

# A New Approach of Lossy Image Compression Based on Hybrid Image Resizing Techniques

Jau-Ji Shen<sup>1</sup>, Chun-Hsiu Yeh<sup>2,3</sup>, and Jinn-Ke Jan<sup>2</sup>

<sup>1</sup>Department of Management Information Systems, National Chung Hsing University, Taiwan

<sup>2</sup>Department of Computer Science and Engineering, National Chung Hsing University, Taichung

<sup>3</sup>Department of Information Management Systems, Chung Chou University, Taiwan

**Abstract:** *In this study, we coordinated and employed known image resizing techniques to replace the widely applied image compression techniques defined by the Joint Photographic Experts Group (JPEG). The JPEG approach requires additional information from a quantization table to compress and decompress images. Our proposed scheme requires no additional data storage for compression and decompression and instead of using compression code it uses shrunken images that can be read visually. Experimental results indicate that the proposed method can coordinate typical image resizing techniques effectively to yield enlarged (decompressed) images that are better in quality than JPEG images. Our novel approach to lossy image compression can improve the quality of decompressed images and could replace the use of JPEG compression in current image resizing techniques, thus enabling compression to be performed directly in the spatial domain without the need for complex conversion in the frequency domain.*

**Keywords:** *Differential image, Image compression, Image rescaling, Image resolution improvement.*

*Received July 14, 2015; accepted October 24, 2016*

## 1. Introduction

Image compression [17] involves using less data than that in an original image to express image content in a way that reduces storage requirements and decreases network transfer time. Image compression includes both lossless and lossy compression, and lossy compression is currently more broadly applied. The most commonly used techniques include compressed algorithms designed by the Joint Photographic Experts Group (JPEG). Typically, the JPEG image compression standard proposed by Wallace [25] (1991) is applied to compress full-color and grayscale images at a high compression ratio. JPEG encoding takes the raw image file and segments it into 8×8 non-overlapping regions, then uses Discrete Cosine Transformation (DCT) to convert the pixels in each region to a frequency field to produce DC and AC coefficients for quantization and entropy coding processes. The compression quality of a JPEG image is determined by the element value in the quantization table.

JPEG uses a lossy compression technique. Many studies are currently underway to improve compression methods and achieve better quality after decompression. When JPEG regions are quantified by coefficients, entropy coding is used to reduce the size of the data. However, some image regions might still have similarity. Barnsley and Sloan [2] introduced the concept of finding the most similar regions and using code to enter them into the quantization table or using the DCT similarity of the coefficients in these regions

to improve JPEG code quality. Rufai *et al.* [18] used singular value decomposition and wavelet difference reduction, both of which offer higher compression ratios and better decompression quality than the JPEG 2000 compression standard, which enhanced the resizing and editing features of JPEG.

In this paper, we propose a method, based on simple concepts but a unique perspective, for improving image quality. When an image is compressed using techniques such as a quantization table, the original file is converted into compressed code. However, the compressed code cannot determine if it adequately represents the original image, and therefore, the image must be decompressed to compare its quality with the original. In the process of image compression, the original image is shrunk without degrading its quality to an unacceptable level. As such, image compression can be seen as shrinking an image into not simply a set of codes but a clearly discernible smaller version of the original. Similarly, the decompression process is essentially enlarging the same image. Many people use image processing software, e.g., Photoimpact, Photoshop, or Matlab image design software, to enlarge/shrink images to different sizes. Researchers have proposed many image enlarging techniques such as interpolation, Neighboring Pixels Average Value (NPAV), Iterative Curve-Based Interpolation (ICBI), Edge-Directed Interpolation (EDI), and New Edge-Directed Interpolation (NEDI), which we describe in section 2 below. In our proposed method, we combine current image resizing techniques to

replace the most common JPEG image compression technique, as illustrated in Figure 1.

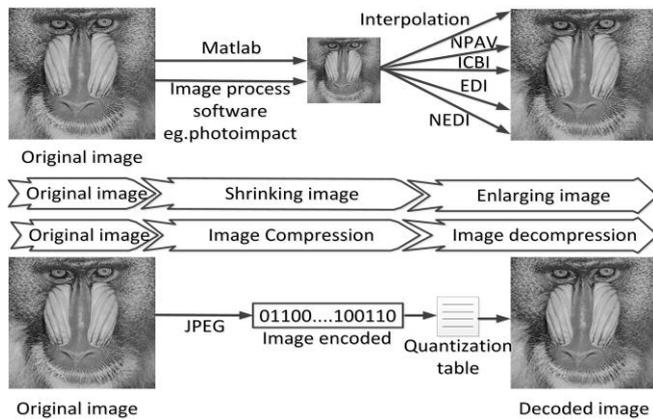


Figure 1. Corresponding image resizing and image compression techniques.

Our proposed method (upper part of Figure 1) does not require the storage of extra data (e.g., quantization tables) and images can be restored using simple calculations. Pixel loss is inevitable during compression, which leads to an inability to completely restore images through decompression, which then leads to image distortion. Many studies of image enlarging techniques [3, 7, 23] have improved image quality during the resizing of high-resolution images by using continuous image data from multiple images and extracting regions with similar signature values. With this approach, image regions are continuously repaired to achieve a restoration variance that is close to the original in building a high-resolution image. In this study, we primarily use the concept of a differential image to effectively consolidate current resizing techniques and achieve higher quality image compression and decompression. The remainder of this paper is organized as follows. We discuss related work in section 2 and describe our proposed methods and algorithms in section 3. In section 4, we present our experimental results and we present our conclusions and future works in section 5.

## 2. Related Work

As shown in Figure 1, image compression and decompression is essentially shrinking and enlarging an image. Shrinking involves skillfully discarding some pixel values in the original image to obtain a smaller version of the image. Therefore, when shrinking an image, the quality of the original is inevitably reduced. In this case, while the pixel values of the shrunken image display a smaller version ratio of the original image, it also critically influences the image quality after decompression. Hence, if the shrunken image can better represent the original, the quality of the enlarged image can be improved.

In this study, we shrank (compressed) tested images by first segmenting  $n \times n$  non-overlapping regions.

