Regular Article

A Novel Joint Blocking and Ringing Artifact Reduction using Beltrami-Based Texture Maps

Thai Van Nguyen[1], **Tuan Hong Do**[1], **Dung Trung Vo**[1,2]

[1] Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology, Vietnam
[2] Samsung Research America, Irvine, CA, USA 92626

Correspondence: Thai Van Nguyen, thai1279@gmail.com
Communication: received 28 September 2016, revised 14 June 2017, accepted 15 June 2017
Online publication: 30 October 2017, Digital Object Identifier: 10.21553/rev-jec.136
The associate editor coordinating the review of this article and recommending it for publication was Prof. Lam Son Phung.

Abstract– **This paper proposes a novel image enhancement approach using advanced texture maps together with isotropic or directional fuzzy filters. Texture maps of original images or compressed images are estimated and then are used to control the filter's strength. The aim is to reduce blocking and ringing artifacts while preserving the sharpness of compressed images. The spread parameter of fuzzy filters plays an important role to control deblocking and deringing process. Thus, the spread parameter is optimized to obtain the highest image quality. The proposed algorithm also simultaneously combines deblocking and deringing schemes to reduce the algorithm's complexity. Simulation results show that the proposed fuzzy filtering scheme achieves the best visual quality, SSIM, and PSNR values among existing methods.**

Keywords– **artifact, texture, compression, fuzzy filter.**

1 Introduction

Rapid image and video transmission growth in mobile data and internet traffic has led to an inevitable requirement of compressing images and videos to reduce storage space and channel bandwidth. The ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) standardization organizations introduce many block-based compression standards such as JPEG, MPEG, H.26x, etc. to meet this requirement. However, block-based compression suffers from undesirable blocking, ringing, mosquito and flickering artifacts [1], especially at low bit rates. When blocks are compressed independently, correlation of pixels at the block boundaries along vertical and horizontal directions may be broken. This causes blocking artifacts. Beside that, ringing artifacts occur due to coarse quantization and truncation of high-frequency Discrete Cosine Transform (DCT) coefficients and mostly happen along strong edges.

Coding artifacts cause uncomfortableness to viewers' human visual system (HVS). Hence, blocking and ringing artifact removal in compressed images becomes a very essential task. It can be considered as a quality enhancement process. In general, image quality enhancement techniques can be implemented either at encoding side or decoding side. At the encoding side, each method requires the scheme of its own such as DCT/SQ coding method [2], the Lapped Orthogonal Transform (LOT) [3], etc. One disadvantage of these methods is that they are not compatible to the existing video or image compression standards. Therefore, postprocessing techniques at the decoding side have received much more attention due to its compatibility with existing compression standards.

Most of previous postprocessing methods mainly base on filtering, both in the pixel domain and the tranform domain. The authors in [4] use low pass filters to reduce the blocking artifacts. However, because the high frequency components are eliminated during filtering, processed images are blurred. In an alternative approach, Chen [5] proposes a postfilter in the transform domain. In this method, image blocks are classified by the HVS masking effect. The adaptive filter is then applied to the transform coefficients of classified blocks having low or high activities. Different window sizes are implemented to adapt to different blocking activities. In particular, for low activity areas where blocking artifacts appear more remarkable, a large window is utilized to efficiently smooth out blocking artifacts. In the other hand, image details are more visible in high activity areas so these are processed with a smaller window to keep the details. Furthermore, the authors in [6] expand the work in [5] by proposing an HVS-based measurement of blocking artifacts. The blocky edges are classified according to their measured local visibility of blocking effect. Based on block edge types, an adaptive filter is then used to improve the transform coefficients. In another research, the authors in [7] alleviate blocking artifacts by increasing the inter-block correlations of three lowest frequency DCT coefficients in each coded block.

For image quality evaluation, objects such as edges, texture, text, etc. significantly affect the HVS response. Construction of these object maps thus play an essential role in order to control the filter's strength. The authors

in [8], [9], [10] and [11] use edge maps to adapt the fuzzy filter artifact reduction. In these methods, the variance and standard deviation value ([12], [13]) are used to construct the edge map. But these operators are rather sensitive to noise. Furthermore, the authors in [1] use the Sobel operator to classify pixels into edge pixels and non-edge pixels. Artifact filtering using this classification may still blur the image due to the lack of texture information. This paper focuses on using the enhanced texture maps to maintain the image details during the deblocking and deringing.

Constructing texture maps is a challenging issue since it is very difficult to define texture based on mathematical terms. The Beltrami method in [14], [15] is used to locate texture for image segmentation. Recently a texture map is estimated using the enhanced Beltrami method in [16] to improve the noise resistance of the estimation. For the first time, the authors in [17] utilize the texture map to remove blocking artifacts only. Simulation results from that approach have proven that the texture map usage is capable of preserving the texture of compressed images in artifact reduction. However, this method has not been considered to suppress ringing artifacts.

In this paper, a novel method is proposed to enhance image quality of compressed images by reducing both blocking and ringing artifacts. Isotropic or directional fuzzy filters are implemented to remove coding artifacts. At first, object maps of blocking artifacts, ringing artifacts, and texture are constructed and then are used to control the filter's strength by adapting the spead parameter of fuzzy filters. Furthermore, an optimization is proposed for this spread parameter to obtain the highest PSNR, SSIM, and visual quality. The proposed approach also considers combining the deblocking and deringing schemes. This is to reduce two separate blocking and ringing rounds into one joint round. The remainings of this paper are organized as follows. Section 2 reviews fuzzy filter background. Section 3 introduces blocking and ringing map estimation. Texture map estimation and enhancement are presented in Section 4. The proposed enhancement method by using object maps is mentioned in Section 5. Parameter optimization is discussed in Section 6. Simulation and conclusion are presented in Section 7, and Section 8, respectively.

2 Fuzzy Filters

2.1 Isotropic Fuzzy Filters

Fuzzy filters are used to remove artifacts while preserving the image details. An isotropic fuzzy filter in [1] is applied to the input image I to formulate the output image I' as

$$I'(x,y) = \frac{\sum_{m,n\in\Omega} h\left(I\left(x,y\right), I\left(x+m,y+n\right)\right) \times I\left(x+m,y+n\right)}{\sum_{m,n\in\Omega} h\left(I\left(x,y\right), I\left(x+m,y+n\right)\right)}, \quad (1)$$

where Ω are the neighbours of the pixel of interest $I(x,y)$; $h(I(x,y), I(x+m,y+n))$ is the response function of the fuzzy filter. The filter response $h(I(x,y), I(x+m,y+n))$ must follow the constraints as in (2), (3), and (4)

$$\lim_{|I(x,y)-I(x+m,y+n)|\to 0} h(I(x,y), I(x+m,y+n)) = 1, \quad (2)$$

$$\lim_{|I(x,y)-I(x+m,y+n)|\to +\infty} h(I(x,y), I(x+m,y+n)) = 0, \quad (3)$$

and

$$\begin{aligned} &if |I(x,y) - I(x+m_1,y+n_1)| \\ &\geq |I(x,y) - I(x+m_2,y+n_2)|, \\ &h(I(x,y), I(x+m_1,y+n_1)) \\ &\leq h(I(x,y), I(x+m_2,y+n_2). \end{aligned} \quad (4)$$

Gaussian function is one of the functions that fulfills the requirements in (2), (3), and (4)

$$h(I(x,y), I(x+m,y+n)) = \exp\left(\frac{(I(x+m,y+n) - I(x,y))^2}{2\sigma^2}\right), \quad (5)$$

where the spread parameter σ controls the contribution of the Gaussian function. The larger σ value is, the higher corelation between the fuzzy filter ouput and neighbours of input is. Conversely, the smaller σ value is, the more isolated from its neighbours the input $I(x,y)$ is. Therefore, σ value is used to adapt the filter's strength at different activity levels such as smooth or detail areas.

