ADAPTIVE REPAIR OF COMPRESSED VIDEO SIGNALS USING LOCAL OBJECTIVE METRICS OF BLOCKING ARTIFACTS

Ihor O. Kirenko , Ling Shao

Philips Research Laboratories, Eindhoven, The Netherlands

ABSTRACT

The paper presents a new objective metrics of blocking artifacts visibility in MPEG compressed video sequences. The metric estimates a local visibility of the block grid by analyzing the discontinuity of a pixel intensity trend across the block edge and by comparing it against pixel activities within blocks. Application of the local blockiness metrics for adaptive control of low-pass filtering provides significant reduction of coding artifacts without blurring of image details. The results of our experiments show the high efficiency of the proposed approach.

Index Terms**—** image enhancement, compression, quality metric

1. INTRODUCTION

A variety of “lossy” video compression techniques are known to reduce the amount of image data that must be stored or transmitted. Sophisticated compression schemes such as JPEG, MPEG attempt to truncate spatial frequency information that is not crucial to the perception of a viewer. With high compression ratios, image artefacts may appear in the decompressed image. One of such artefacts is blocking, in which the blocks used for compression become visible in the displayed, decompressed image. Due to quantization during compression, differences between blocks in luminance and/or chrominance are created that are not present in the original image, but are visible in the decompressed image.

State-of-the-art block visibility metrics, used in methods to reduce the visibility of blocking effects, are based on the observation that the sharp block edges are more visible in smooth image areas, while in areas with high spatial activity (e.g. texture) the block grid, if present, is masked and thus is less visible. Block visibility is estimated by measuring the ratio of a pixel gradient over the block grid (i.e. between blocks) to an average pixel gradient within blocks. Such a method is disclosed in [1], where the visibility of a block edge is judged by comparing pixel level differences between pixels at either side of a block edge with pixel level differences within the blocks.

However, the standard block visibility metrics estimate blockiness not always adequately. There is, therefore, a need for an improved method of estimating and suppressing blocking artefacts that arise in decompressed pictures.

2. LOCAL BLOCKINESS METRICS

The main disadvantage of known block visibility metrics is the fact that it is based on the assumption that the blockiness is due to a visibility of a high-frequency grid only. In reality, blockiness is visible also due to a difference of pixel structures within blocks and due to abrupt changes of pixel intensity trends between two blocks. The state-of-the-art metrics are not able to reflect those changes properly.

Figure 1 shows an example of 1D pixel vector V1-V6 from two blocks divided by a block edge between pixels V3 and V4. A vertical axis shows the luminance intensities of pixels, horizontal axis – spatial position of pixels within one row or column of an image.

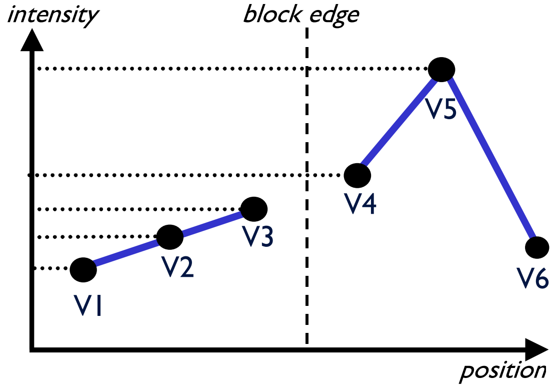

Fig. 1. Example of pixel intensity discontinuity due to the blocking artifact

According to state-of-the-art metrics [1, 2], the local visibility of block edge depicted in Figure 1 is defined as:

$$\mathrm{Bl}=\frac{|v3-v4|}{\dfrac{|v1-v2|+|v2-v3|+|v4-v5|+|v5-v6|}{4}} \quad (1)$$

In reality, blockiness is visible not only due to a relatively high value of the gradient |v3-v4|, but also due to different pixel structures within adjacent blocks.

Figure 2 illustrates an extreme example of such blockiness. There is no visible high-frequency edge if one takes only the difference in intensity into consideration, since the intensities of the blocks are almost the same at the edge. However, the blockiness is very well visible.

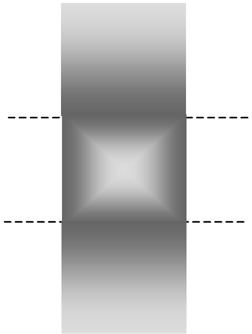

Fig. 2. Example of blockiness

Figure 3 illustrates a less extreme example of this effect. The blockiness depicted in Figure 3 would be estimated by conventional blockiness metrics as non-visible, because the difference in pixel intensity at either side of the block edge |v3-v4|=0. However, the blockiness is still visible due to a break in the pixel intensity trend.

