Coding Artifact Reduction Based on Local Entropy Analysis

Ling Shao, Ihor Kirenko

Abstract — *Coding artifacts are very annoying in highly compressed images and video sequences. Most artifact reduction techniques blur the details of the images while removing various coding artifacts. In this paper, we propose a novel and explicit approach for classifying blocks into detailed regions, intermediate regions and smooth regions. The classification is based on the information content of the underlying region. The information content of a region is quantized by local entropy, which is calculated on the PDF of the pixel intensity distribution. Local entropy is used as an indicator of how much smoothing is needed for a certain region. It is well known that blocking artifacts are more visible in flat regions than in detailed regions. We apply mild low-pass filters on detailed regions to preserve the sharpness, and strong low-pass filters on flat regions to remove the severe blocking artifacts. Experimental results show that our proposed algorithm can preserve the details and reduce coding artifacts better than more expensive state of the art techniques*[1].

Index Terms — Coding artifact reduction, entropy, blocking artifacts, adaptive filter.

I. INTRODUCTION

Today, High Definition Television (HDTV) based on liquid crystal (LCD) and plasma (PDP) technologies offer high picture resolution with very good details. However, the source video materials are not always satisfactory. In order to improve the picture quality of source materials, high-end display manufacturers are putting much effort on designing various video processing algorithms. One of such techniques is coding artifact reduction. Due to the bandwidth limit of the broadcasting channels and the capacity limit of the storage media, video materials are usually compressed using various compression standards, such as MPEG-2 and H.264. These block-based transform coding standards divide an image or a video frame into non-overlapping blocks (usually with the size of 8 x 8 pixels), and apply Discrete Cosine Transform (DCT) on them. The DCT coefficients are thus quantized independently. In low bit rate image or video coding, the coarse quantization will usually result in various noticeable coding artifacts, including the blocking, ringing and mosquito artifacts. Among them, the most annoying are the blocking artifacts, which exhibit artificial discontinuities at block boundaries.

To reduce the blocking artifacts, many deblocking techniques have been proposed in the last decade. These methods can be classified in two types: pixel-domain filtering [2,3,9] and frequency-domain filtering [1,5,6]. For the consideration of computational complexity, pixel-domain filtering is preferred for real time applications. Such techniques usually apply low-pass filters at blocking boundaries to remove the discontinuities. The undistinguishing use of low-pass filtering will smooth the blocking boundaries, but at the same time reduce the sharpness of the image or video sequence. Some deblocking methods attempt to use different modes according to the characteristic of the underlying pixel areas, but these mode-differentiating schemes are heuristic, and usually not effective. Pan et al. [7] proposed a flatness measure of one block based on zero crossings. The total number of zero-crossings is obtained and divided by the total number of crossings to give the flatness measure in a particular region. In [2], the mode decision is characterized by the difference values between neighboring pixels. If the number of pixels which differ to a nearest neighbor is smaller than a threshold in a region, it is considered to be a flat region; otherwise, a detailed region. In [9], a similar mode decision scheme is employed. The only difference is that they use a smaller region that is adjacent to the blocking boundary to calculate the difference values, and three modes are determined.

In this paper, a novel content adaptive blocking artifact and noise reduction algorithm is proposed. We first explicitly distinguish the image into detailed regions, flat regions and intermediate regions based on the information content of the regions. Our purpose is to preserve all the details in the image when artifact reduction filtering is applied, because sharpness and details are very important information in video materials, especially for High Definition (HD) signal. We specifically demonstrate our region classification algorithm on blocking artifact reduction. It is well known that blocking artifacts are most obvious in flat regions, while in detailed regions the artifacts are barely noticeable. This is because of two reasons. The first reason is that an encoder usually allocates more bits for detailed regions than for flat regions, which results in fewer artifacts in detailed regions. The second reason is due to the masking effect, which makes the artifacts or noise less noticeable for human eyes in detailed regions. Therefore, strong smoothing should be applied on flat regions, and little or no smoothing should be used on detailed regions.

