Objective Measurements of Artifacts, Produced by Modern Video Coding Standards

Dmitriy Vatolin*
Graphics & Media Lab,
MSU

Sergey Grishin†
Graphics & Media Lab,
MSU

Alexey Moskvin ‡
Graphics & Media Lab,
MSU

Alexander Parshin**
Graphics & Media Lab,
MSU

Abstract

Algorithms for video content compression have been developed very fast during last years. But most of them are still lossy. This fact leads to different artifacts appearing in compressed video due to information losses. Objective metric for blocking artifacts detection and a number of methods for blurring estimation are proposed in this article. They can work without reference (original) video and are rather simple for hardware implementation. Such features of proposed algorithms allow to use them in many areas of video processing: from quality control for video broadcasting to high-quality adaptive deblocking algorithms during video playback.

CR Categories:

Keywords: MPEG-4, H.264, blocking effect, blurring effect, video quality control, objective metrics.

1 Introduction

The task of video quality assessments often appears during digital video processing. Quality degradation is a result of lossy compression, losses during transmission over networks or different mediums damages. Any artifacts estimation method should take into account that the final video user is a human being. This means that all video quality assessments methods should be adjusted in accordance with human visual system (HVS) [Nadenau et al, 2002]. But it is difficult to meet this condition because of HVS extreme complexity

----

*e-mail: dmitriy@graphics.cs.msu.ru
†e-mail: sgrishin@graphics.cs.msu.ru
‡e-mail: amoskvin@graphics.cs.msu.ru
**e-mail: aparshin@graphics.cs.msu.ru

The ideal solution could be to use people themselves for video quality assessments [ITU, 2002]. Unfortunately, this approach is not suitable in most cases because of time costs, economic reasons, lack of repeatability. That is why more than two decades objective metrics coordinated with HVS are being developed to estimate video quality adequately [Xiao F., 2000][Wang Z et al, 2004].

All objective video quality metrics could be separated into two groups: metrics, which use reference video, and metrics requires only processed sequences. The main advantage of the former is availability of information about original sequence, which significantly increases possibilities of artifacts analysis. But often it is not possible to transmit not only original video, but also any information about it. In such situations these methods are inapplicable.

Second group of methods are widely being used for video quality control on client side during video broadcasting. It is not necessary to transmit reference video, which simplifies assessments workframe, but limits analysis possibilities and decreases algorithms effectiveness.

One of the ways to fill up reference sequence absence is to use information about possible artifacts types. For example, the only source of artifacts in modern video compression standards such as MPEG-4 ASP or MPEG-4 AVC/H.264 is quantization of transform coefficient [Ghanbari, 2003] (in some cases not so strong artifacts can appear during color spaces conversion). Quantization can entail number of effects in encoded video, but such information is enough for more effective quality assessments.

Methods described in this article are intended to estimate different compression artifacts separately, not overall video quality. The most noticeable of compression artifacts are blocking and blurring effects. The former appear because of transform is applied separately to video blocks instead of whole picture. The latter are a consequence of information losses during the transform coefficients quantization or appear after deblocking algorithms.

Early developed blurring [Marziliano et el, 2004] and blocking metrics [Castango et el, 1996] [Castango et el, 1998] [Hwang Y. et el, 2002] work only with predefined types of images or are rather complex for hardware implementation. The proposed metrics can be used in a wide range of application since they do not use any information about original sequence. Moreover, the proposed metrics are of rather low computational complexity and can be easily implemented in hardware.

2 Blurring metric

Blurring effect is one of video compression artifacts. The main source of this artifact is transform coefficients quantization during encoding. High-frequency component of information suffers during this process in the first place. In spite of low perceptibility of HVS to high-frequency band, such artifacts are often visible to video viewers. Another source of blurring effect is deblocking algorithms. Trying to smooth colors along the block border, these algorithms can smooth some object borders because of algorithms mistakes. This leads to damaging of important, critical for HVS border information.

