TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 1, February 2020, pp. 394~406
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i1.12837

Halftoning-based BTC image reconstruction using patch processing with border constraint

Heri Prasetyo[1], Chih-Hsien Hsia[2], Berton Arie Putra Akardihas[3]
[1,3]Department of Informatics, Universitas Sebelas Maret (UNS), Indonesia
[2]Department of Computer Science and Information Engineering, National Ilan University, Taiwan

Article Info

Article history:

Received Apr 4, 2019
Revised Nov 19, 2019
Accepted Nov 30, 2019

Keywords:

Error diffusion
Halftoning-BTC
Order dithering
Sparse representation

ABSTRACT

This paper presents a new halftoning-based block truncation coding (HBTC) image reconstruction using sparse representation framework. The HBTC is a simple yet powerful image compression technique, which can effectively remove the typical blocking effect and false contour. Two types of HBTC methods are discussed in this paper, i.e., ordered dither block truncation coding (ODBTC) and error diffusion block truncation coding (EDBTC). The proposed sparsity-based method suppresses the impulsive noise on ODBTC and EDBTC decoded image with a coupled dictionary containing the HBTC image component and the clean image component dictionaries. Herein, a sparse coefficient is estimated from the HBTC decoded image by means of the HBTC image dictionary. The reconstructed image is subsequently built and aligned from the clean, i.e. non-compressed image dictionary and predicted sparse coefficient. To further reduce the blocking effect, the image patch is firstly identified as "border" and "non-border" type before applying the sparse representation framework. Adding the Laplacian prior knowledge on HBTC decoded image, it yields better reconstructed image quality. The experimental results demonstrate the effectiveness of the proposed HBTC image reconstruction. The proposed method also outperforms the former schemes in terms of reconstructed image quality.

Corresponding Author:

Heri Prasetyo,
Department of Informatics,
Universitas Sebelas Maret (UNS),
Surakarta, Indonesia.
Email: heri.prasetyo@staff.uns.ac.id

1. INTRODUCTION

Block truncation coding (BTC) and its variants have been playing an important role on image processing and computer vision applications, such as image/video compression [1-3], image watermarking [4, 5], data hiding [3, 6], image retrieval and classification [7-10], image restoration, [11-13] etc. Many efforts have been focused on further improving the performance of BTC and its variants, including the computational complexity reduction, decoded image quality improvement, and its applications, as reported in [1, 2, 7-9, 12, 13]. The BTC-based image compression finds a new representation of an image to further reduce the storage requirement, and achieve a satisfactory coding gain. It is classified as a lossy image compression, in which a given image block is processed to yield a new representation consisting of two color quantizers and the corresponding bitmap image. The two color quantizers and bitmap image produced at the encoding stage are then transmitted to the decoder. The typical BTC techniques determine the two color quantizers, namely low and high means, by maintaining the intrinsic statistical properties of an image such as

first moment, second moment, etc. The corresponding bitmap image is simply obtained by applying the thresholding operation on each image block with the mean value of this processed block. This bitmap image consists of two binary values (0 and 1), in which the value 0 is replaced with the low mean value, whereas the value 1 is substituted with high mean value, in the decoding process.

The BTC-based image compression can provide low computational complexity, however, it often suffers from the blocking effect and false contour issues [1, 2, 7, 12]. These problems make it less satisfactory for human perception. A new type of technique, namely halftoning-based block truncation coding (HBTC), has been proposed to overcome these problems. Figure 1 depicts the schematic diagram of HBTC technique. The HBTC substitutes the BTC bitmap image with the halftone image produced from specific image halftoning methods such as void-and-cluster halftoning [1], dithering approach [7-9], error diffusion technique [2, 10, 11], dot diffused halftoning [12], etc. This technique compensates the false contour and blocking effect problems by enjoying peculiar dither effects from the halftone image. In addition, the HBTC offers a lower computational complexity during the process of the two color quantizers determination. Herein, the color quantizers are simply replaced with the minimum and maximum pixel values found in an image block. Two popular HBTC methods, namely the ordered dithered block truncation coding (ODBTC) [1] and error diffusion block truncation coding (EDBTC) [2], have been developed and reported in the literature. The ODBTC and EDBTC change the BTC bitmap image with the halftone image produced from the ordered dithering and error diffused halftoning methods, respectively. Both of the ODBTC and EDBTC schemes yield better image quality compared to that of the classical BTC method as reported in [1, 2]. The two methods can be applied to other image processing and computer vision applications, including low computational image compression [1, 2, 11], content-based image retrieval [7-10], recognition of color building [14, 15], blood image analysis [16], object detection and tracking [17], etc.

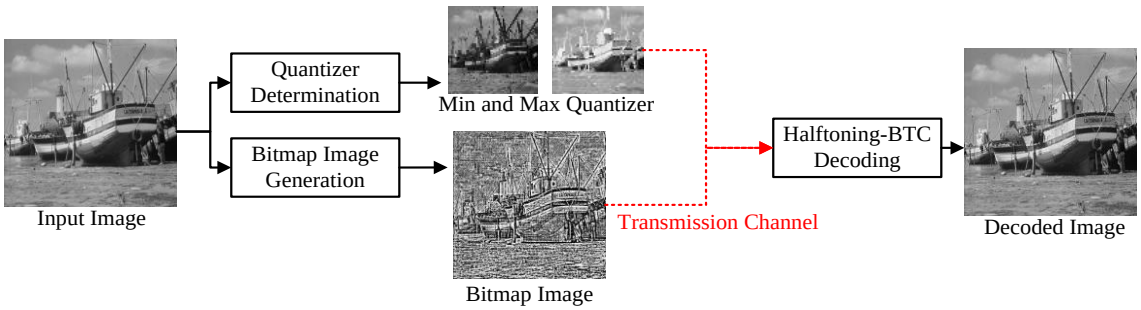


Figure 1. Schematic diagram of the halftoning-based BTC

Although the ODBTC and EDBTC significantly reduce the blocking effect and false contour issues occurred in classical BTC technique, the impulsive noise is always present at considerably high level. To reduce the impulsive noise and mitigate the boundary effect, an additional step can be applied for the HBTC decoded images. The noise filtering is a simple and naïve approach to suppress the appeared impulsive noise, in which a specific window size and kernel value are applied. The Gaussian filter is an example of noise filtering. It performs global filtering, i.e. all pixels are processed in the same manner regardless their statistical intrinsic properties. However, it has a limited effect in reducing the noise levels. An extended Gaussian filtering has been proposed in [13], namely variance classified filtering. This approach applies various kernel functions for various pixels, and the choice of kernel function is determined by the variance within an image block. In this particular method, a set of kernel functions can be iteratively offline-trained using the least-mean-squared (LMS) over various images training set. These set of kernel functions are recorded as a look-up-table (LUT) for further usage. As reported in [13], the variance classified filter yields a significantly better image quality compared to that of the global Gaussian lowpass filtering with the trade-off of higher storage requirement.

