IMPROVING THE PREDICTION ACCURACY OF PSNR BY SIMPLE TEMPORAL POOLING

Christian Keimel and Klaus Diepold

Technische Universität München, Institute for Data Processing, Arcisstr. 21, 80333, Munich, Germany
christian.keimel@tum.de, kldi@tum.de

ABSTRACT

PSNR is still one of the most often and universally used visual quality metrics. Although it is not very well suited to describe the human perception of visual quality, its simplicity and familiarity lead to its extensive use in many applications. We propose to improve the predication accuracy of PSNR by simple temporal pooling and thus not only using the mean PSNR, but also to exploit other statistical properties. In order to support this approach, we conducted extensive subjective testing of HDTV video sequences at typical bit rates for consumer and broadcasting applications. Using temporal pooling, we were able to achieve an improvement of nearly 10 % in the predication accuracy of PSNR for visual quality while not increasing the computational complexity significantly. Also this approach may be extendible to other frame-based metrics.

Index Terms— Visual quality, video quality metric, PSNR, temporal pooling

1. INTRODUCTION

Many visual quality metrics for image and video processing have been proposed so far. Nevertheless, the peak signal to noise ratio, PSNR, is still the most widely used visual quality metric even though its shortcomings with respect to human perception of visual quality are well known. Particularly outside the specific research area of visual quality metrics alternative metrics are neither used nor apparently even known.

The popularity of PSNR is not only due to the familiarity of researchers and developers with PSNR, but also because of its simplicity and therefore easy implementation compared to more sophisticated metrics. Moreover its computational complexity is rather low. Hence it is used in a vast area of applications: from the conception and development of new video coding standards to the every day use in consumer products e.g. video cameras or other devices that encode video, but also in the broadcasting industry to monitor signal quality or to support business decisions on offered service quality and equipment acquisition. Although PSNR is clearly a full reference metric, there has been extensive and quite promising research into extending PSNR into a no reference metric by estimating the PSNR from bit stream features of encoded videos: [1–3] use DCT coeffients to estimate the PSNR, Shim et.al. [4] use integer transform coefficents and Ichigaya et.al. use DCT coeffients and picture energy in [5]. A known limitation of these no-referene PSNR estimation techniques, however, is that they usually only work for certain encoders e.g. MPEG-2 [2, 5] or AVC/H.264 [3, 4] due to their dependance on bitstream features and their statistical distribution. Still, they are able to predict PSNR in a known no-reference enviroment very well. Thus PSNR can also be used in a known no-reference environment like video streaming for IPTV or signal distribution in broadcasting networks.

Usually the individual PSNR of each frame is averaged over a complete video sequence, producing one PSNR value representing the visual quality of the whole sequence. But averaging is insufficient to describe the statistical distribution, especially if the individual values are not normally distributed. In this contribution we propose therefore to improve the prediction accuracy of PSNR significantly by using different, but simple temporal pooling functions and demonstrate the effectiveness of temporal pooling on a set of HDTV sequences representing different consumer and broadcasting applications. Our results are supported by carefully done subjective testing.

This contribution is organized as following: First we will discuss temporal pooling in general before introducing the different encoder scenarios. Then we will describe the conducted subjective tests before applying temporal pooling to PSNR. Finally we present the results and discuss them in the conclusion.

2. TEMPORAL POOLING

Visual quality metrics determine the visual quality by combining and evaluating one or more features that are extracted from distorted videos, no matter if those features were chosen to model the human visual system (HVS), or to represent some distortion in the video. This feature extraction can either be full reference, reduced reference or no reference. Most video quality metrics follow this process of combining a set of features into one numerical value representing the overall visual quality of a video sequence. The features that are extracted may be based on knowledge of the HVS, or based on typical distortions, or based on properties of a video, such as the amount of details in the images.