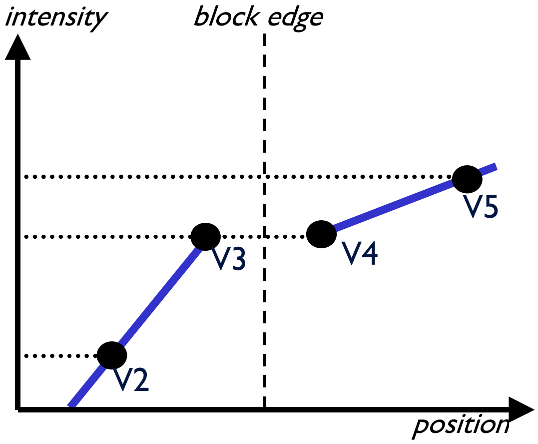

Fig. 3. Example of pixel intensity discontinuity due to the blocking artifact

We propose a novel measure for pixel intensity discontinuity over the block grid $F_{grid}(v)$, namely:

$$F_{grid}(v)=\left|(v3-v4)-\frac{v2-v3+v4-v5}{2}\right| \qquad (2)$$

The difference $F_{grid}(v)$ is a difference between a pixel gradient value across a block edge (v3-v4) and an average difference between pixel gradient values within blocks at either side of the block edge 0.5*((v2-v3)+(v4-v5)).

In other words, $F_{grid}(v)$ corresponds to the pixels gradient over a block edge, compensated by the value of an intensity mismatch at both sides of the block edge. If the intensity of pixels v2-v5 is changing gradually (e.g. from dark to bright with equal steps), or in other word if the values of pixels v2-v5 form one line of intensity (|v3-v4|=|v2-v3|=|v4-v5|), then $F_{grid}(v)$=0, even if |v3-v4|>0. At another hand, if there is a change of an intensity trend, then $F_{grid}(v)$>0 even if |v3-v4|=0.

Generally speaking, the blockiness metric is an estimation of visibility of a pixel intensity discontinuity over the block edge. The visibility of such discontinuity depends on the pixel activity within blocks. The more regular are intensity changes within blocks, more visible is a pixel intensity discontinuity over the block edge. Using the proposed definition of the pixel discontinuity $F_{grid}(v)$ (1), the local blockiness Bl is estimated in the following way:

$$\text{Bl}=\frac{F_{grid}(V)}{F_{non-grid}(V)}=\left|\frac{(v3-v4)-\dfrac{v2-v3+v4-v5}{2}}{\dfrac{\dfrac{v1-v2+v3-v2}{2}+\dfrac{v4-v5+v6-v5}{2}}{2}}\right|,$$

where

$$F_{non-grid}(v)=\left|\frac{\dfrac{v1-v2+v3-v2}{2}+\dfrac{v4-v5+v6-v5}{2}}{2}\right|$$

corresponds to the activity of pixels within blocks, taking into account changes in intensity trends within those blocks.

The analysis window for calculation of local blockiness depends on the particular filtering method, which uses the proposed metric. If the kernel size of a de-blocking filtering method is larger than six, then more pixels should be included in the calculation of $F_{non-grid}(v)$.

The behaviour of $F_{non-grid}(v)$ is following:

$F_{non-grid}(v)$ is small if pixels v1,v2,v3 and/or v4, v5, v6 construct one intensity trend (e.g. gradual change of luminance from dark to bright or vice versa). Then the blockiness is more visible and metric Bl has higher value.

$F_{non-grid}(v)$ is large if pixels v1,v2,v3 and/or v4, v5, v6 do not construct one intensity trend (e.g. fluctuations of luminance from dark to bright and vice versa). Then the blockiness is visually masked by fluctuations of intensities within blocks. In this case, the blockiness metric Bl will have smaller values.

In case values of all pixel pair gradients are equal (|v3-v4|=|v2-v3|=|v4-v5|=|v5-v6|), the state-of-the-art metric would have the same value for both above cases, but subjectively, the difference of visibilities is large. The proposed here metric is able to reflect this difference.

The function Bl takes into account the consistency of an intensity trend across two adjacent blocks. That means that if the block with pixels v1,v2,v3 has transition e.g. from bright to dark, but pixels v4, v5, v6 – from dark to bright, than the value $F_{non-grid}(v)$ will be large, thus making the value of a block visibility relatively small. Otherwise, if pixels v1-v6 form the same intensity trend (e.g. from dark v1 to bright v6), then any abrupt changes at the grid ($F_{grid}(v)$) are more visible. This is reflected in the proposed metric: in this case block visibility metric Bl would have large values.

