

Multi-resolution Lossless Image Compression for Progressive Transmission and Multiple Decoding Using an Enhanced Edge Adaptive Hierarchical Interpolation

Yenewondim Biadgie¹, Min-sung Kim², and Kyung-Ah Sohn^{1*}

¹Department of Software and Computer Engineering, Ajou University
Suwon, 16499 - Rep.of Korea

[e-mail: wondim@ajou.ac.kr, kasohn@ajou.ac.kr]

²Department of Software, Ajou University
Suwon, 16499 - Rep.of Korea

[e-mail: kimmsql@ajou.ac.kr]

*Corresponding author: Kyung-Ah Sohn

*Received October 19, 2016; revised April 19, 2017; accepted August 9, 2017;
published December 31, 2017*

Abstract

In a multi-resolution image encoding system, the image is encoded into a single file as a layer of bit streams, and then it is transmitted layer by layer progressively to reduce the transmission time across a low bandwidth connection. This encoding scheme is also suitable for multiple decoders, each with different capabilities ranging from a handheld device to a PC. In our previous work, we proposed an edge adaptive hierarchical interpolation algorithm for multi-resolution image coding system. In this paper, we enhanced its compression efficiency by adding three major components. First, its prediction accuracy is improved using context adaptive error modeling as a feedback. Second, the conditional probability of prediction errors is sharpened by removing the sign redundancy among local prediction errors by applying sign flipping. Third, the conditional probability is sharpened further by reducing the number of distinct error symbols using error remapping function. Experimental results on benchmark data sets reveal that the enhanced algorithm achieves a better compression bit rate than our previous algorithm and other algorithms. It is shown that compression bit rate is much better for images that are rich in directional edges and textures. The enhanced algorithm also shows better rate-distortion performance and visual quality at the intermediate stages of progressive image transmission.

Keywords: Hierarchical prediction, multi-resolution image coding, multiple decoding, lossless image coding, progressive image transmission, pyramid data structure

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT, and Future Planning (MSIP) [NRF-2014R1A1A3051169, NRF-2016R1D1A1B03933875]

1. Introduction

Data compression reduces the transmission time across a low bandwidth connection. Based on image transmission methods, image encoding schemes can be classified as sequential and hierarchical. In a sequential scheme, an image is transmitted as a sequence of rows of an image from top to bottom and from left to right as shown in Fig. 1. When a large image is transmitted sequentially across a low bandwidth connection, users should wait for a long time in order to inspect the image visually in different IP-based applications, such as browsing World Wide Web visually, interactive visual search from a remote data base, and remote surveillance. Hence, a hierarchical image coding system is designed to support these applications by transmitting the image progressively [1].

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51

Fig. 1. Sequential image coding and transmission scheme for a 2-dimentional image representation with a spatial resolution of 3 by 17.

A hierarchical image coding transforms an image into a hierarchical representation that embeds a base layer and multiple detail or enhanced layers as shown in Fig. 2(a). Since there is a redundant information between consecutive levels of the hierarchy, only incremental information is stored to build the content of the next detail level. This coding scheme generates an embedded bit stream by encoding an image into a single file as a layer of bit streams as shown in Fig. 2(b). Each level of the hierarchy is encoded, transmitted and decoded independently. An approximate version of the original image is transmitted first. Then, the quality of the image is progressively improved by adding only the incremental information over a number of successive transmissions. This gives the user different previews of the original image. This encoding method can reduce the transmission time by terminating the transmission at any time when the intermediate quality is satisfactory for the user. In this encoding method, the spatial resolution or the quality of an image is updated progressively. Hence, hierarchical encoding methods can be divided into two types: progressive resolution (multi-resolution) encoding and progressive quality (multi-quality) encoding [1].

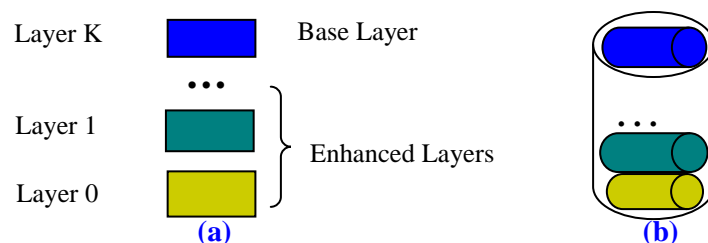


Fig. 2. Hierarchical image coding and transmission scheme; (a) hierarchical data structure with different detail levels; (b) an embedded bit stream file generated by a hierarchical image encoder as a layer of bit streams.

In progressive quality image coding, the spatial resolution of each layer of the hierarchical data structure is identical to that of the original image as shown in Fig. 2(a). However, the

precision of each pixel value increases across the layer of the hierarchy from top to bottom. Initially, the lower precision form of an image is transmitted. In subsequent stages, the precision is gradually increased until the image reaches the full precision level. For example, the bit plane coding of an image transmits each binary digit of each pixel from the most significant bit to the least significant bit. Since the spatial resolution of a reconstructed image is the same across all the layers of the hierarchical data structure, progressive quality image coding technique is also called single resolution progressive image encoding technique.

On the other hand, in progressive resolution encoding, the spatial resolution of each layer increases across the layer of the hierarchy from top to bottom [2]. This coding scheme uses a special type of hierarchical data structure called pyramidal data structure [3]. However, the precision of each pixel value is the same across all the layers of the pyramidal data structure. Hence, an image with reduced resolution is transmitted first and displayed at a small size. The information to obtain a high-resolution image from a lower resolution image is transmitted progressively.

In a multiple decoding environment, each decoder has different capabilities ranging from a handheld device (with a low bitrate network connection and a low-resolution display) to a PC with a high bitrate connection and a high definition display. Consequently, an image is transmitted to heterogeneous decoders by adapting the size of the image according to their own capabilities [4, 5, 6] as shown in Fig. 3. If a single resolution encoding method is used, the encoder should generate and store several files with different sizes for each decoder. This, in-turn, takes more storage space on the encoder.

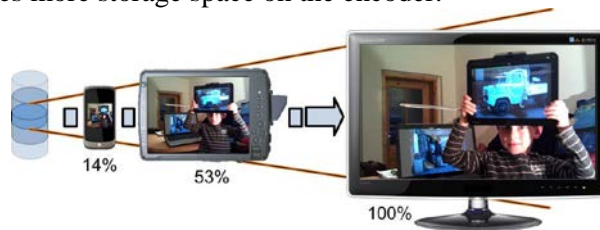


Fig. 3. Multi-resolution image coding scheme to deliver the image by adapting the content of the image to different viewing devices in multiple decoding environment [5].

Image compression methods are usually classified as lossy and lossless. In a lossless image compression method, every single bit of the image remains the same after the image file is uncompressed. Hence, all the information is completely retained. However, lossy compression reduces the original file by permanently removing certain information. In medical imaging and remote sensing applications, the volume of the data is huge for practical storage or transmission, and hence a lossy image coding method is needed. However, legally, medical images should be compressed in lossless mode [7, 8, 9]. The diagnostic information must be stored in the same state as during the initial diagnosis stage to allow their reconsideration in case of judicial proceedings. For example, if some losses are created when a radiological image is compressed in a lossy image coding scheme, the creation of artifacts may seriously mislead the radiologist. Therefore, for ethical reasons, medical images must be stored without any loss for medical professionals to make clinical diagnosis with accuracy. Any minor distortion introduced by lossy compression may lead to serious consequences to patients.

In medical imaging applications, to deal with the dilemma faced between the lossy and lossless versions of a sequential image-coding scheme, a hierarchical or progressive image-coding scheme is used as a compromise. This encoding scheme generates an embedded bit stream from which various rate-distortion levels can be obtained progressively. Hence, users can get progressively reconstructed images ranging from lossy to lossless. In the past, the

focus of the research community was to get higher compression efficiency by designing lossy and sequential image compression algorithms instead of hierarchical image compression algorithms. However, currently, hierarchical image coding scheme is a very hot research topic in tele radiology, where a physician typically requests portions of an image at increased quality including lossless construction while accepting initial renderings and unimportant portions at lower quality, and thus reducing the overall bandwidth requirements. Hence, lossless multi-resolution image coding is very important to access huge volume of biomedical images remotely.

The multi-resolution encoding scheme addresses both the bandwidth limitation of communication systems and the resource limitation of heterogeneous decoding devices simultaneously [6]. To address these problems further, in our previous work [10], we proposed an edge adaptive hierarchical interpolation (EAHINT) algorithm for lossless image coding and progressive transmission. In this paper, we extend it by improving its compression bit rate and visual quality at the intermediate stages during progressive transmission. This is achieved by adding the following three major components as described below.