The sparse representation learns an over-complete dictionary from a set of image patches as training data [18]. The K-Singular value decomposition (KSVD) sparse representation technique [19, 20] offers stable results over the existing convex relaxation approaches for the sparse coding learning and approximation. As reported in [19], the KSVD approach outperforms the matching pursuit [21], orthogonal matching pursuit [22], basis pursuit [23], and maximum a priori (MAP) approach [24]. The sparse representation has been demonstrated to yield a promising result in several image processing and computer vision applications such as image denoising [19, 25], image restoration [26, 27], etc.

In this paper, a new method on HBTC image reconstruction is developed using the sparsity-based approach. Herein, the impulsive noise levels are reduced by means of coupled dictionaries, in which one dictionary is created from the HBTC images, while the other is learned from clean images (uncorrupted images). In the sparse coding stage, the sparse coefficients are firstly estimated from the decoded image containing high impulsive noise levels. The reconstructed image is then predicted using the clean image dictionary with the predicted sparse coefficients.

The rest of this paper is organized as follows. Section 2 delivers the VQ-based HBTC image reconstruction. Section 3 presents the sparsity-based HBTC image reconstruction. Extensive experimental results are reported at section 4. Finally, the conclusions are drawn at the end of this paper.

2. VECTOR QUANTIZATION-BASED IMAGE RECONSTRUCTION

This section elaborates the patch-based processing for HBTC image reconstruction using vector quantization (VQ) approach. Herein, the image patch refers to the processed image block. In this method, a single image patch (HBTC decoded image) is replaced with an image patch obtained from VQ codewords based on the closest matching rule [11]. In our proposed method, the closest matching is conducted under the Euclidean distance similarity criterion. The smaller score of Euclidean distance indicates the more similar. The selected image patch effectively improves the quality of HBTC decoded image. Different image patches require different processes based on their properties, i.e. "border" or "non-border" type. Figure 2 shows three possible border processing scenarios/constraints, i.e. horizontal, vertical, and corner borders. As shown in this figure, suppose that there are four different image blocks denoted as two gray image blocks and two white image blocks. These image blocks are produced from the HBTC process independently. Each image block is uncorrelated with the other. The characteristics of one image block is dissimilar with the other block even though they are adjacent neighbors. Herein, an image patch with border constraint refers to a set of pixels which lay on two or several different image blocks. The red part of Figure 2 (a) depicts a set of pixels laying on several image borders. Since these image pixels are in horizontal position, we regard it as an image patch with a horizontal border. Figures 2 (b) and (c) are examples of image patch in vertical direction and corner part, respectively. Image patches excluded in these three border constraints are considered as non-border image patches.

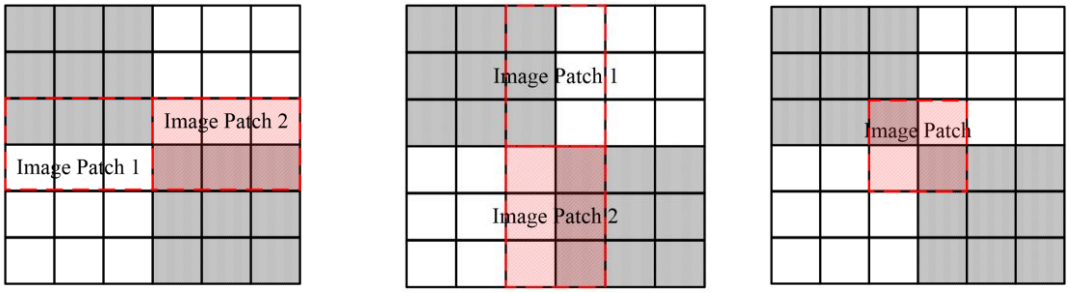


Figure 2. Types of image patch processing: (a) horizontal, (b) vertical, and (c) corner border

The VQ-based method removes the impulsive noise on the border and non-border cases by means of a trained visual codebook, as explained below. This method replaces the HBTC decoded image patch with the visual codebook generated from clean images. The clean images can be simply obtained from natural images, i.e. some images without HBTC processing. Suppose $T = \{i_k(x, y)\}$ are the clean images which are used as the training set, where $k = 1, 2, \dots, K$. The symbol K denotes the number of training image patches. The VQ clustering iteratively processes this clean image set to produce the representative visual codebook $C = \{C_1, C_2, \dots, C_{N_c}\}$, containing N_c codewords.

Let $o(x, y)$ be a halftoned-based BTC decoded image patch. The VQ-based approach firstly identifies whether the image patches are of border or non-border type. If an image patch is classified as non-border, then it is processed with the non-border visual codebook. Conversely, when the image patch is classified as a border patch, then it is processed with either horizontal, vertical, or corner border visual codebook. Figure 3 illustrates the VQ-based approach for HBTC decoded image reconstruction. Figure 4 displays some examples of the visual codebook over horizontal, vertical, and corner borders. After border and non-border image patch determination, this image patch is then processed using VQ-closest matching as defined below:

$$c^* = \underset{c=1,2,\ldots,N_c}{\arg\min} \left\| o(x,y) - C_c^{\theta} \right\|_2^2 \quad (1)$$

where C_c^{θ} denotes the selected visual codebook, and θ indicates whether the current processed image patch is of "border" or "non-border" type. The symbol c^* represents the visual codeword index with the lowest distortion (the most similar) to the image patch $o(x,y)$.