Different methods to estimate blurring artifact were analyzed by the authors. There were two fast algorithms, which contain only few operations, and more complex method, producing better results. These algorithms are described in this section: gradient magnitude estimation, cosine of angle between planes and Laplacian calculation.

2.1 Gradient magnitude

The first analyzed method to estimate picture smoothness was calculation of brightness change in the neighborhood of current pixel. Considering video frame as continuous function I(x,y), one can calculate function gradient:

$$\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right).$$

Magnitude of brightness change can be estimated as magnitude of gradient:

$$V = |\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}.$$

Difference derivations should be used instead of exact solution in case of discrete picture. We used central difference derivation. Formula below shows approximation of partial X derivative:

$$\frac{\partial I}{\partial x} \approx \frac{I(x-1,y) + I(x+1,y)}{2}.$$

Additionally, following formula were used to approximate gradient magnitude:

$$|\nabla I| = \left|\frac{\partial I}{\partial x}\right| + \left|\frac{\partial I}{\partial y}\right|.$$

Such approximation allows to avoid complex operation of square root calculation and doesn't decrease precision significantly. As a result only four pixels, three adding and two modulus operation are used to calculate metric value for each pixel (Fig. 1):

$$V_{blurring} = abs(A_1 - A_2) + abs(B_1 - B_2).$$

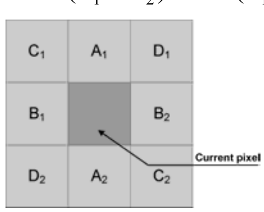


Figure 1. Pixels, which are using to calculate metric's value for current pixel.

To increase accuracy at diagonal edges, gradient magnitude calculation for others directions can be used:

$$V = |\nabla I| = \sqrt{\left(\frac{\partial I}{\partial e_1}\right)^2 + \left(\frac{\partial I}{\partial e_2}\right)^2},$$

$$e_1 = \frac{1}{\sqrt{2}}(x+y), \quad e_2 = \frac{1}{\sqrt{2}}(x-y).$$

The following formula can be used for discrete pictures:

$$V_{blurring} = abs(C_1 - C_2) + abs(D_1 - D_2).$$

We can modify described methods to work both with and without reference sequence. In the former case it is possible to calculate the same values for reference pictures and then analyze difference between them. This allows to estimate spatial areas where blurring increase after compression and deblocking algorithms. Fig. 4,C shows an example of described metric visualization using reference video.

2.2 Cosine of angles between planes

The next method to estimate smoothness in a given pixel is to use cosine of angle between perpendiculars to planes in adjacent pixels (Fig. 2). Consider a picture as continuous brightness function I(x,y). Consider plane Oxz. After projection perpendiculars in points (x-1, y) and (x+1, y) to this plane, we will get the following 2D vectors N_1 and N_2 (Fig. 3):

$$N_1 = \left(-\frac{\partial I}{\partial x}(x-1,y), 1\right)$$

$$N_2 = \left(\frac{\partial I}{\partial x}(x+1,y), 1\right).$$

Cosine of angle between N_1 and N_2 is

$$\cos(N_1 N_2) = \frac{(N_1, N_2)}{|N_1| \cdot |N_2|}.$$

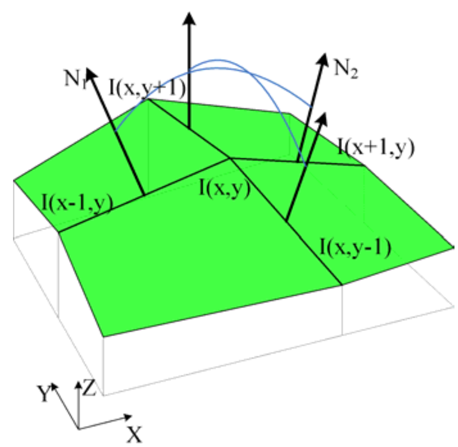


Figure 2. Angles between perpendiculars in adjacent pixels.

