# Blocky Artifacts Detection Algorithm for Compressed Digital Image

Suhaila Sari [a *], Mohamad Affendi Nordin[b], Hazli Roslan[c], Zarina Tukiran[d], Siti Zarina Mohd Muji[e], Nan Mad Sahar[f], Tasiransurini Ab Rahman[g]

EmbCos Research Group, Department of Computer Engineering,

Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia (UTHM),

86400 Parit Raja, Batu Pahat, Johor, Malaysia

[a] suhailas @uthm.edu.my, [b] keypassleypass @gmail.com,

{[c]hazli, [d]zarin, [e]szarina, [f]nan,[g]surini} @uthm.edu.my

**Keywords:** Image processing, Image compression, Blocky artifacts detection

**Abstract.** Image compression is a very important issue for many applications in the field of visual communications. The purpose of image compression is to reduce the storage and transmission costs while maintaining the image quality. Nowadays, image previewing on a mobile device is a practice that takes a great implementation of everyday live. However, the image compression with very low bit rates that is typically used on mobile platforms, usually may introduce visible compression artifacts, which is referred to blocky artifacts. This displaced blocky artifacts, although visually noticeable and annoying, is particularly difficult for automated detection because its location is a priori unknown, and its appearance might be easily mistaken for some real edges or fine details in the image. Therefore, the detection of the blocky artifacts is important to ensure the deblocking process is performed on the blocky artifacts only, not on real edges or fine details of the image. This paper proposes the development of a technique which aims in blocky artifacts detection for compressed digital image. The detection of blocky artifacts presented in this paper is performed in two parts that are vertical and horizontal detections utilizing proposed detection algorithms. The effectiveness of to detect the blocky artifacts detection without including the image edges and fine details is depending on the threshold value setting in the vertical and horizontal detections algorithms. After both detections are completed, then the last step is to combine both edgemaps into a new image which includes the blocky artifacts at the boundary between two different luminance gradients. The algorithm is developed in MATLAB software. The analysis for the result is made based on qualitative observation. For blocky artifacts detection, the proposed technique has achieved its objectives in detecting blocky artifact at the boundary between two different luminance gradients. Thresholding process separated the unwanted image fine details and edges, providing an output image a view of clearer blocky artifacts existing in the compressed image. It is also found that the proposed technique could detect blocky artifacts more effectively (without including image fine details and edges) in comparison to conventional techniques.

## Introduction

Image previewing on a mobile device is a practice that nowadays takes a great share of everyday living. However, image compression with very low bit rates that is typically used on mobile platforms usually introduces visible compression artifacts. This paper proposes the development of a method which aims in blocky artifacts detection for compressed digital image. Blocky artifacts appear when digital image is encoded with block based compression scheme, which may appear at the borders of subdivided pixels blocks (fixed blocky artifacts, as in still images). Blocky artifacts, although visually noticeable and annoying, is particularly difficult for automated detection. This is because its location is a priori unknown, and its appearance might be easily mistaken for some real edges or details in the image. Therefore, detection of blocky artifacts is important for deblocking to improve the visibility of compressed image.

Generally used methods in reducing blocky artifacts can be categorized into implementation in two domains. Based on previous researcher on image compression, techniques used [1]-[5] are implied either in spatial domain or frequency domain. In [2], 1D smoothing filter is used for complex regions, while 2D adaptive filtering is used in smooth regions. Most of the researches proposed iterative methods [6], where initially closed convex constraint sets are defined. Iterative computations of alternating projection onto these convex sets recover the original image from coded image. However, these methods usually have high computational complexity, thus are difficult to adapt to real time image processsing application.

In [7], G.A. Triantafylidis et.al proposed a novel frequency-domain technique for image blocky artifacts detection and reduction. This method detects regions of image which present visible blocky artifacts. This detection is performed in frequency domain and uses estimated relative quantization error calculated when the discrete cosine transform (DCT) coefficients are modeled by a Laplacian probability function. It is constrained by the quantization upper and lower bound.

