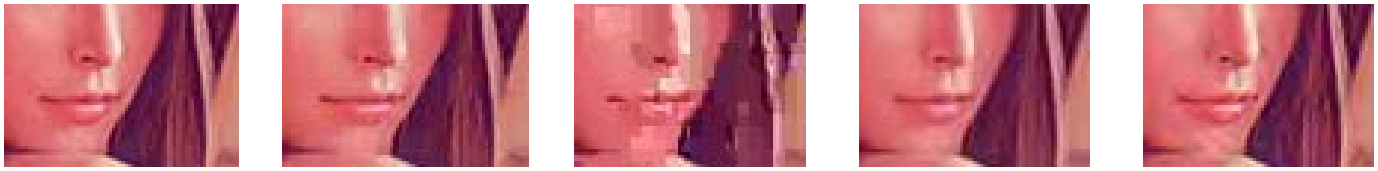
In this subsection, the rate distortion curves of different methods are compared. We employ the PSNR (19) and a recent proposed image quality assessment index based on gradient similarity (GBIQA) [22] as objective quality measurement metrics. Although the PSNR is relatively conventional image quality assessment approaches, it is still referential in some aspects. For the GBIQA, we find this quality index a novel and convincing method for its ability to measure the gradient similarity in a reasonable way. It is worth noting that for color images, the quality assessment is committed in YC_bC_r color space due to the fact that the differences in luminance and chrominance channels

(a) (b) (c) (d)

Fig. 13 The visual comparison of the “lena” image under 1 bpp. (a) The proposed method. (b) H.264 intra coding. (c) JPEG. (d) JPEG2000.

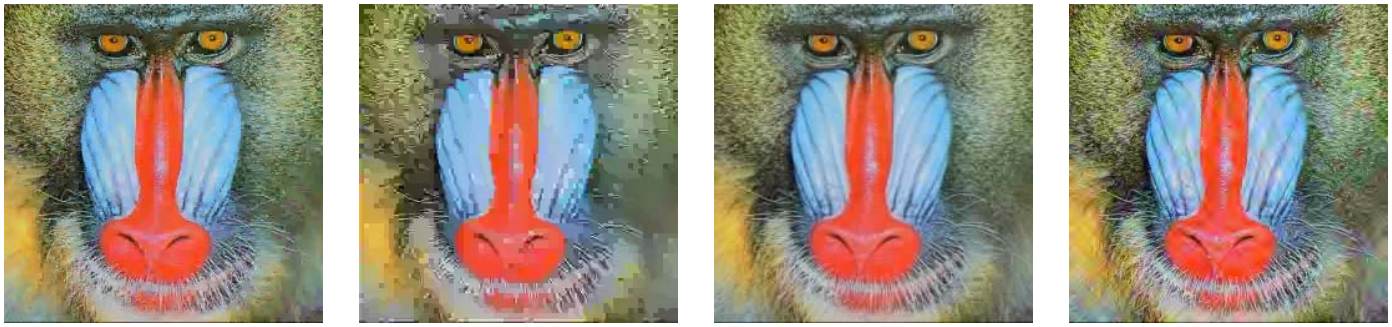
(a) (b) (c) (d)

Fig. 14 The visual comparison of the “lena” image under 0.6 bpp. (a) The proposed method. (b) H.264 intra coding (0.8 bpp). (c) JPEG. (d) JPEG2000.