The remainder of this paper is organized as follows. Section 2 describes the region classification technique based on local entropy analysis. Basically, regions with high entropy, i.e. more informative regions, are classified as detailed regions. The local entropy is calculated on the probability density function (PDF) of the underlying pixel intensity distribution.

[1] L. Shao and I. Kirenko are with Philips Research Laboratories, High Tech Campus 36, Eindhoven 5656 AE, The Netherlands (e-mail: l.shao@philips.com).

Manuscript received March 13, 2007 0098 3063/07/$20.00 © 2007 IEEE

In Section 3, we propose a multi-mode deblocking algorithm according to the region classification of Section 2. Some results and evaluation are presented in Section 4. Finally, Section 5 concludes this paper.

II. Entropy Analysis Based Region Classification

The performance of adaptive spatial filtering methods for image enhancement or artifact reduction is often limited due to a lack of accuracy and robustness in the content classification scheme employed. Those methods usually tune the settings of the algorithms according to some local properties such as the mean gradient or variance of pixel intensities, which are heuristic and only effective for certain occasions. We attempt to design a general-purpose approach for region classification in this section. Our approach is based on information theory, and specifically exploits the fact that a detailed region contains more information than a flat region. The information content of one region is quantized by the local entropy of the probability density function of the pixel intensity distribution inside the region. The PDF is approximated by the histogram of pixel intensities. We can observe that a detailed region has a flat and spread out histogram, while the histogram of a smooth region only has a few peaks. Fig. 1 shows the histograms of two regions, one on the eye and the other on the shoulder of Lena (Lena is a JPEG decoded image). The histogram of the eye region is a lot more spread out and distributed than the shoulder region. Note that the distribution of the histogram is dominated by the local structure of the region, i.e. noise and coding artifacts will not affect the overall distribution of the histogram.

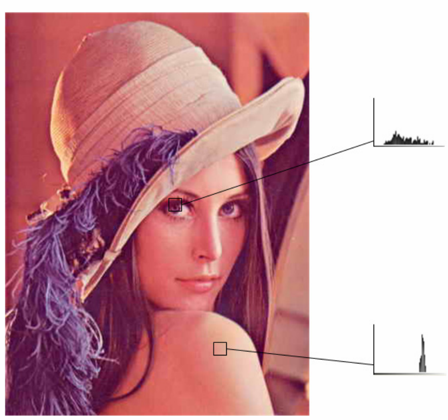

Fig. 1. Histograms of luminance intensity for two different regions.

The local entropy of a region can be defined as:

$$H_{D,R} = -\sum_{i} P_{D,R}(d_i)\log_2 P_{D,R}(d_i) \tag{1}$$

where $P_{D,R}(d_i)$ is the probability of descriptor D taking the value d_i in a local region R. The descriptors can be luminance intensity, color, orientation, phase, etc. A scale invariant version of the local entropy has been used for salient region detection and object retrieval in [8]. For simplicity and the consideration of the context of video processing, we employ luminance intensity as the descriptor. Therefore, the entropy calculation can be revised as follows:

$$H = -\sum_{i=1}^{N} P_R(i)\log_2 P_R(i) \tag{2}$$

where i indicates the bin index in the histogram, N is the total number of bins and R is a local region inside which the entropy is calculated. According to the information theory, H has a higher value for a distributed histogram than a peaked one, i.e. the entropy value of a detailed region tends to be larger than that of a smooth region. As mentioned above, the entropy value is dependent on the information content or structure of the underlying region, and noise or coding artifacts would only deviate the entropy value in a small range. Fig. 2 shows two squared regions of 8x8 pixels on Lenna compressed by JPEG using different quality levels. The entropy values of both the detailed region and the flat region do not change much with the increase of compression rate.

