IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 21, 2020, accepted January 2, 2021, date of publication January 6, 2021, date of current version January 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3049465

A Center-to-Edge Progression for Equirectangular Projected 360° JPEG Images

OH-JIN KWON, JOONHYUNG CHO, AND SEUNGCHEOL CHOI
Department of Electrical Engineering, Sejong University, Seoul 05006, Republic of Korea

Corresponding author: Seungcheol Choi (choisc@sju.ac.kr)

This work was supported by the Institute for Information & Communications Technology Promotion (IITP) grant by Korean Government through the Ministry of Science and ICT (MSIT) (Development of the next-generation 360-degree image/video format) under Grant 2017-0-01667.

ABSTRACT Imaging devices capable of capturing 360° images were recently released into the market. The spherical images acquired by imaging devices, such as 360° cameras and multiple camera smartphones, are projected onto the 2D image plane to allow the use of the existing image coding standard. The 360° image market is currently dominated by the equirectangular projection format coded by the JPEG image coding standard. Spherical images are characterized by their large size, and they are typically viewed by displaying a default viewport of the entire image first, followed by navigation by the user. Therefore, progressive image coding capable of transmitting the default viewport first is in high demand. However, the progressive image coding methods proposed thus far, such as quality-based and resolution-based progressive coding, do not support the spatial region-based progressiveness necessary to first display the default viewport. This paper proposes a center-to-edge progressive mode for the JPEG image coding standard, which enables transmittance of the discrete cosine transform blocks including the default viewport first by reordering the blocks' scanning path. This progressive mode was designed to maintain all other encoding structures of the JPEG standard, such that use of the already established toolchain of the JPEG is maintained. Results of the test performed with our proposed mode using eleven 360° images in the equirectangular projection format and six objective image quality parameters show that transmission of default viewports is notably faster than existing JPEG modes and negligible effects on JPEG image coding efficiency.

INDEX TERMS 360° image, JPEG image coding, equirectangular projection, center-to-edge progression.

I. INTRODUCTION

The enhanced availability of multiple image sensor devices, such as 360° cameras and dual camera smartphones to consumers led to the recent widespread use of 360° images in various applications, such as 3D street or scenery viewing services [1]–[4]. Presently, 360° images are typically obtained by capturing multiple images of a scene surrounding a point of view and subsequently stitching them to form a spherical image. The spherical image is then projected to a 2D image plane to enable application of the existing image coding standard. The equirectangular projection (ERP) format [5], [6] coded by the conventional JPEG image coding standard (ISO/IEC 10918–1), denoted as ERP-JPEG in this paper, currently dominates the 360° image market [7] and has been recently standardized by the JPEG working group under the name of JPEG 360 (ISO/IEC 19566–6).

The main differences between these ERP-JPEG images and conventional 2D JPEG (C-JPEG) images are as follows:

1) The size of ERP-JPEG images is typically significantly larger. 4K- (3840 × 2160 pixels) and even 8K- (7680 × 3840 pixels) sized images are already popular in the commercial field.
2) Whereas the typical aspect ratios of C-JPEG images are 4:3, 3:2, and recently 16:9, ERP-JPEG images are generally in a ratio of 2:1.
3) Whereas the entire image is displayed at once in C-JPEG image services, only a part of the ERP-JPEG image is typically unfolded and rendered since the human visual system does not cover the entire 360° field of view. The rendered region is called a viewport, and the default viewport is the region that has to be rendered first.

The associate editor coordinating the review of this manuscript and approving it for publication was Yudong Zhang.

FIGURE 1. Sample ERP-JPEG images: (a) ERP 4K (3840 × 1920), (b) ERP 6K (5760 × 2880), and (c) ERP 8K (7680 × 3840).

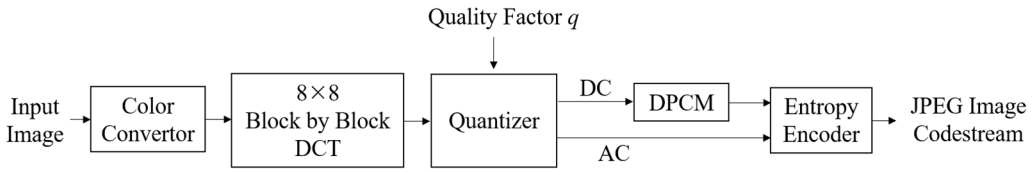

FIGURE 2. Basic encoding structure of JPEG image coding [8].

4) Whereas the sampling intervals in pixels of the C-JPEG image are equally spaced, those of ERP-JPEG are not. Fig. 1 shows sample ERP-JPEG images. The ERP maps latitudes and longitudes to the vertical and horizontal axis, respectively, by spacing them equally on the 2D plane [5], [6]. Therefore, the resulting image is neither equal-area nor conformal. It contains redundant pixels and exhibits horizontal stretching near the pole area.

New coding methods for ERP images, which are more effective than the JPEG image coding, have been recently researched due to these differences [9]. However, JPEG image coding still dominates the ERP image coding market, and it cannot be replaced by another image coding method in the near future, as the JPEG standard has been predominantly used over a long time and established a matured toolchain in the digital imaging industry.

The basic encoding structure of JPEG image coding is shown in Fig. 2. The input image is converted to the YCbCr color coordinate system. Chroma components, Cb and Cr, are optionally downsampled to the 4:2:2 or 4:2:0 type [8]. Each of the Y, Cb, and Cr color components are divided into non-overlapping 8×8 blocks. All the blocks are ordered in a left-to-right and top-to-bottom (LRTB) scanning path. The 8×8 discrete cosine transform (DCT) is applied to each color component and then quantized. Whereas the AC coefficients of each block are losslessly entropy-coded independently, the DC coefficients are encoded first using a differential pulse code modulation (DPCM) method to reduce the redundancy between DC coefficients of consecutive blocks, after which they are entropy-coded. The DC coefficient of a block reflects the averaged pixels on a block-by-block basis, and the redundancy exists in consecutive blocks since adjacent blocks are highly correlated. Lossy compression is performed by a quantizer whose compression rate is controlled by a so-called quality factor denoted as q in Fig. 2.