The following formula express cosine value using right difference derivation for point I(x-1,y) and left difference derivation for point I(x+1,y):

$$\cos(N_1 N_2) = \frac{1 - (a - b_1)\cdot(a - b_2)}{\sqrt{\left((a-b_1)^2 + 1\right)\cdot\left((a-b_2)^2 + 1\right)}},$$

where a, b_1 and b_2 are I(x,y) values in points (x,y), (x-1,y) and (x+1,y) accordingly.

The smaller the angle between vectors the smoother is the surface. Cosine of the angle is good characteristic of picture smoothness near a given point.

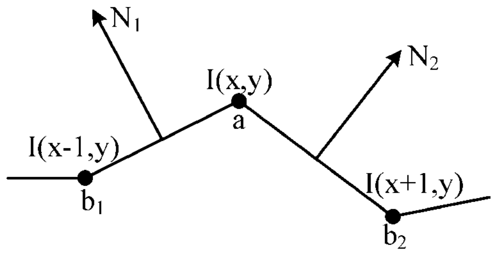


Figure 3. Projection to *Oxy* plane of perpendiculars to picture near pixel (x,y).

It is possible to simplify cosine calculation in case of small angle between perpendiculars using Taylor series:

$$\cos(N_1N_2) = 1 - 2\frac{1}{\left(p^2+1\right)^2}\delta^2 - 2\frac{2p^2-1}{\left(p^2+1\right)^4}\delta^4 + O(\delta^6),$$

where $p = \frac{b_2 - b_1}{2}$, $\delta = a - \frac{b_2 + b_1}{2}$.

As a result the number of operations for metric calculation using this method decreases significantly. An example of results for this algorithm is shown on Fig. 4,D.

2.3 Laplacian calculation

Laplacian of continuous two-dimensional function I(x,y) can be calculated using the following formula:

$$\Delta I = \nabla(\nabla I) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}.$$

Laplacian is a good integral characteristic of brightness change near a given point. But being second order characteristic, Laplacian is not so stable for noisy images. Laplacian can produce "double edges" effect, especially for smooth but large edges. Using second difference derivation

$$\frac{\partial^2 I}{\partial x^2} \approx I(x-1,y) - 2I(x,y) + I(x+1,y)$$

we can obtain that Laplacian calculation is equal to convolution with the following kernel

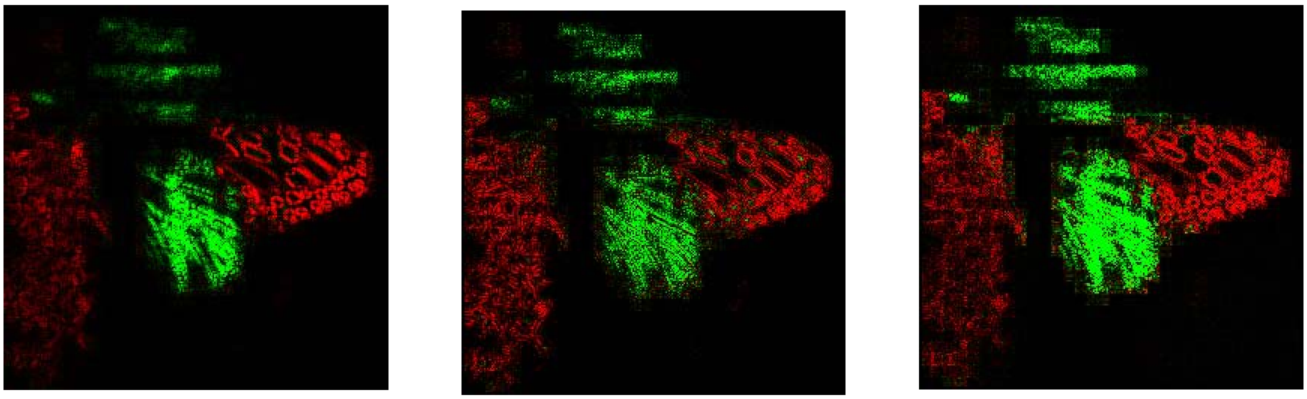
$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

3x3 neighborhood is analyzed using this method. Filter kernel has five points pattern and can be calculated very fast. At the same time this method shows good results for wide range of images. An example of results for this method is shown on Fig. 4,E.