First, the prediction accuracy in EAHINT is improved by refining the predicted value using context-conditioned error modeling technique as a feedback. The conditional probability distribution of prediction errors is modeled by a two-sided geometric distribution. However, the center of this distribution is shifted from its zero center by offset δ due to inaccuracy of the initial prediction. This implies that the distribution has a bias parameter δ . By estimating the bias cancelation parameter within each context, the accuracy of the initial prediction is improved. Context modeling of prediction errors, refinement of initial prediction and estimation of bias cancelation parameter are described in detail in [sections 4.2 and 4.3](#).

After removing inter-pixel redundancy by refining the initial predictor, there is still data redundancy among prediction errors within the context in terms of their arithmetic sign. This redundancy can be reduced by flipping the sign of prediction errors using the sign of the bias cancelation parameter δ of the error model. This, in turn, sharpens the conditional probability of prediction errors, and increases the compression efficiency of a context adaptive entropy coder. The description of sign flipping is presented in [section 4.4](#).

For 8-bit gray scale image, the total number of pixel values is 256. However, the values of prediction errors fall in the closed interval $[-255, 255]$, and the size of the alphabet becomes 511. This shows that the number of entries of a table, which represents the conditional histogram of prediction errors, is doubled. This, in turn, affects the compression efficiency of the entropy coder by creating the context dilution problem in estimating the conditional probability of prediction errors. To address this problem, after error flipping, the number of prediction errors is reduced into 256 by applying error re-mapping function. This is described in detail in [section 4.5](#).

Experimental results on benchmark data sets reveal that the enhanced algorithm achieves a better compression bit rate than our previous algorithm and other algorithms. It is shown that the compression bit rate is much better for images that are rich in directional edges and textures. The enhanced algorithm also shows better rate-distortion performance and visual quality at the intermediate stages of progressive transmission.

This paper differs from our previous work [10] in the following aspects. First, for the initial prediction, the enhanced algorithm uses the predictor of the EAHINT algorithm. In this paper, the above three main components are added to improve the prediction accuracy by capturing higher order local image structures such as edges and textures. Other components of the complete encoding system are also added. Second, in our previous work, we evaluate the performance of our algorithm with other algorithms theoretically by computing the

unconditional (zero order) entropy of prediction errors. In this paper, we evaluate both prediction systems practically using the actual bit rates that are generated by context adaptive arithmetic coder. Third, in our previous work, we compared the image quality of the prediction algorithm with other algorithms using only Peak Signal-to-Noise Ratio (PSNR). This is a pixel based image quality metric does not correlate well with the human perception. In this paper, we use the recently developed image quality metrics, namely, Structural Similarity Index [11] and Feature Similarity Index metrics [12]. These metrics are better correlated with the human perception. Fourth, the performance of the enhanced algorithm is also compared with a new image-upscaling algorithm that is published recently after the publication of our previous work [13].

The rest of the paper is organized as follows. In section 2, a review of lossless multi-resolution image coding algorithms is given. In section 3, an overview of the enhanced encoding system is described. In section 4, each components of the enhanced coding system is described in detail. The experimental results are presented in section 5. Finally, the conclusions are derived in section 6.

2. Related Work

2.1 An Overview of Lossless Multi-resolution Image Compression Scheme

For a better explanation throughout this paper, it is important to refer to the typical multi-resolution image compression scheme which supports both progressive image transmission and multiple decoding functionalities. As Fig. 4 shows, an input image I is encoded into a set of down-sampled layers such that $\{I_0, I_1, \dots, I_K\}$, and is then transmitted progressively from I_K to I_0 . The sender initially encodes I_K into the k -th layer bit stream B_K and transmits B_K to the receiver; and then the receiver decodes it into J_K . At the next lower layer $k-1$, the bit stream B_{k-1} is encoded by an entropy coder using the difference between the I_{k-1} and the interpolated I'_{k-1} from I_k , and is then transmitted to the receiver. On the decoder side, B_{k-1} is decoded, J_k is interpolated to generate J'_{k-1} , and finally J_{k-1} is generated from the sum of J'_{k-1} and the decoded B_{k-1} . The above process takes place repeatedly until I_0 . Once the receiver satisfies the quality of the decoded image, it can stop the transmission at any layer to reduce the long transmission delay. This implies that the encoder transmits the image to a single decoder progressively by adapting the content of the image to the available bandwidth. In short, first, an encoder constructs a pyramidal data structure from bottom to top by applying image downscaling operation, and then it applies image upscaling operation in the opposite direction from top to bottom using interpolation algorithm. Hence, encoders apply both downscaling and upscaling. However, decoders does only upscaling operation.

Multi-resolution image compression also supports multiple decoding functionality to adapt the content of the image to the receiver's capacity ranging from cell phones to PC. As a result of this, the encoder will send only the first few layers of bit streams to a decoder with restricted resources. All the layers of the bit streams from B_K to B_0 can be sent to a decoder with the highest processing power and display. The encoder constructs a pyramid data structure from bottom to top as follows. Let R and C be the number of rows and columns of an input image. Let $I_o(r, c)$ be the original image, where $r=0, 1, 2, \dots, R-1, c=0, 1, 2, \dots, C-1, R=p \times 2^K, C=q \times 2^K$; p, q , and K are positive integers. The set of reduced-resolution images can be obtained by sub-sampling the original image recursively such that

$$I_l(i, j) = I_{l-1}(2i, 2j) = I_o(i \times 2^l, j \times 2^l), \quad (1)$$

for $l=1, 2, 3, \dots, K, i=0, 1, 2, 3, \dots, (R/2^l)-1$, and $j=0, 1, 2, 3, \dots, (C/2^l)-1$.

In the lossy version of multi-resolution image coding, the image downsampling operation is applied after smoothing the original image by applying image smoothing filters such as a bilateral filter [14], a 2x2 mean filter [15] and the Lagrange filter [16]. Hence, at each layer i , the original image I_i and reconstructed image J_i are not identical in Fig. 4. However, in the lossless version, the image downsampling operation is applied directly on the original image without using any image smoothing operation so that I_i and J_i are exactly identical.

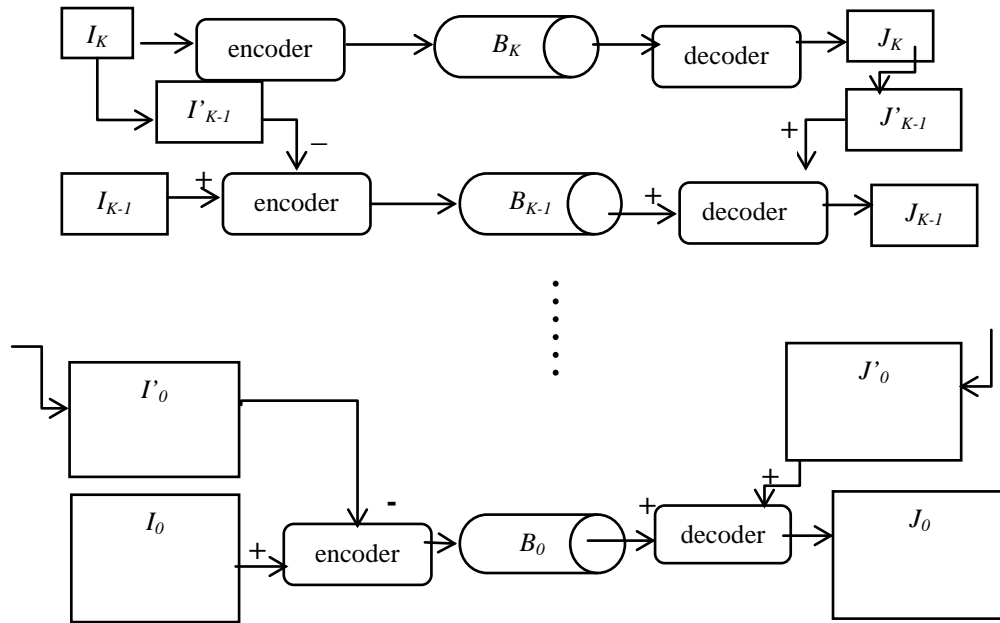


Fig. 4. Pyramidal data structure for multi-resolution image coding and progressive transmission

2.2 Review of Lossless Multi-resolution Image Compression Algorithms

Recently, several spatially adaptive interpolation techniques have been proposed to find the optimal coefficients of a linear predictor by means of least-squares optimization [17, 18, 19, 20, 21, 22, 23]. They are effective for an arbitrary edge direction. However, finding the optimal coefficients is computationally very expensive because the least-squares covariance matrix for each pixel is computed from a large window. Hence, the use of these complex algorithms is limited to develop progressive transmission system for heterogeneous decoding devices because these devices require simple interpolation algorithm.