In this study, we note that the LRTB block scanning path employed in the existing JPEG image coding standard is inconvenient for displaying ERP-JPEG images. When a large ERP-JPEG image transmitted through a band-limited channel is displayed, we must wait until the codestream corresponding to all the blocks covering the default viewport region arrives. We mitigate this inconvenience by providing a progressive mode, which enables transmission of the default viewport region first, using a center-to-edge block scanning path.

This paper is organized as follows: Sect. II describes the proposed progressive mode; Sect. III presents the experimental results showing the JPEG image coding gain influenced by the proposed mode; and Sect. IV presents the conclusions.

II. PROPOSED PROGRESSIVE IMAGE CODING MODE

We propose a center-to-edge progressive mode to improve the C-JPEG standard for ERP-JPEG images. Fig. 3 illustrates the comparison between typical progressive image coding modes and the proposed one. The box in each image denotes the default viewport of each progressive mode displayed as time passes through t_0, t_1, and t_2. The non-progressive mode is the default C-JPEG mode using the LRTB block scanning path. C-JPEG provides the quality progressive mode optionally, by either encoding low-to-high frequency DCT coefficients or employing a coarse-to-fine quantizer [8]. The resolution progressive mode is typically provided by wavelet-based or pyramid-structured image coding methods, such as the JPEG2000 image coding standard (ISO/IEC 15444). The proposed center-to-edge progressive mode is the mode first encoding the default viewport and extending the encoding region to the edge. Compare to the quality-based and the resolution-based progressive mode, the benefit of the proposed mode is that it provides the full-quality default viewport at t_0, whereas all other modes achieve this at t_2.

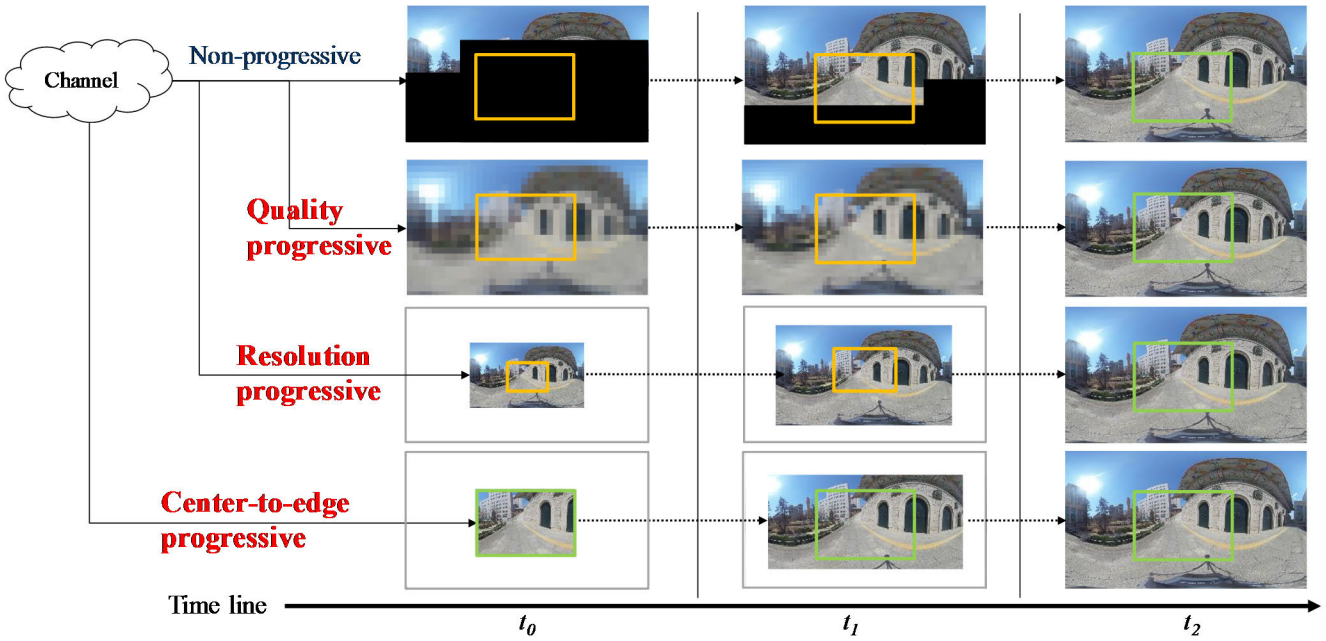

FIGURE 3. Proposed progressive mode compared to existing modes. The box in each image denotes the default viewport. The green box denotes the time at which the full-quality default viewport is displayed in each mode.

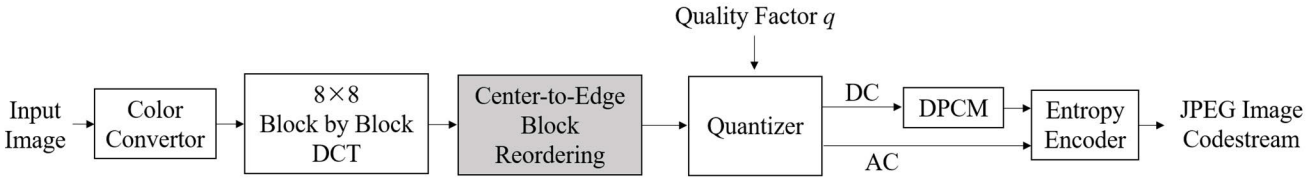

FIGURE 4. Proposed encoding structure of JPEG image coding.