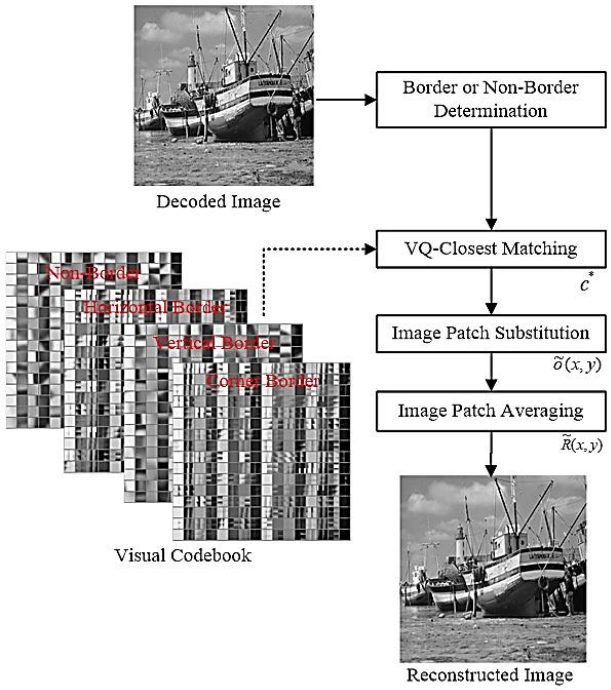
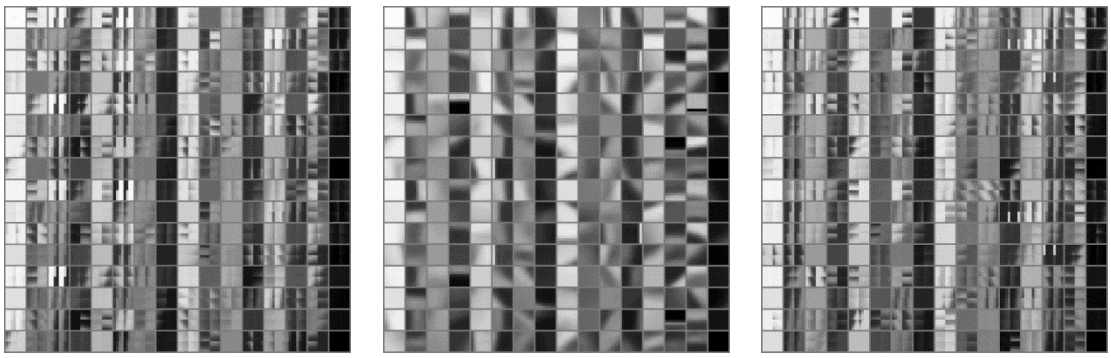


Figure 3. Schematic diagram of VQ-based image reconstruction



(a) (b) (c)

Figure 4. Visual codebook generated from: (a) horizontal, (b) vertical, and (c) corner borders

The selected visual codeword $C_{c^*}^{\theta}$ replaces the image patch $o(x,y)$ by considering border or non-border information. The image patch replacement is denoted as:

$$\tilde{o}(x,y) = C_{c^*}^{\theta} \quad (2)$$

where $\tilde{o}(x,y)$ is the restored image patch on position (x,y). Notably, the VQ-reconstruction is still implemented as a pixel-by-pixel process, not as block-wise process. Consequently, the replacement process is

an overdetermined problem. The averaging process over several restored images patches yields a single HBTC reconstructed image. This process is denoted as:

$$\tilde{R}(x,y) = \frac{\sum \tilde{o}(x,y)}{\sum R(x,y)^T R(x,y)} \quad (3)$$

where $R(x,y)$ denotes the operator of image patch processing [18, 19, 26]. This operator denotes the number of certain pixels employed on the closest matching process and image patch substitution. This operation indicates the number of currently processed pixels used in the image patch computation. The size of matrix $R(x,y)$ is identical to the size of image patch or VQ codewords. The result of $\tilde{R}(x,y)$ in (3) is intuitively identical to that of the simple averaging computation in arithmetics. Thus, the image patch averaging operation is employed at the end of the VQ-based image reconstruction to yield a single value in the overdetermined system.

3. SPARSITY-BASED IMAGE RECONSTRUCTION

This section presents the proposed HBTC image reconstruction methods using a sparsity-based approach. Herein, the image patch is firstly extracted from HBTC decoded image. The sparsity-based approach simply replaces this decoded image patch with the closest match of clean image dictionary. This method also considers the border constraint on HBTC image reconstruction.

The sparsity-based method utilizes two learned coupled dictionaries. An image patch is firstly determined and investigated whether it falls into the border or non-border region as already introduced in the VQ-based processing. Figure 5 displays the HBTC image reconstruction using sparsity-based approach. Figure 6 gives some example of learned dictionary. Similarly, to the VQ-based post processing approach, the sparsity-based method utilizes both non-border dictionary and border dictionary. The learned dictionary is generated from a set of image patches by considering the non-border and border constraints. The border dictionary can be classified as horizontal, vertical, and corner border. Each dictionary contains HBTC decoded dictionary and clean image dictionary.