Consequently, a number of simple pyramid data structures have been evaluated for lossless multi-resolution image coding scheme, namely, the Hierarchical INTERpolation (HINT) [24], the Interleaved Hierarchical INTERpolation (IHINT)[25] and the New Interleaved Hierarchical INTERpolation (NIHINT) [26]. There are also works to encode only chrominance components of the color image using hierarchical interpolation [27, 28]. A detailed review of these hierarchical interpolative algorithms is available in [10]. However, HINT, IHINT, and NIHINT do not use techniques to identify the directions of strong intensity correlations near the target pixel so as to exploit the directional correlations among neighboring pixels. The coefficients of these linear predictors are not adaptive to local structures around edges. As a result, interpolation errors along sharp edges are large and cannot be encoded efficiently by entropy coders. These interpolative algorithms also tend to generate interpolated images with blurred edges and annoying artifacts near the edges. As a

result of this, we proposed an Edge Adaptive Hierarchical INTerpolation (EAHINT) to reduce the entropy of interpolation errors in our previous work [10]. The algorithm exploits the directional correlations around a target pixel by using the local variance of the causal context as a measure of the local activity level. Based on the local variance, three statistical decision rules are designed to classify the local edge into a strong, a weak, or a medium edge. Hence, the EAHINT algorithm is a switching predictor among three linear predictors according to the activity level of the local area.

Recently, an image upscaling algorithm is proposed in [13]. Similar to our previous algorithm in [10], this algorithm uses a linear predictor with eight adaptive coefficients to generate higher resolution image from a lower resolution image. Let $w_1, w_2, w_3, w_4, w_5, w_6, w_7,$ and w_8 be weights assigned to the 8 neighboring pixels of a lower resolution image. Let $b_1, b_2, b_3, b_4, b_5, b_6, b_7,$ and b_8 be the associated 2×2 blocks for each neighboring pixel. The weight w_i is assigned by counting the number common pixels when a 3×3 sliding window, which is centered at the current pixel, intersects with block b_i , for $1 \leq i \leq 8$. This algorithm does not use the intensity value to determine weights and to identify the directions of strong intensity correlations near the target pixel. The coefficients are fixed and remain the same from block to block throughout the image. Due to this reason, interpolation errors along sharp edges are large and cannot be encoded efficiently.

In this paper, we extend our previous EAHINT algorithm by improving the compression bit rate of prediction errors and the visual quality of the reconstructed images at intermediate stages during progressive image transmission. This is achieved by incorporating additional components as described in detail [section 3](#) and [section 4](#).

3. Overview of the Enhanced Hierarchical Encoding System

The compression efficiency of the predictor of EAHINT algorithm is enhanced by incorporating additional components to it as shown in [Fig. 5](#). These components are based on the causal context of the current pixel p_i , denoted by $C_i = [p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8]$. The components of the enhanced system are implemented according to the following steps.

In step 1, the initial prediction p_i of the current pixel p_i is estimated as a function of its eight neighbouring pixels using the predictor of EAHINT algorithm [10]. To make this paper complete by itself, this predictor is described briefly in [section 4.1](#). *In step 2*, the context in which the current pixel p_i occurs is determined to model the prediction errors. There are three types of error modeling contexts: entropy coding context (\mathbf{E}) which captures the strength of the error in a local context, higher-order image structure context (\mathbf{H}) which captures higher-order image structures such as texture, corners and edge patterns, and prediction refinement context (\mathbf{R}) which is formed by combining the two contexts using Cartesian product method. The detailed description of these error modeling contexts is presented in [section 4.2](#).

Normally, prediction errors are modeled by a context-conditioned two-sided geometric distribution with two parameters, namely exponential decay λ and bias δ [29]. The bias parameter δ indicates the extent to which the center of this distribution is shifted from zero. This shifting occurs due to the inaccuracy of the value of p_i . Hence, *in step 3*, in order to cancel the bias δ , the bias cancellation parameter B_c is estimated within each prediction refinement context. The estimation of B_c is described in detail in [section 4.3](#). *In step 4*, the initial prediction p_i is modified into a new refined value p_r by adding B_c to p_i . *In step 5*, the refined prediction error e_r is computed by subtracting p_r from the original pixel p_i . After prediction, statistical redundancy can exist in the arithmetic sign of prediction errors. Hence, *in step 6*, this

redundancy is reduced by flipping the sign of prediction errors according to the sign of the B_c parameter. This, in turn, sharpens the conditional histogram of the prediction errors. The prediction error sign flipping is presented in section 4.4.

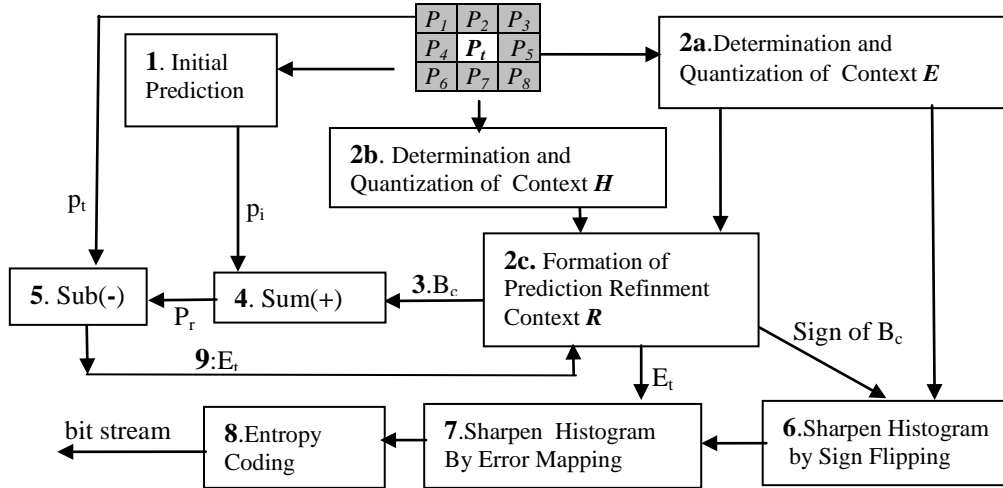


Fig. 5 . A Block Diagram of Enhanced EAHINT Encoding System

In step 7, the conditional probability distribution of prediction errors is further sharpened by reducing the number of symbols of prediction errors as described in section 4.5. In step 8, the prediction error e_r is entropy coded by adaptive arithmetic entropy coder. Finally, in step 9, before encoding the next pixel p_{t+1} , the prediction error e_r of the current pixel p_t is used to update the B_c parameter for the next occurrence of the same context. Since context adaptive arithmetic entropy coder is used to encode the prediction errors, the prediction error e_r also updates the conditional histogram by increasing the frequency of the error by one.

As the image is processed pixel by pixel, the bias cancelation parameter B_c refines the initial prediction p_i , and, in turn, the refined prediction error e_r updates the bias cancelation parameter B_c . This is depicted by a closed loop structure in Fig. 5. Consequently, the B_c parameter is initialized to zero before encoding starts. Similarly, for each entropy coding context, we begin with a table in which all frequency counts are initiated by 1 before encoding starts, and the prediction error updates this table as depicted in Fig. 5.

4. Detailed Description of the Enhanced Encoding System

4.1 Initial Prediction

The initial predictor estimates the value of the current pixel p_t by determining the strength and direction of the local edge. The variance (σ^2) of the causal neighbors of the current pixel is used to estimate the activity level of the prediction context. Using the value of the local variance, the local area can be classified as an edge area or a smooth area. Hence, the initial predictor switches between a non-directional static weighting interpolator for a smooth area, a one-directional interpolator for an edge area. A multi-directional interpolator is also a compromise between the two extreme cases.

By adding the prediction refinement, error sign flipping and error mapping modules to the pseudo code of the original EAHINT algorithm, the pseudo code of the enhanced EAHINT (E-EAHINT) algorithm is presented as follows.

Pseudo Code: Enhanced Edge Adaptive Hierarchical Interpolation (E-EAHINT)

Step 0: Let $C_t = [P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8]$ be the causal context as shown [Fig. 5](#).

Step 1: $\mu = \frac{\sum_{i=1}^8 P_i}{8}$; Mean of the local window

Step 2: $\sigma^2 = \frac{\sum_{i=1}^8 (P_i - \mu)^2}{8}$; Variance of the local window

Step 3: Let G_1 and G_2 be two empty sets

Step 4: **if** $\sigma^2 > T_2$, **then**
 for each $P_i \in X$ **do**
 if $P_i > \mu$ **then**
 $G_1 = G_1 \cup P_i$
 else $G_2 = G_2 \cup P_i$
 Compute σ_1^2 and σ_2^2 for each group G_1 and G_2
 if $\sigma^2 > \sigma_1^2 + \sigma_2^2$, **then**
 One Directional Interpolator to C
 else
 Multi Directional Adaptive Wighting Interpolator to C_t
else if $T_1 < \sigma^2 \leq T_2$, **then**
 Multi Directional Adaptive Weighting Interpolator to C_t
else
 Non-Directional Static Weighting Linear Interpolator to C_t

Step 5: Prediction Refinement Using Error Modeling Context ([section 4.2](#))

Step 6: Sharpening Conditional Histogram of Errors by Error Modeling ([section 4.4](#))

Step 7: Sharpening Conditional Histogram of Errors by Mapping Errors([section 4.5](#))

Step 8: Entropy Coding of prediction Errors

As indicated in the above pseudo code, the initial predictor of the EAHINT algorithm is enhanced by adding the last four components or steps. Since the detailed description of the first four steps is presented in [10], we describe only the last four components in the following sections.