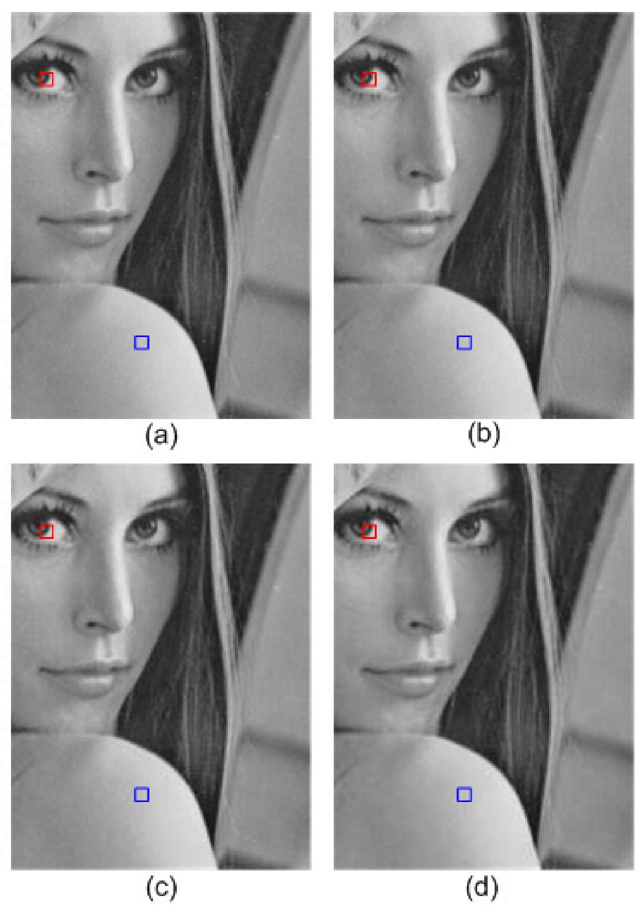

Fig. 2. Local entropy values of two regions on Lenna at different compression rates, H(1) denotes the entropy value of the eye region, and H(2) denotes the entropy value of the shoulder region. (a) Uncompressed, H(1)=3.89, H(2)=1.08; (b) JPEG quality: 80, H(1)= 3.89, H(2)=1.35; (c) JPEG quality: 50, H(1)= 3.82, H(2)= 1.17; (d) JPEG quality: 20, H(1)= 3.79, H(2)= 1.37.

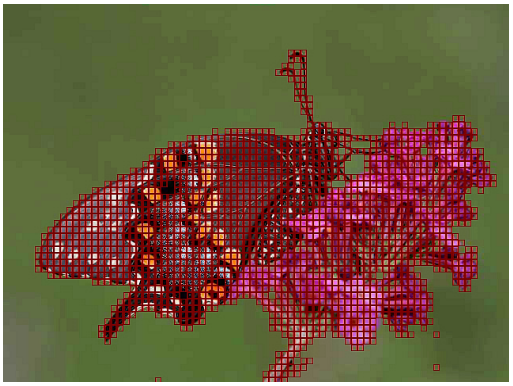

Fig. 3. Classification of regions based on local entropy. The superimposed squares indicate regions with an entropy value higher than a threshold.

Generally, the above entropy *H* can be used as an indicator of what kind of filtering should be applied on a region either for image enhancement or coding artifact reduction. We first divide the image into *N* x *N* blocks as the block-based transform coding does. On each block, we construct the histogram using the luminance intensity of the pixels within the block. Then, the local entropy of that block is calculated on the histogram using Equation (2). Some thresholds can be selected for the use of different filters on a specific block based on how much detail needs to be preserved and the quality of the video being processed. Fig. 3 depicts the classification of 8 x 8 blocks based on a single threshold of 2.8 on a decoded image containing blocking artifact. The superimposed squares indicate the blocks with an entropy *H* larger than 2.8. From Figure 3, we can see that all the detailed regions are selected, while all the smooth regions containing blocking artifacts are ignored.