Extracting a set of features from a video can be seen as a different representation of this video. The content of the video is not represented using the single values of each pixel, or a combination of motion information and residual error, but the representation is done using features that do have a relationship to visual quality. As video does have a temporal dimension, features that were extracted for a certain time instance e.g. one frame must be combined into one quality value by temporal pooling. Most metrics do this by calculating the mean value over time [6–10]. One of the few metrics that use different pooling functions for the single features is the VQM as described in Annex D of ITU-T J.144 [11]. In related works to temporal pooling, Minkowski summation [12] or exponentially weighted Minkowski summation [13] are used. Both exhibit a high computational complexity due to their use of exponential functions. This limits their applicaiton in consumer applications. In a recent contribution, Rimac-Drlje et.al. compared temporal pooling methods for CIF-size videos and for different visual quality metrics including PSNR to subjective test results [14]. Simple statistical properties as suggested in this contribution, however, were only exploited for a subset of frames. Also details about the conducted subjective tests used to verify the results are missing. In particular the outlier ratio and confidence intervals are not provided.

This pooling step results in a number of parameters which are derived from the extracted features. Calculating only the mean value for e.g. PSNR is not sufficient to describe the statistical distribution of this feature in time and space. As a result, the description of these videos using only the mean values is too coarse and hence more parameters are needed. We need to find those parameters that co-vary with the visual quality.

3. TEMPORAL POOLING OF PSNR

Using PSNR as a quality indicator requires to calculate a PSNR or mean squared error (MSE) value on a frame by frame basis. The resulting values are then averaged to determine a single value for the entire video sequence. A rudimentary adaptation to the HVS is made by evaluating the PSNR in the YUV color space. For simplicity, mean PSNR is mostly calculated only in the Luma channel (PSNR^Y_{Mean}), also known as Luma PSNR. Calculating PSNR^Y_{Mean}, however, does not lead to a useful representation of the videos with respect to quality evaluation. The use of other parameters derived from PSNR results in a even more accurate visual quality evaluation.

To illustrate this, we consider the following two different sequences: although the PSNR^Y_{Mean} values for both sequences are extremely similar (31.39 dB vs. 31.53 dB), the visual quality measured in our subjective tests is very different (0.95 vs. 0.69 on a scale between 0 and 1). In particular, the sequence with the slightly higher value for PSNR^Y_{Mean} has the overall lower visual quality.

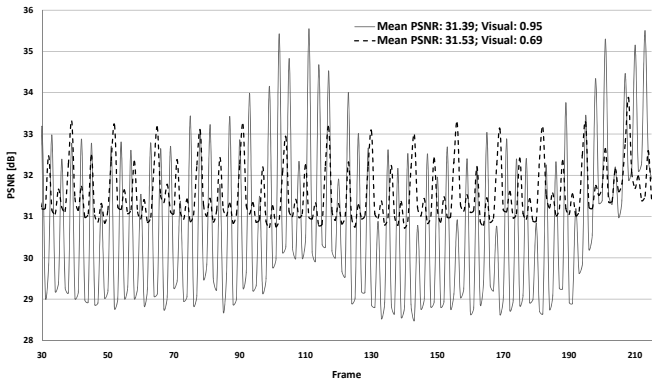

Fig. 1: PSNR over time for two sequences

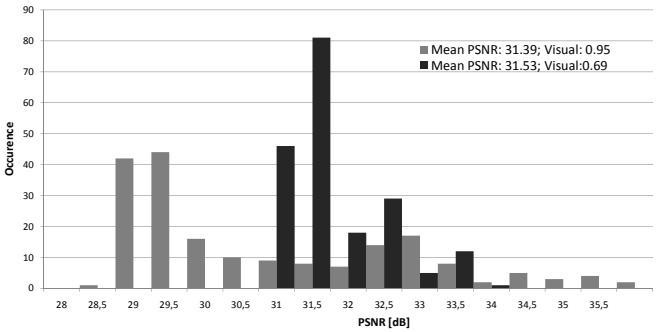

Fig. 2: Histogram of PSNR values for two sequences

If we examine the PSNR^Y over time in Fig. 1 and also the histogram of the PSNR^Y values for each sequence in Fig. 2, we can easily see that the PSNR^Y distribution of both squences differ significantly and especially shows that the PSNR^Y values over all frames are not normally distributed. Therefore we propose to evaluate not only the mean PSNR, but also the minimum value, the maximum value, the standard deviation, the 90% percentile and the 10% percentile, denoted as PSNR^Y_{Min}, PSNR^Y_{Max}, PSNR^Y_{sDev}, PSNR^Y_{90}, and PSNR^Y_{10}, respectively.