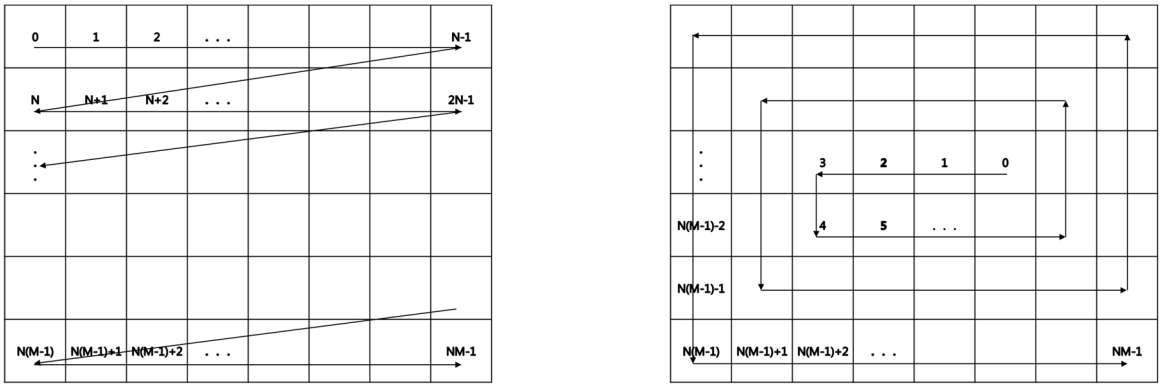

FIGURE 5. Proposed block scanning path compared to LRTB scanning path: (a) LRTB scanning path and (b) proposed scanning path.

We propose a simple reordering method only modifying the LRTB block scanning path used in C-JPEG, as shown in Fig. 4. We keep all other encoding structures of C-JPEG. Thereby, we may guard the merit of C-JPEG, namely the usage of the already established toolchain of C-JPEG. Fig. 5 illustrates our proposed block scanning path compared to the LRTB scanning path. (In this study, we consistently assume that the default viewport is centered, and the block most to the right and bottom corner is the last block in the proposed scanning path. When the default viewport is not

centered, a pre-processing may be applied to the ERP-JPEG image for centering the viewport in the image before the proposed reordering. We may use a method centering the viewport in the image by applying a simple block shifting.) An example of a pseudocode that reorders C-JPEG blocks into the proposed order is listed in the Appendix.

In JPEG image coding, the AC coefficients of each block are encoded independently. Therefore, changing the block scanning path has a negligible effect on the coding performance. However, DC coefficients are DPCM encoded. Therefore, modifying the block scanning path may change the redundancy between consecutive DC coefficients and thus affect coding performance.

In Fig. 6, we compare the redundancy between consecutive DC coefficients of the proposed scanning path to those of the LRTB path. Denoting the DC coefficient of the i-th block by X_i, Fig. 6 exemplifies the scatterplot of X_i and X_{i+1} of Fig. 1(a) for each of Y, Cb, and Cr color component at two typically employed subsampling modes: 4:4:4 and 4:2:0. Notably, the Y component of 4:4:4 subsampling is same as that of 4:2:0 subsampling in JPEG image coding, hence we only depict the scatterplot in the case of 4:4:4 subsampling. The difference in the scatterplots between two scanning paths is negligible for all cases. Therefore, we may conclude that the redundancy between consecutive DC coefficients of the proposed scanning path is comparable to that of the LRTB scanning path.

Fig. 6 also compares the correlation coefficients (denoted by r) of consecutive DC values of two scanning paths for the ERP-JPEG image. Notably, the LRTB scanning path has only one direction, namely left-to-right, and illustrates the discontinuity between end points: the last block in one line, and the first block in the following line, as shown in Fig. 5(a). In contrast, the proposed scanning path has four directions: left-to-right, right-to-left, up, and down directions, and it has no discontinuous blocks, as shown in Fig. 5(b). Therefore, it is expected that the proposed scanning path would exhibit relatively less redundancy between consecutive DC coefficients than the LRTB path for horizontally correlated images and more redundancy for vertically correlated images. (Fig. 1(a) depicts the case of a more horizontally correlated image.) Nevertheless, we may conclude that the differences in the correlations of consecutive DC coefficients between proposed and LRTB scanning paths are almost negligible. Therefore, the coding performance difference between two scanning paths is likewise expected to be almost negligible.