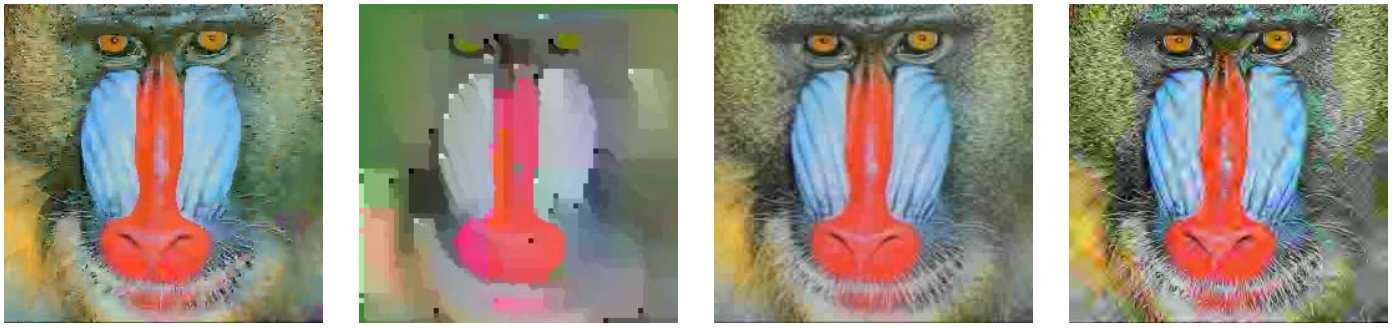
(a) (b) (c) (d) (e)

Fig. 15 The regional crops of the “Lena” image. (a) The original patch. (b) The proposed method. (c) H.264 intra coding. (d) JPEG. (e) JPEG2000.



(a) (b) (c) (d)

Fig. 16 The visual comparison of the “baboon” image under 1 bpp. (a) The proposed method. (b) H.264 intra coding. (c) JPEG. (d) JPEG2000.



(a) (b) (c) (d)

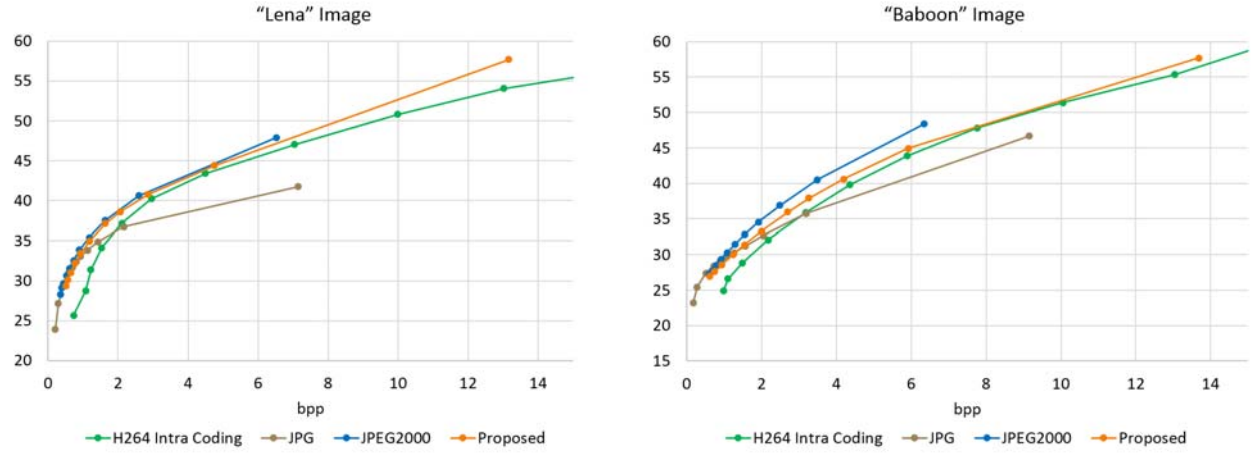
Fig. 17 The visual comparison of the “baboon” image under 0.6 bpp. (a) The proposed method. (b) H.264 intra coding (0.8 bpp). (c) JPEG. (d) JPEG2000.

(a) (b) (c) (d)

Fig. 18 The visual comparison of the “pepper” image under 0.9 bpp. (a) The proposed method. (b) H.264 intra coding (0.8 bpp). (c) JPEG. (d) JPEG2000.