When the compression rate was 25%, the size of each region was  $2 \times 2$ , and when the compression rate was 6.25%, the size of each region was  $4 \times 4$ . We selected the following three shrinking techniques, as shown in Figure 2:  $V_1 = (P_1 + P_2 + \dots + P_{n \times n}) / (n \times n)$  (i.e., average block Figure 2.a), the top-left region's pixel value (Figure 2-b), and the lower-right region's pixel value (Figure 2-c).

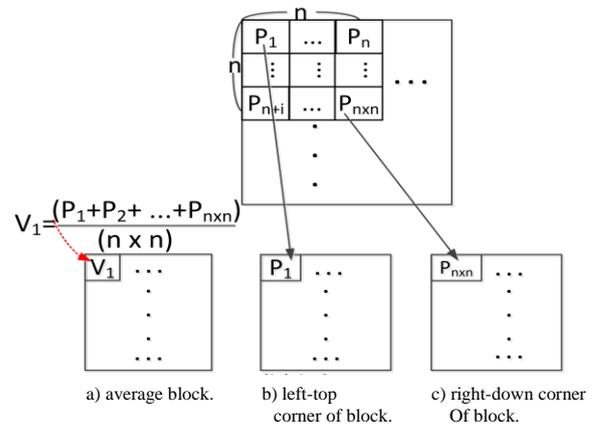


Figure 2. Three types of image compression.

Image enlarging, which plays a vital technical role in image compression and decompression, uses known pixels in the original image to calculate the unknown pixels in the enlarged image. Image enlarging techniques have been proposed in many studies, and the most commonly used are interpolation methods [8, 9, 13, 14, 16, 21] in which the values of known pixels are used to predict the values of unknown pixels (Figure 3). Image resizing techniques, a crucial research topic, involve basic image processing operations [9, 22]. Interpolation techniques can be divided into two types: non-adaptive and adaptive interpolation.

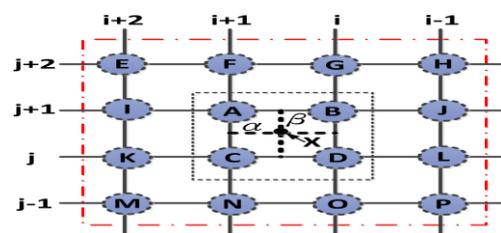


Figure 3. Non-adaptive interpolation.

The more commonly used non-adaptive approaches include nearest-neighbor interpolation [21], bilinear interpolation [9], and bicubic interpolation [13]. Nearest-neighbor interpolation uses the distance of four nearby points with the closest pixel values to distance  $X$  as the value after enlarging. Bilinear interpolation uses the location of four neighboring pixels to obtain the unknown pixel, by the following:  $X = (1 - \beta) \times (1 - \alpha) \times A + \alpha \times B + \beta \times (1 - \alpha) \times C + \alpha \times D$ . Bicubic interpolation uses contributions from 16 existing neighboring pixels to calculate point  $X$  when enlarging to obtain greater computational precision. Its core

function  $h_c(X)$  is as follows:

$$h_c(x) = \begin{cases} 1 - 2|x|^2 + |x|^3, & 0 \leq |x| < 1 \\ 4 - 8|x| + 5|x|^2 - |x|^3, & 1 \leq |x| < 2 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$X = \sum_{m=1}^2 \sum_{n=1}^2 x_{i+m, j+n} h_c(n - \beta) h_c(\alpha - m) \quad (2)$$

Shen *et al.* [20] proposed the NPAV method, in which A, B, C, and D are four known pixel values in the original image, and X is the unknown pixel in the enlarged image, whose value is the average of these known four pixel values. However, non-adaptive approaches do not consider the characteristics of image edges, so the edges are prone to blurring and can obscure the presence of artifact problems.

To improve image quality, many studies have exploited the features of image edges, and several different adaptive interpolation algorithms have been proposed (Figure 4). EDI was first proposed by Allebach and Allebach and Wong [1] in 1996, who detected the direction of edges during image enlarging. NEDI, proposed by Li and Orchard [14], first determines the borders of an image and then performs interpolation along the borders to prevent blurring. ICBI, proposed by Giachetti and Asuni [6, 7] in 2008 and 2011, is an adaptive edge interpolation method that uses edge direction to perform image enlarging and allows the enlarged image to retain most of its original features, providing a more natural rendering of image texture. Lai *et al.* [15] proposed an adaptive image interpolation algorithm that allows enlarged images to retain a certain level of image texture and superior image quality. All these algorithms are forms of adaptive interpolation. Non-adaptive interpolation algorithms are useful for treating smooth images, whereas adaptive interpolation algorithms are useful for treating images with obvious edges.

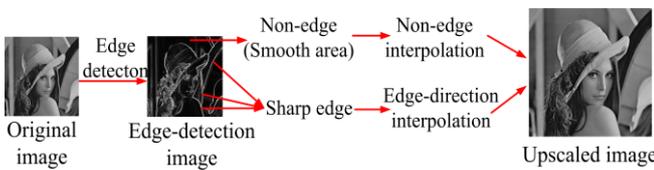


Figure 4. Adaptive edge interpolation.

In addition to the above techniques, single-image super-resolution high-quality image effects [3, 4, 7, 23] have been the focus of image research, which mainly utilize image processing techniques to produce high-resolution images that more closely represent real-world images using current image data such as multimedia video, astronomy images, and medical imaging. Interpolation is used in image enlarging to increase image resolution or create super-resolution images by reconstructing high-resolution images from continuous low-resolution images or extracting image data from similar scenes. [3, 4, 5] used single-image

nearest-path computing and local self-examples from an assigned filter bank to create a high-resolution image amplification technique. The Iterative Back Projection (IBP) algorithm, proposed by Irani and Peleg [10], takes a distorted image and uses repeated simulated low-resolution images. Figure 5 illustrates the error generated by using an observed low-resolution image to reconstruct high-resolution images. IBP uses continuous iteration to adjust image pixel values to achieve a high-resolution image, as shown in Figure 6. The high-resolution iterative formula of the IBP method can be expressed as follows:

$$X^{(n+1)} = X^{(n)} + \frac{1}{k} \sum_{k=1}^k T_k^{-1} ((y_k - y_k^n) \uparrow S) \times h^{BP} \quad (3)$$

Where  $h^{BP}$  is the back projection function,  $T_k^{-1}$  is the geometric antitransform function,  $\uparrow S$  is the up-sampling procedure,  $y_k^{(n)}$  is the observed image k of the  $n_{th}$  iterative process, and  $X^{(n)}$  is the high-resolution image obtained after the  $n_{th}$  iterative process.