III. Multi-mode Deblocking

In this section, a deblocking algorithm is proposed based on the above region classification. With the region classification, we can apply adaptive low-pass filters on different regions. In our implementation, 8x8 blocks are used for region classification. For video sequences encoded using different block sizes or scaled materials, some grid detection methods, e.g. the technique described in [4], could be used first to detect the grid position and the block size. We define three modes for detailed regions, intermediate regions and flat regions. In our implementation, non-overlapping blocks are used for computing the entropy. All pixels in a particular block are allocated the same mode. For example, if the entropy of one block is very high, we deem all the pixels inside that block to be informative; otherwise, we consider all the pixels inside that block to be less important. This implementation is efficient and is suitable for real time video processing. We can also calculate the entropy value for each pixel using the neighborhood of that pixel, but it is computationally more demanding. As we are targeting real time applications, the block based mode classification is preferred. The different low-pass filters are described in the following.

A. Deblocking for Detailed Regions

As we know, blocking artifacts are hardly perceivable in detailed regions. Besides, we want to preserve the sharpness of the detailed regions, because detailed regions contain the most information in an image. Based on the above two arguments, we apply weak low-pass filtering on detailed regions with the entropy higher than a certain threshold. The selection of this threshold can be varied according to how much detail the system should preserve. In our implementation, a threshold $T1$=2.3 is used, i.e. all the blocks with the entropy higher than 2.3 are filtered using the following weak low-pass filter.

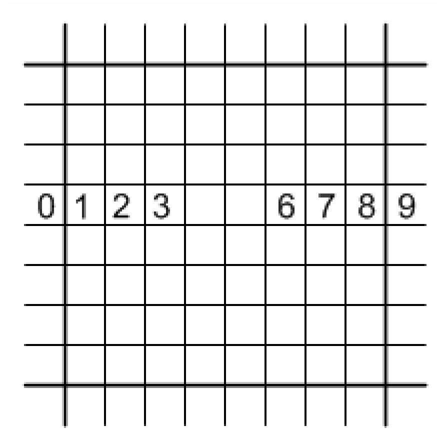

Fig. 4. Illustration of pixels that are adjacent to the block boundaries.

We explain the horizontal deblocking in the following, and the vertical deblocking is similar. Fig. 4 illustrates an 8 x 8 block, 0~3 are four pixels at two sides of the left block boundary, and 6~9 are four pixels at two sides of the right block boundary. We define the difference between two pixels adjacent to the block boundaries as:

$$\text{offset\_l} = v(0) - v(1)$$
$$\text{offset\_r} = v(9) - v(8)$$

where v(*i*) is the intensity of pixle *i*. The pixels 1~3 and 6~8 can be updated as follows:

$$\text{If offset\_l} < Th: \quad v(1) = v(1) + \text{offset\_l}/3$$
$$v(2) = v(2) + \text{offset\_l}/6;$$
$$\text{If offset\_r} < Th: \quad v(7) = v(7) + \text{offset\_r}/6$$
$$v(8) = v(8) + \text{offset\_r}/3.$$

where *Th* indicates the threshold for updating the pixels. The values of v(1)~v(3) and v(6)~v(8) are then adjusted to be in the range of (0, 255). We only update the pixels when the offset is smaller than *Th*, because when the offset is large it might be a real edge.

B. Deblocking for Intermediate Regions

We define blocks with an entropy higher than $T2$=1.8 but lower than $T1$=2.3 to be intermediate regions. Those regions are not as important as the detailed regions since they are less informative. And, there may exist more visible blocking artifacts in these regions than in detailed regions. Therefore, we apply the low-pass filtering which does not change too much of the detail but removes most of the blocking artifacts.

We define the offsets of pixels adjacent to the block boundaries the same as in the previous section. The pixels 1~3 and 6~8 can be updated as follows:

If offset_l < *Th* : v(1) = v(1) + offset_l/2
v(2) = v(2) + offset_l/4
v(3) = v(3) + offset_l/8
If offset_r < *Th* : v(6) = v(6) + offset_r/8
v(7) = v(7) + offset_r/4
v(8) = v(8) + offset_r/2

The values of v(1)~v(3) and v(6)~v(8) are then adjusted to be in the range of (0, 255).