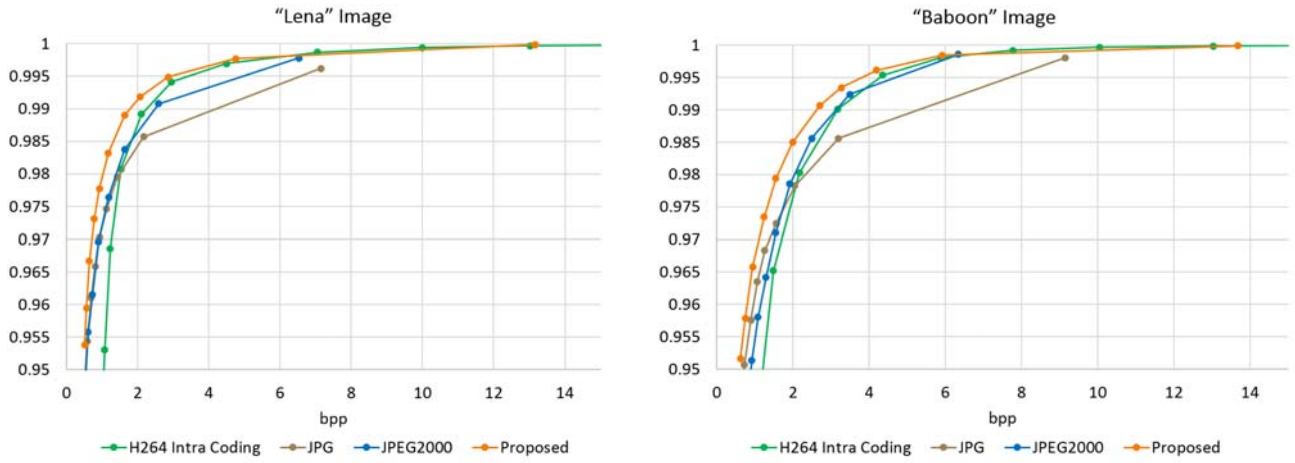
(a) (b) (c) (d)

Fig. 19 The visual comparison of the “girls” image under 1 bpp. (a) The proposed method. (b) H.264 intra coding. (c) JPEG. (d) JPEG2000.



(a) (b)

Fig. 20 The PSNR performances of (a) “Lena” image and (b) “baboon image.



(a) (b)

Fig. 21 The GBIQA [22] performance of the (a) "Lena" image (b) "baboon image.

respectively have stronger physical meaning than the differences in the R, G, and B color channels.

$$PSNR=10\log_{10}\left(\frac{L^2}{MSE}\right) \tag{19}$$

and the *MSE* is defined as

$$MES=\frac{1}{MNC}\sum_{c}\sum_{i}\sum_{j}\left(I_{i,j,c}-I_{ref,i,j,c}\right)^2 \tag{20}$$

where L is the maximum level of pixel values, which is 255 in our case; I and I_{ref} are the compressed and the original images, respectively.

The PSNR performance comparisons of the "Lena" and the "Baboon" images are provided in Fig. 20. From the plots, we can observe that the performance of our method have higher PSNR values than other methods except the JPEG2000, which is the only non-block-based approach and considered as the ceiling of the block-based approaches.

On the other hand, the GBIQA comparison in Fig. 21 suggests a different aspect. Based on gradient similarity measurement, which is closer to the human visual system characteristics, the shape-adaptively compressed images receive higher quality scores with respect to other methods. This result echoes the fact we find in the visual comparison that texture smoothing in shape-adaptive image compression is less disturbing than additional artifacts like blocks or ghosts.

V. Conclusion

In this paper, we proposed a shape-adaptive image compression system using lossy shape coding, shape-adaptive versions of the image prediction, transform coding and deblocking techniques. Different from most shape-adaptive methods with lossless region coding, we find it more efficient if some tolerable errors are allowed to record the shape information. To prevent the label overlapping or emptiness caused by lossy shape coding, we use an additional refinement procedure to retrieve the missing labels. The proposed region-wise image prediction allows us to estimate the image more precisely. In addition, due to the prediction scheme, the non-optimal PO-SA-DCT becomes more useful compared to other SA-DCT forms. While reconstructing the compressed images, we employ the deblocking filter and modify it to shape-adaptive version to reduce the potential blocking artifacts. In subjective visual comparisons, we find the results from the proposed method have better visual qualities and more reasonable degradations at high compression ratios. As for the objective quality measurements, though the PSNR measurements do not have significant performance compared to the JPEG2000, in the quality index measured by gradient similarities, the proposed method achieves the best quality scores among other compared methods in all bitrates.