4.2 Types of Conditional Error Models

Our previous predictor is a linear predictor, which uses a 3x3 prediction context (window) to model the unknown current pixel P_t as a linear combination of its neighboring pixels $P_{t-1} \dots P_{t-8}$. Since the initial predictor captures a linear relationship among neighboring pixels, it removes only low-frequency correlations or smoothness related redundancy in natural images. The predictor may fail to remove all the correlations among neighboring pixels if the relationship among neighbouring pixels is complex and non-linear within that prediction context. Because of this, still there exists some hidden structures among neighbouring pixels in the error image I_e within that prediction context. This implies that the initial predictor cannot model higher order image structures such as textures, corners and edge patterns. The shape of objects or local regions can be arbitrarily in the real scene. However, to simplify the computation of prediction, the shape of the prediction context is

a rectangular. This implies that the prediction context can contain pixels from both smooth regions and edge areas. Hence, there will be a high variability among neighboring pixel values. This variability also affects the probability distribution of the prediction errors.

To address the above two weakness of the initial predictor, it is necessary to refine the initial prediction p_i by modeling the prediction error using a local context. In other words, we transform the maximization of the conditional probability $P(P_t|P_{t-1}\dots P_{t-8})$ into the maximization of the conditional probability $p(e_t|p_{t-1}\dots p_{t-8})$, where $e_t = P_t - P_i$.

Normally, the conditional probability of errors is modeled by a two-sided geometric distribution centered at zero with only one parameter called exponential decay λ [29]. However, due to the above two reasons, the center of the conditional probability is shifted from zero, and the shifting parameter is called bias of the model. To remove the bias (offset), we model the prediction refinement context $P(e_t|R)$ by combining higher order context $P(e_t|H)$ and error energy strength context $P(e_t|E)$, where e_t is a refined prediction error. Using this hybrid context, the final predictor becomes context-sensitive and non-linear predictor.

4.2.1 Determination and Quantization of Entropy Coding Context

At any time t , let $C_t = P_{t-n}, \dots, P_{t-1}$ be an n -order entropy coding context to model the conditional probability of prediction errors. In an enhanced E-EAHINT algorithm, the value of n is 8. If the number of bits for each pixel is m , the total number of distinct contexts will be 2^{8m} . For an 8-bit gray scale image, the number of possible contexts will be 2^{64} . This is a very large number. Hence, each context cannot provide sufficient samples of prediction errors to estimate conditional probability of errors because the count statistics spread over too many contexts. Within a local context, the strength of the local error has strong correlation with the local variance of the original pixel values. Hence, the number of conditional entropy coding contexts can be reduced by representing each context using its error strength [29,30,31]. The strength of error can be estimated using the local gradients which are pre-computed during the initial prediction. Except the top layer of the pyramid, each layer is scanned two times, and the interpolation is done at two stages of interpolation as described in detail in [10].

Let g be the local error strength in a given context C_t for the current pixel p_t . Let $g_v, g_h, g_d,$ and g_a be local gradients along vertical, horizontal, diagonal and anti-diagonal directions, respectively. Then, the strength of local error g in a given context C_t can be estimated from the four local gradients as follows.

$$\text{First Stage of Interpolation: } g = g_a + g_d \quad (2)$$

$$\text{Second Stage of Interpolation: } g = 0.95(g_v + g_h) + 0.05(g_a + g_d) \quad (3)$$

The computation of $g_v, g_h, g_d,$ and g_a is described in detail in [10]. Since the gradient captures the smoothness of the local area surrounding the current pixel, it determines the statistical behavior of prediction errors. Since the error strength g can take a large number of possible values, the number of possible contexts still remains large. To reduce these contexts, the range of g is quantized into k number of connected subintervals, $Q(g) = I_i, 1 \leq i \leq k$. The objective of this quantization is to maximize the mutual information between the current pixel p_t and its context C_t [30, 31]. For 8-bit images, prediction errors is minimized when when $K=7$. The quantization intervals are $[0,4],[5,14],[15,26],[27,48],[49,106],[107,140]$ and $g \geq 141$. The number of quantization levels and intervals are determined empirically off-line based on training data set by minimizing the conditional entropy of prediction errors.

4.2.2 Determination and Quantization of Prediction Refinement Context

When the conditional probability distribution of initial prediction errors E_i is modeled by two-sided geometric distribution, the distribution has a bias parameter δ . This parameter can be canceled from each context by subtracting a bias cancelation value B_c from the value of E_i . This implies that the refined prediction error will be $E_r = E_i - B_c$. Since the bias δ is created due to the inaccuracy of the initial prediction p_i , it can be also canceled by adding the value of B_c to the value of p_i . Hence, the refined prediction will be $P_r = P_i + B_c$, where P_r is a refined prediction of the current pixel p_r . The equality of these two concepts is justified as follows.

$$E_r = p_i - p_r \quad (4.1)$$

$$E_r = p_i - (p_i + B_c) \quad (4.2)$$

$$E_R = (p_i - p_i) - B_c \quad (4.3)$$

$$E_r = E_i - B_c \quad (4.4)$$

The B_c parameter is computed from prediction refinement context R . The context R is formed by the combination of the higher-order image structures context (H) and the error strength context (E). This context is also called entropy-coding context. In section 4.2.1, we determined the context E. In this section, we determine the context H as follows.

Recently, we proposed an algorithm in [32] to detect higher order image structures such as edges and corners using a circular mask. In computer vision, many edge and corner detectors were developed using a circular mask [33, 34, 35]. This motivates us to incorporate these higher order image structures into prediction refinement contexts. Let $C_r = [p_1, p_2, \dots, p_7, p_8]$ be a circular mask of the central pixel p_r . The central pixel may lie on an edge, corner or smooth area. To identify the type of local area, the circular region is modeled by two parameters, namely the geometric threshold and the brightness threshold [35]. The geometric threshold G controls the shape of edges and corners. The brightness threshold T controls the number of detected local edges and corners. Using these two parameters, these higher order local image structures are determined using the following steps.

First, all pixels in a circular area are quantized into a binary value by comparing their brightness with the brightness of the initial prediction value p_i by Eq. (5). This captures the local texture information [30, 31]. *Second*, the total running value $n(p_i)$ is computed as a measure of similarity of a central pixel with its surrounding pixels by Eq. (6). This value gives the proportion of pixels that are similar to the brightness of the central pixel. *Third*, the edge or corner response $R(p_i)$ of the central pixel is computed as a measure of dissimilarity of the central pixel by comparing the value of $n(p_i)$ with the geometric threshold G by Eq. (7).

Experimentally, it was found that to detect local edges, t is set to 24 and G is set to three-fourth of the maximum possible value of $n(p_i)$, n_{max} . However, to detect local corner structures, the value of G is half of the maximum value of n [35].

$$b(p_k, p_i) = \begin{cases} 1 & \text{if } |p_k - p_i| \leq t \\ 0 & \text{if } |p_k - p_i| > t \end{cases} \quad (5)$$

$$n(p_i) = \sum_{k=1}^8 b(p_k, p_i) \quad (6)$$

$$R(p_i) = \begin{cases} G - n(p_i), & \text{if } n(p_i) \leq G \\ 0 & \text{otherwise,} \end{cases} \quad (7)$$

Since we are detecting local edges and corner structures simultaneously, we can set the value of G equal to n_{max} . Hence, for our 3x3 local circular window, the value of G is 8, and the

value $R(p_i)$ ranges from 0 to 8. These values are further quantized into three connected non-overlapping intervals to characterize the three types of local regions as follows.

$$\text{Local Region} = \begin{cases} \text{smooth area,} & \text{if } 0 \leq R(p_i) \leq 2 \\ \text{edge area,} & \text{if } 3 \leq R(p_i) \leq 5 \\ \text{corner area,} & \text{if } 6 \leq R(p_i) \leq 8 \end{cases} \quad (8)$$