2.4 Blurring metrics comparison

Examples of results for all described algorithms are shown on Fig. 4. Usage of gradient magnitude estimation after blurring areas with sharp edges produces pixels with higher magnitude value. This fact shows that magnitude estimation is not quite adequate for those areas. Laplacian calculation produces artifacts, but our studies show that they are not as strong as in case of gradient method The cosine calculation is the best method to estimate picture blurring, but it requires a bigger number of operations per pixel.

A B



C D E

Figure 4. Examples of blurring metrics. A. Original frame. B. Processed frame with blurring and contrast increasing n different areas. C. Gradient magnitude estimation. D. Cosine of angles calculation. E. Laplacian calculation.
Red color corresponds to smoothness increasing; green color corresponds to smoothness decreasing.

3 Blocking metric

3.1 Algorithm description

Most modern algorithms of video compression including MPEG-2, MPEG-4 ASP, H.263, MPEG-4 AVC/H.264 and some others divide each frame into blocks of predefined size. Motion compensation technique is applied to each block after transform of estimated residual. The purpose of transform is to reduce dependencies between block's pixels. Resulting coefficients are quantizing and coding using lossless compression. Information loss during quantization produces number of artifacts in compressed video such as blocking effect, blurring effect, Gibbs effect, etc.

Blocking effect appears because of separate blocks transformation. Adjacent blocks distort independently, resulting in big brightness differential at the blocks boundaries in decoded sequences. This effect becomes stronger simultaneously with increasing quantize coefficient (decreasing information after quantization).

Visibility of blocking artifact is additionally connected with features of HVS. It is well known that high-frequency artifacts (including blocking) are better visible in smooth areas than in high-detailed areas. This HVS feature was taken into account in metric's algorithm with the help of area contrast estimation.

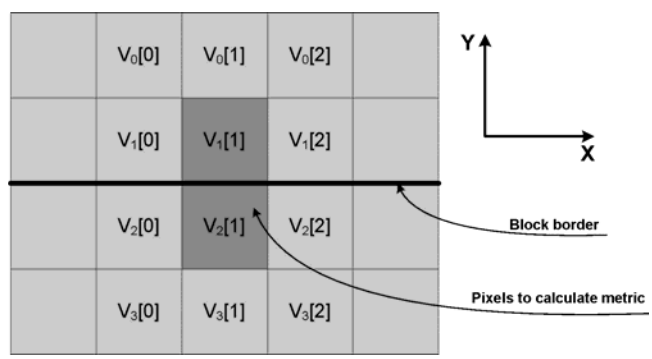


Figure 5. Picture's pixels, which are using for blocking metric calculation in points $V_1[1]$ and $V_2[1]$.

Metric is calculated for pixels at boundaries of 8x8 blocks. The metric value is the same for each two adjacent to blocks boundary pixels (dark gray pixels at Fig. 5). That value depends on two factors: magnitude of color difference at block's boundary and picture contrast near boundaries. The former is calculated using the following expressions:

$$A = V_0 - V_1$$
$$B = V_1 - V_2$$
$$C = V_2 - V_3$$
$$D = C - \frac{A+B}{2}$$
$$M = abs(D[1] + D[2] + D[3]).$$

Consider lines produced by values of two pixels from each side of block boundary. Each component of vector D is the difference between prolongations of these lines to block boundary (Fig. 6). So, geometric sense of vector D is the magnitude of color difference at block's boundary.