2.2 Directional Fuzzy Filters

Ringing artifacts occur strongest at pixels which are very close to strong edges and become weaker at further-from-edge pixels. Based on this directional characteristic, the authors in [1] propose a directional fuzzy filter to effectively remove ringing artifacts. The general cosine based form of the spread parameter is defined as [1]

$$\sigma(\theta_c) = \sigma_m(\alpha + \beta\cos^2(\theta_c) \quad (6)$$

where α and β are constant factors. The $\sigma(\theta_c)$ value depends on the spread parameter amplitude σ_m and the angle θ_c. This angle is defined as $\theta_c = \theta - \theta_0$, where θ and θ_0 are the angles defined as in Figure 1. Obviously, $\sigma(\theta_c)$ has maximum value if $\theta = \theta_0$. It means that the strongest filtering strength is applied to the direction perpendicular to the edge. Assume G_x and G_y are the horizontal and vertical derivative, respectively, image edges are detected by comparing the gradient magnitude $G = \sqrt{G_x^2 + G_y^2}$ to a predefined threshold. Its correspoding direction is determined by $\theta_0 = a\tan(\frac{G_y}{G_x})$.

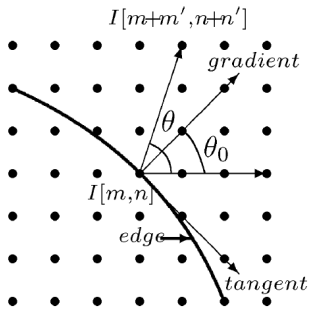


Figure 1. Angles θ and θ_0 of the directional fuzzy filter [1].

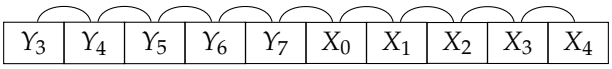


Figure 2. Vertical boundary gap calculation.

3 Blocking Map and Ringing Map

To deblock artifacts, the authors in [16] propose method to detect pixels suffering from blocking artifacts and only deblock these pixels. In block-based compression, blocking artifacts occur at the boundary of blocks along vertical and horizontal directions. Hence, the gray difference at the block boundary is used to detect blocking aritifact pixels. Figure 2 shows a vertical boundary gap calculation where X_i and Y_i are the pixel in the same row but different image blocks. The vertical gap differences are calculated as follows: $C_0 = |X_0 - Y_7|$, $L_i = |Y_{7-i} - Y_{6-i}|$ and $R_j = |X_{j+1} - X_j|$, $(i, j = 0, ..., 3)$. If $\max(L_0, L_1, L_2, L_3) < C_0$ or $\max(R_0, R_1, R_2, R_3) < C_0$ then X_0 and Y_7 are the gap pixels. If gap pixels does not belong to strong edges then a 2D-fuzzy filter is applied to them. At the same time, neighbours of these blocking pixels in a 3×3 window are also considered during the deblocking process. Figure 3(c) shows an example of a blocking map of the compressed JPEG image in Figure 3(a) in which black color pixels are blocking pixels.

The ringing artifacts usually occur close to strong edges. Therefore, neighbours of strong edge pixels may be suffered from ringing artifacts. The strong edge map [10] is chosen to construct the ringing map. Figure 3(d) shows a ringing map with search window = 8, where green color pixels are ringing pixels and red color pixels are strong edge pixels. The search window in this case determines the distance from strong edge pixels to their neighbours which are specified as ringing pixels.

4 Texture Maps

Texture maps are constructed by classifying pixels based on texture features. Normally, the texture map accuracy depends on the classification feature choice. Pixels in the texture map are generally classified as strong edges, weak edges, strong texture, weak texture and flat areas. Usually, the texture map is estimated based on operators such as standard deviation, Sobel, etc. The original RGB image is first converted to YUV components. The texture feature is estimated only from Y component to simplify the computation load. However for postprocessing enhancement, the original images are not available at the decoder. Hence, texture estimation must base on the compressed images. With the fact that details such as texture are lost during compression, texture estimation based on the compressed images is thus a more difficult task. Most of the previous texture estimation methods focus on estimation on original images in high quality ([1, 9, 10]). In this paper, the compressed images are used while original images are also used for accuracy comparison. Texture feature based on pixel by pixel is sensitive to noise, so the texture map using this scheme cannot obtain high accuracy. The estimation utilizing patch areas ([18, 19]) is proposed to be more robust to noise. But texture maps based on patches are not highly accurate since the classification error is high. In [10], the texture feature $F(x, y)$ based on window derivative is introduced to obtain the texture map with high accuracy as in (7). Let $W(x, y)$ be a window of $(2R + 1) \times (2R + 1)$ pixels. The texture feature is defined in [15] as follows

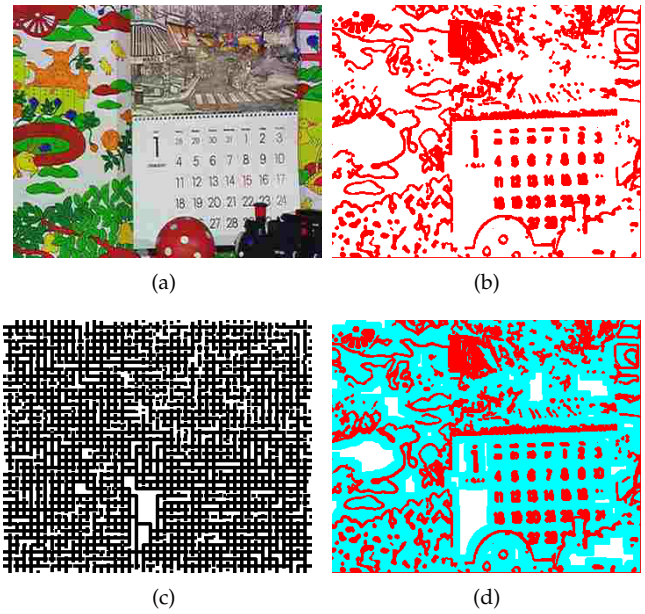


Figure 3. An example of the blocking map and ringing map. (a) Compressed Mobile image; (b) Strong edge map; (c) Blocking map; (d) Ringing map.