The method in [8] introduced a novel and enhanced form of mean square of different slope (MSDS) in frequency domain which involves all neighbouring blocks, including diagonally located neighbouring blocks. A novel blockiness detection method which reduce the time and computational load of deblocking algorithm is proposed. This method has complex and long operation due to its high precision step.

I.O. Kirenko et.al [9] proposed a new method blocky artifacts reduction of MPEG compressed video sequences. The blocky artifacts reduction method can be applied in a system, where encoded bit-stream is not available. This method adapts filtering automatically based on a local spatial analysis and a block grid visibility. Thus, no external control parameters are required.

In [10], A. Gandam and J.S Sidhu proposed a new post-processing algorithm based on signal adaptive filtering along with corner outlier detection or replacement scheme. The proposed method uses signal adaptive filtering, corner outlier detection and replacement scheme. The 2-D median filter is used in filtering of smooth area. The algorithm reduces the blocky artifacts, such as grid noise, staircase noise and corner outlier, without degradation of image details.

In [11], K. Singh and P. Kumar attempt to further improvement on [12] by adding the concept of corner outlier detection and replacement algorithm. The corner outlier is visible at the corner point of 8 x 8 block at intermediate mode. In this mode, only pixels near the boundary are selected for the filtering window. Grayscale values are modified within the specified range around the grayscale values of neighbouring pixels. This method is simple and involves no change to wavelet transform.

With the use of existing blocky artifacts detection methods, some amount of blocky artifacts can be detected but at the same time some edges and fine details of image are included in detected edgemap. Therefore, a development of detection method which can balance the trade-off between blocky artifacts detection and without mistakenly including edges and fine details is still required. The main objective is to design a detection algorithm for blocky artifacts at the block border in compressed digital image. The results of detected blocky artifacts are analyzed and the performance of proposed method is compared to conventional detection methods. The proposed method is developed by using MATLAB software and simulated for digital natural images.

## Proposed Detection Method

The flow of this method can be divided into 3 parts: vertical detection, horizontal detection and combination of both vertical detection and horizontal detection to form the final edgemap image.

### Vertical Detection

Pixel continuity is tested by calculating luminance gradient values that act as low pass filter so that small detail is not mistaken as pixel discontinuity. This is the most important part to separate edges and blocky artifacts in proposed method. Two variables are calculated :

$$A\,(i,j) = [\,f\,(x,y) - f\,(x,y+1)\,] - [\,f\,(x,y-1) - f\,(x,y)\,] \tag{1}$$

$$B\,(i,j) = [\,f\,(x,y) - f\,(x,y+1)\,] - [\,f\,(x,y+1) - f\,(x,y-2)\,] \tag{2}$$

$A(i,j)$ and $B(i,j)$ are discontinuity vertical luminance gradients at pixel location $(x,y)$ and $(x,y+1)$ repectively. $A(i,j)$ and $B(i,j)$ are checked based on threshold value, $Th$. If the condition is met, then the boundary between $f(x,y)$ and $f(x,y+1)$ is considered as pixel discontinuity. Then edgemap of vertical detection, $g(i,j)$ is generated based on:

$$g(i,j) = \begin{cases} 255 & \text{if } (A(i,j) \geq Th) \text{ and } (B(i,j) \geq Th) \\ 0 & \text{if } (A(i,j) < Th) \text{ and } (B(i,j) < Th) \end{cases} \quad (3)$$

**Horizontal Detection**

Next the horizontal detection is performed in horizontal direction:

$$C(i,j) = [f(x,y) - f(x+1,y)] - [f(x-1,y) - f(x,y)] \quad (4)$$

$$D(i,j) = [f(x,y) - f(x+1,y)] - [f(x+1,y) - f(x-2,y)] \quad (5)$$

Both variables are checked based on threshold value, $Th$. If the condition is met, the boundary between $f(x,y)$ and $f(x,y+1)$ is considered as pixel discontinuity. The new edgemap image of horizontal detection $h(i,j)$ is generated based on:

$$h(i,j) = \begin{cases} 255 & \text{if } (C(i,j) \geq Th) \text{ and } (D(i,j) \geq Th) \\ 0 & \text{if } (C(i,j) < Th) \text{ and } (D(i,j) < Th) \end{cases} \quad (6)$$