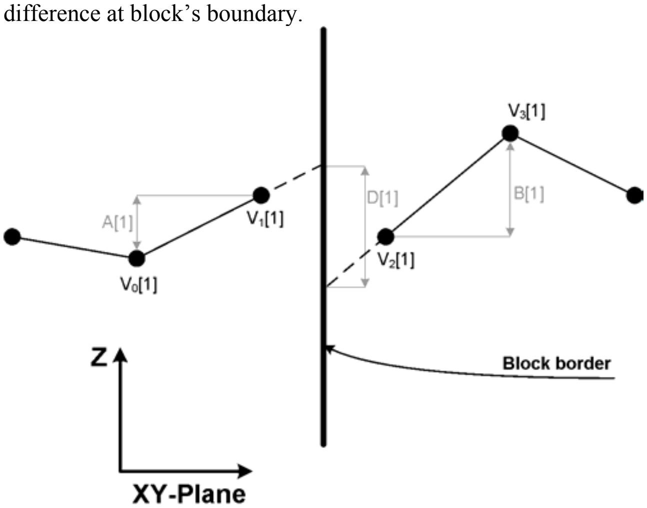


Figure 6. Geometric sense of D.

Contrast near block's boundary is calculated using the following formulas:

$$W_1 = W(abs(A[0]) + abs(B[0]))$$
$$W_2 = W(abs(A[1]) + abs(B[1]))$$
$$W_3 = W(abs(A[2]) + abs(B[2]))$$
$$W_R = (W_1 + W_3) \cdot W_2$$

The higher contrast value the lower is a contrast coefficient W_R. Such coefficient behavior achieved with the help of shape of function W(x). Important feature of this function is slow decreasing speed at low values of argument. Contrast coefficient is near one in smooth areas and doesn't influence on resulting metric's value. On the other hand, contrast coefficient is low for contrast areas, which decrease resulting metrics value.

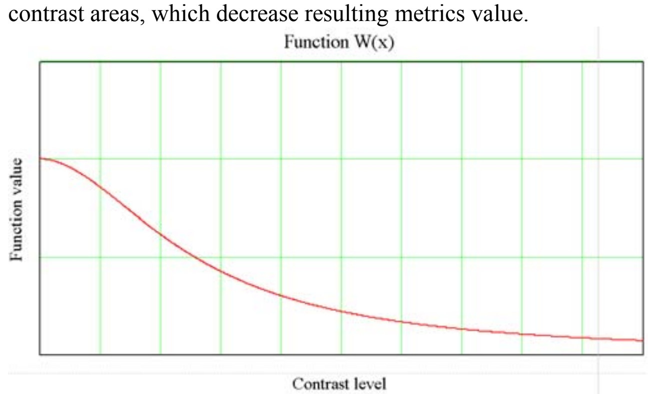


Figure 7. Shape of function W(x).

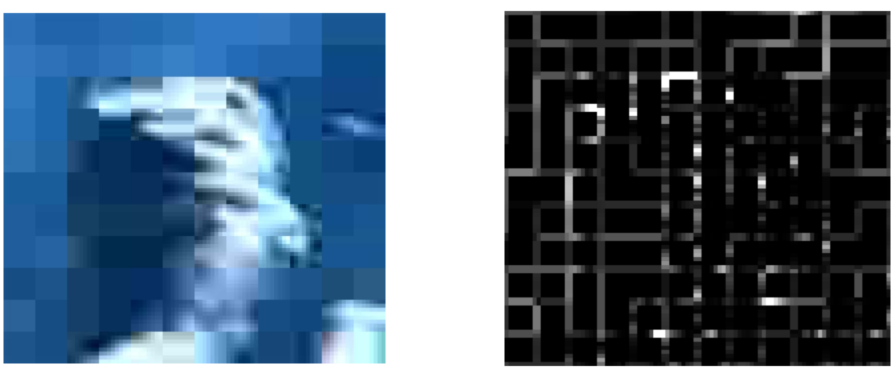
Resulting metric's value $V_{blocking}$ can be obtained by multiplying color break value M and contrast coefficient W_R:

$$V_{blocking} = M \cdot W_R$$

3.2 Results

"Battle" sequence (from film "Terminator 2") is used to show example of metric's work. That sequence is rather complex to compress because of very fast motion.