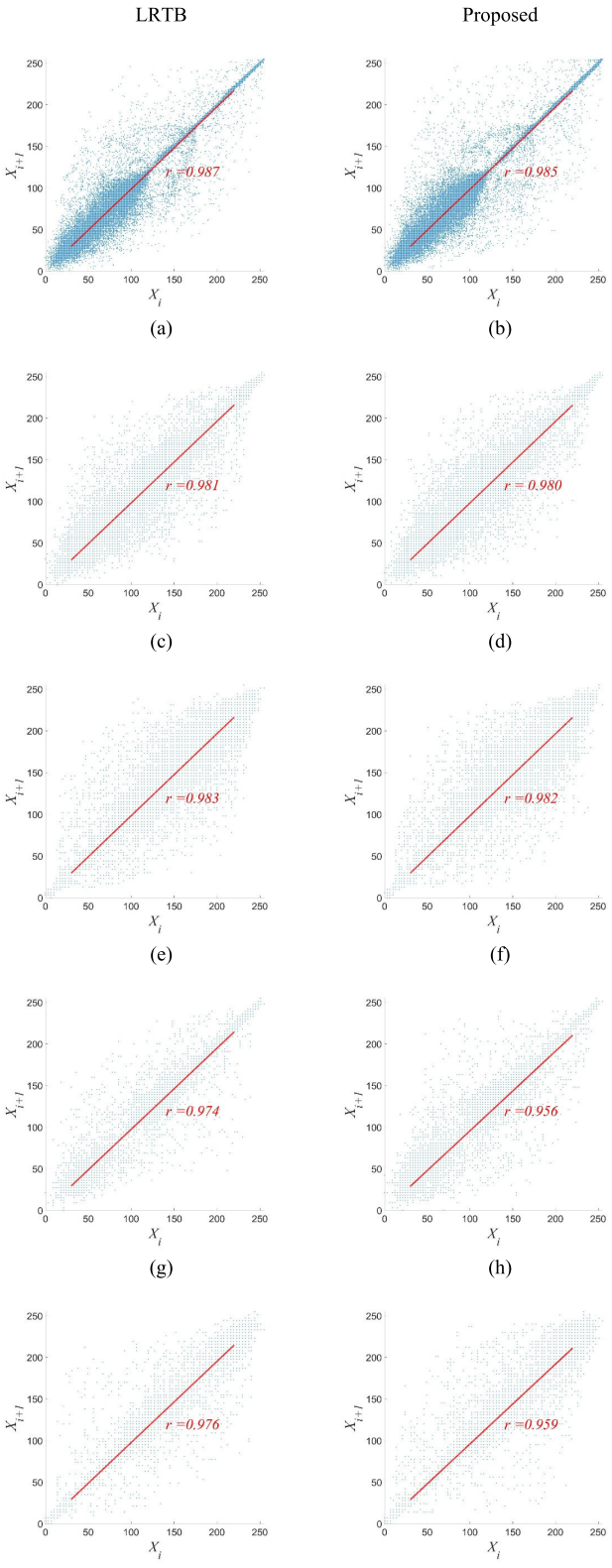

FIGURE 6. DC Scatterplot: 4:4:4 subsampling of (a) LRTB Y, (b) proposed Y, (c) LRTB Cb, (d) proposed Cb, (e) LRTB Cr, and (f) proposed Cr, and 4:2:0 subsampling of (g) LRTB Cb, (h) proposed Cb, (i) LRTB Cr, and (j) proposed Cr.

III. EXPERIMENTAL RESULTS

A. INFLUENCE OF PROPOSED PROGRESSIVE MODE ON JPEG IMAGE CODING PERFORMANCE

We compare the JPEG image coding performance using the proposed progressive mode with that obtained with the C-JPEG coding. Eleven sample images were chosen for our test, three of which are shown in Fig. 1. The remaining eight images are shown in Fig. 7. They were captured by authors and chosen to represent ERP images of various size, brightness, colorfulness, and amount of detail.

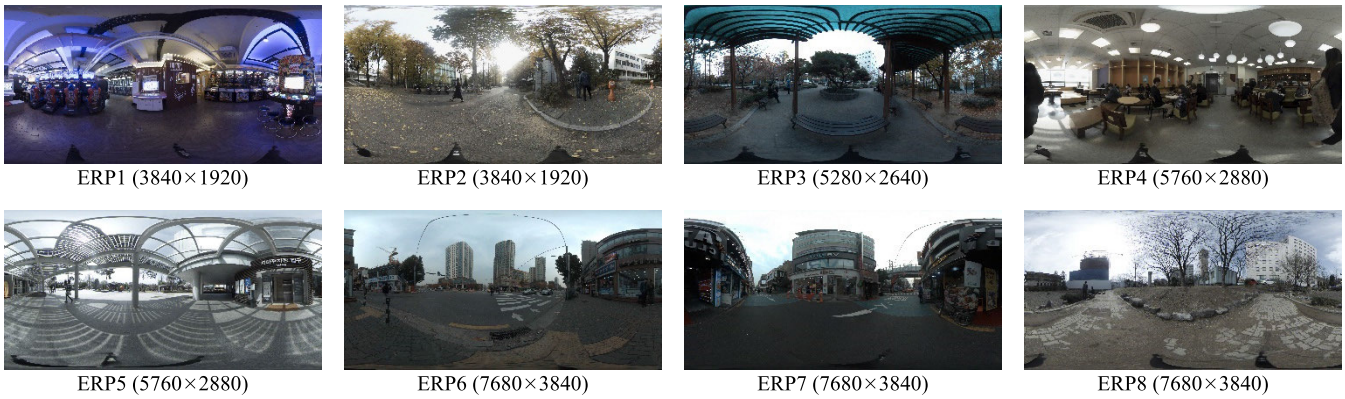

FIGURE 7. Eight out of eleven sample images selected for our test. The first three images are shown in Fig.1.

We perform the comparison using objective image quality metrics that measure the distortion between the original and coded images. We selected six metrics, which include mostly adopted metrics in the field of image coding research and additional metrics recently recommended by the JPEG working group [10], [11]. These comprise the following: 1) PSNR [12] calculated for color images [13]; 2) structural similarity index metric (SSIM) [14] focusing on the structural image difference; 3) multi-scale SSIM (MS-SSIM) [15] computing the SSIM over different resolutions to represent the quality at different resolutions and viewing conditions; 4) feature-similarity index metric (FSIM) [16] using the phase congruency and gradient magnitude to assess local image quality; 5) visual information fidelity (VIF) [17] using natural image statistics exploiting characteristics of the human visual system; and 6) CIEDE2000 [18] calculating the difference between two colors in the CIELab color space.

Our test results are shown in Fig. 8. We performed our test for the JPEG image coding quality factors of 90, 85, 80, 75, 70, and 65. Notably, PSNR and CIEDE2000 metrics count chroma components for their measurement, whereas SSIM, MS-SSIM, FSIM, and VIF metrics use only the luminance component for their measurement. Therefore, we provided two chroma subsampling cases only for PSNR and CIEDE2000 metrics in Fig. 8. The results shown in Fig. 8 indicate that the differences in the coding performance of LRTB and proposed scanning path are negligible for all cases of sample images, employed metrics, subsampling types, and JPEG quality factors. Furthermore, we also performed the pre-processing for centering the default viewport before encoding and inversely converted it to the original ERP image after decoding, and then measured the coding performance. We found that the influence of the pre-processing is also negligible.