As the Variations of PSNR^Y differ significantly over time for the two sequences, we also calculate

$$\text{dPSNR}^Y_i = \left|\text{PSNR}^Y_i - \text{PSNR}^Y_{i-1}\right|, \qquad (1)$$

representing the difference between the PSNR^Y of two consecutive frames. Once again we calculate the mean value of dPSNR^Y and also calculate the mean, maximum, minimum, standard deviation and percentiles. Hence PSNR^Y is not only represented by PSNR^Y_{Mean}, but by a set of 12 different values derived from the individual frames' PSNR^Y.

As shown in Table 1 exemplary for the luma channel, only a few of the values for both PSNR^Y and dPSNR^Y are close to each other for the two sequences. A similar process is applied to PSNR^U and PSNR^V, representing the PNSR values for the

Table 1: Different PSNR^Y based parameters for two sequences

	Sequence 1	Sequence 2
Visual Quality [MOS]	0.69	0.95
PSNR^Y_{Mean} [dB]	31.53	31.39
PSNR^Y_{Min} [dB]	30.70	28.49
PSNR^Y_{Max} [dB]	33.87	36.43
PSNR^Y_{10} [dB]	30.88	28.86
PSNR^Y_{90} [dB]	32.52	34.84
PSNR^Y_{sDev} [dB]	0.70	2.49
dPSNR^Y_{Mean} [dB]	0.69	2.29
dPSNR^Y_{Min} [dB]	0.00	0.01
dPSNR^Y_{Max} [dB]	2.20	5.44
dPSNR^Y_{10} [dB]	0.10	0.14
dPSNR^Y_{90} [dB]	1.35	4.53
dPSNR^Y_{sDev} [dB]	0.55	1.82

two chroma channels. Such a brute-force approach results in a total number of 36 PSNR different parameters.

Keep in mind that no significant additional computational complexity is introduced by this temporal pooling step compared to the commonly used PSNR_{Mean}. There the computational most expensive step is the separate calculation of PSNR for each frame in the video sequence. The only cost increase compared to the traditional PSNR_{Mean} might occur in memory usage as the PSNR of each frame must be stored until the overall statistical values are computed and a slight increase in computational load as not only the mean but also other values must be computed. This, however, only occurs once per sequence and not for every frame in the sequence separately.

4. SUBJECTIVE TESTING

In order to validate our simple temporal pooling approach for PSNR, we conducted extensive subjective tests. These tests were performed in the video quality evaluation laboratory of the Institute for Data Processing at the Technische Universität München. We employed 17 naïve viewers (all students with no or very little experience in video coding) and one expert viewer, all of them screened for visual acuity and color blindness. The tests itself were performed in a test room compliant with recommendation ITU-R BT.500 [15], using a professional LCD display with 1080 lines (Cine-tal Cinemage display). The decoded videos were converted to 4:2:2 YUV by bilinear upsampling of the chrominance channels of the 4:2:0 decoder output. A HD-SDI link was used to connect the video server to the display. To maintain the unique viewing experience that can be achieved with HD video, the distance between the screen and the observers was only three times the picture height. To allow stable viewing conditions for all participants, only two viewers took part in the test at the same time.

The tests were carried out using a variation of the standard DSCQS test method as proposed in [16] . This Double Stimulus Unknown Reference (DSUR) test method differs from the standard DSCQS test method, as it splits a single basic test cell in two parts: the first repetition of the reference and the processed video is thought to allow the test subjects to decide which is the reference video. Only the second repetition is used by the viewers to judge the quality of the processed video in comparison to the reference. To allow the test subjects to differentiate between relatively small quality differences, a discrete voting scale with eleven grades ranging from 0 to 10 was used (later rescaled to 0 to 1). In order to verify if the test subjects were able to produce stable results, a small number of test cases were repeated during the test. Processing of outlier votes was done according to Annex 2 of [15], and the votes of one test subject were removed based on this procedure. To gain one visual quality value for each test case all valid votes were simply averaged. The 95% confidence intervals of the subjective votes are below 0.07 on a scale between 0 and 1 for all single test cases, the mean 95% confidence interval is 0.04.

5. ENCODER SCENARIOS

We selected four different bit rates from 5.4 Mbit/s to 30 Mbit/s to represent different real life HDTV consumer and broadcasting applications from IPTV at the lower end to Blu-ray on the upper end on the bit rate scale.