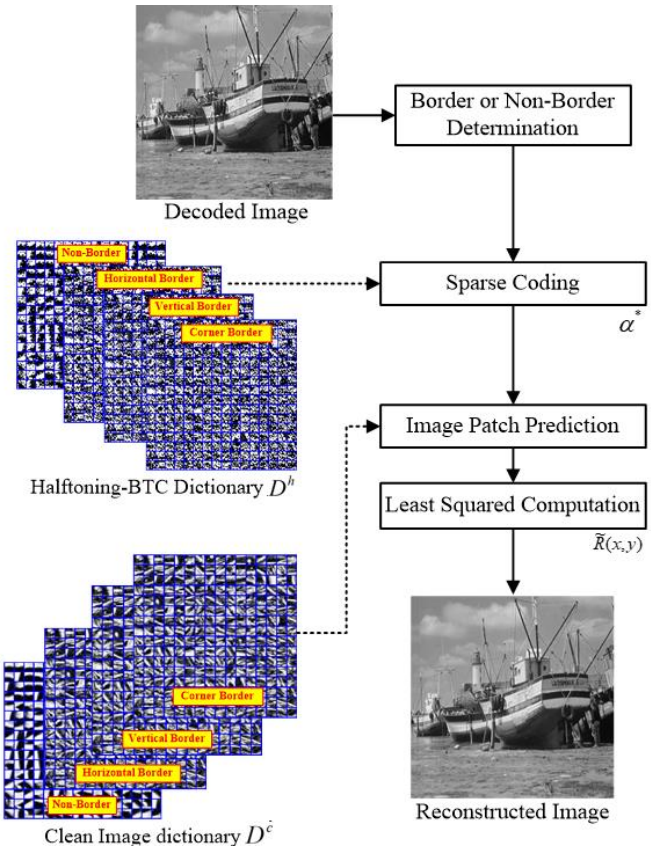


Figure 5. Schematic diagram of sparsity-based image reconstruction

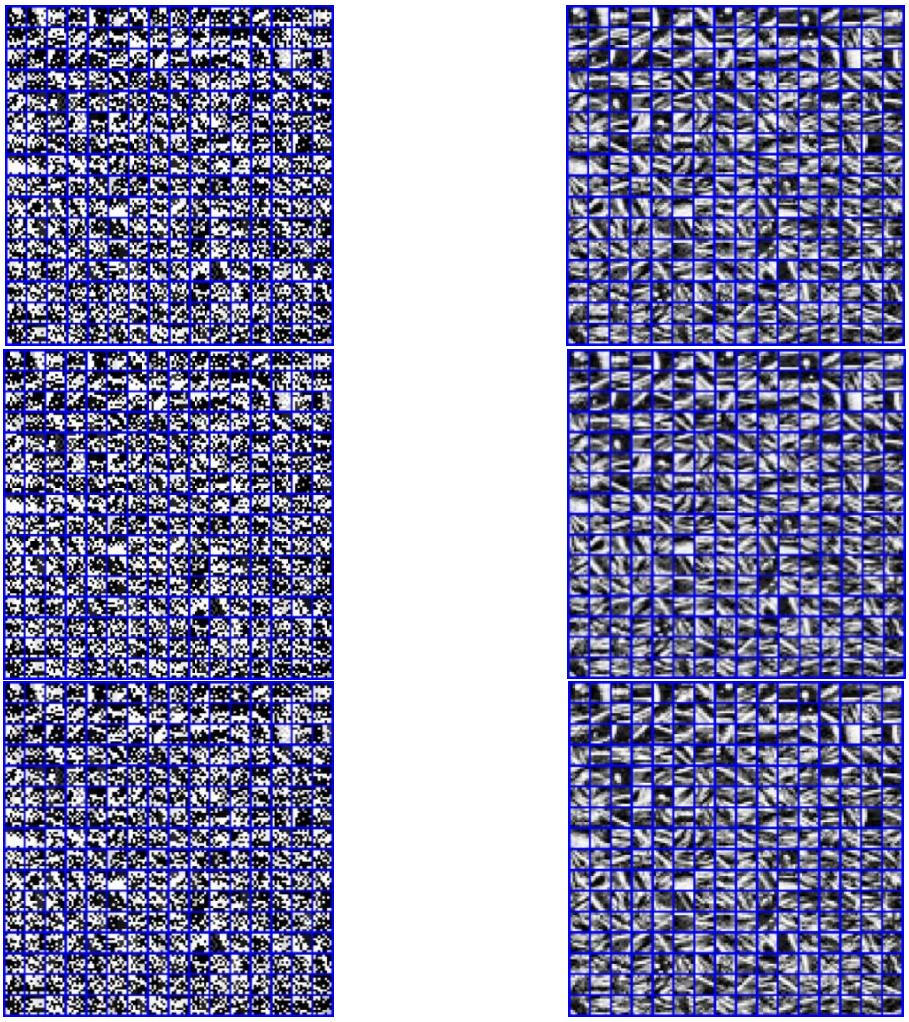


Figure 6. Learned dictionaries generated from horizontal, vertical, and corner border as indicated in the first to the last row. The left and right columns denote the coupled dictionaries generated from halftoned-BTC and clean image patch, respectively

Let $T = \{i_k(x,y), o_k(x,y)\}$ be a training set. This set contains K clean image patches $i_k(x,y)$ and their corresponding HBTC decoded image patches $o_k(x,y)$. The following optimization procedure learns two dictionaries, i.e. clean image dictionary (D^c) and HBTC image dictionary (D^h), from training set T. To obtain the sparse coefficients α, please refer [19] for further detailed explanation of optimization process. This process is denoted as:

$$\min_{D^c, D^h, \alpha} \{\|i - D^c\alpha\|_2^2 + \|o - D^h\alpha\|_2^2\} + \lambda\|\alpha\|_1 \quad (4)$$

where α denotes sparse coefficients and λ is sparse regularization term. Since two dictionaries D^c and D^h share the same spare coefficient α, the optimization in (4) can also be performed as:

$$\min_{D,\alpha}\{\|Y - D\alpha\|_2^2\} + \lambda\|\alpha\|_1 \quad (5)$$

where $D = [D^c; D^h]$ and $Y[i^T, o^T]^T$ denote the concatenated dictionary and concatenated image patch, respectively. The dictionary D contains the clean and HBTC decoded component dictionaries. The matrix Y consists of the clean and HBTC decoded image patches. The KSVD [19] or the other dictionary learning algorithms can be exploited to effectively solve this optimization problem in (5). After deciding the border or non-border region, the next step is to determine the sparse coefficient of the HBTC image patch $o(x,y)$ with a sparse coding step. The sparse coefficient $o(x,y)$ can be predicted with a help of $D^{\theta h}$ as follow:

$$\min_{\alpha^*}\left\{\left\|o(x,y)-D^{\theta h}\alpha\right\|_2^2\right\}+\lambda\|\alpha\|_1 \quad (6)$$

where $D^{\theta h}$ denotes the HBTC decoded image under border or non-border region θ. The symbol α^* represents the predicted sparse coefficient. Several specific algorithms, such as matching pursuit (MP) [21], orthogonal matching pursuit (OMP) [22], Basis Pursuit (BP) [23], maximum a posteriori (MAP) [24], or others, can be exploited to predict α^*. By means of clean image dictionary $D^{\theta c}$ with border or non-border region θ, the HBTC decoded image can be replaced and aligned as follow:

$$\min_{\tilde{R}(x,y)}\left\{\left\|R(x,y)\tilde{R}(x,y)-D^{\theta c}\alpha^*\right\|_2^2\right\} \quad (7)$$

where $\tilde{R}(x,y)$ denotes the HBTC reconstructed image. Similarly, to the VQ-based post processing approach, $\tilde{R}(x,y)$ is an overdetermined system. Thus, the $\tilde{R}(x,y)$ can be solved using the ordinary least squared method (averaging process) as follow:

$$\tilde{R}(x,y)=\{\sum R(x,y)^T R(x,y)\}^{-1}\sum R(x,y)^T D^{\theta c}\alpha^* \quad (8)$$

The image patch is extracted using $R(x,y)$ operator. It can be expected that the HBTC image reconstruction will yield better image quality since it utilizes the clean image dictionary. In addition, the centralized sparse representation [26] can also be embedded into the sparse representation module to further improve the quality of the HBTC decoded image. It is based on the observation that the original image (without HBTC compression) and the HBTC decoded image, $\alpha_x-\alpha_y$, have high probability to fit the Laplace distribution. The prospective readers are suggested to refer to [26] for the full description of centralized sparse representation. Figure 7 depicts the distribution of sparse coding between the original image and HBTC decoded image. Thus, incorporating the Laplacian prior knowledge on the sparse representation framework may produce better quality on the HBTC reconstructed image. The centralized sparse representation [26] can be deployed into the proposed method by considering a border or non-border region.