Acknowledgment

This work was supported by Ministry of Science Technology, under the contract of 104-2221-E-002-097-MY3.

References

[1] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. 18-34, Feb. 1992.

[2] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp.1103-1127, Nov. 2000.

[3] T. Sikora, "MPEG-4 video standard verification model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 19-31, Feb 1997.

[4] J. Ostermann, E. S. Jang, J. S. Shin, and T. Chen, "Coding of arbitrarily shaped video objects in MPEG-4," *IEEE International Conference on Image Processing*, vol. 1, pp.496-499, Oct. 1997.

[5] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560-576, July 2003.

[6] I. E. Richardson, *The H.264 Advanced Video Compression Standard*, 2nd ed. John Wiley & Sons, 2011.

[7] E. van Bilsen, "Advanced image coding." Available: http://www.bilsen.com/aic.

[8] T. Sikora and B. Makai, "Shape-adaptive DCT for generic coding of video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, pp. 59-62, Feb. 1995.

[9] P. Kauff and K. Schuur, "Shape-adaptive DCT with block-based DC separation and ΔDC correction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, pp. 237-242, June 1998.

[10] Y. Zheng, X. Wang, and C. Y. Wang. "Shape-adaptive DCT and its application in region-based image coding," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 7, no. 1, pp. 99-108, 2014.

[11] J. J. Ding, Y. W. Huang, P. Y. Lin, S. C. Pei, H. H. Chen, and Y. H. Wang, "Two-dimensional orthogonal DCT expansion in trapezoid and triangular blocks and modified JPEG image compression," *IEEE Trans. Image Processing*, vol. 22, issue 9, pp. 3664-3675, 2013.

[12] G. Fracastoro, F. Verdoja, M. Grangetto, and E. Magli, "Superpixel-driven graph transform for image compression," *International Conference on Image Processing*, pp. 2631-2635, Sep. 2015.

[13] Q. Liu and K. N. Ngan, "Arbitrarily shaped object coding based on H.264/AVC," *Intelligent Signal Processing and Communication Systems*, pp. 343-346, Jan. 2009.

[14] G. Raja and M. J. Mirza, "In-loop deblocking filter for H.264/AVC video," *International Conference on Signal Processing, Robotics and Automation*, pp. 235-240, Stevens Point, Wisconsin, USA, March 2006.

[15] H. Malvar and G. Sullivan, "Transform, Scaling & Color Space Impact of Professional Extensions," *ITU-T/ISO/IEC JVT Document JVT-H031*, May 2003.

[16] J. J. Ding, C. W. Hsiao, and L. A. Chen, "Advanced contour compression algorithm using weighted curvature, Lagrange curve approximation, and improvement adaptive arithmetic coding," *International Conference on Information, Communications and Signal Processing*, Singapore, Dec. 2015.

[17] ISO/IEC JTC 1/SC 29, *Information Technology – Coded Representation of Picture and Audio Information – Progressive Bilevel Image Compression*, International Organization for Standardization, 1st edition, 1993.

[18] H. Freeman, "Computer processing of line drawing images," *ACM Computing Surveys*, vol. 6, pp. 57-59, March 1974.

[19] R. Baran and A. Kleszcz, "The efficient spatial methods of contour approximation", *Signal Processing: Algorithms, Architectures, Arrangements, and Applications*, pp. 116-121, Sep. 2014.

[20] I. F. Lu, J. J. Ding, and H. Y. Ko, "High accuracy and high robust natural image segmentation algorithm without parameter adjusting," *Computer Vision, Graphics, and Image Processing*, Yilan, Taiwan, Aug. 2015.

[21] Z. Wang, A. C. Bovik, H.R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600-612, April 2004.

[22] A. M. Liu, W. S. Lin, and M. Narwaria, "Image quality assessment based on gradient similarity," *IEEE Tran. Image Processing*, vol. 21, no. 4, pp. 1500-1512, April 2012.