# REMOVING COMPRESSION BLOCK ARTIFACTS WITH ORTHOGONAL EDGE BASIS

*Daben Liu*
BBN Systems and Technologies
70 Fawcett Street
Cambridge, MA 02138

*Sos Agaian*
Division of Engineering
U. of Texas at San Antonio
San Antonio, TX 78249

*Joseph P. Noonan*
Dept. of EECS
Tufts University
Medford, MA 02155

## ABSTRACT

A technique to remove block artifacts in compressed images is presented. This technique is based on a local representation of the image using an orthogonal edge basis. Small image regions are classified into several prototypes according to the coefficients of the representation. Three algorithms are proposed. One technique reduces block effects by adjusting the coefficients. The other two techniques classify the local image into several prototypes and introduces different edge-preserving filtering strategies for different types. Experiments have shown promising results with improved visual qualities and higher peak-signal-to-noise-ratio (PSNR).

## 1 INTRODUCTION

DCT-based block image coding has always suffered due to block artifacts in the compressed image caused by individually processing blocks. Several approaches have been proposed for dealing with block artifacts since the DCT-based JPEG and MPEG standards have been widely used in image coding. These approaches can be classified into four categories:

1. Processing at the encoding end. New Transforms are developed as modifications of the DCT, e.g. LOT. This approach however cannot deal with the images reconstructed from the standard JPEG[5].
2. Putting constraints on the DCT. This method is based on some *a priori* knowledge of the original image. This obviously puts a limitation on its own efficiency since it needs information which sometimes is not available.[7]
3. Postprocessing at the decoding end. This is the simplest way of removing block artifacts. These methods use linear or nonlinear filters. Nonlinear filters show better performance. The disadvantage of filters is that they tend to smear the edges which cause a loss of image quality. [1] [5] [7]. In [1], we proposed a new non-linear edge-preserving filter with image gradients, which yielded better results compared to the conventional filters.
4. One interesting approach combines 1 and 3 [4]. At the encoding side, a certain percentage of random noise is added to the image. Then at the decoding side, the reconstructed image is filtered to get rid of the noise and block effects which results in a very good visual quality. This, however, has the effect of lowering the PSNR.

Block artifacts occur in small image regions because of the independent block-by-block processing during compression. As a result, efficient block removing is also done in small regions. To generalize the procedure, we propose a new filtering strategy which is based on the local representation of image with orthogonal edge basis. Since human eyes are more sensitive to the image edge than to the texture, it's advantageous to represent the image region based on edge behavior.

## 2 SHAPE ( or EDGE) IMAGE REPRESENTATION SYSTEM

We can map any local image into a combination of different shapes or edge prototypes with carefully selected orthogonal bases ($\boldsymbol{V_n}$). The mapping coefficients construct a system of a new domain which should have the following properties:

- can enhance and/or detect edges for given direction
- can show the edge direction and type of the local region
- invertible or pseudo-invertible
- small computational complexity

Note that, similar to what we have done in conventional transforms, which map signals into frequency or scale domain, here we map the local image into the "shape domain".

## 3 LOCAL REPRESENTATIONS

Let the local image (for example, 3x3 or 5x5) be $f$ and the orthogonal basis be $\mathbf{V} = \{V_0, \ldots, V_{N-1}\}$, both have local support $D$. The representation of the image would be:

$$f \cong \sum_{n=0}^{N-1} a_n V_n \qquad (1)$$

where $a_n$ is the coefficients and $N$ is the size of basis. The mean square error (MSE) of the representation would be

$$e^2 = \sum_{D} [\, f - \sum_{n=0}^{N-1} a_n V_n \,]^2 \qquad (2)$$

where $D$ is the admissible region of $f$.

Formula (2) provides an objective criteria for evaluating the representations. Let's investigate the following two cases.

1. Ideally, we want the error to be 0, namely,

$$f = \sum_{n=0}^{N-1} a_n V_n \qquad (3)$$

Multiply $V_i$ on both side of the equation and note that $V_n$ 's are orthogonal to each other. We get

$$f V_i = a_i V_i \qquad (4)$$

which means that the bases have to be the eigenvectors of image $f$ and the coefficients are the eigenvalues, respectively. The usability of this case is limited because the representation is data-dependent.