Part of decoded frame and blocking metric visualization are shown on Fig. 8. It is easy to see that metric's value is lower at contrast areas comparing with areas without any details.



A **B**

Figure 8. A. Decoded frame. B. Visualization of blocking metric (the brighter is point, the more visible is blocking).

4 Testing of proposed metrics

Proposed metrics were used for MPEG-4 AVC/H.264 [JVT, 2003] codecs testing [VATOLIN et al, 2005]. Metrics values were measured for more than 1100 video files, received after compression of 7 different sequences at 10 bitrates.

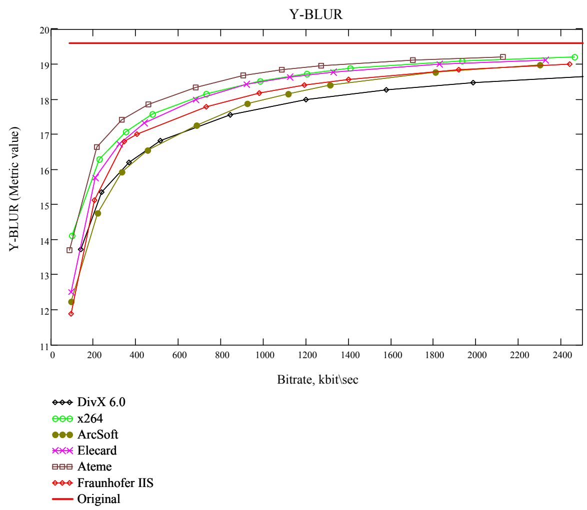


Figure. 9. Bitrate dependence of blurring metric for Y-PSNR. Red horizontal line is blurring for original sequence.

Examples of blurring and blocking metric are shown accordingly at Fig. 9 and at Fig. 10. “Foreman CIF” sequence was used to create these graphs. Bitrate dependence of metrics values is shown for different codecs. Horizontal red line is metric’s value for original sequence.

Analysis of received data shows that proposed metrics are rather adequate at different sequences and with different codes. Metrics are monotonous relative to bitrate in most cases: blurring metric values decrease and blocking values increase simultaneously with bitrate growing.

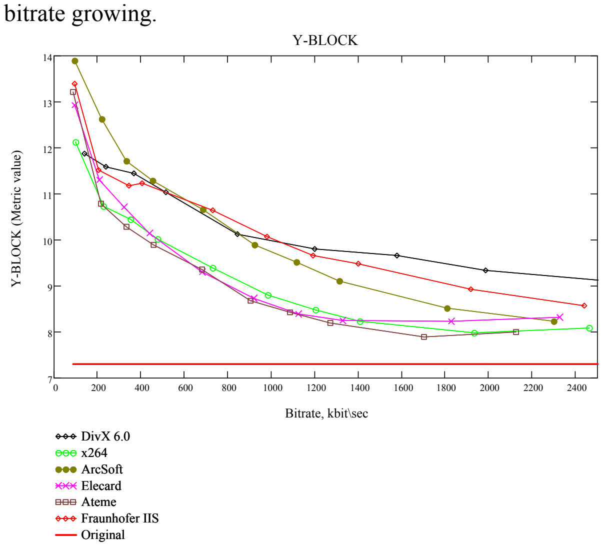


Figure 10. Bitrate dependence of blocking metric for Y-PSNR. Red horizontal line is blocking for original sequence.

3 Conclusion

A number of metrics are described in this article. These metrics estimate artifacts of modern video compression standard codecs. Three metrics of blurring estimation and blocking metric were developed and tested. All proposed metrics could work without reference video and are rather simple for hardware implementation. They could be used for real time video quality measurements during broadcasting or as a part of high-quality deblocking algorithms.