Since each pixel in the circular region is quantized into binary values, this model captures the texture context. Unlike texts, natural images usually contain locally homogenous areas in local regions, unless there is a strong edge boundary, corner or random noise. Due to the existence of such type of pixels, the repetition of the central pixel within the local window is not usually exact [36]. Given the center pixel P_c and a neighboring pixel P_n , we can say that P_c is a T -approximate match to P_n if $|P_c - P_n| \leq T$, where T is an integer threshold as indicated in [Eq. \(5\)](#). This can be generalized for context strings. Let $C_1 = [p_m, p_m, p_{m-1} \dots p_1]$ and $C_2 = [q_m, q_{m-1} \dots q_1]$ be an order m -contexts. Then, these two contexts are a T -approximate matching iff

$$|p_m - q_m| \leq T \wedge |p_{m-1} - q_{m-1}| \leq T \wedge \dots \wedge |p_1 - q_1| \leq T \quad (9)$$

In text pattern matching, the value of T considers the position of each symbol. However, in natural images, the value of T considers the gray level values of the pixels. This implies that using L_I -metric, [Eq. \(9\)](#) can be modified into [Eq. \(10\)](#). Hence, C_1 and C_2 are T -approximate matching contexts if the following holds true.

$$\sum_{i=1}^m |p_i - q_i| \leq K \quad (10)$$

For example, unlike text pattern matching, two contexts $C_1 = [80, 47, 79, 202]$ and $C_2 = [78, 45, 77, 201]$ are approximate matches for $T=2$ and $K=7$ using L_I -metrics. For the order-8 context, the 8-bit binary patterns (contexts) which are obtained by [Eq. \(5\)](#) are clustered or quantized into 8 clusters based on their distance from a reference context $C = [1, 1, 1, 1, 1, 1, 1, 1]$ using [Eq. \(7\)](#). The 8 contexts are further clustered into three clusters using [Eq. \(8\)](#) and [Eq. \(10\)](#). The value of T is determined empirically from training test images. Initially, we start the searching of the current context, by setting $T=0$ (exact pattern matching). We increase the value of T in steps of 1 by observing the change in compression bit-rate and PSNR at intermediate levels of progressive transmission. According to our statistical analysis, we observed that the optimal value of T has direct relationship with the entropy (complexity) of test images. Since the maximum value of the entropy of 8-bit gray scale images is 8 bits per pixel, we found that the reasonable range of T is $0 \leq T \leq 8$. Finally, through extensive experiments, we found that $T=3$ works very well for all test images. Thus, we used this threshold value in our experimental results.

4.3 Estimation of the Bias Cancelation Parameter

As we described in [section 4.2](#), when prediction errors are modeled by context-conditioned two-sided geometric distribution, the center of the distribution is shifted from zero [29]. The conditional sample mean E_m is the most likely prediction error within each prediction refinement context [29,30,31]. Hence, adding E_m to the initial prediction p_i removes the bias parameter δ and refines the initial prediction p_i . An accumulator of errors A and a counter N on the occurrence of each quantized prediction refinement context are maintained to compute the conditional sample mean E_m of prediction errors within each context.

4.4 Sharpening Conditional Probability of Errors by Sign Flipping

Within a given entropy coding context $Q(g)$, although spatial or interpixel redundancy is reduced by refined prediction, sign redundancy still exists among the prediction errors. This redundancy can be removed by flipping the sign of errors using the sign of the conditional sample mean E_m [29,30,31]. If the sign of E_m is negative, the sign of the refined prediction error E_r is flipped, and then it is entropy coded. However, if the sign of E_m is positive, the refined error E_r is entropy coded without sign flipping. Hence, the magnitude of E_m is used to refine the initial prediction and its sign is used to sharpen the conditional probability of prediction errors $p(e | Q(g))$ as depicted in Fig. 5.

4.5 Sharpening Conditional Probability of Errors by Remapping Errors

For n -bit gray level images, the total number of distinct gray level pixel values is 2^n , and the possible value of pixels ranges from 0 to $(2^n)-1$. On the other hand, the total number prediction error values is $(2^{n+1})-1$, and their values ranges from $(-2^n)+1$ to $(2^n)-1$. Hence, the number of entries of a table, which represents the conditional histogram of prediction errors, is doubled from 2^n to $2^{n+1}-1$. This affects the compression efficiency of the entropy coder by creating the context dilution problem in estimating the conditional probability of prediction errors, $p(e | Q(g))$. To reduce this problem, the prediction errors can be remapped to take values from 0 to $(2^n)-1$. This is explained as follows.

The predicted value P must fall in the closed interval $[0, 2^n-1]$. For a given prediction P , the corresponding prediction error E must fall in the closed interval $[-P, (2^n-1) - P]$ [30]. By partitioning the closed interval $[0, 2^n - 1]$, P belongs to one of the two sub-intervals: $[0, 2^{n-1}]$ or $[2^{n-1}+1, 2^n-1]$. When P belongs to the first subinterval $[0, 2^{n-1}]$, its corresponding prediction error E belongs to one of the three subintervals: $[-P, -1]$, $[0, P]$ or $[P+1, (2^n-1)-P]$. Similarly, when P belongs to the second subinterval $[2^{n-1}+1, 2^n-1]$, its corresponding E belongs to one of the three intervals: $[-P, -((2^n-1)-P)-1]$, $[-((2^n-1)-P), -1]$ or $[0, (2^n-1)-P]$.

When E belongs to the subinterval $[P+1, (2^n-1) - P]$, E can be expressed as $E = P+i$, for some i such that $1 \leq i \leq j$ and $j = (2^n-1) - 2P$. The value of j is computed from $P+j = (2^n-1)-P$. On the other hand, when E belongs to the subinterval $[-P, -((2^n-1)-P)-1]$, E can be expressed as $E = -((2^n-1) - P) - i$, for some i such that $1 \leq i \leq j$ and $j = 2P - (2^n-1)$. The value of j is computed from $-((2^n-1) - P) - j = -P$. Based on this analysis, the C++ pseudocode of the error-mapping algorithm is given as follows.

ErrorRemapping(P,E)

```
{
  if 0 ≤ P ≤ 2n-1
  {
    if -P ≤ E ≤ -1, then E = 2 | E | -1.
    if 0 ≤ E ≤ P, then E = 2E.
    if P+1 ≤ E ≤ (2n -1) -P
    {
      j = (2n-1) -2P;
      for (i=1; i ≤ j; i++)
        if E = (P+i), then E = 2P+i
    }
  }
  else if (2n-1)+1 ≤ P ≤ 2n -1
  {
    if 0 ≤ E ≤ (2n -1)-P, then E = 2E
    if -((2n -1)-P) ≤ E ≤ -1, then E = 2 | E | -1.
    if -P ≤ E ≤ -((2n -1)-P) -1
    {
      j = 2P - (2n-1);
      for (i=1; i ≤ j; i++)
        if E = -((2n -1)-P) -i, then E = 2 | -(2n -1)-P | +i
    }
  }
} // End of ErrorRemapping
```

The above analysis shows that when $P \leq 2^{n-1}$, the set of all possible prediction errors $\{-P, -(P-1), -(P-2), \dots, -1, 0, 1, \dots, P, P+1, P+2, \dots, (2^n-1)-P\}$ can be mapped into an ordered list $\{0, +1, -1, +2, -2, \dots, +P, -P, P+1, P+2, \dots, (2^n-1)-P\}$. In addition to minimizing the context sparsity problem, the remapping operation arranges errors from the highest probability to the lowest probability. This, in turn, increases the compression efficiency of the entropy coder. This also holds true when $P \geq 2^{n-1}+1$.

5. Experimental Results and Discussion

In this section, we evaluate the performance of the enhanced EAHINT (E-EAHINT) algorithm in comparison to the previous algorithm (EAHINT). It is also compared with a recently published image up scaling algorithm [13]. Five common standard test images with 8 bits per pixel and size of 512x512 are used in our experiments as shown in Fig. 6. In lossless multi-resolution image coding, the input image is encoded only once as a layer of bit streams. Hence, the total lossless bit rate is a common criterion to measure the performance of the encoder. The decoder can also up-scale each layer of the pyramid from the lower resolution to the original higher resolution. Therefore, the rate distortion and the visual quality at intermediate stages during the progressive transmission are also used to evaluate the performance of the interpolation algorithms.



Fig. 6. Images used for the experimental evaluation of the proposed encoding system

5.1 Total Lossless Compression Bitrate

Before compression, the number of bits per pixel (bpp) is 8 for our test images. Hence, the compression bit rate is the reduced number of bits per pixel. To compute the total lossless compression bit rate, a different codebook was assumed to encode each pyramid level l . As shown in Fig. 4, the progressive bit rate up to level l , $R(l)$, is the total accumulated bit rate up to level l , such that

$$R(l) = R(l+1) + r(l), \quad (11)$$

where $R(l+1)$ is the progressive bit rate up to its upper level $l+1$ and $r(l)$ is the bit rate used for transmitting the data at level l [3]. However, the bit rate of each pyramid level $r(l)$ is given in terms of bits per node (bpn), and it must be normalized to the actual number of bits per pixel (bpp) with respect to the size of the original image. Hence, we use the equivalent number of bits per pixel, bpp_{eq} , for comparison purposes such that

$$r(l) = bpp_{eq}(l) = (N_n/N_p) bpn(l), \quad (12)$$

where N_n is the number of nodes at level l and N_p is the number of pixels of the original image.