C. Deblocking for Smooth Regions

The blocking artifacts are most noticeable in smooth regions. Strong low-pass filtering is preferred in order to remove the severe discontinuities between blocks and the noise. Most deblocking algorithms only update pixels close to the block boundaries, which is not sufficient for smooth regions, because on smooth regions the intensity value in a block is almost constant, and only updating the pixels close to the block boundaries will bring in other artifacts in the centre of the block. Therefore, we update all the pixels inside the block. The low-pass filter we employ is a 3 x 3 bilateral filter. Each pixel in a smooth region is updated according to the following pseudo code:

```
int n = 0, sum = 0;
for (int i = 1; i <= 9; i ++)
{
    if (abs(v(i) – v(5)) < Sigma)
    {
        n ++;
        sum += v(i);
    }
}
v(5) = sum/n;
```

where v(1)~v(9) represent the intensities of 9 pixels in the 3 x 3 window, and v(5) is the intensity of the central pixel. For simplicity, we employ a constant value for Sigma. Sigma can be also dependent on the local pixel variations. The use of bilateral filtering preserves the fine details in smooth regions. A block is deemed to be smooth if the entropy of that block is lower than *T2*=1.8.

In order to remove blocking artifacts and noise in even smoother regions, we apply a 5 x 5 averaging filter on blocks with the entropy smaller than *T3*=1.5.

The thresholds *T1~T3* can be varied depending on resolution, bit-rate and some image quality metrics. In our implementation, constant values of *T1~T3* are employed for simplicity. When *T1~T3* are adaptive to specific sequences, the performance of our algorithm can be further improved.

IV. Results and Evaluation

We evaluated our proposed algorithm on a number of video sequences with the resolutions of CIF, SD and HD compressed by an MPEG-4 encoder at different bit-rates. The objective metrics we use are Peak Signal-to-Noise Ratio (PSNR) and Generalized Block-Edge Impairment Metric (GBIM) [10]. For benchmarking, we compared the results of our algorithm with three state of the art coding artifact reduction techniques. The first technique was proposed by Kirenko [3] based on the spatial analysis of luminance and chrominance. The second is the algorithm presented by Nosratinia [6] using re-application of JPEG. The third is a two-mode technique developed by Kim et al. [2]. All methods are computationally more expensive than our proposed algorithm. The results of PSNR and GBIM of the three algorithms are presented in Tables 1 and 2. For GBIM, the average of the horizontal GBIM and vertical GBIM is used. The tables show that our proposed algorithm performs better than the methods in [3, 2] both in PSNR and GBIM for all the different bit-rates. The performance of the method in [6] is comparable to ours, but our algorithm can preserve the details and reduce the artifacts better at higher bit-rates.

TABLE 1
PSNR COMPARISON OF TEST SEQUENCES.

Sequence	Bit-rate (Mbit/s)	Ref [3]	Ref [6]	Ref [2]	Proposed
Foreman (CIF)	0.1	29.75	30.02	29.83	29.88
	0.2	31.38	32.01	31.56	31.98
	0.5	32.87	34.35	33.24	34.59
	1.0	33.68	36.06	34.47	36.62
Girlsea (SD)	1.0	31.71	33.06	32.23	32.80
	2.0	33.28	35.69	34.08	35.43
	3.0	33.99	37.18	35.39	37.09
	4.0	34.41	38.14	36.54	38.29
	5.0	34.64	38.90	37.16	39.15
Fashion (HD)	4.0	40.18	40.31	40.20	40.25
	6.0	41.05	41.14	41.11	41.17
	8.0	41.47	41.62	41.56	41.68
	10.0	41.62	41.80	41.79	41.87

TABLE 2
GBIM COMPARISON OF TEST SEQUENCES.