B. COMPARISON TO JPEG PROGRESSIVE MODE

In order to quantify the improvement of the proposed scanning mode compared to existing JPEG non-progressive and progressive modes, we established a simple HTTP server providing image services whose client is the web browser displaying 360-degree images. The client requests downloading JPEG images over an emulated 15Mbps network corresponding to typical 4G service speed, decodes the downloaded codestream block by block, and displays each block immediately when it becomes available on the client side. We measured the delay time to display the full-quality default viewport thoroughly. The viewport was set to 75° × 75° whose corresponding sizes in pixels are 1816 × 1816, 1336 × 1336, and 856 × 856 for 8K, 6K, and 4K sample images,

TABLE 1. Delay times for displaying full-quality default viewports. (Units are given in seconds).

Modes	Images	Quality Factors					
		65	70	75	80	85	90
Non-progressive JPEG	ERP1	0.39	0.43	0.48	0.53	0.64	0.78
	ERP2	0.73	0.76	0.83	0.9	1.06	1.14
	ERP3	0.81	0.85	0.92	1.06	1.17	1.5
	ERP4	0.64	0.69	0.75	0.88	1.02	1.15
	ERP5	0.87	0.95	1.04	1.18	1.17	1.16
	ERP6	0.99	1.08	1.17	1.26	1.23	1.98
	ERP7	0.93	0.99	1.08	1.25	1.22	1.8
	ERP8	1.26	1.26	1.89	2.12	2.22	2.16
	Average	*0.83*	*0.88*	*1.02*	*1.15*	*1.22*	*1.46*
Progressive JPEG	ERP1	0.56	0.56	0.59	0.66	0.79	0.96
	ERP2	0.82	0.89	0.94	1.04	1.21	1.33
	ERP3	1.01	1.09	1.17	1.3	1.45	1.72
	ERP4	0.90	0.93	1.05	1.14	1.33	1.54
	ERP5	1.21	1.26	1.34	1.47	1.51	1.86
	ERP6	1.43	1.5	1.75	1.79	2.34	2.47
	ERP7	1.36	1.52	1.61	1.69	1.59	2.43
	ERP8	2.37	2.24	2.42	2.59	2.5	3.47
	Average	*1.21*	*1.25*	*1.36*	*1.46*	*1.59*	*1.97*
Proposed	ERP1	0.28	0.34	0.35	0.34	0.37	0.39
	ERP2	0.37	0.35	0.36	0.38	0.41	0.42
	ERP3	0.37	0.37	0.38	0.45	0.43	0.46
	ERP4	0.35	0.36	0.37	0.38	0.4	0.43
	ERP5	0.38	0.42	0.38	0.42	0.45	0.47
	ERP6	0.62	0.67	0.65	0.68	0.72	0.81
	ERP7	0.61	0.63	0.65	0.67	0.73	0.78
	ERP8	0.67	0.69	0.73	0.8	0.89	0.98
	Average	***0.46***	***0.48***	***0.48***	***0.52***	***0.55***	***0.59***