$$F(x, y) = \exp\left(-\frac{\det(g_{xy})}{\delta^2}\right), \tag{7}$$

where δ is a scaling parameter, the value of g_{xy} ([10, 14, 15]) is defined as

$$g_{xy} = \begin{pmatrix} 1 + (\partial_x W(x,y))^2 & \partial_x W(x,y).\partial_y W(x,y) \\ \partial_x W(x,y).\partial_y W(x,y) & 1 + (\partial_y W(x,y))^2 \end{pmatrix},$$

where the window derivatives ([10]) are defined as

$$\partial_x W(x,y) = \sqrt{\frac{\sum_{m=-R}^{R} \sum_{n=-R}^{R} [W(m+1,n) - W(m,n)]^2}{(2R+1) \times (2R+1)}}$$

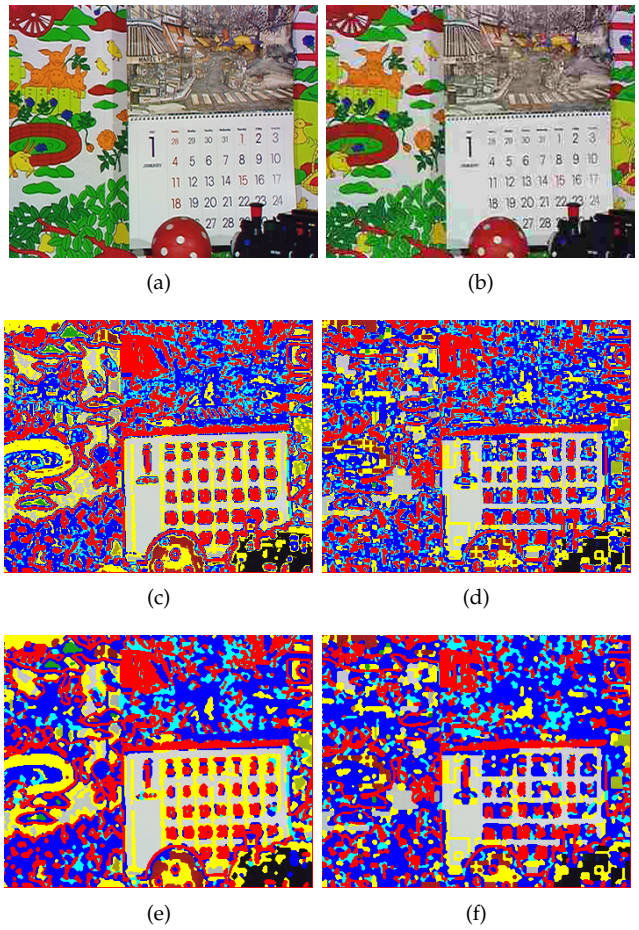


Figure 4. An example of the texture map. (a) Original Mobile image; (b) Compressed Mobile image; (c) The texture map of Figure 4(a); (d) The texture map of Figure 4(b); (e) The enhanced texture map of Figure 4(c); (f) The enhanced texture map of Figure 4(d).

and

$$\partial_y W(x,y) = \sqrt{\frac{\sum_{m=-R}^{R}\sum_{n=-R}^{R}\left[W(m,n+1)-W(m,n)\right]^2}{(2R+1)\times(2R+1)}}.$$

Texture map based on window derivative is estimated as

$$Pel-type = \begin{cases} Strong-edge & F(x,y) < 10^{-4} \\ Weak-edge & 10^{-4} \le F(x,y) < 8.10^{-3} \\ Strong-texture & 8.10^{-3} \le F(x,y) < 0.5 \\ Weak-texture & 0.5 \le F(x,y) < 0.95 \\ Flat & otherwise \end{cases} \quad (8)$$

Figure 4 shows an example of the texture map based on the enhanced Beltrami method with $\delta = 15$ based on both original and compressed images, where quality factor of compressed image is 12.5. Colors of the texture map are defined as follows: Red: strong edge; Green: weak edge; Blue: strong texture; Yellow: weak texture; Others: flat. As can be seen in Figure 4(c) and Figure 4(d), there are many isolated texture pixels which do not correctly represent the texture area because they are generally located in groups. So isolated pixels in the texture map should be removed since textures are geometric structures while noise is not ([10]). As can be seen in Figure 4(e) and Figure 4(f), the enhanced texture maps are more accurate and cleaner than the texture maps in Figure 4(c) and Figure 4(d), respectively for the original and compressed image.

5 Proposed Deblocking and Deringing Using Texture Map

Figure 5 shows the flow chart of the proposed deblocking and deringing method. This method implements a directional fuzzy filter to reduce ringing artifacts for ringing pixels. The other pixels, an isotropic fuzzy filter is applied to remove the artifacts. The input of the flow chart is a compressed image, which suffers from blocking and ringing artifacts. Object maps including blocking map, ringing map and texture map are first estimated and later are used to control the filtering process. Texture map is constructed as described in Section 4 based on compressed images. Next, ringing map is constructed by using the strong edge map as described in Section 3, which can be extracted from the texture map. Blocking map as mentioned in Section 3 is also estimated. At each pixel, based on blocking map, the blocking artifact pixels and their neighbours in a 3×3 window are firstly processed by the 2D-fuzzy filter.

Spread parameter group $\{\sigma_{11}, \sigma_{12}, \sigma_{13}, \sigma_{14}, \sigma_{15}\}$ is selected to adapt the filter's strength based on the advanced texture map. If the pixel is a blocking pixel, it is filtered with a strong 2D isotropic filter with highest σ_{11} value to removed the strong blocking. Then, its 3×3 neighbour pixels are considered. Depending on the texture types (which are weak edges, strong texture, weak texture, and flat), corresponding isotropic 2D-fuzzy filter with different σ value is selected to adapt the filtering strength. The σ value is ranged from highest value σ_{11} for blocking areas to lowest value σ_{15} for flat areas, corresponding to strongest filtering level to weakest filtering level.

Next, ringing map is considered to remove ringing artifact. After filtering blocking artifact pixels and their neighbour pixels in 3×3 window, the process is moved to deringing step. Spread parameter group $\{\sigma_{21}, \sigma_{22}, \sigma_{23}, \sigma_{24}, \sigma_{24}\}$ controls filter's strength to reduce ringing artifacts. Similar to deblocking step, if the pixel is classified as a ringing pixel, it is filtered by a directional fuzzy filter with highest value σ in the deringing group σ. This ensures the ringing artifacts are removed using the strongest filtering. If the pixel is not a ringing pixel, then it is filtered adaptively based on its type based on the texture map (weak edge, strong texture, weak texture, flat areas). Different than deblocking step, the deringing step only considers the texture type of the pixel of interest, not the types of neighboring pixels. The spread parameter values for deringing are selected similar as for deblocking. Highest value for σ_{21} and lowest value for σ_{26} are selected adaptively for the filter in removing the most ringing artifact while still keeping the most details. The deblocking and deringing steps are combined to achieve the highest exploit of the