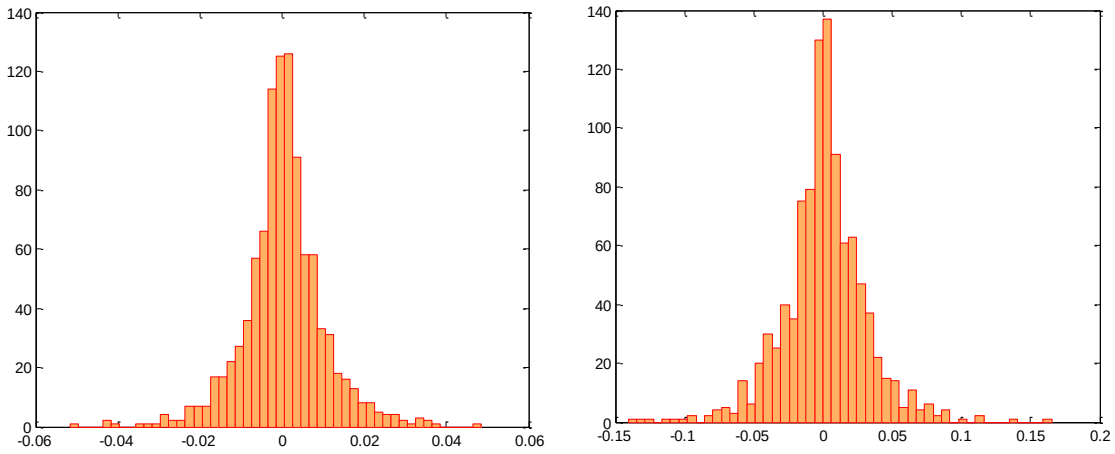


Figure 7. The distribution of sparse coding from HBTC reconstructed image obtained from: (a) ODBTC decoded image, and (b) EDBTC decoded image

4. EXPERIMENTAL RESULTS

This section demonstrates the effectiveness and usefulness of the proposed method. Figure 8 shows the training and testing sets utilized for the experiment. The training set consists of sixteen grayscale images, whereas the testing set contains twenty grayscale images. The training and testing images are with various image conditions such as high-frequency, low-frequency, dark and light brightness, etc. The proposed method also employs a set of training image sets as similarly used in [13] to generate several visual codebooks and dictionaries. The training image set consists of twenty grayscale images over various conditions such as various illumination conditions, different lighting, frequency and activity, etc. Each training image is of size

512×512. We extract image patches from all training images using overlapping strategy. Herein, the overlapping strategy refers to the process in which the current processed image patch is moved to the new position under one pixel different. In this occasion, one image pixel may be processed several times since several image patches employ this pixel.

In a result, we obtain more than five million image patches as training set for codewords generation and codebooks learning in the VQ and sparsity-based approach, respectively. This amount of dataset is sufficient to satisfy the variability aspect of image patch in the training process. Thus, the proposed method can be directly applied to a general case even though the training data is not from the same or specific dataset. The peak-signal-noise-ratio (PSNR) score evaluates the performance of proposed method and former schemes objectively. The PSNR is formulated as:

$$PSNR = 10\log_{10}\frac{255^2}{\frac{1}{MN}\sum_{y=1}^{M}\sum_{x=1}^{N}[I(x,y)-\tilde{R}(x,y)]^2} \quad (9)$$

where $\tilde{R}(x,y)$ and $I(x,y)$ denote the HBTC decoded image and original image, respectively. Higher value of PSNR indicates higher similarity between two images, making it more preferable for human vision. Thorough experiment, the image blocks are set as 8×8 and 16×16 for both ODBTC and EDBTC methods. The image quality improvement after post processing is considered based on the increasing PSNR value obtained after applying the post-processing step to the HBTC decoded image against the case of not applying that step.

(a) (b)

Figure 8. A set of images used in this experiment as: (a) training set, and (b) testing set

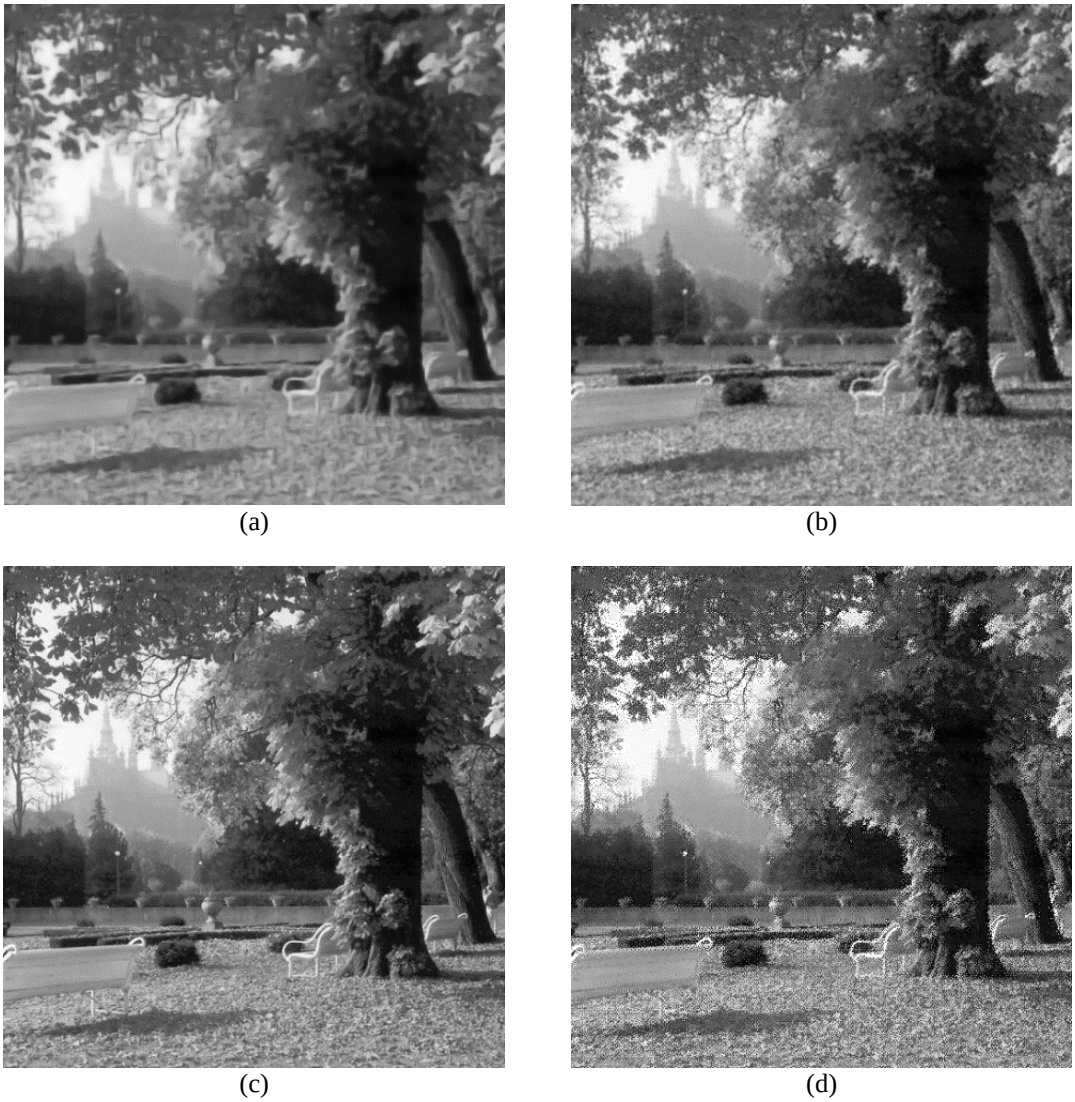
4.1. Effectiveness of the proposed method

This subsection reports the experimental results on the HBTC decoded image reconstruction. The performance of the proposed method is visually judged based on the image quality of the post processing results. The proposed method yields better reconstructed image quality compared to the original HBTC decoded images. Herein, a single image is firstly encoded using either the ODBTC or the EDBTC method to yield two color quantizers and a bitmap image. These two methods simply set the image block size as 8×8. Then, the decoding process is further applied to the two color quantizers and the bitmap image to obtain the ODBTC or EDBTC decoded image. The post processing is conducted on the ODBTC or EDBTC decoded image to further examine the performance of the proposed reconstruction method in terms of specific imaging tasks such image compression, retrieval and classification, etc.