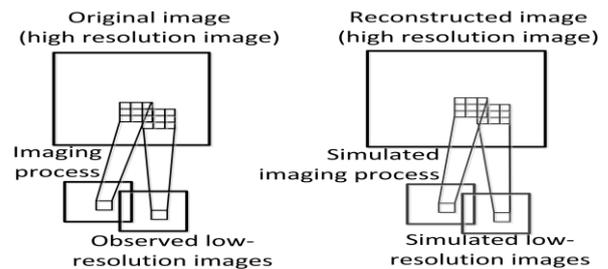


Figure 5. Relative relationship between high- and low-resolution images [10].

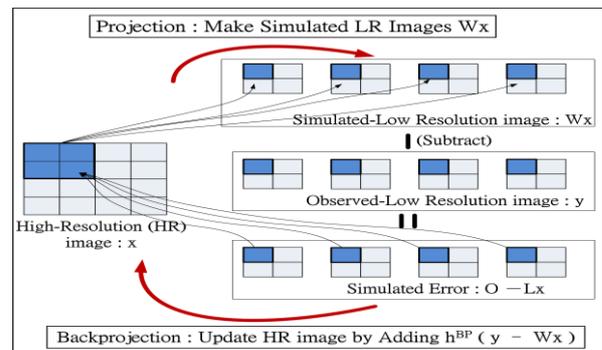


Figure 6. IBP concept [19].

All image enlarging/shrinking or compression/decompression image processing techniques strive to achieve higher image restoration quality. Currently, the JPEG compression method is the most common image processing method used. Therefore, in this study, we performed simple comparisons of JPEG compression results with those of other image enlarging techniques. Using the Lena image (256x256) as an example, we used the JPEG method to conduct 25% and 6.25% image compression and decompression. We then compared the resulting shrunken images with the results of other image enlarging techniques by these compression rates. When

the compression rate was 25%, the JPEG image quality was 29.64 dB, and when we used bicubic interpolation for image enlarging, the resulting Peak Signal-to-Noise Ratio (PSNR) of 29.94 dB was better than that of JPEG, as shown in Table 1. When the compression rate was 6.25%, the PSNR of the JPEG image was 24.58 dB. Table 2 shows the image qualities achieved by the other image enlarging techniques.

Table 1. PSNR comparison of JPEG and other image enlarging methods; Lena image compression/decompression of 25%.

LENA image compression/decompression of 25%							
Upscaling method	JPEG	Bilinear	Bicubic	NAPV	EDI	ICBI	NEDI
PSNR (dB)	29.64	29.06	29.94	27.57	28.72	27.76	27.38

Table 2. PSNR comparison of JPEG and other image enlarging methods; Lena image compression/decompression of 6.25%

LENA image compression/decompression of 6.25%							
Upscaling method	JPEG	Bilinear	Bicubic	NAPV	EDI	ICBI	NEDI
PSNR (dB)	24.58	24.28	24.63	22.46	23.36	22.34	22.43

With respect to quality, the results from the existing image enlarging techniques are inferior to those of JPEG. Therefore, in this study, we used a concept similar to IBP to combine various image compression/decompression techniques to achieve higher image quality after enlarging (decompression).

### 3. Proposed Scheme

In general, when images are enlarged, image quality will differ based on the enlarging technique employed. Zigzagging or blurring in reconstructed images will be more apparent at higher compression ratios. However, different enlarging techniques produce different characteristics in enlarged images. To effectively determine the lost-image characteristics of various image processing methods, we considered images that had been enlarged by different methods as an important data source for image reconstruction. Regardless of how many times an image is compressed and decompressed, the original shrunken image retains many characteristics of the high-resolution image. When considering how to increase the quality of enlarged images without increasing the processing requirements, we used the Revised IBP (RIBP) concept shown in Figure 7. When an image is processed, the quality of the enlarged image can be improved. Our proposed method primarily involves shrinking the original image into an observed image and using different enlarging techniques to produce multiple enlarged images. Then, for a reconstructed image, we use these enlarged images to reconstruct an enlarged image whose image quality has not been adjusted (as shown in Figure 5). Shrinking the reconstructed image creates a simulated image, and we then use the IBP

concept to create a differential image that is based on the differences between the observed and simulated images. We then use this back-projection to adjust the quality of the reconstructed image.

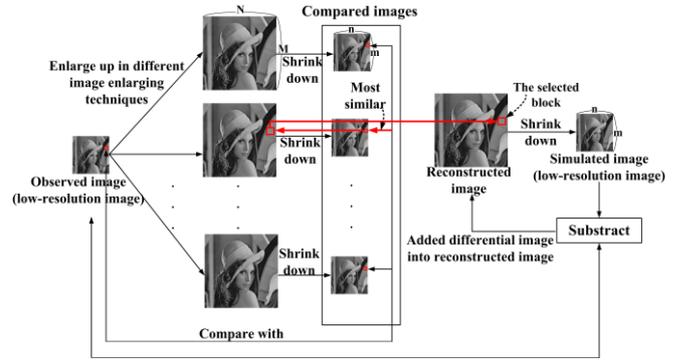


Figure 7. Frequent query patterns.

The proposed method addresses four major issues:

1. How to use shrinking techniques to obtain a Tdown (O) (see Table 3 for term definitions) that retains the original image’s pixel value.
2. How to use image Tdown(O) to restore a compressed image.
3. How to use RIBP to obtain pixel values lost in compression/decompression processes,
4. How to use a differential image D to improve the quality of the enlarged image by reconstructing lost image characteristics.

Figure 8 shows the conceptual schema of the proposed method. The key procedures involve finding suitable image shrinking techniques to replace image compression and finding a suitable region to construct images in the enlarging techniques. The pixel values of this region are used to reconstruct and enlarge images.