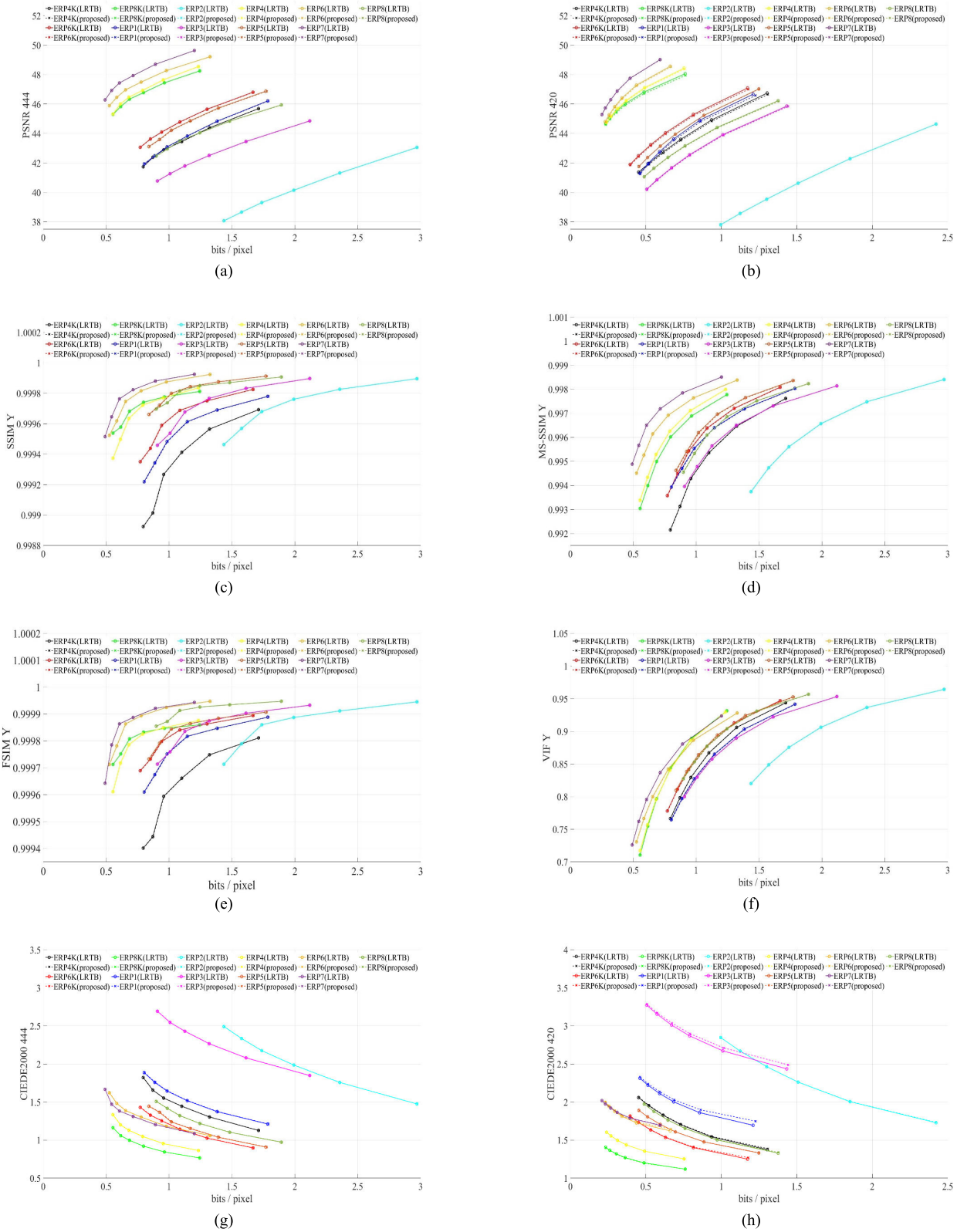

FIGURE 8. Comparison of coding performances of LRTB and proposed scanning path: (a) PSNR with 4:4:4 subsampling, (b) PSNR with 4:2:0 subsampling, (c) SSIM Y, (d) MS-SSIM Y, (e) FSIM Y, (f) VIF Y, (g) CIEDE2000 with 4:4:4 subsampling, and (h) CIEDE2000 with 4:2:0 subsampling.

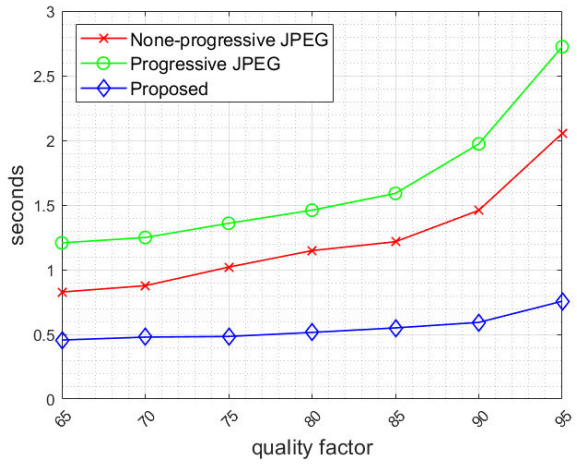

FIGURE 9. **Average delay times for displaying full-quality default viewports.**

respectively [19]. It is noted that, in our experiments, both the time for signaling between the server and the client and the time due to the client's processing time for displaying are negligibly short compared to the delay time to transmit the image.

The results compared to JPEG non-progressive and progressive modes are given in Table 1 and Figs. 9 and 10. Table 1 shows the delay times for displaying full-quality default viewports in each mode when the original image is JPEG encoded with variable quality factors. Fig. 9 shows the delay times averaged over sample images. Fig. 10 illustrates an example of viewport images displayed over time. It is noted that we measured the delay time for displaying the full-quality default viewport, whereas JPEG progressive mode provides quality progressiveness. Therefore, the JPEG progressive mode delay in Table 1 is longer than JPEG non-progressive mode, whereas the low-quality viewport of JPEG progressive mode is displayed earlier than JPEG non-progressive mode in Fig. 10.

We may conclude that the proposed mode is about two times faster than existing JPEG modes for transmitting full-quality $75° \times 75°$-sized default viewports and it will be faster for the smaller-sized default viewports.

Finally, we also evaluate the computational complexity of block reordering for the proposed scanning path. The execution times of reordering were calculated on a platform powered by quad-core 3.6-GHz CPU and 20-GB RAM. Table 2 lists the execution times for our sample ERP images