Firstly, we compare the VQ and sparsity-based approaches with other schemes such as lowpass filtering and variance-classified filtering. The lowpass filtering approach employs a Gaussian kernel of size 11×11 with $\mu = 0$ and $\sigma = 1$ to suppress the halftoning impulsive noise [13]. On the other hand, the variance-classified filtering approach uses 13 optimized kernels of size 7×7 [13]. The VQ-based technique exploits the 1024 optimal visual codebooks, while the sparsity-based method utilizes 1024 dictionary atoms. The sparsity-based image reconstruction employs two learned dictionaries. In this approach, the first dictionary is for suppressing the impulsive noise while the other dictionary for reducing the noise occurred in the HBTC

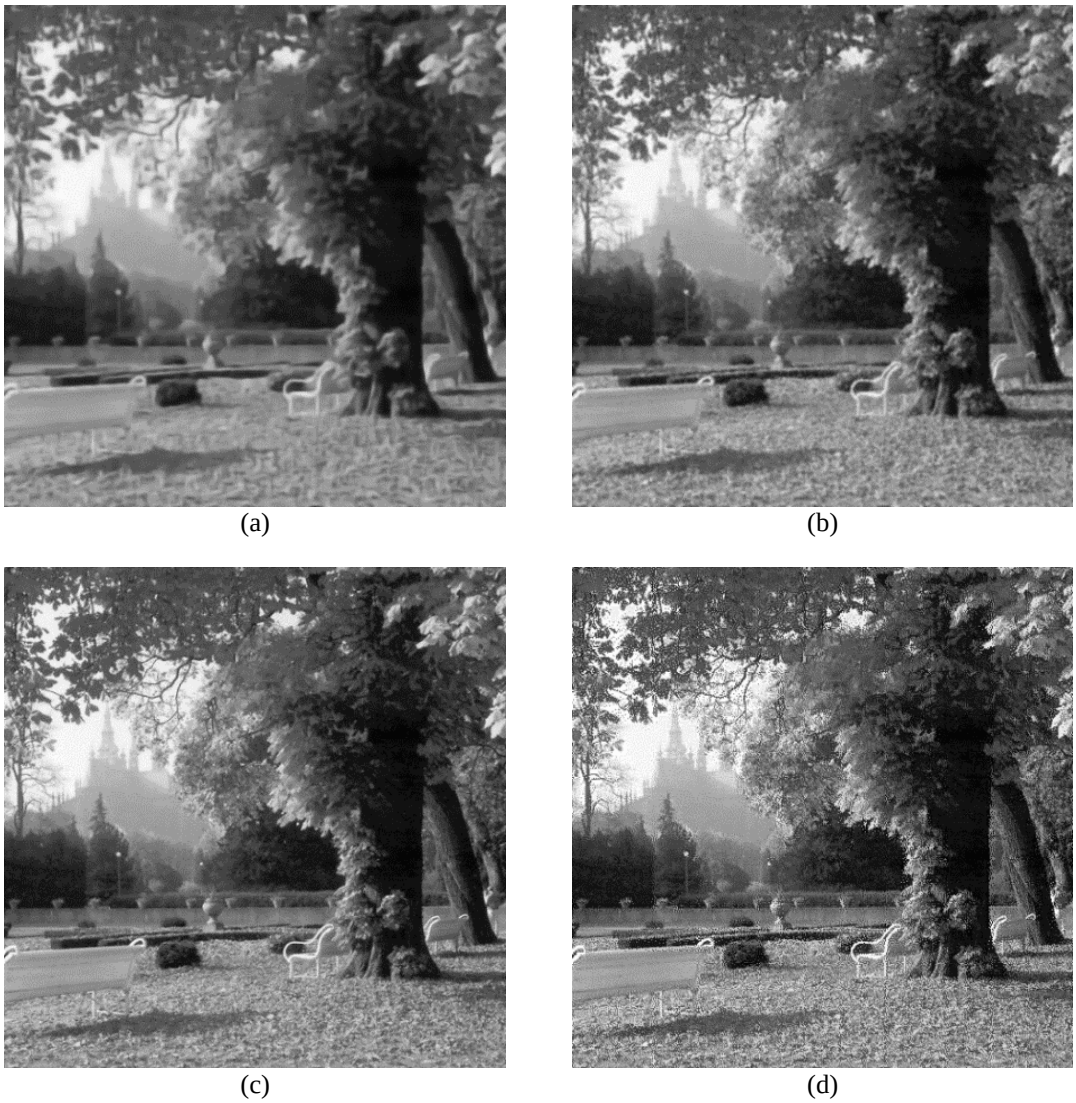
image border. The second dictionary considers the condition of HBTC image border such as mixing horizontal, vertical, and corner borders. For each image border, a coupled dictionary will be produced containing the HBTC decoded and clean image components. The sparse coefficient is predicted from the ODBTC or EDBTC decoded image by means of HBTC dictionary, while the post processing image is composed and aligned using the clean image dictionary with predicted sparse coefficient. The image patch is initially classified as of "border" or "non-border" type.

Figure 9 depicts the image quality comparison after applying the post processing methods on ODBTC decoded image. As shown in this figure, the VQ-based technique produces better image quality in comparison with the lowpass filtering approach. Some impulsive noises are successfully reduced by applying lowpass filtering, but the resolution of the reconstructed image is deteriorated due to the blurring effect of the lowpass filters. The sparsity-based method offers the best ODBTC reconstructed image compared to the other schemes as demonstrated in Figure 9. The post processing technique for EDBTC decoded image is reported in Figure 10. Similarly, to the ODBTC case, the sparsity-based method for EDBTC decoded images outperforms the lowpass filtering and VQ-based post processing. Thus, the sparsity-based method can be regarded as a satisfactory post processing technique for improving the HBTC decoded image quality. Using the proposed method, the HBTC is expected to achieve low computational complexity in the HBTC encoding stage and, at the same time, produce satisfactory decoded image at the HBTC decoding side.