In terms of image compression, our proposed method employs the Tdown algorithm assuming a compression rate of  $\delta\%$ . If we assume an original image O of size  $N \times M$  pixels, we can use the Tdown algorithm to shrink image O to  $n \times m$  pixels, where  $n = N \times (\sqrt{\delta\%})$  and  $m = M \times (\sqrt{\delta\%})$ . Then, N and M are the sizes of the original image O, and n and m define the size of the small Tdown(O) image. In the following, we describe the proposed decompression algorithm.

Table 3. Summary of terms.

Term	Description
O	Original image
Tup	Image enlarging algorithm
Tdown	Image shrinking algorithm
Tup(O)	Image O is enlarged (decompressed) by the Tup algorithm (high-resolution image)
Tdown(O)	Image O is shrunk (compressed) by the Tdown algorithm (low-resolution image)
D	Differential image (difference between the observed and simulated images)
Tup(D)	D is enlarged with the Tup algorithm to obtain an enlarged differential image

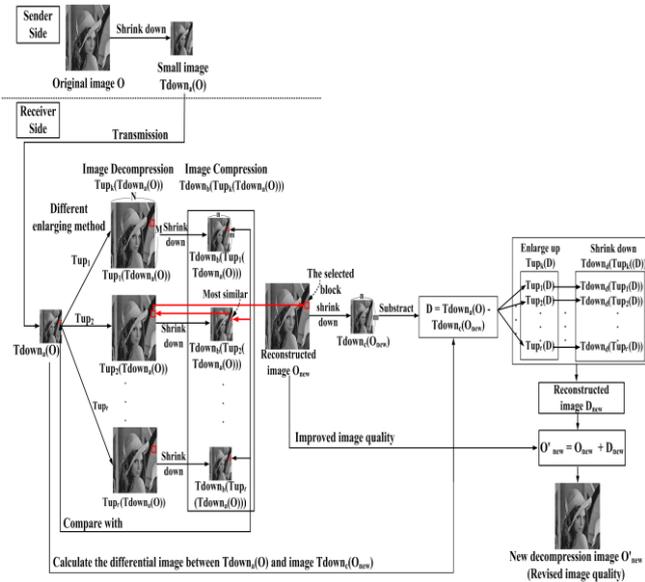


Figure 8. Conceptual schema of the proposed method.

#### Decompression Algorithm 1:

*Input:*  $Tdown_a(O)$  image of  $n \times m$  pixels; compression rate  $\delta\%$

*Output:* Better quality decompressed image  $O'_{new}$

**Step 1:** Decompress image  $Tdown_a(O)$  with different  $Tup_k$  algorithms ( $k$  denotes different enlarging techniques;  $k = 1, \dots, t$ ) to generate different decompressed images  $Tup_1(Tdown_a(O))$ ,  $Tup_2(Tdown_a(O))$ , ...,  $Tup_t(Tdown_a(O))$ ; an  $n \times m$  image is enlarged to an  $N \times M$  image, where  $N = n / (\sqrt{\delta\%})$  and  $M = m / (\sqrt{\delta\%})$ ;

**Step 2:**  $Tdown$  algorithm takes image  $Tup_1(Tdown_a(O))$ ,  $Tup_2(Tdown_a(O))$ , ...,  $Tup_t(Tdown_a(O))$  for shrinking, obtaining  $t$  enlarged image shrunken images  $Tdown_b(Tup_1(Tdown_a(O)))$ ,  $Tdown_b(Tup_2(Tdown_a(O)))$ , ...,  $Tdown_b(Tup_t(Tdown_a(O)))$ ;  $N \times M$  images are shrunk to  $n \times m$  images, where  $n = N \times (\sqrt{\delta\%})$  and  $m = M \times (\sqrt{\delta\%})$ ;

**Step 3:** For  $i = 1$  to  $n$ ;

**Step 4:** For  $j = 1$  to  $m$  ( $n$  and  $m$  define the size of the small image);

**Step 5:** For all  $t$  shrunken images with pixels on  $(i, j)$ ,  $Tdown_b(Tup_1(Tdown_a(O)))_{i,j}$ ,  $Tdown_b(Tup_2(Tdown_a(O)))_{i,j}$ , ...,  $Tdown_b(Tup_t(Tdown_a(O)))_{i,j}$  with the original image's pixels on the shrunken images located at  $(i, j)$  of  $Tdown_a(O)_{i,j}$ , calculate the Euclidean distance between the two;

**Step 6:** Choose the pixel with the smallest distance  $Tdown_b(Tup_{min}(Tdown_a(O)))_{i,j}$ ;

**Step 7:** Take pixel  $Tdown_b(Tup_{min}(Tdown_a(O)))_{i,j}$ , the relative enlarged image and its  $Tup_{min}(Tdown_a(O))$  image region; reconstruct all pixel values in the region in the same locations on the enlarged image  $O_{new}$ ;

**Step 8:** End;

**Step 9:** End; (until shrunken images  $Tdown_a(O)_{i,j}$  and all pixel values are compared to produce a new image  $O_{new}$ );

**Step 10:** Shrink down  $O_{new}$  by  $Tdown$  algorithm, obtain  $Tdown_c(O_{new})$ ;

**Step 11:** Calculate differential image  $D$  by  $Tdown_a(O) - Tdown_c(O_{new})$ ;

**Step 12:** Enlarge up  $D$  with different  $Tup_k$  algorithms ( $k$  denotes

different enlarging techniques;  $k = 1, \dots, t$ ) to generate different decompressed images  $Tup_1(D)$ ,  $Tup_2(D)$ , ...,  $Tup_t(D)$ ; an  $n \times m$  image is enlarged to an  $N \times M$  image, where  $N = n / (\sqrt{\delta\%})$  and  $M = m / (\sqrt{\delta\%})$ ;

**Step 13:** For  $x = 1$  to  $n$ ;

**Step 14:** For  $y = 1$  to  $m$  ( $n$  and  $m$  define the size of the small image);

**Step 15:** For all  $t$  shrunken images with pixel on  $(x, y)$ ,  $Tdown_d(Tup_1(D))_{x,y}$ , ...,  $Tdown_d(Tup_t(D))_{x,y}$  with the original image's pixel on the shrunken images located at  $(x, y)$  of  $(D)_{x,y}$ , calculate the Euclidean distance between the two;

**Step 16:** Choose the pixel with the smallest distance  $Tdown_d(Tup_{min}(D))_{x,y}$ ;

**Step 17:** Take pixel  $Tdown_d(Tup_{min}(D))_{x,y}$ , the relative enlarged image and its  $Tup_{min}(Tdown_d(D))$  image region; reconstruct all pixel values in the region in the same locations on the enlarged image  $D_{new}$ ;

**Step 18:** End;

**Step 19:** End;

**Step 20:** Return image  $O'_{new}$  by taking the sum of differential image  $D_{new}$  and image  $O_{new}$ .

In the above algorithm,  $Tdown$  may or may not be the same image shrinking method in any given instance. In the compression and decompression processes, appropriate enlarging and shrinking techniques must be employed to effectively improve the image quality of the image after compression. Note that  $Tup_1$ ,  $Tup_2$ , ...,  $Tup_t$  refers to different image enlarging techniques.