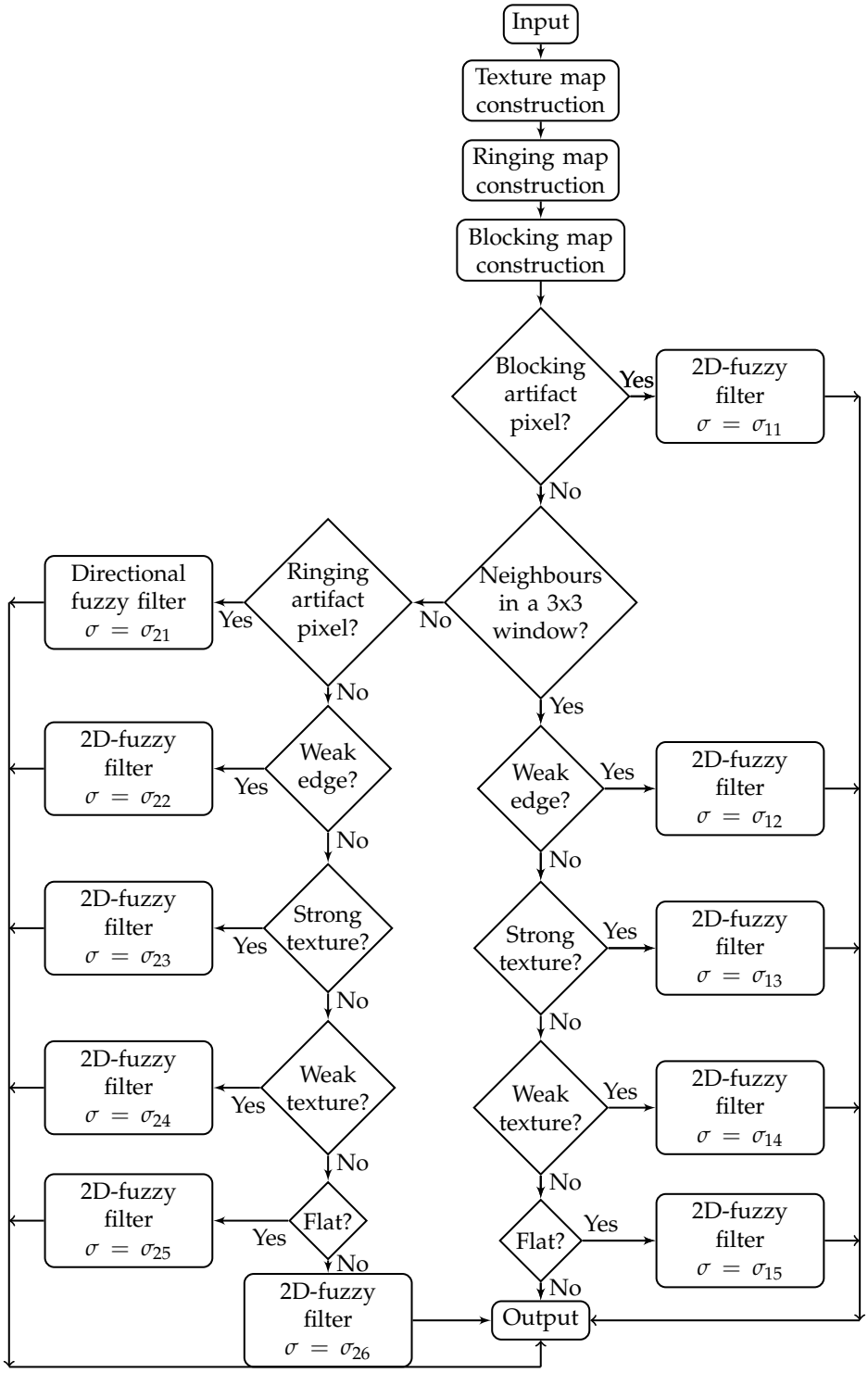


Figure 5. The flow chart of the proposed deblocking and deringing method.

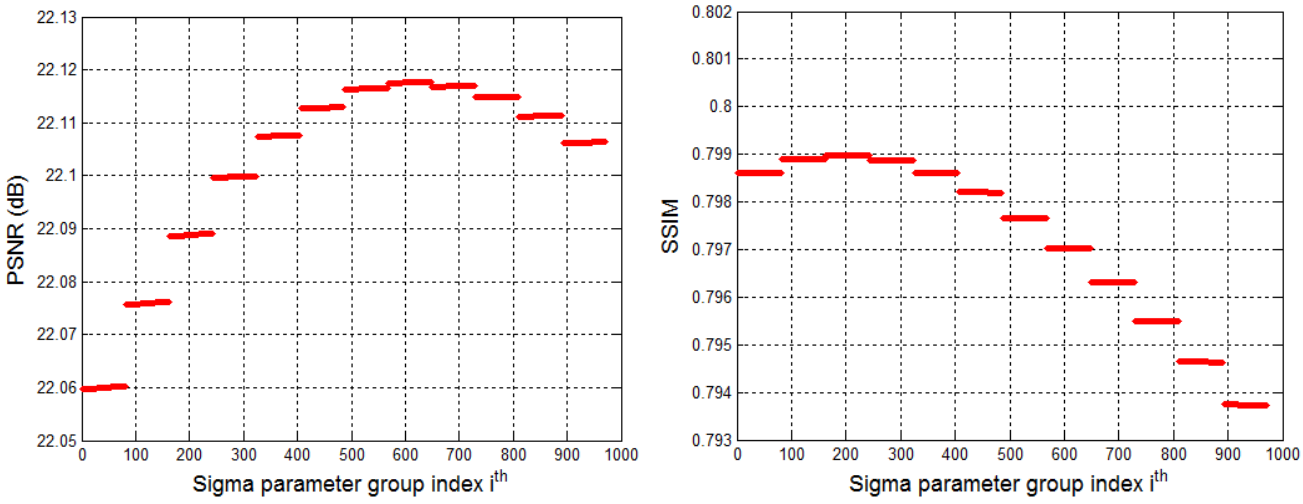
texture map and obtain one round processing for both deblocking and deringing steps. The selection of the spread parameter for the fuzzy filters is later discussed in Section 6.

6 Parameter Optimization

The spread parameter is selected so that PSNRs, SSIMs, and visual quality are highest. The spread parameter groups of $\{\sigma_{11}, \sigma_{12}, \sigma_{13}, \sigma_{14}, \sigma_{15}\}$ and $\{\sigma_{21}, \sigma_{22}, \sigma_{23}, \sigma_{24}, \sigma_{25}\}$ control filter's strength in the deblocking and deringing processes, respectively. In general, spread parameter ranges are selected based on experimental studies and are shown in Table I. The larger ranges and finer interval between group selection may result in a better solution but, simulation running time significantly increases. The spread parameters are selected to be integer values for simplification purpose although they can be real values. The spread parameter group is