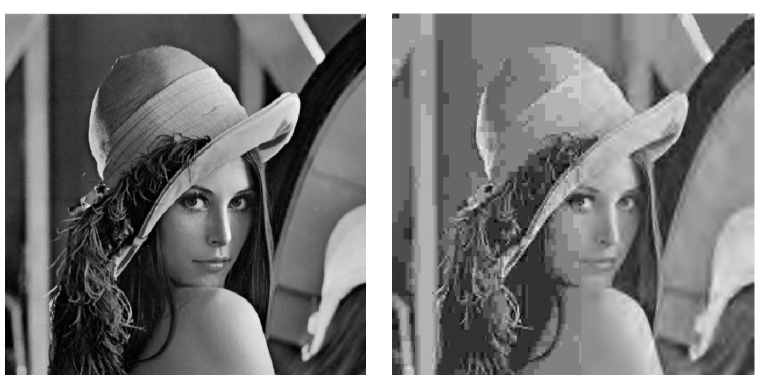
**Combination Detection**

After vertical and horizontal detection are completed, both $g(i,j)$ and $h(i,j)$ edgemaps generated are examined again for every pixel locations to be combined for final image. The final image generated is referred to the last edgemap image produced denoted as $k(i,j)$.

$$k(i,j) = \begin{cases} 255 & \text{if } g(i,j) = 255 \text{ or } h(i,j) = 255 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

**Result And Discussion**

For comparison purposes, proposed method is tested by using natural images from the SIDBA database. For simplicity, results for "Lena" image with the size of 256 x 256 is shown for discussion purpose. The image is partially compressed using DCT compression of 8-by-8 sub-blocks division to investigate the effectiveness of proposed method to differentiate blocky artifacts with real edges or details. The performance is evaluated by using subjective image quality evaluation method, which is visual effects comparison of restored images. Proposed method is also compared with conventional detection methods such as zerocross detector and Prewitt detector.

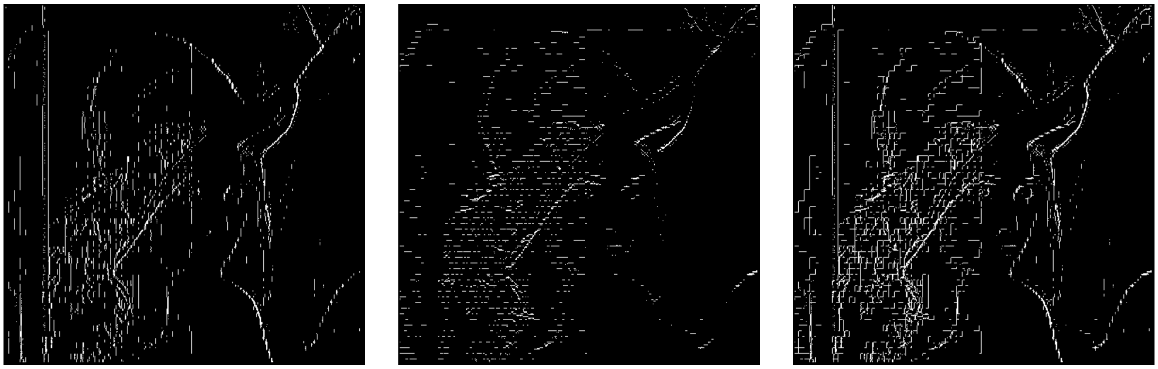


(a) Original image "Lena" (b) Test image "Lena"

Fig. 1 : Utilized image for proposed method analysis

**Vertical, Horizontal and Combination Detections' Edgemap Results**

For test image "Lena" shown in Fig. 1, the threshold value settings is 5. From resulting edgemap in Fig. 2, if the threshold values is equal to 5, both blocky artifacts and strong edges are detected by the proposed method. However, we can observed that fine details has been excluded succesfully.