For the purpose of enhancing the quality of the enlarged images, we compared the zoomed image and the shrunken version of the original image to determine the difference in the pixel values between the two processes. We label the pixel value thus acquired for the differential image, which is essential for correcting the quality of the enlarged images. The differential image in this study refers to the concept of Irani and Peleg [10, 11] whereby the differential value is projected by IBP onto the high-resolution image. The applicability of the differential image is subject to some use constraints in the existing shrink-down approaches with respect to image compression. We applied three shrinking approaches (Figure 2), i.e., the lower-right region (Right), top-left region (Left), and the average (Avg). As illustrated in Figure 4, the corrective image quality operations apply the image resizing techniques four times in the image reductions, i.e.,  $Tdown = \{Tdown_a, Tdown_b, Tdown_c, Tdown_d\}$ .

If  $Tdown$  is applied to reduce the image, the four operations of image reduction must avoid the combinations of {Left, Left, Left, Left} and {Right, Right, Right, Right}. Assuming that the reduction technique used by  $Tdown_a$  is Left, then  $Tdown_b$ ,  $Tdown_c$ , and  $Tdown_d$  must not all use Left for image reduction, because the original image was reduced by Left in the first-time operation. If the shrunken image generated by Left is used for decompression, the resulting image will have the same pixel values as the

original image compressed by Left, thus producing a differential image of 0 and a failure in image quality enhancement. However, if we use {Avg, Avg, Avg, Avg} for image reduction, we will not experience the problem of a zero differential image. Therefore, based on the above concept, if we use Avg to reduce the original image, then we can use Right, Left, and Avg to generate the three shrunken image versions of the enlarged image. Ultimately, the six image enlargement techniques will generate a total of 18 shrunken images versions. We can then compare their pixel values with those of the shrunken image version of the original image to identify the version with pixel values closest to those in the original image.

Using the Lena image as an example, we set the average block defined in Figure 2-a as  $Tdown_a$  for shrinking (compression) to obtain the shrunken image  $Tdown_a(Lena)$ . We then applied the six enlarging techniques described above (i.e.,  $Tup_1, \dots, Tup_6$ ) to obtain six enlarged images:  $Tup_1(Tdown_a(Lena)), Tup_2(Tdown_a(Lena)), \dots, Tup_6(Tdown_a(Lena))$ . We used the average block, top-left region, and lower-right region (Figure 3) shrinking algorithms as  $Tdown_b$  to compress the six enlarged images, resulting in 18 shrunken images  $Tdown_b(Tup_1(Tdown_a(Lena))), Tdown_b(Tup_2(Tdown_a(Lena))), \dots, Tdown_b(Tup_6(Tdown_a(Lena)))$ . For comparison, we then used each pixel value in  $Tdown_a(Lena)$  to rank the 18 shrunken images to determine the most similar pixel values of the shrunken images and the corresponding enlarged image. We then used  $Tdown_b(Tup_{min}(Tdown_a(Lena)))$  to process the shrunken images to determine the relative position in the enlarged image of  $Tup_{min}(Tdown_a(Lena))$  and used the image region for placement on the enlarged image  $Lena_{new}$  in the same location. To improve the decompressed image quality, we employed the RIBP method to obtain another shrunken  $Tdown_c(Lena_{new})$  from the reconstructed enlarged image  $Lena_{new}$ . Then, we calculated the differential values for  $Tdown_a(Lena)$  to obtain a differential image  $D$ . We then applied the six enlarging techniques to enlarge  $D$  to obtain six enlarged images,  $Tup_1(D), Tup_2(D), \dots, Tup_6(D)$ . Taking the six enlarging images processed by  $Tdown_d(D)$ , we ranked the 18 shrunken images  $Tdown_d(Tup_1(D)), Tdown_d(Tup_2(D)), \dots, Tdown_d(Tup_6(D))$  for comparison to determine the most similar pixel values in each with the corresponding differential image. Then, we filled  $D_{new}$  back into the pixel values of image  $Lena_{new}$  to obtain the enlarged image with an adjusted quality  $Lena'_{new}$ . Figure 9 illustrates the above process with the Lena example.

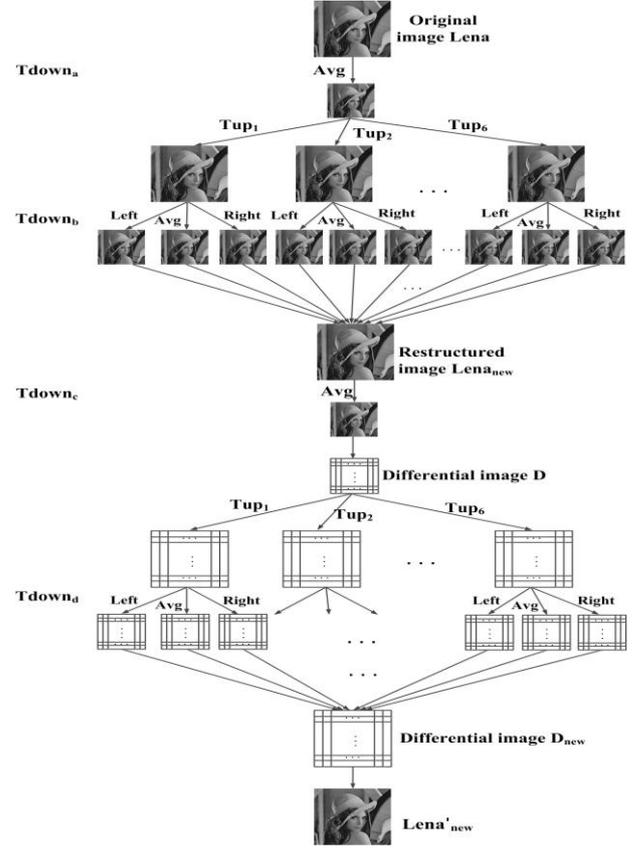


Figure 9. Tree diagram of  $Tdown$ .