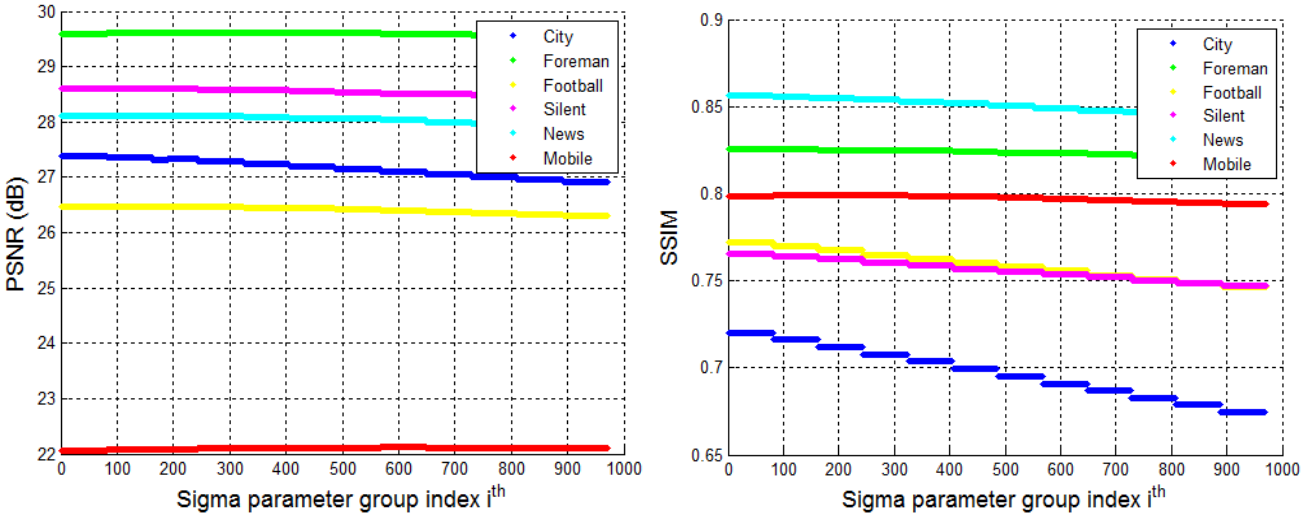
Table I
SIGMA PARAMETER RANGES.

σ_{11}, σ_{21}	σ_{12}, σ_{22}	σ_{13}, σ_{23}	σ_{14}, σ_{24}	σ_{15}, σ_{25}
[15, 17]	[10, 12]	[10, 12]	[10, 12]	[8, 10]
[18, 20]	[13, 15]	[13, 15]	[13, 15]	[11, 13]
[21, 23]	[16, 18]	[16, 18]	[16, 18]	[14, 16]
[24, 26]	[19, 21]	[19, 21]	[19, 21]	[17, 19]



(a) (b)

Figure 6. Deblocking simulation results of the 5^{th} Mobile frame for different σ selection.



(a) (b)

Figure 7. Deblocking simulation results of many images for different σ selection.

the order combination of the spread parameter values at per row of the Table I. Each row of this table contains $3\times3\times3\times3\times3 = 243$ groups of spread parameters. For total 4 rows, the total number of groups is 972. In Table I, the index i of the spread parameter group selection for the first row is from 1 to 243, for the second row is from 244 to 486, for the third row is from 487 to 729, for the fourth row is from 730 to 972.

Based on the proposed deblocking and deringing filtering scheme, each of the spread parameter group is simulated to get a pair of PSNR and SSIM [20].

Assume PSNR and SSIM values between the j^{th} enhanced and original images using the enhanced and original images using the i^{th} spread parameter group are $\text{PSNR}_{i,j}$ and $\text{SSIM}_{i,j}$, respectively. The $\text{PSNR}_{i,j}$ and $\text{SSIM}_{i,j}$ values are calculated when running the proposed algorithm on a large image database with all selection of spread parameter groups. Deblocking results on PSNR and SSIM are shown in Figure 6 for compressed Mobile image and Figure 7 for a larger set of compressed images. From these results, PSNRs and SSIMs do not obtain maxi-

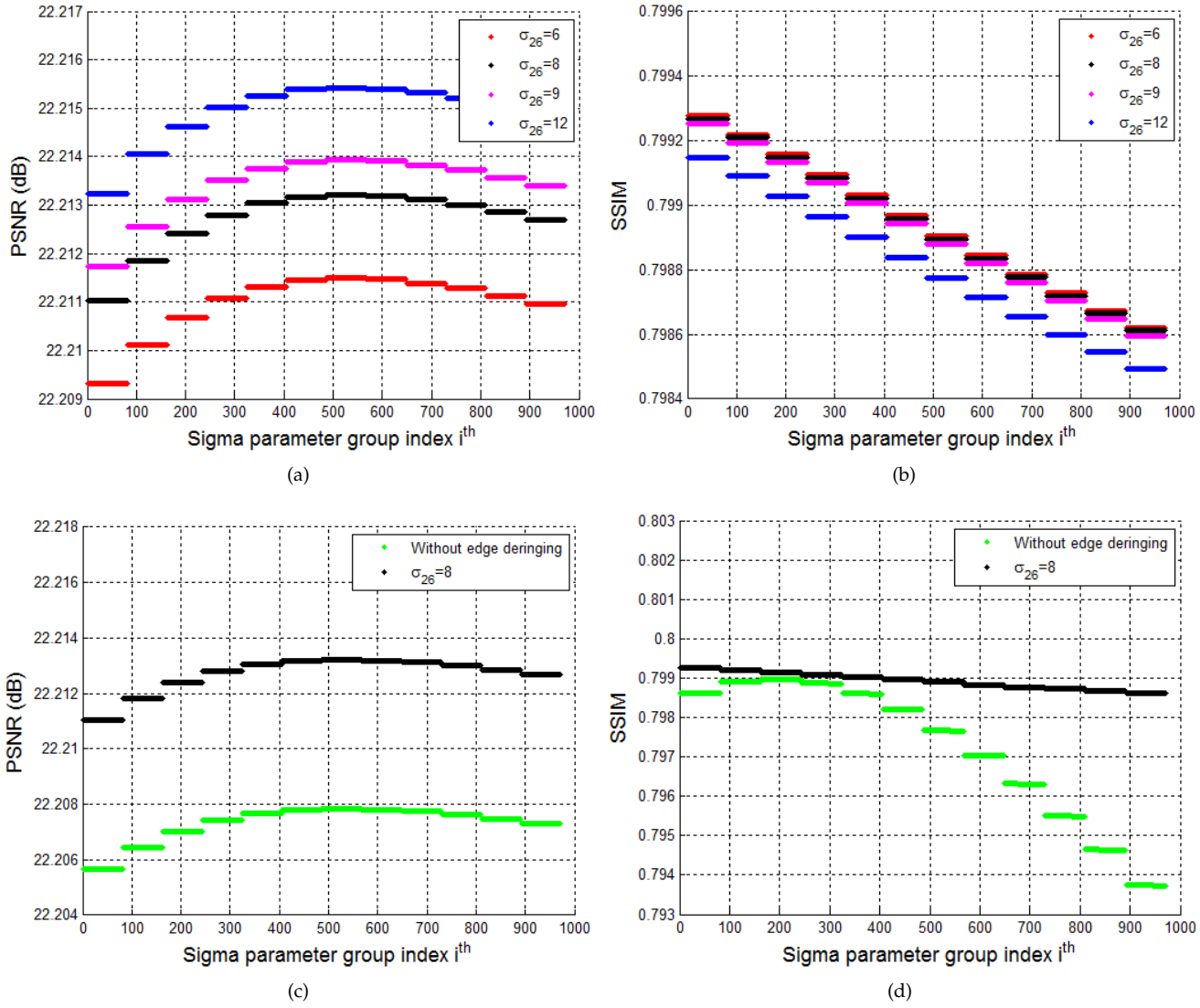


Figure 8. Deringing and deblocking results of Mobile image with different σ_{26} parameters.