2. More general case is to minimize the MSE, i.e., find the partial derivative of (2) with respect to $a_n$ which leads to

$$a_n = \sum_{D} f V_n \,/\, \sum_{D} V_n^2 \qquad (5)$$

Here $\mathbf{V}$ can be arbitrary orthogonal sets. However only with $a_n$, perfect reconstruction cannot be done. Formula (1) can be expressed as

$$f = \sum_{n=0}^{N-1} a_n V_n + \beta \qquad (6)$$

where $\beta$ is the resulting error. Formula (6) provides a generalized tool for the analysis of image $f$. With a carefully chosen $\mathbf{V}$, the coefficients $a_n$ can separate different properties that will be individually treated.

## 4 ORTHOGONAL EDGE BASES

Conventionally, the information of gray-scaled image is categorized into edge and texture. Edges are critical to image perception[2] which motivates a new representation scheme that uses edge prototypes as the basis.

Studies on edge-detection have thoroughly investigated different edge behaviors and some successful prototypes have been presented by Frei and Chen[3]. A 3x3 set of prototypes are expressed as local masks shown in Fig. 1

Taking these prototypes as the orthogonal bases to decompose a local image region, the resulting

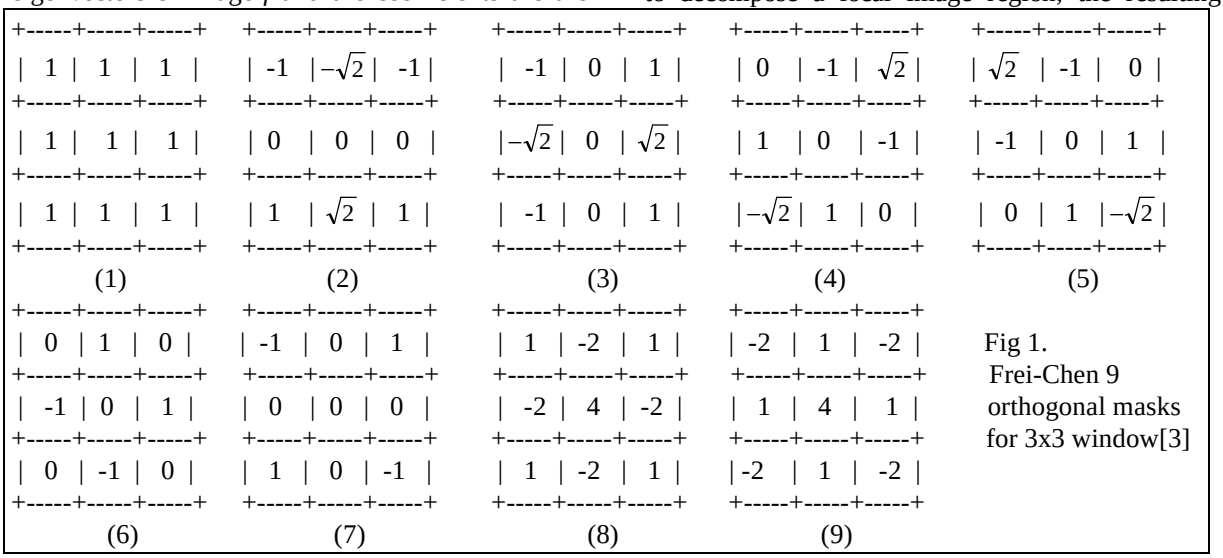


Fig 1. Frei-Chen 9 orthogonal masks for 3x3 window[3]

coefficients can give us separable properties which may be useful for further processing. For example, the 2nd and 3rd masks will get larger coefficients when masking a horizontal and vertical edged image, respectively. The 4th and 6th masks are sensitive to step edges. These different behaviors on different kinds of edges are very useful for edge detection and other related applications.

## 5 REMOVING THE BLOCKING ARTIFACTS

In DCT coded images, almost all undesirable effects occur on the boundaries of compression blocks. Local representations of such effects provide an easy and efficient way of reducing these effects, since such representation facilitates the use of effective algorithms.

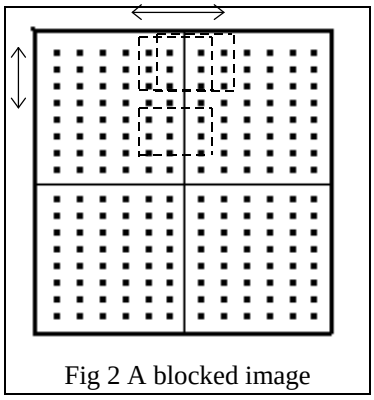

Fig 2 A blocked image

Fig 2 shows how the blocking artifacts are going to be processed. The two solid lines represent the blocking artifacts. The dots are the pixels and the dashed boxes are the analysis window which will shift along or across the block boundaries as indicated with the two-side arrow lines.