#### 4. Experimental Results and Analyses

We selected six grayscale images, i.e., Lena, Babala, Baboon, Boat, GoldHill, and Guitar as samples (Figure 10) to test our proposed algorithm. For the given  $512 \times 512$  test images, we shrank each image to  $256 \times 256$  and  $128 \times 128$  images, and then used the above decompression techniques to decompress them to  $512 \times 512$  images at decompression rates of 25% and 6.25%. For all cases, the size of the decompressed image was  $512 \times 512$  and we could calculate the PSNR of each image with respect to the original test image, and then use the Structural Similarity (SSIM) index method to determine the similarity between the two.

To evaluate image quality, we used the PSNR (a higher PSNR indicates better quality), which is defined as follows:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (4)$$

Mean Square Error (MSE) is defined as follows:

$$MSE = \frac{1}{M \times N} \sum_{p=1}^M \sum_{q=1}^N (X_{p,q} - X'_{p,q})^2 \quad (5)$$

Where  $M \times N$  is the image size,  $X_{p,q}$  is the pixel value at its position in the original image ( $p, q$ ), and  $X'_{p,q}$  is the pixel value at its location in the restored image.

The SSIM [24] is based on the idea that the human visual system is highly adapted to process structural information, and the algorithm attempts to measure the

change in this information between a reference and a distorted image. The SSIM value is a decimal number between -1 and 1, for which a value of 1 indicates that two images have an identical structure.

In our experiment, to decompress images, we applied the six interpolation methods, including three non-adaptive (bilinear, bicubic, and NPAV) and three edge-direction (EDI, ICBI, and NEDI) techniques (i.e.,  $Tup_1, Tup_2, \dots, Tup_6$ ). We obtained the images used for decompression by the bilinear and bicubic approaches from Matlab. We obtained the NPAV, EDI, and NEDI algorithms from published papers [1, 14, 20] and used Matlab to implement them. For the ICBI method, we referenced papers hosted at <http://www.andreagiachetti.it/icbi/>, including related code and experimental images (i.e., the guitar image). To evaluate images after decompression, we used an objective comparison chart to determine the advantages and disadvantages of the proposed method.

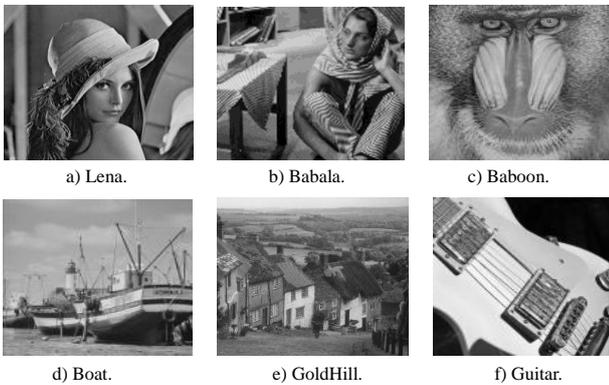


Figure 10. Test images

Figures 11 and 12 show comparisons of the proposed method and JPEG results at the two compression rates, respectively, when applied to the six images. For example, the sample image Lena from the proposed method did not require the storage of any extra data for image compression and decompression and achieved a PSNR of 34.23 dB. The JPEG approach resulted in a PSNR of 32.83 dB. As such, the proposed method improves the PSNR by 1.4 dB (Figure 11). For a compression rate of 6.25%, the proposed method resulted in a decompressed image quality of 29.96 dB, whereas the JPEG approach achieved 27.42 dB. With the increased compression rate, the experimental results show that the proposed method is superior in quality to JPEG (Figure 12).

Using the Boat image as an example, with a compression rate of 25%, the proposed method resulted in an SSIM of 0.9113 when comparing the decompressed image with the original image, and the JPEG method resulted in an SSIM of 0.8875. The proposed method thus achieved a higher SSIM than the JPEG method by 0.0238 (Figure 13). With a compression rate of 6.25%, the proposed method resulted in a SSIM of 0.8129 compared with the original image, whereas the JPEG method achieved 0.7514.

Overall, for both the 25% and 6.25% compression rates, the experimental results demonstrate that images produced using the proposed method are superior in quality to those by JPEG (Figure 14).

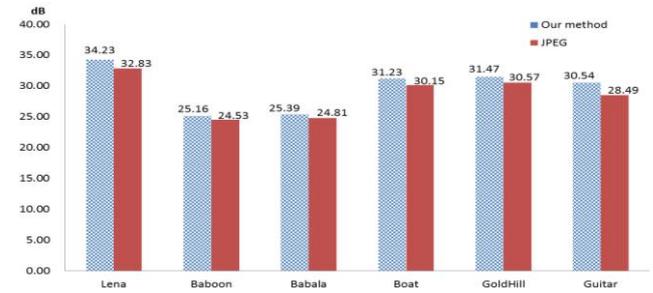


Figure 11. Experimental images; comparison of image quality of the proposed method and JPEG method after decompression (compression rate = 25%).

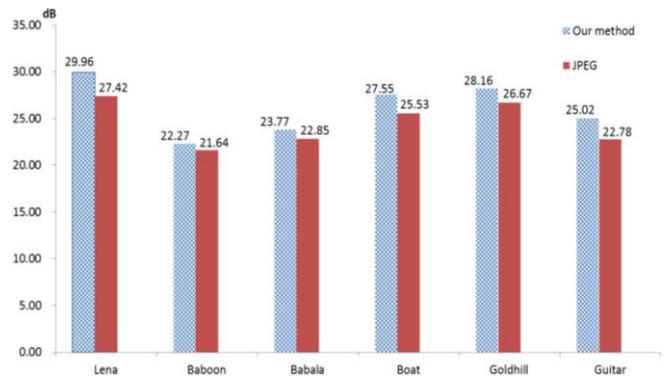


Figure 12. Experimental images; comparison of image quality of the proposed method and JPEG method after decompression (compression rate = 6.25%).