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

Figure 9. Effectiveness of the proposed method in OBTC image reconstruction using: (a) lowpass filtered technique, (b) VQ-based image reconstruction, (c) sparsity-based image reconstruction using two learned dictionaries, and (d) the original ODBTC decoded image with image block size 8×8



(a) (b)

(c) (d)

Figure 10. Effectiveness of the proposed method in EBTC image reconstruction using: (a) lowpass filtered technique, (b) VQ-based image reconstruction, (c) sparsity-based image reconstruction using two learned dictionaries, and (d) the original EDBTC decoded image with image block size 8×8

4.2. Proposed method performance using two dictionaries

In this experiment, the performance of the proposed method and other schemes is objectively assessed in terms of PSNR score. The image block sizes are set as 8×8 and 16×16 for both ODBTC and EDBTC techniques. The sparsity-based method simply exploits two dictionaries, i.e. the impulsive noise dictionary and the image border dictionary. Each dictionary consists of a couple dictionaries (HBTC image dictionary and clean image dictionary). The experimental setting for low-pass and variance-classified filtering remains identical to that of section 4.1.

Table 1 presents the performance of the proposed sparsity-based method on HBTC image reconstruction against low-pass and variance-classified filtering, VQ-based post-processing and absence of post-processing. In this table, the compared value (denoted as PSNR score) is the average value of PSNR over all testing images. The variance classified filtering technique offers better image reconstruction compared to that of the lowpass filtering. Whereas the VQ-based approach is superior compared to filtering methods (lowpass filtering and variance-classified filtering). As shown from this table, the sparsity-based method yields the best performance for ODBTC and EDBTC decoded image over image block size 8×8 and 16×16. Thus, the sparsity-based method with two learned dictionaries is suitable to improve the image quality of HBTC decoded image.

Table 1. Performance of the proposed method with two learned dictionaries in terms of PSNR score, one dictionary is from noisy image patch, and the other dictionary is from image patch extracted on the border

Method	ODBTC		EDBTC	
	8×8	16×16	8×8	16×16
Without Post Processing	19.23	16.10	20.14	16.54
Lowpass Filtered [13]	29.14	28.46	30.76	30.11
Variance Classified [13]	29.48	28.63	31.63	30.82
VQ-Based	33.96	33.89	34.68	34.37
Sparsity-Based	34.77	34.63	34.85	34.80

4.3. Proposed method performance using several dictionaries

This experiment examines the usefulness of the proposed sparsity-based post processing method under four learned dictionaries. In this experiment, the sparsity-based method generates four learner dictionaries, in which all dictionaries are for suppressing impulsive noise, and for dealing with horizontal, vertical, and corner border. Each dictionary is comprised of two dictionaries coupled to each other, namely the HBTC image dictionary and the clean image dictionary. The performance is simply investigated in terms of average PSNR score over all testing images. Table 2 shows the performance comparison between the proposed method using four learned dictionaries and former existing schemes. Herein, the image block size is set at 8×8 and 16×16 for both ODBTC and EDBTC techniques. As it can be seen from this table, the sparsity-based method produces the best average PSNR value compared to the other schemes. The sparsity-based method with four learned dictionaries offers better reconstructed image compared to the sparsity-based scheme with two learned dictionaries. The sparsity-based technique is suitable for the HBTC decoded image reconstruction task.

Table 2. Performance of the proposed method with four learned dictionaries, one dictionary is from noisy image patch, and the other dictionaries are from image patch extracted on the horizontal, vertical, and corner border

Method	ODBTC		EDBTC	
	8×8	16×16	8×8	16×16
Without Post Processing	19.23	16.10	20.14	16.54
Lowpass Filtered [13]	29.14	28.46	30.76	30.11
Variance Classified [13]	29.48	28.63	31.63	30.82
VQ-Based	34.01	33.92	34.74	34.45
Sparsity-Based	34.98	34.71	34.91	34.83

4.4. Proposed method performance incorporating laplacian prior knowledge

In this experiment, the sparsity-based method injects the Laplacian prior knowledge in order to generate four learned dictionaries. Herein, the sparse coding noise is assumed to obey the Laplace distribution [26]. Thus, the centralized sparse representation is more suitable for learning the dictionaries by a given set of training images. The performance is measured in terms of average PNSR value over all testing images. Table 3 reports the performance comparisons, where the proposed method considers the Laplacian prior to generate the learned dictionaries and to perform the closest matching between the HBTC decoded image patch with the learned dictionaries. The proposed method with Laplacian prior outperforms the former existing schemes as reported in Table 3. The proposed sparsity-based method with Laplacian prior also offers better average PNSR value compared to that of the sparsity-based without Laplacian prior. Thus, the sparsity-based method with Laplacian prior is the best candidate for the HBTC image reconstruction.

Table 3. Performance of the proposed method by incorporating laplacian prior on sparse coding stage

Method	ODBTC		EDBTC	
	8×8	16×16	8×8	16×16
Without Post Processing	19.23	16.10	20.14	16.54
Lowpass Filtered [13]	29.14	28.46	30.76	30.11
Variance Classified [13]	29.48	28.63	31.63	30.82
VQ-Based	34.01	33.92	34.74	34.45
Sparsity-Based	35.00	34.82	35.03	34.97

4.5. Computational time comparison

This subsection delivers the performance comparisons in terms of computational time. In this experiment, a single image reconstruction is conducted to measure computational burden in terms of CPU time.

The execution time is measured in units of seconds. To make a fair comparison, the computing environments are identically selected over all post-processing methods. Herein, the experiments are performed under Personal Computer with Intel Core 2 Quad CPU @2.40 GHz and 4.0 GB RAM memory. All image reconstruction methods are developed and run in the Matlab R2010b.

Table 4 summarizes the computation time comparisons. Our results suggest that the lowpass filtering approach needs the lowest computation time since it simply performs filtering in blind strategy. The lowpass filtering ignores the image characteristic and noise statistics therefore requiring the lowest effort in the filtering process. The variance classified filtering requires higher computational time compared to that of lowpass filtering since it needs to investigate an underlying image statistic and characteristic. The variance classified filtering computes the statistical metric of the variance to determine a suitable filter coefficient thereby requiring relatively higher computational time. The VQ-based approach requires longer computational time compared to the low-pass and variance-classified filtering in the HBTC image reconstruction. The VQ-based method employs the closest matching and image patch substitution in the image reconstruction stage. In addition, the VQ-based approach also performs the image patch averaging at the end of the reconstruction process. On the other hand, sparsity-based method requires the highest computational time to reconstruct the HBTC decoded image. This method estimates the sparse coefficient and replaces the image patch based on the predicted sparse coefficient. The sparsity-based approach also requires an additional overhead computing cost on the least-square calculation at the end of the image reconstruction procedure. Even though the sparsity-based approach is associated with the highest computational time, it produces the best reconstructed image for the HBTC decoded image. This effect can be considered as a trade-off between the quality of the HBTC decoded image and the processing time.