Sequence	Bit-rate (Mbit/s)	Ref [3]	Ref [6]	Ref [2]	Proposed
Foreman (CIF)	0.1	1.63	1.28	1.75	1.50
	0.2	1.48	1.20	1.58	1.31
	0.5	1.46	1.16	1.53	1.18
	1.0	1.45	1.15	1.50	1.13
Girlsea (SD)	1.0	1.45	1.21	1.55	1.47
	2.0	1.28	1.13	1.36	1.25
	3.0	1.25	1.11	1.31	1.15
	4.0	1.23	1.09	1.27	1.10
	5.0	1.21	1.08	1.23	1.05
Fashion (HD)	4.0	1.25	1.15	1.32	1.03
	6.0	1.20	1.11	1.24	0.94
	8.0	1.18	1.08	1.21	0.90
	10.0	1.16	1.08	1.18	0.88

We also performed subjective evaluation on a variety of test sequences with different resolutions and bit-rates. Fig. 5 shows the screen-shots of a decoded frame of the sequence "Foreman" at the bit-rate of 200kbit/s before and after applying different coding artifact reduction algorithm. We can see that the proposed deblocking algorithm can reduce blocking artifacts and preserve object details better than the other methods.


(a)


(d)


(b)


(e)


(c)

Fig. 5. A cropped frame of the "Foreman" sequence before (a) and after coding artifact reduction by (b) the method in Ref [3], (c) the method in Ref [6], (d) the method in Ref [2], and (e) our proposed algorithm.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel coding artifact reduction method based on a specific region classification. Regions with high entropy are processed using mild low-pass filters to preserve the details, and regions with low entropy are processed using strong low-pass filters to remove severe blocking artifacts and noise. The experimental results show that our proposed algorithm outperforms computationally more complex techniques. In terms of hardware implementation, only eight lines of memory are required for our method.

Besides coding artifact reduction, the proposed region classification method can be also employed for image and video enhancement, i.e. apply sharpness enhancement algorithms, e.g. LTI, on detailed regions, and apply low-pass filtering on smooth regions which contain mostly noise. Another application of the region classification method is image and video compression, for example adaptive quantization, i.e. assign more bits for detailed regions and less bits for smooth regions.

REFERENCES

[1] T. Chen, H. R. Wu and B. Qiu, "Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts", IEEE Trans. Circuits and Systems for Video Technology, vol.11 no. 5, pp. 584-602, Aug. 2001.

[2] S. Kim, J. Yi, H. Kim, J. Ra, "A deblocking filter with two separate modes in block-based video coding", IEEE Trans. Circuits and Systems for Video Technology, vol. 9, 1999.

[3] I. Kirenko, "Reduction of coding artifacts using chrominance and luminance spatial analysis", In Proceedings of the International Conference on Consumer Electronics, Las Vegas, USA, Jan. 2006.

[4] I. Kirenko, R. Muijs and L. Shao, "Coding artifact reduction using non-reference block grid visibility measure", In Proceedings of IEEE International Conference on Multimedia and Expo, Toronto, Canada, July 2006.

[5] Y. Luo and R. K. Ward, "Removing the blocking artifacts of block-based DCT compressed images", IEEE Transactions on Image Processing, Vol.12(7), pp. 838-842, Jul. 2003.

[6] A. Nosratinia, "Denoising of JPEG images by re-application of JPEG", Journal of VLSI Signal Processing, Vol.27, pp. 69-79, 2001.

[7] F. Pan, X. Lin, S. Rahardja, W. Lin, E. Ong, S. Yao, Z. Lu and X. Yang, "A locally-adaptive algorithm for measuring blocking artifacts in images and videos", IEEE International Symposium on Circuits and Systems, 2004.

[8] L. Shao and M. Brady, "Specific object retrieval based on salient regions", Pattern Recognition, vol.39(10), 2006, pp. 1932-1948.

[9] S. Tai, Y. Chen, S. Sheu, "Deblocking filter for low bit rate MPEG-4 video", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, 2005.

[10] H. R. Wu and M. Yuen, "A generalized block-edge impairment metric for video coding", IEEE Signal Processing Letters, Vol.4(11), pp. 317-320, Nov. 1997.