The test sequences were chosen from the SVT high definition multi format test set [17] with a spatial resolution of 1920×1080 pixel and a frame rate of 25 frames per second (fps) was used. The particular sequences are 'CrowdRun'(CR), 'ParkJoy'(PJ), 'IntoTree'(IT) and 'OldTownCross'(OTC). Each of those videos was encoded at the selected four different bit rates. This results in a quality range from 'not acceptable' to 'perfect', corresponding to mean opinion scores (MOS) between 0.19 and 0.96 on a scale ranging from 0 to 1. The artifacts introduced into the videos by this encoding include pumping effects i.e. periodically changing quality, a typical result of rate control problems, obviously visible blocking, blurring or ringing artifacts, flicker, banding i.e. unwanted visible changes in color and similar effects. An overview of the sequences and bit rates is given in Table 3.

The sequences were encoded using the AVC/H.264 reference software [18] version 12.4. Two significantly different encoder settings were applied to represent the different complexity of different application areas. The first setting is chosen to simulate a low complexity (LC) AVC/H.264 encoder representative of consumer devices using a 'Main' profile according to Annex A of the AVC/H.264 Standard: many

Table 2: Correlation values for PSNR based parameters to results of subjective tests

PSNR Parameter	Pearson Correlation
PSNR^{Y}_{Mean}	0.688
PSNR^{Y}_{Min}	0.753
PSNR^{Y}_{10}	0.720
PSNR^{U}_{Mean}	0.588
PSNR^{U}_{Min}	0.606
PSNR^{V}_{Mean}	0.603
PSNR^{V}_{Min}	0.635

tools that account for the high compression efficiency are disabled. In contrast to this a high complexity (HC) setting aims at getting the maximum possible quality out of this coding technology, using a 'High' profile representing sophisticated broadcasting grade encoders. In addition to AVC/H.264, we used the 'Dirac' encoder [19] in order to investigate if temporal pooling is sensible for different coding technologies. The development of 'Dirac' was initiated by the British Broadcasting Cooperation (BBC) and it is a wavelet based video codec, originally targeting at HD resolution video material. For 'Dirac', the standard settings for the selected resolution and frame rate were used. Only the bit rate was varied to encode the videos. The used software version for Dirac is 0.7. Selected encoding settings for AVC/H.264 are given in Table 4.

6. RESULTS

Using the results of the conducted subjective tests for the different encoder scenarios, we now examine how well these alternatives to PSNR_{Mean} describe the visual quality of video sequences. To evaluate the prediction accuracy, we calculate the Pearson correlation coefficient between the different pooling parameters and the experimentally determined visual quality from our tests described in section 4 for a total of 48 data points.

As we can see in Table 2, we are able to achieve an improvement of nearly 10 % for PSNR^{Y} , and still between 3 % to 5 % for PSNR^{U} and PSNR^{V}, respectively.

This may not seem like a huge improvement and indeed does not even come close to corresponding correlations of more sophisticated metrics like SSIM [10] that usally achieve a correlation to visual quality between 0.8 to 0.9 and higher. We should, however, keep in mind that we got this improvement basically for free, without a noticeable increase of computational complexity, compared to the usual calculation of PSNR^{Y}_{Mean}.

7. CONCLUSION

We have seen that mean PSNR alone does not sufficiently describe the visual quality of video sequences. The video sequences were encoded to real life bit rates to represent consumer and broadcasting applications by using different encoders or different encoder settings. Even two sequences with extremely similar mean PSNR values can exhibit vastly different visual qualities.

If we, however, exploit already existing per-frame knowledge of the video sequences by more extensive temporal pooling, the prediction accuracy of PSNR can be increased by nearly 10 %. Although our current results are only based on a limited test set of 48 data points, we can see that alternative temporal pooling methods can describe the statistical properties of the video sequence better than the traditional approach of only averaging the PSNR over all frames. Furthermore the added computational complexity compared to averaging is negligible.

While we have only considered PSNR in this contribution it may very well be that other frame-based visual quality metrics (including, but not limited to [6–10]) when applied to video sequences might also benefit from this simple temporal pooling approach.

8. REFERENCES

[1] T. Brandão and M. P. Queluz, "Blind PSNR estimation of video sequences using quantized DCT coefficient data," in *Proc. Picture Coding Symposium*, Nov. 2007.