Table 4. Computational time comparison (in seconds) between the proposed method and former scheme in the halftoning-based BTC image reconstruction

Method	ODBTC		EDBTC	
	8×8	16×16	8×8	16×16
Lowpass Filtered [13]	3.161	3.162	3.163	3.162
Variance Classified [13]	3.412	3.405	3.45	3.466
VQ-Based	20.803	23.99	21.536	23.89
Sparsity-Based	35.12	35.88	35.44	35.12

5. CONCLUSION

A new method has been presented for improving HBTC image quality under sparse representation framework. The HBTC image often contains impulsive noise which may distort human vision perception of this decoded images. It induces unpleasant condition for human visual perception. To further alleviate the impulsive noise, the HBTC decoded image is then predicted, aligned, and modified using two learned dictionaries, i.e. HBTC and clean image dictionaries. In the proposed method, the sparse coefficient is simply estimated from the HBTC decoded image by means of HBTC image dictionary. Subsequently, the reconstructed image is composed from the clean image dictionary with the predicted sparse coefficient. The HBTC image patch is initially classified as border or non-border image patch before applying the sparse representation. The experiment finding suggests that the proposed method yields a promising performance compared to former existing schemes. As documented in the experimental results, the proposed method can provide superior results compared to the former related schemes. To further reduce the computational time, several techniques such as the fast codewords matching, simple sparse coding calculation, etc., can be exploited for the proposed method. The Euclidean distance computation can be replaced with linear time closest matching technique. The sparse coding and estimation techniques can be replaced with the recent advance technique on sparse representation technique for the proposed sparsity-based method. To reduce computational time, the proposed method can be implemented in the parallel computation framework, in which different image blocks are processed independently.

REFERENCES

[1] Guo J. M. and M. F. Wu, "Improved block truncation coding based on the void-and-cluster dithering approach," *IEEE Transactions on image processing*, vol. 18, no. 1, pp. 211-213, 2009.
[2] Guo J. M., "Improved block truncation coding using modified error diffusion," *Electronics Letters*, vol. 44, no. 7, pp. 462-464, 2008.
[3] Liu X., et al., "Joint Data Hiding and Compression Scheme Based on Modified BTC and Image Inpainting," *IEEE Access*, vol. 7, pp. 116027-116037, 2019.

[4] Chen Y. Y. and K. Y. Chi, "Cloud image watermarking: high quality data hiding and blind decoding scheme based on block truncation coding," *Multimedia Systems*, vol. 25, no. 5, pp. 551-563, 2019.
[5] Singh D., and S. K. Singh, "Block truncation coding based effective watermarking scheme for image authentication with recovery capability," *Multimedia Tools Applications*, vol. 78, pp. 4197-4215, 2017.
[6] Huynh N. T., et al., "Minima-maxima preserving data hiding algorithm for absolute moment block truncation coding compressed images," *Multimedia Tools Applications*, vol. 77, no. 5, pp. 5767-5783, 2018.
[7] Guo J. M. and H. Prasetyo, "Content-based image retrieval using features extracted from halftoning-based block truncation coding," *IEEE Transactions on image processing*, vol. 24, no. 3, pp. 1010-1024, 2015.
[8] Guo J. M., H. Prasetyo, and H. S. Su, "Image indexing using the color and bit pattern feature fusion," *Journal of Visual Communication Image Representation*, vol. 24, no. 8, pp. 1360-1379, 2013.
[9] Guo J. M., et al., "Image retrieval using indexed histogram of void-and-cluster block truncation coding," *Signal Processing*, vol. 123, pp. 143-156, 2016.
[10] Guo J. M., H. Prasetyo, and J. H. Chen, "Content-based image retrieval using error diffusion block truncation coding features," *IEEE Transactions on Circuits Systems for Video Technology*, vol. 25, no. 3, pp. 466-481, 2015.
[11] Guo J. M., H. Prasetyo, and K. Wong, "Halftoning-based block truncation coding image restoration," *Journal of Visual Communication Image Representation*, vol. 35, pp. 193-197, 2016.
[12] Prasetyo H. and H. Kurniawan, "Reducing JPEG False Contour Using Visual Illumination," *Information*, vol. 9, no. 2, pp. 41, 2018.
[13] Guo J -M., C. C. Su, and H. J. Kim, "Blocking effect and halftoning impulse suppression in an ED/OD BTC image with optimized texture-dependent filter sets," *Proceedings 2011 International Conference on System Science and Engineering*, 2011.
[14] Mustaffa M. R., et al., "A colour-based building recognition using support vector machine," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 17, no. 1, pp. 473-480, 2019.
[15] Al-Nima R. R. O., M. Y. Al-Ridha, and F. H. Abdulraheem, "Regenerating face images from multi-spectral palm images using multiple fusion methods," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 17, no. 6, pp. 3110-3119, 2019.
[16] Memon M. H., et al., "Blood image analysis to detect malaria using filtering image edges and classification," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 17, no. 1, pp. 194-201, 2019.
[17] Attamimi M., T. Nagai, and D. Purwanto, "Object detection based on particle filter and integration of multiple features," *Procedia computer science*, vol. 144, pp. 214-218, 2018.
[18] Zhang Z., et al., "A survey of sparse representation: algorithms and applications," *IEEE access*, vol. 3, pp. 490-530, 2015.
[19] Aharon M., M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311-4322, 2006.
[20] Etienam C., R. V. Velasquez, and O. Dorn, "Sparse Multiple Data Assimilation with K-SVD for the History Matching of Reservoirs," in *Progress in Industrial Mathematics at ECMI Conference paper*, pp. 567-573, 2019.
[21] Mallat S. G. and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3397-3415, 1993.
[22] Pati Y. C., R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," *Proceedings of 27th Asilomar conference on signals, systems and computers, IEEE*, 1993.
[23] Chen S. S., D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129-159, 2001.
[24] Lewicki M. S. and T. J. Sejnowski, "Learning overcomplete representations," *Neural computation*, vol. 12, no. 2, pp. 337-365, 2000.
[25] Golla M., and S. Rudra, "A Novel Approach of K-SVD-Based Algorithm for Image Denoising," *Histopathological Image Analysis in Medical Decision Making*, pp. 154-180, 2019.
[26] Dong W., et al., "Nonlocally centralized sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1620-1630, 2013.
[27] Hu Y. C., K. K. R. Choo, and W. L. Chen, "Tamper detection and image recovery for BTC-compressed images," *Multimedia Tools and applications*, vol. 76, no. 14, pp. 15435-15463, 2017.