mum values at the same selection of spread parameter group. Therefore, it is a compromise to select a best spread parameter group so that PSNRs, SSIMs and visual quality are highest. Let $\text{PSNR}_{max,j}=\max_i(\text{PSNR}_{i,j})$, $\text{SSIM}_{max,j}=\max_i(\text{SSIM}_{i,j})$ and $\sigma_{opt,j}$ be the optimal spread parameter group for the j^{th} image. For each j^{th} input image, $\sigma_{PSNR,j}$ and $\sigma_{SSIM,j}$ are spread parameter groups that $\text{PSNR}_{max,j}=\text{PSNR}(\sigma_{PSNR,j})$ and $\text{SSIM}_{max,j}=\text{SSIM}(\sigma_{SSIM,j})$. $\sigma_{opt,j}$ value for that j^{th} image is calculated as follows

$$\sigma_{opt,j}=\frac{\sigma_{PSNR,j}+\sigma_{SSIM,j}}{2} \tag{9}$$

It is assumed that J value is the numbers of input images in the database, which are used to find the optimal spread parameter group. For J input images, σ_{opt} value is calculated as follows

$$\sigma_{opt}=\frac{\sigma_{opt,1}+\sigma_{opt,2}+\cdots+\sigma_{opt,J}}{J} \tag{10}$$

Based on the deblocking simulation results of Mobile, City, Foreman, Football, Silent and News, the index of the optimal spread parameter group is 118. So, spread parameter values that optimally control deblocking process are $\sigma_{11}=16$, $\sigma_{12}=11$, $\sigma_{13}=11$, $\sigma_{14}=10$, $\sigma_{15}=8$.

For the directional fuzzy filter, to adapt to different areas having different activity levels, the amplitude of the spread parameter is defined as

$$\sigma_m(x,y)=\sigma_{21}\left((1-\gamma)\left(\frac{F_{\max}-F(x,y)}{F_{\max}-F_{\min}}\right)+\gamma\right), \tag{11}$$

where $F_{\min}$ and $F_{\max}$ are minimum and maximum values of all $F(x,y)$ values defined as in (7), γ is a scaling factor in $[0,1]$ and σ_{21} is the maximum spread parameter value. The parameters in [1] are chosen as α=0.5, β=3.5 and γ=0.5. The σ_{26} parameter controls the spread parameter of the isotropic fuzzy filter, which removing artifacts on strong edges. Figure 8 shows the deringing and deblocking simulation results of the Mobile image with different σ_{26} values. The green curves are PSNR and SSIM plots without filtering artifacts on strong edges. The red, black, magenta and blue curves are PSNRs and SSIMs plots in case of filtering artifacts on strong edges with $\sigma_{26}=6$, $\sigma_{26}=8$, $\sigma_{26}=9$ and $\sigma_{26}=12$, respectively. Based on these results, filtering

artifacts on strong edges with $\sigma_{26} = 8$ is chosen to get the best PSNRs and SSIMs.

The next step is to determine the optimal spread parameter parameter group for the deringing filters $\{\sigma_{21}, \sigma_{22}, \sigma_{23}, \sigma_{24}, \sigma_{25}\}$. The ranges of σ values are the same as in deblocking process and shown in Table I. The procedure to find the optimal spread parameter group for deringing process is also similar to the procedure of deblocking process. Deringing results are simulated for Mobile, Akiyo, Lena, Football, and City. The index of the optimal spread parameter group is 118. The spread parameter values of the 118^{th} group for deringing are $\sigma_{21} = 16$, $\sigma_{22} = 11$, $\sigma_{23} = 11$, $\sigma_{24} = 10$, $\sigma_{25} = 8$.

7 Simulation Results

Different methods of the postprocessing quality enhancement are compared in terms of PSNR, SSIM, and visual quality. The simulation programs are implemented by on a computer with CPU of 2.4 GHz and RAM of 4 GB.

A large number of testing frames are taken from News, Silent, Foreman, Mobile, Mother, Paris, Akiyo, Tempete, Waterfall, Lena, Ice, Crew, Football, and City sequences. These images are compressed by JPEG standard, where quality factor is 12.5. Table II and Table III show comparision in PSNR values and SSIM values between the proposed deblocking method to Chen [5], Liu [6], and 1D-fuzzy filter [1]. The texture map is constructed from the original images (Proposed 1) or the compressed images (Proposed 2). The average PSNR improvement for Chen's method, Liu's method, 1D-fuzzy filter's method, the Proposed 1, and the Proposed 2 on the compressed images are 0.3330 dB, 0.4182 dB, 0.4806 dB, 0.7633 dB, and 0.7538 dB, respectively. The average SSIM improvement for Chen's method, Liu's method, 1D-fuzzy filter's method, the Propose 1, and the Proposed 2 are -0.0005, 0.0099, 0.0120, 0.0166, and 0.0161, respectively. Compared to the existing methods of Chen's method, Liu's method, 1D-fuzzy filter's method, the PSNR average improvement of using the Proposed 1 is +0.4303 dB, +0.3451 dB, +0.2827 dB, and using the Proposed 2 is +0.4209 dB, +0.3357 dB, +0.2732 dB, respectively. Similarly, the SSIM average improvement of using the Proposed 1 is +0.0171, +0.0067, +0.0046, and using the Proposed 2 is +0.0167, +0.0063, +0.0041, respectively. PSNRs and SSIMs of the Proposed 2 is slightly lower than those of the Proposed 1. The texture map in the Proposed 1 is constructed from the original image, and is more accurately than texture map in the Proposed 2 which is constructed from the compressed image. The proposed deblocking method gives better results on PSNR and SSIM values than the compared methods.

To evaluate the visual quality, deblocking results with different methods on the 5^{th} frame of the Foreman sequence are shown on Figure 9 for a zoomed area. Blocking artifact reduction of proposed deblocking method is significant better than that of Chen's method, Liu's method, and 1D-fuzzy filter's method. To further improve the image, the combined deblocking and deringing method is simulated. Table IV and Table V show the comparision of PSNR values and SSIM values between the proposed deblocking and deringing method to Chen's method, Liu's method, and directional fuzzy filter's method. The average PSNR improvement over JPEG compression of Chen's method, Liu's method, and directional fuzzy filter's method, the Proposed 1, and the Proposed 2 are 0.3330 dB, 0.4182 dB, 0.6654 dB, 0.8594 dB, and 0.8327 dB, respectively. The average SSIM improvement of Chen's method, Liu's method, directional fuzzy filter's method, the Propose 1, and the Proposed 2 are -0.0005, 0.0099, 0.0122, 0.0237, and 0.0230, respectively. Compared to the previous methods in Table V, the average PSNR improvement of using the Proposed 1 is +0.5264 dB, +0.4412 dB +0.1940 dB, and using the Proposed 2 is +0.4909 dB, +0.4057 dB, +0.1585 dB, respectively. Similarly, the average SSIM improvement of using the Proposed 1 over Chen's method, Liu's method, and directional fuzzy filter's method is +0.0242, +0.0138, +0.0115, and using the Proposed 2 over other methods is +0.0235, +0.0131, +0.0108, respectively. PSNR and SSIM values of the proposed deblocking and deringing methods are higher than those of the compared methods.