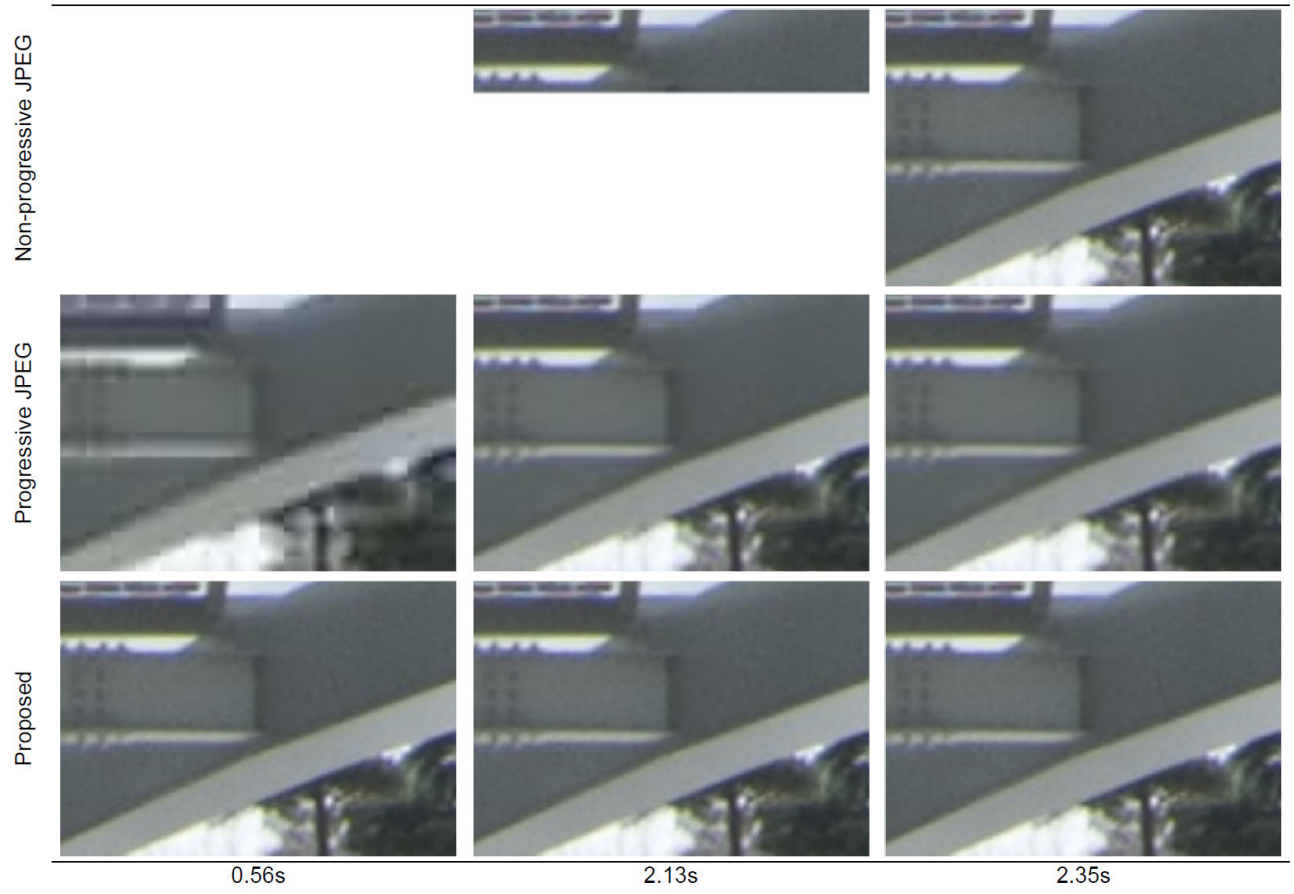

FIGURE 10. **Example of viewport images displayed over time: JPEG non-progressive mode, JPEG progressive mode, and proposed mode captured at 0.56s, 2.13s, and 2.35s, respectively, for the ERP5 sample image encoded with quality factor 95. A part of the viewport is displayed for showing visual differences.**

TABLE 2. Execution times for reordering by proposed scanning path. (Units are given in seconds).

Image size	Averaged execution time	
	4:4:4	4:2:0
3840×1920	0.112	0.104
5280×2640	0.205	0.207
5760×2880	0.243	0.248
7680×3840	0.428	0.437

with two chroma subsampling types: 4:4:4 and 4:2:0. The computing time is highly dependent on the size of the image and increases with an increase in the input image size.

IV. CONCLUSION

A new progressive mode for the JPEG image coding is proposed, in particular for ERP-JPEG images. ERP-JPEG images are characterized in that a default viewport as part of the image is displayed first as they are serviced, whereas C-JPEG image services typically display the entire image at once. The proposed mode encodes the default viewport first employing center-to-edge reordering of blocks. This mode maintains all other encoding structures of C-JPEG, such that the already established toolchain of C-JPEG may be used. We performed the test to determine whether our proposed mode degrades the coding efficiency of C-JPEG and conclude that the influence of the proposed mode on C-JPEG image coding efficiency is negligible.

In summary, the progressive mode proposed in this study was accepted as a part of the methods improving the current JPEG 360 standard in the JPEG working group, and the amendment of JPEG 360 based on the proposed mode is being processed [20]. We further note that the proposed progressive mode is accepted for the next-generation JPEG image coding standard, JPEG XL, which is being standardized for general images and thus not restricted to 360° images [21], [22].

Procedure Rearranging(*C-JPEG_DCT_blocks*)
Input: *C-JPEG_DCT_blocks*
Output: *Rearranged_DCT_blocks*

$N \leftarrow$ number of blocks in width
$M \leftarrow$ number of blocks in height
if $N < M$
  Rotate *C-JPEG_DCT_blocks* 90° clockwise

// Set the position of starting block
if M is even then
  $N_s \leftarrow \mathrm{N} - (\mathrm{M}/2)$
  $M_s \leftarrow (\mathrm{M}/2) - 1$
else
  $Ns \leftarrow Ms \leftarrow (\mathrm{M} - 1)/2$
// N_s is the horizontal position of starting block
// M_s is the vertical position of starting block

Call procedure MakeLUT(*LUTN*, *LUTM*, N_s, M_s)
// *LUTN* contains horizontal index of rearranged blocks
// *LUTM* contains vertical index of rearranged blocks

$idx \leftarrow 0$ // index of *LUTM* and *LUTN*
for $i \leftarrow 0 \ldots (M-1)$ do
  for $j \leftarrow 0 \ldots (N-1)$ do
    Rearranged_DCT_blocks $(i, j) \leftarrow$ *C-JPEG_DCT_blocks* (*LUTN*[*idx*], *LUTM*[*idx*])
    $idx \leftarrow idx + 1$
End of Rearranging

where

Procedure MakeLUT(*LUTN*, *LUTM*, N_s, M_s)
$diffNM \leftarrow (N - M)$
$LUTN[0] \leftarrow N_s$
$LUTM[0] \leftarrow M_s$
$num \leftarrow 1$ // number of runs
$idx \leftarrow 0$ // index of *LUTM* and *LUTN*

for $i \leftarrow 0 \ldots (M-1)$ do
  $N_h \leftarrow num + diffNM$
  $N_v \leftarrow num$

// N_h is the number of horizontal runs
// N_v is the number of vertical runs
for $j \leftarrow 0 \ldots (N_h - 1)$ do
  if M is even then
    if *num* is odd then
      $LUTN[idx+1] \leftarrow LUTN[idx] - 1$
    else
      $LUTN[idx+1] \leftarrow LUTN[idx] + 1$
  else
    if *num* is odd then
      $LUTN[idx+1] \leftarrow LUTN[idx] + 1$
    else
      $LUTN[idx+1] \leftarrow LUTN[idx] - 1$
  $LUTM[idx+1] \leftarrow LUTM[idx]$
  $idx \leftarrow idx + 1$
for $j \leftarrow 0 \ldots (N_v - 1)$ do
  $LUTN[idx+1] \leftarrow LUTN[idx]$
  if M is even then
    if *num* is odd then
      $LUTM[idx+1] \leftarrow LUTM[idx] + 1$
    else
      $LUTM[idx+1] \leftarrow LUTM[idx] - 1$
  else
    if *num* is odd then
      $LUTM[idx+1] \leftarrow LUTM[idx] - 1$
    else
      $LUTM[idx+1] \leftarrow LUTM[idx] + 1$
  $idx \leftarrow idx + 1$
  $num \leftarrow num + 1$
End of MakeLUT

APPENDIX

This appendix introduces an example of a pseudo code reordering C-JPEG blocks into the proposed order.