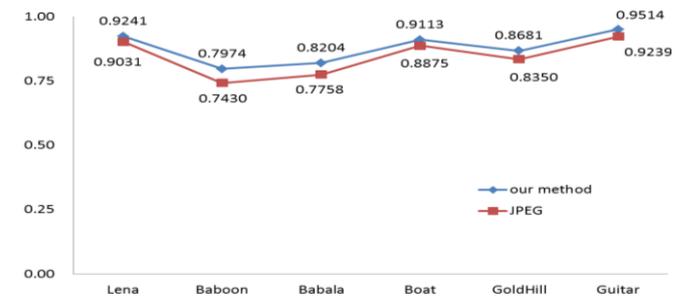


Figure 13. Experimental images; comparison of image SSIM of the proposed and JPEG methods after decompression (compression rate = 25%).

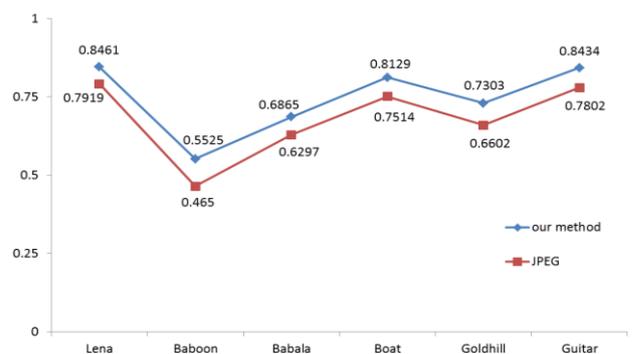


Figure 14. Experimental images; comparison of image SSIM of the proposed and JPEG methods after decompression (compression rate = 6.25%).

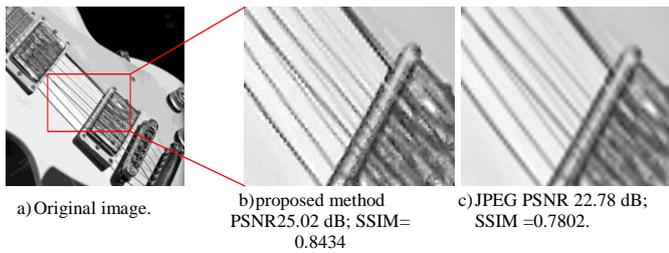


Figure 15. Guitar image (decompression rate 6.25%).

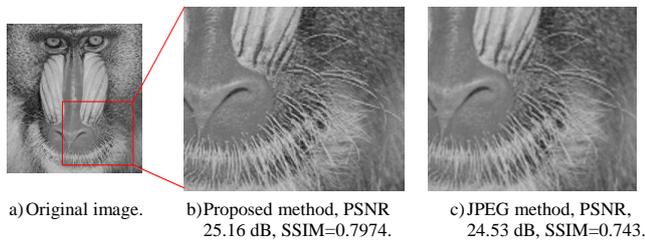


Figure 16. Comparison of a portion of the Baboon image (decompression rate 25%).

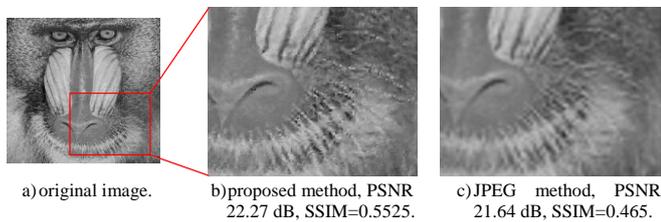


Figure 17. Comparison of a portion of the baboon image (decompression rate 6.25%).



Figure 18. Comparison of the Lena image (decompression rate 6.25%).

For different decompression rates, we hoped that the differential image could repair the features lost during compression. We selected the guitar image [9] for comparison in the following experiment and the obtained image quality values were 30.54 dB and 25.02 dB for decompression rates of 25% and 6.25%, respectively. The compression rate for the guitar image was 6.25%. Figure 15 shows the enlarged image result.

We used partial blocks of the sample image Baboon with compression rates of 25% and 6.25% to compare the differential images produced by the methods described herein. As is evident from the results, using the proposed method with a compression rate of 25%, (for compression and decompression) yielded a PSNR of 25.16 dB, and the JPEG method yielded a PSNR of 24.53 dB. Although these quantitative results do not differ significantly, a subjective assessment indicates that the proposed method result is better than that of the JPEG compression method (Figure 16) in terms of clarity and line presentation. With a compression rate

of 6.25%, when compared with JPEG method results, our method is superior in terms of image quality, and the enlarged image obtained by the proposed method is subjectively clearer than those obtained using the JPEG method (Figure 17). However, the experimental results obtained for the Lena image with a compression rate of 6.25% showed that the difference in PSNR between the proposed and JPEG compression methods was only 0.63 dB. Based on a subjective comparison of these two methods, we consider that for the Lena image, the proposed method yielded brightness and line clarity superior to those of the JPEG method. Note that the JPEG performance for the entire image was subjectively blurry, as shown in Figure 18.

## 5. Conclusions and Future Works

Image resizing and compression techniques are widespread and commonplace. Conceptually, shrinking an image in size is similar to compressing the image and expanding a shrunken image is similar to extracting a compressed image. The JPEG method is the most commonly used image compression technique and has therefore been made available in all current image processing software. We can draw several conclusions from our experimental results. First, the proposed method does not require storage of extraneous data and uses only a simple algorithm to compress and decompress images. Coupled with the RIBP method, our scheme does not create compression code, but rather creates a shrunken image that is visible as the original image. This image is not only a shrunken image of the original but is also a critical component required to restore the image, because its pixel value determines the quality of the restored image.

The results of the six methods, combined with our proposed differential image method, using equal compression ratios (25% and 6.25%) and then decompressing the image showed that the PSNR and SSIM values were superior to those of JPEG decompressed images. The results after compression demonstrate that decompression with the proposed method retains the content of the original image, whereas it was impossible to read the compressed codes in the JPEG results. Therefore, the proposed image quality improvement technique can replace the current JPEG image compression technique. Because color images are composed of three primary color planes, our future work will focus on designing a color image compression method based on the method proposed here.