Figure 9. Comparison of deblocking results on Foreman frame. (a) Original; (b) Compressed; (c) Chen's method; (d) Liu's method; (e) 1D - fuzzy filter's method; (f) Proposed deblocking method.

Figure 10 and Figure 11 show more comparison results for City and Mobile frame.

As can be seen from these figures, the directional fuzzy filter (Figure 10(b) and Figure 11(b)) is able to reduce ringing artifact very effectively. However, this method cannot keep some small details of the image. The proposed deblocking and deringing method significantly preserves much more details of the compressed images than those of the existing methods. It confirms that the proposed deblocking and deringing scheme gives best results on both objective assessment and subjective assessment among the compared methods.

Table II
PSNR Comparison in dB of the Proposed Deblocking Method
(When Using Texure Map from Original and Compressed Image to Other Methods).

Image	JPEG	Chen [5]	Liu [6]	1D-fuzzy filter[1]	Proposed 1	Proposed 2
News	27.3155	27.3159	27.5162	27.7057	28.0964	28.0617
Silent	27.8178	28.4832	28.4414	28.3874	28.6024	28.6019
Foreman	28.6129	29.1498	29.1232	29.2323	29.5582	29.5972
Mobile	21.6516	21.5652	21.6859	21.9215	22.0759	22.0324
Mother	30.9931	31.6901	31.6651	31.8034	32.0812	32.0749
Paris	23.8095	23.8791	23.9597	24.1945	24.2672	24.2374
Akiyo	29.4774	29.9906	30.0904	30.0767	30.6038	30.5925
Tempete	24.6425	24.7513	24.8496	24.9587	25.1718	25.1410
Waterfall	25.7287	26.4732	26.3755	26.4210	26.5432	26.5424
Lena	28.0495	28.8390	28.8535	28.6890	28.9599	28.9588
Ice	30.2158	30.2228	30.5625	30.7223	31.2431	31.2795
Crew	30.0017	30.3633	30.4266	30.3942	30.7730	30.7731
Football	25.7957	26.0223	26.2009	26.2409	26.4607	26.4292
City	27.0038	27.0314	27.2194	27.0967	27.3647	27.3471
Average difference		0.3330	0.4182	0.4806	0.7633	0.7538

Table III
SSIM Comparison of the Proposed Deblocking Method
(When Using Texure Map from Original and Compressed Image to Other Methods).

Image	JPEG	Chen [5]	Liu [6]	1D-fuzzy filter[1]	Proposed 1	Proposed 2
News	0.8244	0.8317	0.8403	0.8415	0.8561	0.8557
Silent	0.7511	0.7621	0.7654	0.7627	0.7639	0.7639
Foreman	0.7867	0.8151	0.8104	0.8137	0.825	0.8254
Mobile	0.7736	0.7499	0.7668	0.7796	0.7989	0.7965
Mother	0.8152	0.8388	0.8366	0.8377	0.8456	0.8455
Paris	0.7859	0.7744	0.7885	0.8003	0.8110	0.8096
Akiyo	0.8368	0.8671	0.8645	0.8641	0.8780	0.8778
Tempete	0.8111	0.7892	0.8122	0.8164	0.8240	0.8228
Waterfall	0.7410	0.7109	0.7370	0.7382	0.7089	0.7088
Lena	0.7847	0.8046	0.8081	0.8058	0.8103	0.8103
Ice	0.8726	0.8867	0.8930	0.8914	0.9080	0.9081
Crew	0.7927	0.7989	0.8077	0.8040	0.8105	0.8104
Football	0.7812	0.7592	0.7807	0.7884	0.7703	0.7697
City	0.7377	0.6986	0.7214	0.7191	0.7166	0.7162
Average difference		-0.0005	0.0099	0.0120	0.0166	0.0161

(a) (b) (c)

Figure 10. Comparison of deblocking and deringing results on City frame. (a) Original; (b) Directional fuzzy filter; (c) Proposed method.

(a) (b) (c)

Figure 11. Comparison of deblocking and deringing results on Mobile frame. (a) Original; (d) Directional fuzzy filter; (c) Proposed deblocking and deringing method.

8 Conclusions

Block-based image compression causes annoying artifacts to visual human. Thus, compressed image quality improvement still attracts many attentions of researchers. This paper proposes a novel method to reduce

Table IV
PSNR in dB Comparison of the Proposed Deblocking and Deringing Method
(When Using Texure Map from Original and Compressed Image to Other Methods).

Image	JPEG	Chen [5]	Liu [6]	Fuzzy filter [1]	Proposed 1	Proposed 2
News	27.3155	27.3159	27.5162	27.9429	28.2013	28.1562
Silent	27.8178	28.4832	28.4414	28.6270	28.7671	28.7722
Foreman	28.6129	29.1498	29.1232	29.5627	29.6105	29.6291
Mobile	21.6516	21.5652	21.6859	22.1075	22.2161	22.1274
Mother	30.9931	31.6901	31.6651	32.0772	32.1099	32.1038
Paris	23.8095	23.8791	23.9597	24.4471	24.3943	24.2954
Akiyo	29.4774	29.9906	30.0904	30.2508	30.5678	30.5476
Tempete	24.6425	24.7513	24.8496	24.9493	25.2831	25.2268
Waterfall	25.7287	26.4732	26.3755	26.5614	26.6633	26.6635
Lena	28.0495	28.8390	28.8535	28.9250	29.0568	29.0542
Ice	30.2158	30.2228	30.5625	31.1701	31.1517	31.1624
Crew	30.0017	30.3633	30.4266	30.5095	30.9242	30.9046
Football	25.7957	26.0223	26.2009	26.3534	26.6423	26.5964
City	27.0038	27.0314	27.2194	26.9470	27.5586	27.5338
Average difference		0.3330	0.4182	0.6654	0.8594	0.8327

Table V
SSIM Comparison of the Proposed Deblocking and Deringing Method
(When Using Texure Map from Original and Compressed Image to Other Methods).