[2] A. Ichigaya, M. Kurozumi, N. Hara, Y. Nishida, and E. Nakasu, "A method of estimating coding PSNR using quantized DCT coefficients," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 2, pp. 251–259, Feb. 2006.

[3] A. Eden, "No-reference estimation of the coding PSNR for H.264-coded sequences," *IEEE Trans. Consum. Electron.*, vol. 53, no. 2, pp. 667–674, May 2007.

[4] S.-Y. Shim, J.-H. Moon, and J.-K. Han, "PSNR estimation scheme using coefficient distribution of frequency domain in H.264 decoder," *IET Electronics Letters*, vol. 44, no. 2, pp. 108–109, 17 2008.

[5] A. Ichigaya, Y. Nishida, and E. Nakasu, "Nonreference method for estimating PSNR of MPEG-2 coded video by using DCT coefficients and picture energy," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 6, pp. 817–826, June 2008.

[6] M. C. Farias and S. K. Mitra, "No-reference video quality metric based on artifact measurements," in *Proc. IEEE International Conference on Image Processing*, vol. 3, Sep. 2005, pp. 141–144.

[7] H.R.Wu and K.R.Rao, Eds., *Digital video image quality and perceptual coding*. CRC, 2006, ch. No-Reference Quality Metric for Degraded and Enhanced Video, pp. 305–324.

[8] I. P. Gunawan and M. Ghanbari, "An efficient reduced-reference video quality metric," in *Proc. Picture Coding Symposium*, Nov. 2007.

[9] C. Lee, S. Cho, J. Choe, T. Jeong, W. Ahn, and E. Lee, "Objective video quality assessment," *SPIE Optical Engineering*, vol. 45, p. 7004, Jan. 2006.

[10] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 121–132, Feb. 2004.

[11] *ITU-T J.144. Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference*, ITU-T Std., Mar. 2004.

[12] A. S. Rmi Barland, *Advanced Concepts for Intelligent Vision Systems*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2005, vol. 3708, ch. A New Reference Free Approach for the Quality Assessment of MPEG Coded Videos, pp. 364–371.

[13] A. M. Rohaly, J. Lu, N. R. Franzen, and M. K. Ravel, "Comparison of temporal pooling methods for estimating the quality of complex video sequences," B. E. Rogowitz and T. N. Pappas, Eds., vol. 3644, no. 1. SPIE, 1999, pp. 218–225.

[14] S. Rimac-Drlje, M. Vranjes, and D. Zagar, "Influence of temporal pooling method on the objective video quality evaluation," in *Broadband Multimedia Systems and Broadcasting, 2009. BMSB '09. IEEE International Symposium on*, May 2009, pp. 1–5.

[15] *ITU-R BT.500 Methodology for the Subjective Assessment of the Quality for Television Pictures*, ITU-R Std., Rev. 11, Jun. 2002.

[16] V. Baroncini, "New tendencies in subjective video quality evaluation," *IEICE Transaction Fundamentals*, vol. E89-A, no. 11, pp. 2933–2937, Nov. 2006.

[17] SVT. (2006, Feb.) The SVT high definition multi format test set. [Online]. Available: http://www.ldv.ei.tum.de/lehrstuhl/team/Members/tobias/sequences

[18] K. Sühring. (2007) H.264/AVC software coordination. [Online]. Available: http://iphome.hhi.de/suehring/tml/index.htm

[19] C. Bowley. Dirac video codec developers' website. [Online]. Available: http://dirac.sourceforge.net

Table 3: Tested video sequences

Sequence	Frame Rate	Bit Rate [MBit/s]
CrowdRun	25 fps	8.4 / 12.7 / 19.2 / 28.5
IntoTree	25 fps	5.7 / 10.4 / 13.1 / 17.1
OldtownCross	25 fps	5.4 / 9.6 / 13.7 / 19.0
ParkJoy	25 fps	9.0 / 12.6 / 20.1 / 30.9

Table 4: Selected encoder settings for AVC/H.264

	LC	HC
Encoder	JM 12.4	
Profile	Main	High
Reference Frames	2	5
R/D Optimization	Fast Mode	On
Search Range	32	128
B-Frames	2	5
Hierarchical Encoding	On	On
Temporal Levels	2	4
Intra Period	1 second	
Deblocking	On	On
8x8 Transform	Off	On