After the encoder transmits the bottom layer of the pyramid, the decoder reproduces the lossless version of the image. Hence, we can use the total lossless bit rate, bpp_L , to compare hierarchical interpolative algorithms. Due to the absence of low-pass filtering, sub-sampling breaks down the local geometry as the value of the base-band level K increases, making it

difficult to process the next level of image decomposition. Based on this reasoning, the base-band level in interpolating pyramid algorithms is usually set to $K=2$ or $K=3$ [10]. We used the based band level $K=3$ in our experiment. **Table 1** shows the number of bits per node (bpn) and the equivalent number of bits per pixel (bpp_{eq}) achieved by EAHINT, E-EAHINT and image upscaling algorithm in [13]. At the base band level $K=3$, all the three hierarchical interpolation algorithms have identical compression bit rate. The reason is that the pixels at the base band layer are predicted using a sequential image prediction technique called median edge detector (MED) from the JPEG-LS standard [29].

The E-EAHINT algorithm reduces the total lossless bit rate bpp_L significantly compared to the EAHINT algorithm for the Barbara, Boat and Lena images, whereas the reduction is marginal for the Baboon image. This result was expected somewhat because the Boat image contains many sharp edges, the Barbara image contains strong directional textures, and the Lena image has fairly sharp edges with a small amount of texture. However, the Baboon image has numerous fine details that are scarcely predictable as shown in **Fig. 6**

Table 1. The number of bits per node (nbp) and the equivalent number of bits per pixel (bpp_{eq}) achieved by [13], EAHINT and E-EAHINT for the base-band level $K=3$. For bpp_L , Gain-1 and Gain-2 represent the percentage of the total bit rate reduction by E-EAHINT compared to [13] and EAHINT algorithms, respectively.

image	l	Ref[13]		EAHINT		E-EAHINT		Gain	
		bpn	bpp_{eq}	bpn	bpp_{eq}	bpn	bpp_{eq}	Gain-1	Gain-2
Baboon	3	7.41	0.11	7.41	0.11	7.41	0.11	-----	-----
	2	7.11	0.33	7.02	0.33	6.71	0.31	-----	-----
	1	7.07	1.32	6.70	1.26	6.37	1.19	-----	-----
	0	7.04	5.29	6.01	4.51	5.67	4.25	-----	-----
	bpp_L	---	7.05	---	6.21	---	5.86	16.88%	5.64%
Barbara	3	6.95	0.10	6.95	0.10	6.95	0.10	-----	-----
	2	6.78	0.31	6.15	0.29	5.81	0.27	-----	-----
	1	6.48	1.28	5.81	1.09	5.48	1.02	-----	-----
	0	6.88	5.16	5.03	3.77	4.70	3.52	-----	-----
	bpp_L	---	6.85	---	5.21	---	4.91	28.32%	6.48%
Boat	3	6.58	0.10	6.58	0.10	6.58	0.10	-----	-----
	2	6.55	0.30	6.01	0.28	5.67	0.26	-----	-----
	1	6.59	1.23	5.37	1.01	5.02	0.94	-----	-----
	0	6.60	4.97	4.77	3.58	4.43	3.32	-----	-----
	bpp_L	---	6.60	---	4.97	---	4.62	30.00%	7.04%
Lena	3	6.25	0.09	6.25	0.09	6.25	0.09	-----	-----
	2	6.14	0.28	5.38	0.25	5.06	0.23	-----	-----
	1	6.23	1.16	4.75	0.89	4.43	0.83	-----	-----
	0	6.31	4.73	4.11	3.09	3.78	2.84	-----	-----
	bpp_L	---	6.26	---	4.32	---	3.99	36.26%	7.64%
Paper	3	6.48	0.10	6.48	0.10	6.48	0.10	-----	-----
	2	6.24	0.29	5.47	0.26	5.14	0.24	-----	-----
	1	6.36	1.19	4.77	0.89	4.43	0.83	-----	-----
	0	6.42	4.82	4.47	3.52	4.14	3.10	-----	-----
	bpp_L	---	6.40	---	4.77	---	4.27	33.28%	10.48%

The better performance of E-EAHINT is also demonstrated visually in **Fig. 7**. In this figure, the prediction errors of base-band pixels and triplets of the interpolation errors of enhanced

layers are arranged recursively into a sub-band like regions. This is illustrated by drawing two differential pyramids of the Barbara test image generated by the EAHINT and E-EAHINT algorithms after three levels of image decomposition. It is clear that the H -band (higher frequency) sub-images generated by the E-EAHINT algorithm show less signal energy. Therefore, they are much less noticeable. The reduction in energy in this case shows that better compression results can be obtained using E-EAHINT algorithm compared to EAHINT.

To show the superior performance of the E-EAHINT algorithm visually along sharp edges and textured areas in comparison to EAHINT, we generated the black-and-white image as shown in Fig. 8. After error mapping, let e_{EE} and e_E be prediction errors in the interval $[0,255]$ which are generated by the EEAHINT and EAHINT interpolative algorithms, respectively. If the value of e_{EE} is less than the value of e_E , black color (the zero intensity value) is assigned for the corresponding pixel p . Otherwise, white color (255) is assigned. Fig. 8 shows that E-EAHINT interpolation scheme is superior to EAHINT along sharp edges and around textured areas. For the based-band level, both error values are equal because they are generated by median edge detector (MED) of the JPEG-LS standard [29]. Consequently, the top left region is completely white.

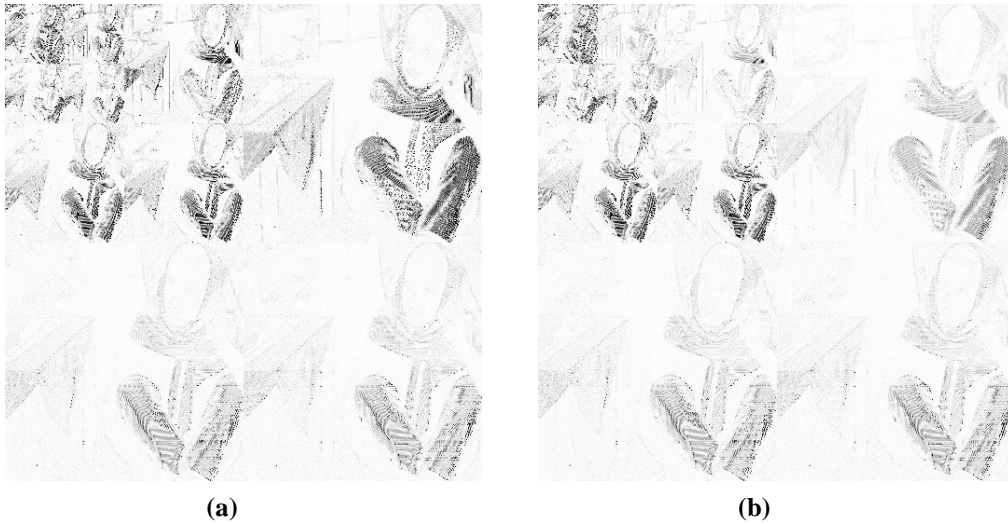


Fig. 7. Differential pyramids of the Barbara image generated by (a) EAHINT and (b) E-EAHINT after three levels of image decomposition. The darker the pixel, the larger the error. As shown in Table 1, the top left image is identical in both algorithms because this image is the base-band level image.

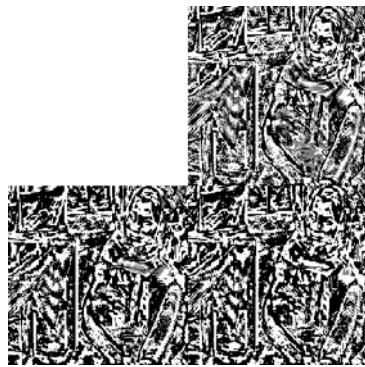


Fig. 8. Comparison of EAHINT and E-EAHINT algorithm results along sharp edges or around textured areas after one level of image decomposition for the Barbara image. The black pixels demonstrate that the E-EAHINT is superior to EAHINT along sharp edges or around textured.