The analysis window each time grabs a local region and the representations of this region is found with formula (5). Three algorithms are proposed to smooth the blocks.

***Algorithm 1 Filtering in the Shape Domain***
Step: 1: Perform the orthogonal local shape representation with formula (5)

Step 2: Multiply the coefficients with some weighting coefficients $w_n$ :

$$a_n^{'} = a_n \cdot w_n \tag{7}$$

Step 3: Perform the inverse transform

$$f^{'} = \sum_{n=0}^{N-1} a_n^{'} w_n V_n + \beta \tag{8}$$

Assume that $V_0$ is corresponding to (1) in Fig 1, $V_1$ to (2), and so on so forth. $f^{'}$ is the smoothed image.

The $w_n$ are chosen empirically. The basic rule is that :

- $w_0$=1. Since the DC component is always the strongest in image, preserve the value of $a_0$ will guarantee that the reconstructed image will have the same energy level as the original.
- $w_1$ and $w_2$ are set according to the current analysis region. If it's on a horizontal effect, $w_1$ is to be smaller than 1. Otherwise $w_2$ is to be reduced. On the cross point, both are to be set less than 1. Same treatment is given to $w_3$ and $w_5$.
- The 9th mask in Fig 1 is sensitive to both horizontal and vertical effects. As a result, $w_8$ is usually set to a small value.

One example of the selected values of $w_n$'s is { 1 0.5 1 0.5 1 1 1 1 0.5}, for removing horizontal block effects.

***Algorithm 2. Edge-preserving filtering***.

The decomposition with formula (5) using the edge bases given in Fig 1 provides an accurate way of edge detection. If the edge information can be retrieved and projected onto these 9 prototypes, different filtering strategies can be applied on different prototypes. In doing so, the original edge information can be preserved while filtering out the blocks.

When choosing edge-preserving filters, one has to be careful on the decision of what kind of edge to preserve because the block effects themselves are also edges.

- When smoothing horizontal edges, we consider three directions: top-bottom, topright- bottomleft and topleft-bottomright and two types: step and line. The smoothing steps are:

Step 1: Perform the orthogonal local shape representation with formula (5)

Step 2: Find the maximum projection :

$$Max\_proj = \max(|a_2|,|a_3|,|a_4|,|a_7|,l)$$

where $l$ is the threshold whose value depends on the image data.

Step 3: Make decision about the edge:

if $Max\_proj = l$ ➔ *No edge*

if $Max\_proj = |a_2|$ ➔ top-bottom step or line

if $Max\_proj = |a_3|$ ➔ topleft-bottomright step

if $Max\_proj = |a_4|$ ➔ topright-bottomleft step

if $Max\_proj = |a_7|$ ➔ line, use $a_6$ determine the direction:

if $a_6 > 0$ ➔ topright-bottomleft line

else ➔ topleft-bottomright line

Step 4. Filtering along the edge direction

$$f(p, dire) = \left[ \sum_{i=0}^{N-1} w_i f^{p}(i) \Big/ \sum_{i=0}^{N-1} w_i \right]^{1/p} \qquad (9)$$

When $p=1$ , the filter is a linear weighted mean filter. When $p>1$, (8) becomes nonlinear. Experiments show that using a weighted mean filter ($p=1$) which puts more weight on the central pixel gives a better result. The *dire* means $f$ is also determined by the edge direction.

Fig 3 shows how the edge direction is related to filtering procedure. Step 3 classifies the local images into 5 cases according to the edge bases. In Fig 3. the first three are steps and the last two are lines.

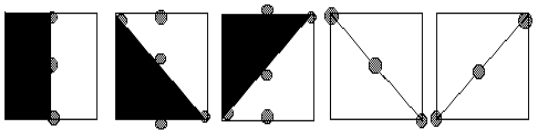

Fig 3 Five classes of edges and different selection of pixels for filtering
(From left to right: top-bottom, topleft-bottomright step, topright-bottomleft step, topleft-bottomright line, topright-bottomleft line)

The gray dots are the pixels to be used in estimating the center pixels. We assume that these pixels are much more correlated to the estimated pixel than those that are not selected. In doing so, we prevent the original edge from being blurred by filtering.