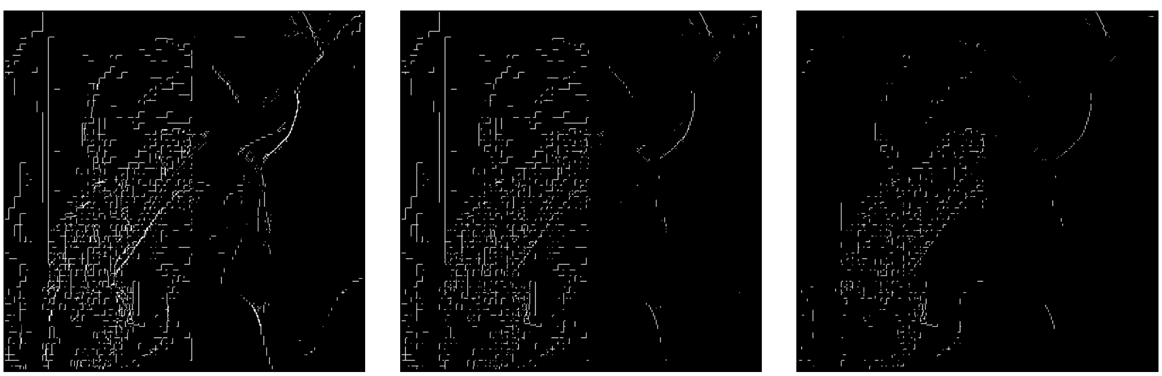


(a) $g(i,j)$ (b) $h(i,j)$ (c) $k(i,j)$

Fig. 2: Proposed method edgemap result: (a) Vertical detection edgemap, $g(i,j)$, (b) Horizontal detection edgemap, $h(i,j)$, (c) Combination of vertical and horizontal detection edgemap, $k(i,j)$

**Optimum Threshold Value Setting**

Optimal threshold value settings, $Th$ is fixed as follows for implementation in practice. Selecting suitable threshold value is crucial for preventing mistakenly including real edges into resulting edgemaps. Threshold values are investigated ranging from 5 to 30. For convenient comparison, Fig. 3 illustrates the final edgemap $k(i,j)$ images by using $Th$ ranges = 10, 20 and 30, respectively.



(a) $Th = 10$ (b) $Th = 20$ (c) $Th = 30$

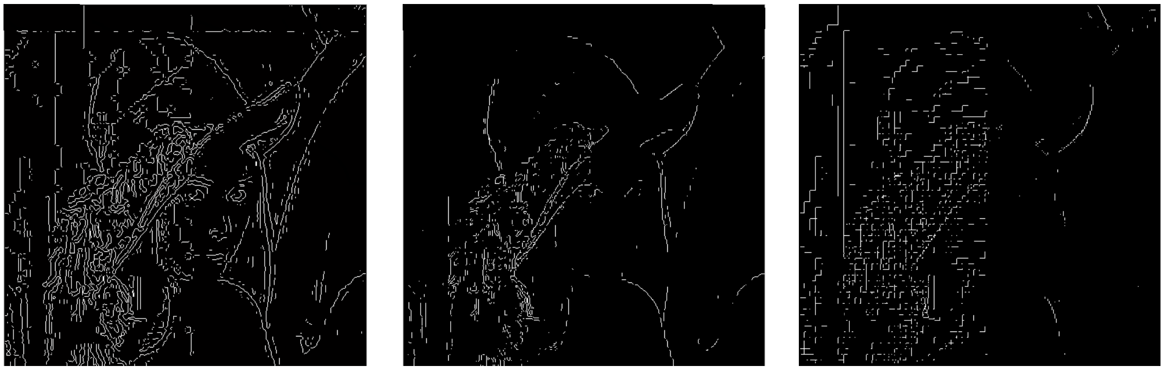
Fig. 3: Final edgemap $k(i,j)$ images by using different $Th$ ranges

It can be observed that when the threshold value is low, blocky artifacts and strong edges appear in the resulting edgemap. On the other hand, larger threshold values caused the image edges to dissappear but the blocky artifacts are affected as well. From observation made to the images, the suitable threshold value for this method is $Th$ = 20 because the image still preserved blocky artifacts while discarding unwanted edges.