We also compared the performance of our algorithm with the recently published image upscaling algorithm in [13]. As described in section 2.2, this algorithm uses the weighted average of 8 neighboring pixels. Each coefficient is determined by counting the number common pixels when a 3x3 sliding window intersects with a 2x2 neighboring block. Since the intensity values are not used to identify the directions of strong intensity correlations near the target pixel, the coefficients are fixed from block to block. As a result, interpolation errors along sharp edges are large and cannot be encoded efficiently as shown in Table 1.

5.2 Rate Distortion and Visual Quality

In multi-resolution coding, given prediction errors up to level l , the receiver can expand the reduced resolution from this level into the original resolution l_0 using an interpolative algorithm without receiving predictions errors from other next levels. Thus, we can compare the distortion of interpolated images at intermediate levels l of the pyramid in comparison to the original image. The experimental results are summarized in Table 2 for the base-band level $K=3$. These tables show objective image quality by upscaling the image from $l=3$ to $l=0$ (from 64x64 to 512x512), from $l=2$ to $l=0$ (from 128x128 to 512x512), and from $l=1$ to $l=0$ (from 256x256 to 512x512). E-EAHINT algorithm shows a better improvement for all test images.

Table 2. The SSIM and FSIM values of the images generated by EAHINT, E-EAHINT and an image upscaling algorithms in [13] for base band level $K=3$. Gain-1 and Gain-2 represent the percentage of the image quality enhancement by E-EAHINT compared to Ref [13] and EAHINT algorithms respectively.

image	l	[13]		EAHINT		E-EAHINT		Gain	
		SSIM	FSIM	SSIM	FSIM	SSIM	FSIM	Gain-1: SSIM/FSIM	Gain-2: SSIM/FSIM
Baboon	3	0.31	0.73	0.41	0.72	0.75	0.86	139.6 / 18.2	84.8 / 19.5
	2	0.50	0.81	0.59	0.83	0.82	0.91	62.9 / 11.6	38.0 / 9.9
	1	0.82	0.91	0.85	0.92	0.94	0.96	13.5 / 5.8	9.5 / 4.4
Barbara	3	0.51	0.76	0.65	0.82	0.84	0.92	63.1 / 22.1	29.3 / 12.1
	2	0.69	0.85	0.76	0.89	0.86	0.94	24.3 / 11.01	12.8 / 5.0
	1	0.88	0.94	0.88	0.96	0.94	0.98	7.1 / 3.67	16.8 / 2.3
Boat	3	0.50	0.73	0.62	0.78	0.88	0.92	74.8 / 25.5	42.2 / 17.4
	2	0.69	0.83	0.79	0.89	0.93	0.95	33.3 / 15.1	17.0 / 7.3
	1	0.91	0.93	0.94	0.97	0.97	0.98	6.8 / 5.5	3.4 / 1.7
Lena	3	0.62	0.77	0.77	0.86	0.92	0.95	47.6 / 22.1	20.0 / 9.8
	2	0.80	0.87	0.89	0.94	0.95	0.97	19.0 / 12.0	6.9 / 3.4
	1	0.95	0.95	0.97	0.99	0.99	0.99	4.1 / 4.2	1.43 / 0.7
Paper	3	0.67	0.77	0.81	0.86	0.94	0.95	40.7 / 23.2	15.2 / 10.6
	2	0.84	0.87	0.92	0.94	0.96	0.97	14.6 / 11.6	4.8 / 3.5
	1	0.95	0.96	0.96	0.98	0.97	0.99	2.7 / 3.2	1.6 / 0.7

In addition to SSIM and FSIM metrics, we also compared the visual quality of these algorithms in Fig. 9 and Fig. 10. These pictures show zoomed-in portions of the Baboon and Boat images. The proposed algorithm preserves sharp edges better and generates images of higher visual quality compared to the EAHINT algorithm. However, in [13], the coefficients

are fixed and remain the same throughout the image. As a result, it tends to generate interpolated images with blurred edges and annoying artifacts near the edges as shown in [Fig. 10](#).

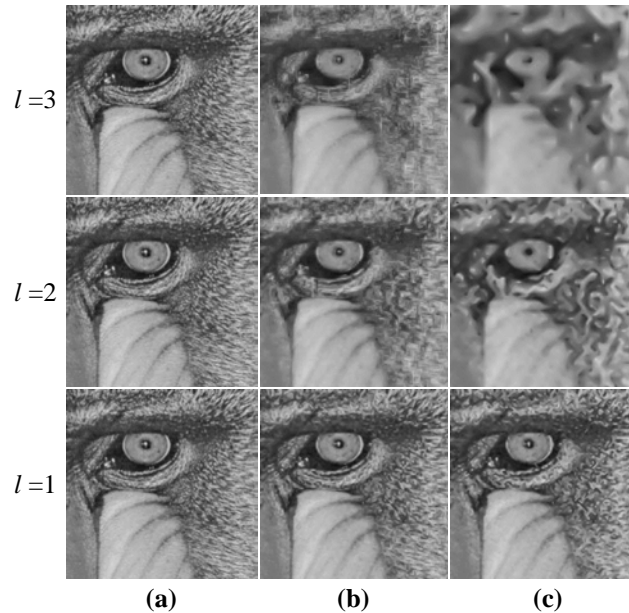


Fig. 9. (a) Zoomed-in portion of the original Baboon image, (b) the image interpolated by E-EAHINT, and (c) the image interpolated by EAHINT from top to bottom of the pyramid levels, $l=3$ (from 64×64 to 512×512), $l=2$ (from 128×128 to 512×512), and $l=1$ (from 256×256 to 512×512).

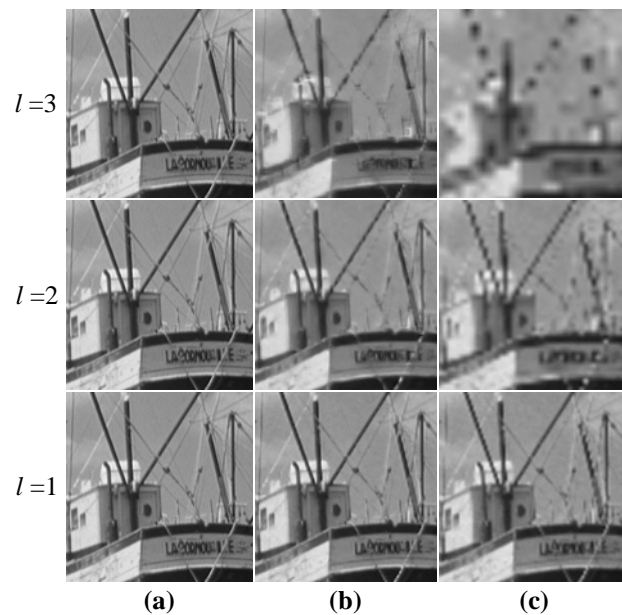


Fig. 10. (a) Zoomed-in portion of the original Boat image, (b) the image interpolated by E-EAHINT, and (c) the image interpolated by Ref[13] from top to bottom of the pyramid levels, $l=3$ (from 64×64 to 512×512), $l=2$ (from 128×128 to 512×512), and $l=1$ (from 256×256 to 512×512).

5. Conclusion

We enhanced our previous multi-resolution image coding scheme by adding context-condition error modeling component. This component refines the predictor of our previous algorithm by canceling the bias parameter of the two sided conditional probability distribution of prediction errors. The sign of this parameter is also used to reduce the sign redundancy among local errors by flipping the sign of prediction errors. This, in turn, sharpens the conditional probability distribution of prediction errors. The proposed context exploit local error strength and higher order local image structures such as texture, edge and corner in a better way by using circular context. As a result of this, the enhanced algorithm achieved better total lossless bit rates. It also generates better image quality at the intermediate stages of progressive image transmission.

The limitation of this algorithm comes mainly from its use of the selected value T by the user to compute the similarity of the central pixel to its neighboring pixels. Another limitation is the quantization of the error strength g into subintervals. The number of subintervals and the boundary of each subinterval is determined empirically based some training data set. These empirically selected thresholds may not achieve the best performance for images that are not included in the training data set. These parameters determine the number of distinct conditional probability contexts of the entire image. They can be optimized globally instead of locally. We used empirically determined threshold values because organizing and searching the previously occurred T -approximate matching contexts to the current context at the time of encoding process (online) is computationally expensive. These limitations remain as future work.