Image	JPEG	Chen [5]	Liu [6]	Fuzzy filter [1]	Proposed 1	Proposed 2
News	0.8244	0.8317	0.8403	0.8561	0.8542	0.8541
Silent	0.7511	0.7621	0.7654	0.7649	0.7793	0.7795
Foreman	0.7867	0.8151	0.8104	0.8274	0.8206	0.8209
Mobile	0.7736	0.7499	0.7668	0.7846	0.7992	0.7948
Mother	0.8152	0.8388	0.8366	0.8491	0.8470	0.8470
Paris	0.7859	0.7744	0.7885	0.8067	0.8098	0.8069
Akiyo	0.8368	0.8671	0.8645	0.8776	0.8717	0.8717
Tempete	0.8111	0.7892	0.8122	0.8044	0.8303	0.8289
Waterfall	0.7410	0.7109	0.7370	0.7074	0.7517	0.7515
Lena	0.7847	0.8046	0.8081	0.8078	0.8094	0.8094
Ice	0.8726	0.8867	0.8930	0.9102	0.8986	0.8987
Crew	0.7927	0.7989	0.8077	0.8100	0.8200	0.8198
Football	0.7812	0.7592	0.7807	0.7702	0.7948	0.7943
City	0.7377	0.6986	0.7214	0.6894	0.7396	0.7391
Average difference		-0.0005	0.0099	0.0122	0.0237	0.0230

blocking and ringing artifacts in compressed images. By applying object maps to adapt the fuzzy filters, the proposed deblocking and deringing method reduces blocking and ringing artifacts, while significantly preserving much more details of the compressed images. The object maps are considered including blocking artifact map, ringing artifact map and enhanced texture map. These maps are used to control the fuzzy filter's strength. Furthermore, the adaptive spread parameters are optimized to obtain the highest quality. In the existing image enhancement methods, the deblocking filter and the deringing filter are separated, this may be over-filtering at some pixels. This paper proposed a combined deblocking and deringing algorithm to improve effect of filtering artifacts and also simplify the implementation complexity. The simulation results show that the proposed method is better than other conventional methods in terms of PSNR, SSIM and visual quality.

References

[1] D. T. Vo, T. Q. Nguyen, S. Yea, and A. Vetro, "Adaptive fuzzy filtering for artifact reduction in compressed images and videos," *IEEE Transactions on Image Processing*, vol. 18, no. 6, pp. 1166–1178, 2009.

[2] M. Kaneko, Y. Hatori, and A. Koike, "Improvements of transform coding algorithm for motion-compensated interframe prediction errors-dct/sq coding," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 7, pp. 1068–1078, 1987.

[3] H. S. Malvar and D. H. Staelin, "The lot: Transform coding without blocking effects," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 4, pp. 553–559, 1989.

[4] T. Jarske, P. Haavisto, and I. Defee, "Post filtering met-

hods for reducing blocking effects from coded images," *IEEE Transactions on Consumer Electronics*, vol. 40, no. 3, pp. 521–526, 1994.
[5] T. Chen, H. R. Wu, and B. Qiu, "Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts," *IEEE transactions on circuits and systems for video technology*, vol. 11, no. 5, pp. 594–602, 2001.
[6] S. Liu and A. C. Bovik, "Efficient dct-domain blind measurement and reduction of blocking artifacts," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 12, pp. 1139–1149, 2002.
[7] S. B. Yoo, K. Choi, and J. B. Ra, "Post-processing for blocking artifact reduction based on inter-block correlation," *IEEE Transactions on Multimedia*, vol. 16, no. 6, pp. 1536–1548, 2014.
[8] H.-S. Kong, Y. Nie, A. Vetro, H. Sun, and K. E. Barner, "Adaptive fuzzy post-filtering for highly compressed video," in *Proceedings of the 2004 International Conference on Image Processing (ICIP'04)*, vol. 3. IEEE, 2004, pp. 1803–1806.
[9] H.-S. Kong, A. Vetro, and H. Sun, "Edge map guided adaptive post-filter for blocking and ringing artifacts removal," in *Proceedings of the International Symposium on Circuits and Systems (ISCAS'04)*, vol. 3. IEEE, 2004, pp. III–929.
[10] T. Van Nguyen, D. T. Vo, and T. H. Do, "Highly noise resistant beltrami-based texture maps with window derivative," in *Proceedings of the 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS)*. IEEE, 2015, pp. 71–75.
[11] D. T. Võ and T. Q. Nguyen, "Localized filtering for artifact removal in compressed images," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 1269–1272.
[12] A. Jain, M. Gupta, S. Tazi *et al.*, "Comparison of edge detectors," in *Proceedings of the 2014 International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom)*. IEEE, 2014, pp. 289–294.
[13] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the sobel operator," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, 1988.
[14] N. Sochen, R. Kimmel, and R. Malladi, "A general framework for low level vision," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 310–318, 1998.
[15] N. Houhou, J.-P. Thiran, and X. Bresson, "Fast texture segmentation based on semi-local region descriptor and active contour," *Numerical Mathematics: Theory, Methods and Applications.*, vol. 2, no. EPFL-ARTICLE-140431, pp. 445–468, 2009.
[16] E. Nadernejad, S. Forchhammer, and J. Korhonen, "Artifact reduction of compressed images and video combining adaptive fuzzy filtering and directional anisotropic diffusion," in *3rd European Workshop on Visual Information Processing (EUVIP)*. IEEE, 2011, pp. 24–29.
[17] T. Van Nguyen, T. H. Do, and D. T. Vo, "Advanced texture-adapted blocking removal for compressed visual content," in *Proceedings of the International Conference on Advanced Technologies for Communications (ATC)*. IEEE, 2015, pp. 285–290.
[18] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2. IEEE, 1999, pp. 1033–1038.
[19] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum, "Real-time texture synthesis by patch-based sampling," *ACM Transactions on Graphics (ToG)*, vol. 20, no. 3, pp. 127–150, 2001.
[20] W. S. Malpica and A. C. Bovik, "Range image quality assessment by structural similarity," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009)*. IEEE, 2009, pp. 1149–1152.

Thai V. Nguyen graduates in telecommunication electronics at the Can Tho University in 2003 and receives M.S degree in electronics engineering from the Ho Chi Minh City University of Technology (HCMUT), HCMC, Vietnam, in 2009. He is currently pursuing the Ph.D. degree at HCMUT. He is working at MobiFone Corporation, Vietnam. His research interests are image and video quality enhancement, MIMO model, cellular network.

Tuan Do-Hong received the B.S. and M. Eng. degrees in electrical engineering from the Ho Chi Minh City University of Technology, Vietnam, in 1994 and 1997, respectively, the M.Sc. and Ph.D. degrees in communication engineering from the Munich University of Technology, Germany, in 2000 and 2004, respectively. Since 1994 he has been with the Faculty of Electrical and Electronics Engineering, the Ho Chi Minh City University of Technology, Vietnam. His research interests include stochastic signal processing and applications for image and video processing.

Dung T. Vo (S'06 - M'09) received the B.S. and M.S. degrees from HCMUT, Vietnam, in 2002 and 2004, respectively, and the Ph.D. degree from the University of California at San Diego, La Jolla, in 2009. He has been a Fellow of the Vietnam Education Foundation (VEF) and is with HCMUT since 2002. He interned at Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, and Thomson Corporate Research, Princeton, NJ, in the summers of 2007 and 2008, respectively. He has been a staff 2 research engineer at the Digital Media Solutions Lab, Samsung Research America, Irvine, CA, since 2009. He receives the Special Merit Awards for Outstanding Paper at IEEE Conference on Consumer Electronics (ICCE) 2011 and 2012. His research interests are algorithms and applications for image and video coding and post-processing.