A more generalised equation for estimation of a local block visibility between two blocks with pixels Ai and Bi:

$$Bl=\left|\frac{(A_0-B_0)-\frac{A_1-A_0+B_0-B_1}{2}}{\frac{\sum_i \frac{A_{i-1}-A_i+A_{i+1}-A_i}{2}+\sum_i \frac{B_{i-1}-B_i+B_{i+1}-B_i}{2}}{2*n}}\right|, \quad (3)$$

A0 and B0 are pixels from two blocks located directly at the block grid, pixels A1 and B1 are located at one pixel distance from the grid, *n* is a number of $\frac{A_{i-1}-A_i+A_{i+1}-A_i}{2}$ or $\frac{B_{i-1}-B_i+B_{i+1}-B_i}{2}$ components.

Block grid might coincide locally with an object edge. Block edges usually have high values of pixel gradients. Without proper protection, such edge gradients will be regarded as an extremely high blockiness. The protection of object edges from being estimated as a block grid is achieved by means of clipping of a value of the pixel intensity discontinuity over the block grid ($F_{grid}(v)$) between two thresholds Ttexture and Tedge:

If $F_{grid}(v)$ > Tedge than $F_{grid}(v)$=Tedge;

and/or

If $F_{grid}(v)$ < Ttexture than $F_{grid}(v)$=Ttexture;

Threshold Tedge defines the maximum value of $F_{grid}(v)$, which can be regarded as an intensity change caused by blocking artefacts.

The analysed image might have a film grain or fine texture, which should be preserved, even if a small blockiness exists. Threshold Ttexture defines the minimum value of $F_{grid}(v)$, which can be regarded as a visible blockiness. The value of Ttexture corresponds to the highest level of film grain, noise or texture we want to preserve in the image.

Obviously, the value of Ttexture is smaller than Tedge.

3. DEBLOCKING USING LOCAL BLOCKINESS VISIBILITY METRICS

The main goal of any local blockiness metric is to control the adaptive low-pass de-blocking filtering. We explain the horizontal deblocking in the following example, and the vertical deblocking is similar. Fig. 4 illustrates parts of two 8 x 8 blocks, where V0~V3 are four pixels at the left block boundary, and V4~V7 are four pixels at the right block boundary.

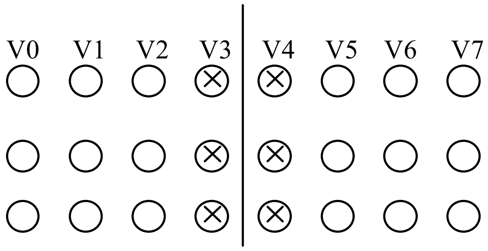

Fig. 4. Parts of two blocks of pixels divided by the block edge between V3 and V4.

The local blockiness metric defines different filtering modes, which should be applied to reduce artifacts.

The **first filtering mode** is used against highly visible luminance blockiness, characterized by a large value of Bl. The filtering is applied if

Thr1 < Bl < Tedge

The filtering in the first mode is the strongest:

$$\begin{aligned} V3' &= (V2 + V3 + V4)/3, \\ V4' &= (V3 + V4 + V5)/3, \\ V2' &= (2*V2 + V3')/3, \\ V5' &= (2*V5 + V4')/3, \end{aligned} \quad (4)$$

where V2’- V5’ are output luminance pixel values. The values of V1~V3 and V6~V8 are then adjusted to be in the range of (0, 255). We only update the pixels when the Bl is smaller than *Tedge*, because when the Bl is large it might be a real edge.

The **second mode** is applied when the visibility of a luminance discontinuity is lower than in the first mode:

Thr2 < Bl <= Thr1

The local blockiness has a lower value either due to smaller pixel discontinuity over the block edge $F_{grid}(v)$, or due to large pixel activity within blocks $F_{non-grid}(v)$. Both cases require less strong filtering than in the first mode. The second mode filter is defined as:

$$\begin{aligned} V3' &= (V2 + 2*V3 + V4)/4, \\ V4' &= (V3 + 2*V4 + V5)/4 \end{aligned} \quad (5)$$

The goal of the **third filtering mode** is to reduce residual block edges. This mode is applied if