References

- [1] Y.-K. Chee, "Survey of progressive image transmission methods," *International Journal of Imaging Systems and Technology*, vol. 10, no. 1, pp. 3-19, January, 1999. [Article\(CrossRef Link\)](#)
- [2] A. Said and W. A. Pearlman, "An image multiresolution representation for lossless and lossy compression," *IEEE Transactions on Image Processing*, vol. 5, no. 9, pp. 1303-1310, September, 1996. [Article\(CrossRef Link\)](#)
- [3] M. Goldberg and L. Wang, "Comparative performance of pyramid data structures for progressive image transmission," *IEEE Transactions on Communications*, vol. 39, no. 4, pp. 540-548, April, 1991. [Article\(CrossRef Link\)](#)
- [4] C. Thiede, H. Schumann and R. Rosenbaum, "On-the-fly device adaptation using progressive contents," in *Proc. of Intelligent Interactive Assistance and Mobile Multimedia Computing*, pp.49-60, November 9-11, 2009. [Article\(CrossRef Link\)](#)
- [5] R. Rosenbaum and B. Hamann, "Raster image Adaptation for mobile devices using profiles," in *Proc. of SPIE 8304, multimedia on mobile devices 2012; and multimedia content access: Algorithms and systems VI, 8340H*, February, 2012. [Article\(CrossRef Link\)](#)
- [6] R. Rosenbaum and H. Schumann, "Progressive raster imagery beyond a means to overcome limited bandwidth," in *Proc. of SPIE 7256, Multimedia on Mobile Devices*, January, 2009. [Article\(CrossRef Link\)](#)
- [7] A. J. Penrose, "Extending lossless image compression," University of Cambridge, Technical Report UCAM-CL-TR-526, December, 2001. [Article\(CrossRef Link\)](#)
- [8] N. M. Banu and S. Sujatha, "3D Medical image compression: a review," *Indian Journal of Science and technology*, Vol 8, no.12, pp. 1- 9, June, 2015. [Article\(CrossRef Link\)](#)
- [9] K. Ranjee and B.R.C. Reddy, "Image Compression: An overview," *International Journal of Electrical, Electronic and Mechanical controls*, Vol.1, no. 1, November, 2012. [Article\(CrossRef Link\)](#)

- [10] Y. Biadgie, Y. Wee1 and J. Choi, "Edge adaptive hierarchical interpolation for lossless and progressive image transmission," *KSII Transactions on Internet and Information Systems*, vol. 5, no. 5, pp. 2068-2086, November, 2011. [Article\(CrossRef Link\)](#)
- [11] 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, April, 2004. [Article\(CrossRef Link\)](#)
- [12] L. Zhang, X. Mou and Zhang, "FSIM: a feature similarity index for image quality assessment," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2378-2386, January, 2011. [Article\(CrossRef Link\)](#)
- [13] M. Frucci, C. Arcelli and G.S. di Baja, "An automatic image scaling up algorithm," in *Proc. of the 4th Mexican Conference on Pattern Recognition*, pp. 35-44, June 27-30, 2012. [Article\(CrossRef Link\)](#)
- [14] J. Han, J. Kim, S. Cheon, J. Kim and S. Ko, "A novel image interpolation method using the bilateral filter," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 1, pp. 175-181, February, 2010. [Article\(CrossRef Link\)](#)
- [15] Y. Wee and H. Shin, "A novel fast fractal super resolution technique," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1537-1541, August, 2010. [Article\(CrossRef Link\)](#)
- [16] Y. Lee and J. Yoon, "Nonlinear image up sampling method based on radial basis function interpolation," *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2682-2692, October, 2010. [Article\(CrossRef Link\)](#)
- [17] X.Li and M. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1521-1527, October, 2001. [Article\(CrossRef Link\)](#)
- [18] N. Asuni and A. Giachetti, "Accuracy improvements and artifacts removal in edge based image interpolation," In *Proc. of 3rd International Conference on Computer Vision Theory and Applications*, Vol. 1, pp. 58-65, January 22-25, 2008. [Article\(CrossRef Link\)](#)
- [19] X. Zhang and X. Wu, "Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation," *IEEE Transactions on Image Processing*, vol. 17, no. 6, pp. 887-896, June, 2008. [Article\(CrossRef Link\)](#)
- [20] W. Tam, C. Kok and W. Siu, "A modified edge directed interpolation for images," in *Proc. of 17th European Signal Processing Conference*, pp. 283-287, August 24-28, 2009. [Article\(CrossRef Link\)](#)
- [21] K. Hung and W. Siu, "Improved image interpolation using bilateral filter for weighted least square estimation," in *Proc. of 17th IEEE Internal Conference on Image Processing*, pp.3297-3300, September 26-29, 2010. [Article\(CrossRef Link\)](#)
- [22] L.-J. Kau and Y.-P. Lin, "Least-squares-based switching structure for lossless image coding," *IEEE Transactions on Circuits and Systems I*, vol. 54, no. 7, pp. 1529-1541, July, 2007. [Article\(CrossRef Link\)](#)
- [23] J.Taquet and C.Labit, "Hierarchical Oriented predictions for resolution scalable lossless and near-lossless compression of CT and MRI Biomedical Images," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2641-2652, May, 2012. [Article\(CrossRef Link\)](#)
- [24] P. Roos, M. Viergever, M. van Dijke and J. Peters, "Reversible intraframe compression of medical images," *IEEE Transactions on Medical Imaging*, vol. 7, no. 4, pp. 328-336, December, 1988. [Article\(CrossRef Link\)](#)
- [25] A. Abrardo, L. Alparone and F. Bartolini, "Encoding-interleaved hierarchical interpolation for lossless image compression," *Signal Processing*, vol. 56, no. 3, pp. 321-328, February, 1997. [Article\(CrossRef Link\)](#)
- [26] B. Zeng, M. S. Fu and C. C. Chuang, "New interleaved hierarchical interpolation with median-based interpolators for progressive image transmission," *Signal Processing*, vol. 81, no. 2, pp. 431-438, February, 2001. [Article\(CrossRef Link\)](#)
- [27] S.Kim and N.I. Cho, "Hierarchical prediction and context adaptive coding for lossless color image compression," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 445-449, January, 2014. [Article\(CrossRef\)](#)
- [28] P.S.Babu and S.Sathappan, "Efficient lossless image compression using modified hierarchical

- Forecast and context adaptive System,” Indian Journal of Science and Technology, vol.8, no.34, December, 2015. [Article\(CrossRef Link\)](#)
- [29] M. Weinberger, G. Seroussi and G. Sapiro, “The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS,” *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309-1324, August, 2000. [Article\(CrossRef Link\)](#)
- [30] X. Wu and N. Memon, “Context-based, adaptive, lossless image coding,” *IEEE Transactions on Communications*, vol. 45, no. 4, pp. 437-444, April, 1997. [Article\(CrossRef Link\)](#)
- [31] X. Wu, “Lossless compression of continuous-tone images via context selection, quantization, and modeling,” *IEEE Transactions on Image Processing*, vol. 6, no. 5, pp. 656-664, May, 1997. [Article\(CrossRef Link\)](#)
- [32] Y. Biadgie and K.-A. Sohn, “Speed-up feature detector using adaptive accelerated segment test,” *IETE Technical Review*, vol.33, no.5, pp.492-504, March, 2017. [Article\(CrossRef Link\)](#)
- [33] Y. Xing, D. Zhang, J. Zhao, M. Sun, and W. Jia, “Robust fast corner detector based on filled circle and outer ring mask,” *IET Image Processing*, vol.10, no.4, pp.314-324, April, 2016. [Article\(CrossRef link\)](#)
- [34] Mair, G. D. Hager, D. Burschka, M. Suppa and G. Hirzinger, “Adaptive and generic corner detection based on the accelerated segment test,” in *Proceedings of the 11th European Conference on Computer Vision: Part II*, pp. 183-196, September 5-11, 2010. [Article\(CrossRef link\)](#)
- [35] S. M. Smith, and J. M. Brady, “SUSAN—a new approach to low level image processing,” *International Journal of Computer Vision*, Vol. 23, no. 1, pp. 45-78, May, 1997. [Article\(CrossRef Link\)](#)
- [36] Y. Zhang and D.A. Adjeroh, “Prediction by partial approximate matching for lossless image compression,” *IEEE transactions on image processing*, vol.17, no.6, pp.924-935, June, 2008. [Article\(CrossRef Link\)](#)



Yenewondim Biadgie received his Ph.D. in Computer Engineering in 2012 from Ajou University, South Korea. He was a post-doctoral research fellow at Ajou University from March 2012 to August 2015. He is currently an assistant professor in the department Software and Computer Engineering, at Ajou University South Korea. His research interest includes digital image processing, Computer Vision, Pattern Recognition, Machine Learning and Big Data Analytics.



Min-Sung Kim received her BS in Information Technology from Sungshin Women’s University, South Korea, in 2015. She is currently MS student in the Department of Software, Ajou University, South Korea. Her research interest includes machine learning and data mining approaches.



Kyung-Ah Sohn received her Ph.D. in Computer Science from Carnegie Mellon University, USA, in 2011. She was with the College of Medicine, Seoul National University, as a research professor in 2012. She is currently an associate professor in the Department of Software and Computer Engineering, Ajou University, South Korea. Her research interest includes statistical machine learning and biomedical informatics.