References

JOINT VIDEO TEAM OF ITU-T AND ISO/IEC JTC 1. 2003. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 ISO/IEC 14496-10 AVC). *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050.*

WIEGAND T., SULLIVAN G. J., BJONTEGAARD G., LUTHRA A. 2003. Overview of the H.264 / AVC Video Coding Standard, *IEEE CSVT.*

INTERNATIONAL TELECOMMUNICATION UNION. 2002. Methodology for the subjective assessment of the quality of television pictures. *ITU-R BT.500-11.*

GHANBARI M. 2003. Standard Codecs: Image Compression to Advanced Video Coding, *IEE PUBLISHING.*

MARZILIANO P., DUFAUX F., WINKLER S., EBRAHIMI T. 2004. Perceptual blur and ringing metrics: application to JPEG2000. *SP:IC(19), No. 2, February 2004, pp. 163-172.*

XIONG Z., ORCHARD M. T., ZHANG Y. Q. A Deblocking Algorithm for JPEG Compressed Images Using Overcomplete Wavelet Representations. *IEEE Transactions on Circuits and Systems for Video Technology, 7, 433-437.*

GOPINATH R.A., LANG M., GUO H., ODEGAM J.E. 1994. Wavelet-Based Post-processing of Low Bitrate Transform Coded Images. *In Proc. of Intl. Conf. in Image Processing, Austin, TX.*

NADENAU M. J., WINKLER S., ALLEYSSON D., KUNT M. 2002. Human Vision Models for Perceptually Optimized Image Processing. *A Review, submitted to Proc. of the IEEE.*

YANG Y., GALATSANOS N., KATSAGGELOS A.K. 1995. Projection-Based Spatially-Adaptive Reconstruction of Block-Transform Compressed Images. *IEEE Trans. on Image Processing, Vol. 4, No. 7, pp. 896-908.*

MEIER T., NGAN K.N., GREBBIN G. 1998. Reduction of Coding Artifacts at Low Bit Rates. *SPIE International Conference on Visual Communications and Image Processing, San Jose, U.S.A., pp. 241-251.*

SUN X., WU F., LI S., GAO W. 2002. In-Loop Deblocking Filter for Block-Based Video Coding. *ICSP 2002, vol. 1, 33-36, Beijing, August.*

CASTANGO R., MARSI S., RAMPONI G. 1998. A Simple Algorithm for the Reduction of Blocking Artifacts in Images and Its Implementation. *IEEE Trans. On Consumer Electrinics, vol.44, no.3.*

CASTANGO R., RAMPONI G. 1996. A Rational Filter for the Removal of Blocking Artifacts in Image Sequences Coded at Low Bitrate. *Proc. of EUSIPCO’96, 10-13 Sept., vol. I, pp. 567-570.*

HWANG Y., JEON B., SULL S. 2002. Post-Processing for Reducing Blocking Artifacts using Adaptive Low-Pass Filtering. *Proc. of ITC-CSCC2002, pp.297-300.*

GUNTURK B. K., ALTUNBASAK Y., MERSEREAU R. M. 2002. Multiframe Blocking-Artifact Reduction for Transform-Coded Video. *IEEE Transactions on Circuits and Systems for Video Technology, pp. 276-282, vol. 12, issue 4.*

VATOLIN D., KULIKOV D., PARSHIN A. 2005. Second Annual MPEG-4 AVC/H.264 codec comparison. *Website www.compression.ru.*

http://compression.ru/video/codec_comparison/mpeg-4_avc_h264_2005.html.

XIAO F. 2000. DCT-based Video Quality Evaluation. *Final Project for EE392J*.

WANG Z., BOVIK A. C., SHEIKH H. R., SIMONCELLI E. P. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing, vol. 13, no. 4*.