Ttexture < Bl <= Thr2

Most deblocking algorithms only update pixels close to block boundaries, which is not sufficient in case a local blockiness is not strongly pronounced. In fact, filtering of only pixels that are adjacent to a block edge might create other artifacts in centers of the blocks. Therefore, we update all the pixels inside the blocks. The low-pass filter used in the third mode is a 3 x 3 bilateral filter. Each pixel Vx in a third mode is updated according to the following pseudo code:

```
Sum=0; n=0;
for (int i = 1; i <= 9; i ++)
 {
  if (abs(Vi – Vx) < Sigma)
   { n ++;
     sum += Vi;}
 }
Vx = sum/n;
```

where V1~V9 represent the intensities of 9 pixels in the 3x3 window, and Vx is the intensity of the central pixel within this window. The value of Sigma dependents on $F_{grid}(v)$:

$$\text{Sigma}= F_{grid}(v)+1;$$

The thresholds *Thr1, Thr2, Tedge, Ttexture* can be varied depending on resolution, bit-rate and sharpness of the decoded image. In our implementation, constant values of the thresholds are employed for simplicity.

4. RESULTS OF EXPERIMENTS

The efficiency of the proposed artifact reduction method was evaluated using test sequences with SD, SIF, and HD resolutions compressed by a MPEG-4 coder at different bit-rates. From the various state-of-the-art algorithms, we choose two methods, which provide best results and represent two different approaches to artifact reduction. The first method is the algorithm proposed in [3], which is based on a spatial analysis of luminance and chrominance. The second algorithm is an efficient but very expensive technique of A. Nostratinia [4]. PSNR results of benchmarking are shown in Table 1.

TABLE 1
PSNR OF PROCESSED TEST SEQUENCES

bit-rate Mbit/s	algorithm [3]	algorithm [4]	proposed algorithm
"Stefan", SIF			
0.1	22.88	22.75	22.89
0.25	23.08	22.98	23.10
0.5	23.60	23.54	23.61
1.0	26.46	26.49	26.51
"Vanessa", SD			
2.0	29.62	29.21	29.64
3.0	31.40	31.00	31.42
4.0	32.78	32.45	32.87
5.0	33.93	33.67	33.98
"Porsche", SD			
2.0	30.51	30.08	30.59
3.0	32.26	31.92	32.30
4.0	33.64	33.41	33.68
5.0	34.68	34.64	34.71

Besides PSNR, the Block Impairment Metric (BIM) [5] was used for evaluation, which provides an objective metrics of blockiness in vertical VBIM and horizontal HBIM directions. Table2 shows VBIM and HBIM values.

According to subjective as well as objective evaluations, the proposed here blockiness metric and the de-blocking filtering provide better visual quality than the method [4] for all bit-rates and spatial resolutions. The performance of [3] is similar to the proposed method at low bit-rates, but at high bit-rates it blurs objects edges and fine texture.

TABLE 2
BIM OF TEST SEQUENCES

bit-rate Mbit/s	method	H BIM	V BIM
"Stefan", SIF			
0.50	algorithm [3]	1.39	1.99
	algorithm [4]	1.66	2.79
	proposed	1.31	1.87
1.00	algorithm [3]	1.20	1.56
	algorithm [4]	1.35	1.92
	proposed	1.18	1.50
"Vanessa", SD			
3.0	algorithm [3]	1.45	1.75
	algorithm [4]	2.71	3.11
	proposed	1.40	1.74
4.0	algorithm [3]	1.29	1.48
	algorithm [4]	2.14	2.46
	proposed	1.18	1.46

5. CONCLUSIONS

In this paper, a new local blockiness metrics and the example of its application for coding artifact reduction have been presented. The metric does not require coding parameters and is based on the spatial analysis of luminance pixel values of a decoded picture. The application of our blockiness metrics for adaptation of de-blocking low-pass filtering is able to significantly reduce blockiness and in MPEG decoded sequences. According to the results of experiments, the proposed algorithm demonstrates good performance for low bit-rate as well as high bit-rate compressed sequences, and outperforms more complicated artifact reduction algorithms. Hardware implementation of our algorithm requires only five lines memory.

REFERENCES

[1] Masahiro Honjo, *"Method of correcting an image signal decoded in block units*", US patent US5337008, 1993.

[2] Muijs R., and Kirenko I., "*A non-reference blocking artifact measure for adaptive video processing*", in Proceedings of the 13th European Signal Processing Conference (EUSIPCO).. Turkey, 2005.

[3] Kirenko I., "R*eduction of coding artifacts using chrominance and luminance spatial analysis*", in Proceedings of the International Conference on Consumer Electronics (ICCE), Las Vegas, USA, January 2006.

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

[5] H.R. Wu and M. Yuen, "*A Generalized Block-Edge Impairment Metric for Video Coding*", IEEE Signal Processing Letters, Vol.4, Issue 11, Nov. 1997, pp.317-320.