ACKNOWLEDGMENT

The authors would like to thank Yasiriqbal for his valuable contribution to the experiments.

REFERENCES

[1] S. Yang, J. Zhao, T. Jiang, J. Wang, T. Rahim, B. Zhang, Z. Xu, and Z. Fei, "An objective assessment method based on multi-level factors for panoramic videos," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, St. Petersburg, FL, USA, Dec. 2017, pp. 1–4.

[2] C. Timmerer, M. Graf, and C. Mueller, "Adaptive streaming of VR/360-degree immersive media services with high QoE," in *Proc. NAB Broadcast Eng. IT Conf.*, Las Vegas, NV, USA, 2017.

[3] A. T. Nasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash, "Adaptive 360-degree video streaming using layered video coding," in *Proc. IEEE Virtual Reality (VR)*, Los Angeles, CA, USA, 2017, pp. 347–348.

[4] Y. Yang, B. Jenny, T. Dwyer, K. Marriott, H. Chen, and M. Cordeil, "Maps and globes in virtual reality," *Comput. Graph. Forum*, vol. 37, no. 3, pp. 427–438, Jun. 2018.

[5] J. P. Snyder, "Flattening the Earth: Two thousand years of map projections," *Isis*, vol. 85, no. 3, pp. 488–489, Sep. 1994.

[6] Y. Ye, *Algorithm Descriptions of Projection Format Conversion and Video Quality Metrics in 360Lib Version 5*, document JVET Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 8th Meeting, JVET-H1004, Macao, CN, Oct. 2017.

[7] (2018). *Final Call for Proposals on JPEG 360 Metadata*. [Online]. Available: https://jpeg.org/items/20180212_cfp_jpeg360.html

[8] W B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*. New York, NY, USA: Van Nostrand Reinhold, 1992.

[9] (2018). *Next-Generation Image Compression (JPEG XL) Final Call for Proposals*. [Online]. Available: https://jpeg.org/items/20180423_cfp_jpeg_xl.html

[10] J. Wassenberg and J. Sneyers, *Core Experiments 6 for JPEG XL*, document JPEG (ISO/IEC JTC 1/SC 29/WG 1) 86th Meeting, N86047, Sydney, Australia, Jan. 2020.

[11] J. Ascenso, P. Akayzi, M. Testolina, A. Boev, and E. Alshina, *Performance Evaluation of Learning Based Image Coding Solutions and Quality Metrics*, docuemnt JPEG (ISO/IEC JTC 1/SC 29/WG 1) 85th Meeting, N85013, San Jose, USA, Nov. 2019.

[12] D. Salomon, *Data Compression: The Complete Reference*. New York, NY, USA: Springer-Verlag, 2004.

[13] S. Choi, O.-J. Kwon, J. Lee, and Y. Kim, "A JPEG backward-compatible image coding scheme for high dynamic range images," *Digit. Signal Process.*, vol. 67, pp. 1–16, Aug. 2017.

[14] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[15] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th IEEE Asilomar Conf. Signals, Syst. Comput.*, Asilomar, CA, USA, Nov. 2003, pp. 1398–1402.

[16] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.

[17] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Montreal, QC, Canada, May 2004, pp. 430–444.

[18] G. Sharma, W. Wu, and E. N. Dalal, "The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations," *Color Res. Appl.*, vol. 30, no. 1, pp. 21–30, 2005.

[19] J. Boyce, E. Alshina, A. Abbas, and Y. Ye, *JVET Common Test Conditions and Evaluation Procedure for 360° Video*, docuemnt JVET Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 7th Meeting, JVET-G1030v2, Torino, IT, Jul. 2017.

[20] S. Choi, A. Kuzma, and O. Kwon, *WD for ISO/IEC 19566-6 AMD1: Addition of New JPEG 360 Image Types and Accelerated ROI Rendering*, document JPEG (ISO/IEC JTC 1/SC 29/WG 1) 84th Meeting, N84012, Brussels, Belgium, Jul. 2019.

[21] J. Wassenberg and J. Sneyers, *JPEG XL Use Cases and Requirements*, document JPEG (ISO/IEC JTC 1/SC 29/WG 1) 82th Meeting, N82009, Lisbon, Portugal, Jan. 2019.

[22] J. Sneyers and J. Wassenberg, *JPEG XL Committee Draft*, document JPEG (ISO/IEC JTC 1/SC 29/WG 1) 84th Meeting, N84043, Brussels, Belgium, Jul. 2019.

OH-JIN KWON received the M.S. degree in electrical engineering from the University of Southern California, Los Angeles, in 1991, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 1994. He worked as a Researcher with the Agency for Defense Development, South Korea, from 1984 to 1989, and operated as the Head of the Media Lab, Samsung SDS Company Ltd., Seoul, South Korea. Since 1999, he has been a Faculty Member with Sejong University, Seoul, where he is currently a Professor. His research interests include image and video fusion, coding, watermarking, analyzing, and processing.

JOONHYUNG CHO is currently pursuing the M.S. degree in electrical engineering with Sejong University, Seoul, South Korea. His research interests include image processing and JPEG.

SEUNGCHEOL CHOI received the B.S. and M.S. degrees in computer science from Sejong University, Seoul, South Korea, in 1998 and 2001, respectively, and the Ph.D. degree in electronic engineering from Sejong University, in 2017. From 2001 to 2013, he worked as a Researcher with Galaxia Communications. Since 2017, he has been a Research Fellow with Sejong University. His research interests include image and video coding, high-dynamic range imaging, image processing, image fusion, and JPEG.