Blocky artifacts are preserved due to the calculation were made in (1), (2), (4) and (5). $A(i,j)$ and $B(i,j)$ are used for vertical detection and $C(i,j)$ and $D(i,j)$ for horozontal detection, respectively. The calculation is simply manipulates the values of pixel intensity and compared it with neighbouring pixels. The difference intensities will determine the luminance gradient that later will be used to construct new edgemap image. Low luminance gradient shows that it includes mostly image edges and high luminance gradient will represent blocky artifacts. Therefore, if the value of luminance gradient is higher than threshold value, blocky artifacts will be preserved.

**Detection Methods Comparison**

The zerocross detector and Prewitt detector are performed to compare the performance of proposed method with conventioned methods for blocky artifacts detection in compressed image. Both methods are implemented in MATLAB and utilizing default parameter settings.



(a) zerocross detector (b) Prewitt detector (c) Proposed method ($Th_{g,h}$ =20)

Fig. 4 : Final edgemap $k(i,j)$ images by using different detection methods

For blocky artifacts detection, it is found that the proposed method could detect blocky artifacts more effectively (without include image edges) in comparison to conventional methods.

**Conclusion**

Though simulation results, the proposed method has achieved its objectives in detecting blocky artifacts at the boundary between two different luminance gradients. Thresholding setting plays an important role to ensure blocky artifacts in an image are detected correctly without including real image details and edges. For further work, quantitative analysis also must be performed in order to verify the results obtained since current results analysis is based on qualitative observation.

**References**


[1] H. Reeve, J. Lim, Reduction of blocking effect in image coding, Int. Conf. Acoustics, Speech, and Signal Processing (1983) 1212-1215.

[2] Y.F. Hsu, Y.C. Chen, A new adaptive separable median filter for removing blocking effects, IEEE Trans. Circuits Systems and Video Technology (1993) 510-513.

[3] T. Meier, K. N. Ngan, G. Crebbin, A region-based algorithm for enhancement of images degraded by blocking effects, IEEE Tencon1 (1996) 405-408.

[4] W. C. Zhang, W. J. Fang, X.Z, Adaptive reduction of blocking artifacts in DCT domain compressed images, IEEE Trans. Consumer Electronics, vol. 50, pp. 647-654, 2004.

[5] G. Lakhani and N. Zhong, Derivation of prediction equations for blocking effect reduction, IEEE Trans. Circuits Systems and Video Technology 9(1999) 415-418.

[6] S. Minami, A. Zakhor, An optimization approach for removing blocking effects in transform coding, IEEE Trans. Circuits Systems and Video Technology 5(1995) 74–82.

[7] G. A. Triantafyllidis, D. Tzovaras, M. G. Strintzis, Blocking artifact detection and reduction in compresse data, IEEE Trans. Circuits Systems and Video Technology 12(10)(2002) 877-890.

[8] I. O. Kirenko, R. Muijis, L. Shao, Coding artifact reduction using non-reference block grid vision measure, IEEE Transactions on Circuits Systems and Video Technology(2006) 469-472.

[9] I. O. Kirenko, R. Muijs, L. Shao, New method for the reduction of artifacts in MPEG compressed video sequences, IEEE Multimedia and Expo7(1)(2006) 469-472.

[10] A. Gandam, J. S. Sidhu, A post-processing algorithm for detection and removal of corner outlier, Int. J. Computer Applications, 4(2)(2010).

[11] K. Singh, P. Kumar,Algorithm for blocking artifact detection and reduction using adaptive filtering in compressed image, Int. J. Computer Applications 4(2)(2010)

[12] S. Singh, V. Kumar, H. K. Verma, Reduction of blocking Artifacts in JPEG compressed Images, Digital Signal Processing 17(2007) 225-243.