## References

- [1] Allebach J. and Wong P., "Edge-Directed Interpolation," in *Proceedings of 3<sup>rd</sup> IEEE*

- International Conference on Image Processing*, Lausanne, pp. 707-710, 1996.
- [2] Barnsley M. and Sloan A., "A Better Way to Compress Images," *Byte Archive*, vol. 13, no. 1, pp. 215-223, 1998.
- [3] Freeman W., Jones T., and Pasztor E., "Example-Based Super-Resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56-65, 2002.
- [4] Freedman G. and Fattal R., "Image and Video Upscaling from Local Self-Example," *ACM Transactions on Graphics*, vol. 30, no. 2, pp. 1-10, 2011.
- [5] Freeman W., Pasztor E., and Carmichael O., "Learning Low-Level Vision," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 25-47, 2000.
- [6] Giachetti A. and Asuni N., "Fast Artifact-Free Image Interpolation," in *Proceedings of International Conference British Machine Vision*, Leeds, pp. 123-132, 2008.
- [7] Giachetti A. and Asuni N., "Real-time Artifact-Free Image Upscaling," *IEEE Transactions on Image Processing*, vol. 20, no. 10, pp. 2760-2768, 2011.
- [8] Gonzalez R. and Woods R., *Digital Image Processing*, Prentice Hall, 2008.
- [9] Hu Y., Su B., Chen W., and Lu W., "Image Zooming for Indexed Color Images based on Bilinear Interpolation," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 7, no. 2, pp. 353-358, 2012.
- [10] Iran M. and Peleg S., "Improving Resolution by Image Registration," *CVGIP: Graphical Models Image Processing*, vol. 53, no. 3, pp. 231-239, 1991.
- [11] Iran M. and Peleg S., "Motion Analysis for Image Enhancement: Resolution, Occlusion, and Transparency," *Journal of Visual Communications and Image Representation*, vol. 4, no. 4, pp. 324-335, 1993.
- [12] Kumar M., "Digital Image Processing," in *Proceedings International Conference Satellite Remote Sensing and GIS Applications in Agricultural Meteorology*, Dehradun, pp. 81-102, 2003.
- [13] Keys R., "Cubic Convolution Interpolation for Digital Image Processing," *IEEE Transactions Acoustics, Speech and Signal Processing*, vol. 29, no. 6, pp. 1153-1160, 1981.
- [14] Li X. and Orchard M., "New Edge-Directed Interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1521-1527, 2001.
- [15] Lai Y., Tzeng C., and Wu H., "Adaptive Image Scaling based on Local Edge Directions," in *Proceedings of International Conference on Intelligent and Advanced Systems*, Kuala Lumpur, pp. 1-4, 2010.
- [16] Maeland E., "On the Comparison of Interpolation Methods," *IEEE Trans Med Imaging*, vol. 7, no. 3, pp. 213-217, 1988.
- [17] Rawat C. and Meher S., "A Hybrid Image Compression Scheme using DCT and Fractal Image Compression," *The International Arab Journal of Information Technology*, vol. 10, no. 6, pp. 553-562, 2013.
- [18] Rufai A., Anbarjafari G., and Demirel H., "Lossy Image Compression using Singular Value Decomposition and Wavelet Difference Reduction," *Digital Signal Processing*, vol. 24, pp. 117-123, 2014.
- [19] Park S., Park M., and Kang M., "Super-resolution Image Reconstruction: A Technical Review," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21-36, 2003.
- [20] Shen J., Yeh C., and Jan J., "The Light-Weighted Coding Schemes for Images Sharing on Mobile Devices," in *Proceedings of 15<sup>th</sup> International Conference Signal and Image Processing*, Banff, pp. 437-440, 2013.
- [21] Teoh K., Ibrahim H., and Bejo S., "Investigation on Several Basic Interpolation Methods for the use in Remote Sensing Application," in *Proceedings of IEEE Conference on Innovative Technologies in Intelligent Systems and Industrial Applications*, Cyberjaya, pp. 60-65, 2008.
- [22] Takeda H., Farsiu S., and Milanfar P., "Kernal Regression for Image Processing and Reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349-366, 2007.
- [23] Wang J., Liang K., Chang S., and Chang P., "Super-resolution Image with Estimated High Frequency Compensated Algorithm," in *Proceedings of 9<sup>th</sup> International Symposium on Communications and Information Technology*, Incheon, pp. 175-180, 2009.
- [24] Wang Z., Bovik A., Sheikh H., and Simoncelli E., "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, 2004.
- [25] Wallace G., "The JPEG Still Picture Compression Standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30-44, 1991.



**Jau-Ji Shen** received his Ph.D. degree from National Taiwan University in 1988, and the master degree from National Chung Hsing University in 1984. His research interests include digital image, software engineering, information security, and data base technique. His work experiences include the Director at National Formosa University Library and the Associate Dean at Chaoyang University of Technology. Now, he is a professor in the Department of Management Information Systems, National Chung Hsing University.



**Chun-Hsiu Yeh** received the B.S. degree in management information systems from Chaoyang University of Technology, Taichung, Taiwan in 1997, and the M.S. degree in information management from Chaoyang University of Technology, Taichung, Taiwan in 2001. She is currently pursuing the Ph.D. degree in information engineering from National Chung Hsing University in Taiwan. Her current research interests include image processing and image restore.



**Jinn-Ke Jan** was born in Taiwan in 1951. He received the B.S. degree in physics from Catholic Fu Jen University in 1974 and the M.S. degree in information and computer science from the University of Tokyo in 1980. He studied Software Engineering and Human-Computer Interface in the University of Maryland, College Park, MD, during 1984-1986. He is presently a professor and the chairman of the department of Computer Science & Engineering at National Chung Hsing University. He is currently a member of the Chinese Association for Information Security. From 1995 to 1997, he was the Director of the Counseling Office for Overseas Chinese and Foreign Students. From 1997 to 2000, he was the Director of the Computer Center at National Chung Hsing University. From 2005 to 2008, he was the Head of the department of Computer Science & Engineering. His research interests include computer cryptography, human factors of designing software and information systems, ideograms I/O processing, data structures and coding theory.