- When smoothing vertical edges, the procedure is the same. Only that now we want to investigate the left-right type edges using $a_1$ instead of the top-bottom type, which is now a degraded artifacts.

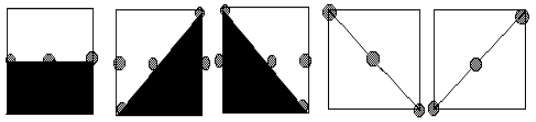

Fig 4 Five classes for removing vertical block effects

Fig 4 shows the classification and pixel selection for filtering in this case.

***Algorithm 3.*** Keep step 1, 2 and 3 of Algorithm 2. In step 4, we use moving nonlinear filtering with a shifting window of 5 pixels. The shifting is consistent with the edge direction obtained from the previous steps.

## 6 EXPERIMENTS AND RESULTS

Experiments were performed on several images with different values of $p$ and different kinds of filters. All showed promising results. Here the results for 256x256 lena image with 256 gray-levels are given. Sixteen by sixteen block DCT coefficients were calculated. The image was then compressed to a ratio of 20:1. In Fig 5, the face part is enlarged so that the blocks can be easily seen.

Results of two proposed methods of removing block artifacts are shown. For comparison Fig 6 gives the image processed with method proposed in [1], which has proved to be better than other conventional filters[1]. Fig 7 shows the de-compressed image which is processed by modifying the representation coefficients. Fig 8 shows the result of using edge-preserving filtering. All three show the smoothed version of lena image. The edge-preserved one has a better visual quality.

PSNR is also calculated in **dB**:

$$PSNR = 10 \cdot \log_{10} \left[ \frac{(255N)^2}{\sum_{N^2} (f - f_c)^2} \right] \qquad (10)$$

where $N$ is the number of gray-levels (256 here), $f$ is the original image and $f_c$ is the processed image. The results are also shown in Fig 6, 7, 8 and 9.

We can see that all methods have slightly improved the PSNR. Edge-preserving filtering achieves a better PSNR than the other two.

## 7 CONCLUSION

We've proposed a local orthogonal edge system and based on the system, directional filters both in shape domain and spatial domain are designed for removing transform block artifacts. All experimental results using the proposed algorithms have shown that they can efficiently get rid of the annoying block artifacts as well as improve the PSNR of the reconstructed image.

## 8 REFERENCE


1. D. Liu, S. Agaian, J.P. Noonan, "Removing DCT Block Artifacts with Nonlinear Filters", Proceeding of EI '97-- IS&T/SPIE's SYMPOSIUM ON ELECTRONIC IMAGING SCIENCE & TECHNOLOGY, Paper Number 3026-30.
2. Agaian and A. Petrosian, "On the Relationship Between Compression Coefficient Precision of Reconstruction and Complexity of Calculations," J. Inform. Process. Cybernet. EIK 7, Germany, 1991, pp335-345
3. Werner Frei and Chung-Ching Chen, "Fast Boundary Detection: A Generalization and a New Algorithm", *IEEE Trans. on Computer*, Vol, C-26, No.10, Pg. 988-997, Oct. 1977
4. Tsuhan Chen, "Elimination of Subband-Coding Artifacts Using The Dithering Technique", *IEEE Proc. 0-8186-6950-0/94*, Pg. 874-877, 1994
5. Jechang Jeong and Byeungwoo Jeon, "Use of a Class of Two-Dimensional Functions for Blocking Artifacts Reduction in Image Coding", *IEEE Proceeding 0-8186-7310-9/95*, Pg. 478-481, 1995
6. Yongyi Yang, N. P. Galatsanos and A. K. Katsaggelos, "Iterative Projection Algorithms For Removing The Blocking Artifacts of Block-DCT Compressed Images", *IEEE Proceeding , 0-7803-0946-4/93*, pg. V-405-408, 1993
7. Robert L. Stevenson, "Reduction of Coding Artifacts in Transform Image Coding", *IEEE Proceeding, 0-7803-0946-4/93*, pg V-405-408, 1993


Fig 5 The compressed blocky image, (PSNR=28.9dB)

Fig 6. The de-blocked image with image gradient proposed in [1] (PSNR=29.7dB)

Fig 7. The de-blocked image with coefficient modification (PSNR=29.6dB)

Fig 8 The de-blocked image with edge-preserving filtering (PSNR = 30.1dB)