INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG4
MPEG VIDEO CODING

ISO/IEC JTC1/SC29/WG4 MPEG/M54896
June 2020, Online

Source	**Poznań University of Technology (PUT), Poznań, Poland**
Status	**Input**
Title	**Even faster implementation of IV-PSNR software**
Authors	Jakub Stankowski, Adrian Dziembowski

Abstract

This document presents faster version of IV-PSNR software.

1 Introduction

In IV-PSNR the quality is calculated as pixel to block comparison. With 5×5 block and two-directional comparison (A vs. B, B vs. A), it requires 50 comparisons instead of 1 for typical PSNR.

The goal of creating IV-PSNR v2.1 was to further decrease computational time (when compared to IV-PSNR v2.0) without changing the output and introduce a new set of features.

Source code for IV-PSNR v2.1 is available on MPEG Git repository (v2.1 tag).

2 IV-PSNR v2.1 software features

The IV-PSNR v2.1 software was developed as improved version of IV-PSNR v2.0.

2.1 Parallel processing

Calculation of IV-PSNR metric have been parallelized by using multiple computing threads. The parallel version uses OpenMP API. The number of threads could be set on runtime by commandline parameter or determined automatically (by detection of the number of CPU logical cores).

2.2 New functionalities

Calculation of PSNR and WS-PSNR have beed introduced. Since, computational complexity of PSNR and WS-PSNR metrics is neglible, we decided to include them in IV-PSNR v2.1 software. It allows to calculate a set of all metrics at a single pass, resulting in reduction of experiment complexity and filesystem burden.

2.2.1 Implementation details – PSNR

The distortion for entire component plane is accumulated in integer numbers domain. Therefore, all floating point inaccurancy errors are eliminated during SSD (sum of squared differences) calculation step.

The quality values for each frame are buffered and accumulated using Kahanand-Babuska-Neumaier summation algorithm.

2.2.2 Implementation details – WS-PSNR

Similar to IV-PSNR calculation, the distortion values for each picture row (multiplied by WS weight) are buffered and accumulated using Kahanand-Babuska-Neumaier summation algorithm in order to improve accurancy.

The quality values for each frame are buffered and accumulated using Kahanand-Babuska-Neumaier summation algorithm.

When compared to [N18069], the choma pixel vertical position (and corresponding value of WS weight) have been corrected. However, the difference in results is negligible.

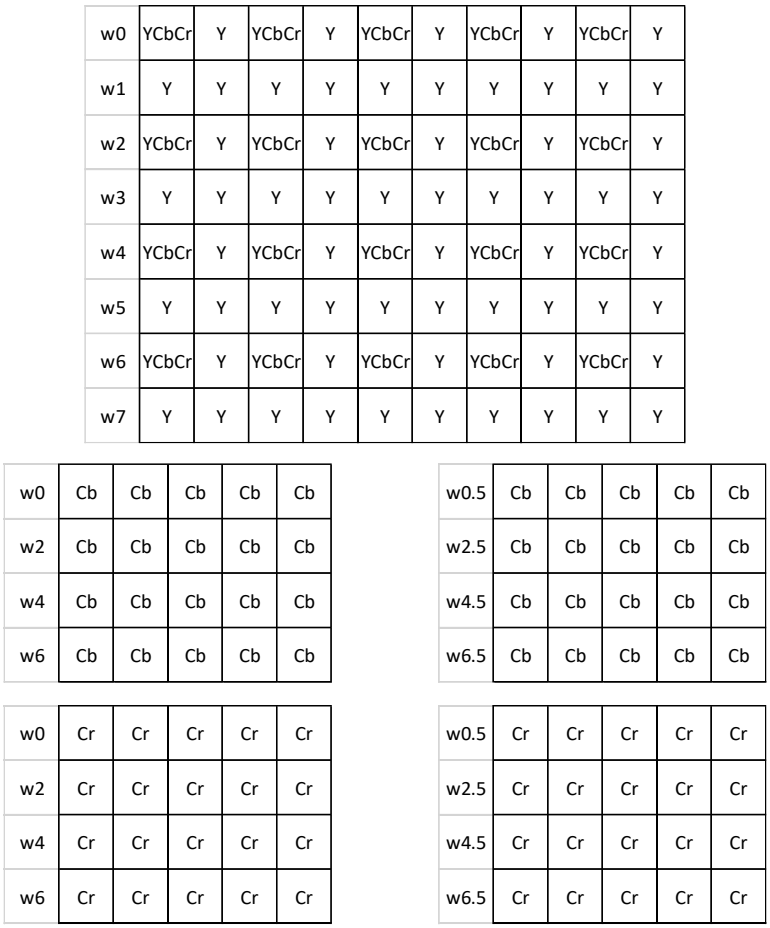

Fig. 1. WS weights for 4:2:0 video: WS-PSNR software (left column) and IV-PSNR v2.1 (right).

In order to emulate behavior of WS-PSNR software (signal peak value set to 1020), the `WSPSNR_COMPATIBILITY` compile-time flag has been implemented.

2.3 Fixed issues

Fixed incorrect WS-weight calculations for ERP sequences with non-180 lateral range (additional commandline parameters `-lor` and `-lar` added).

2.4 Commandline arguments formatting

In order to improve comandline arguments clarity and robustness, the new position-independent arguments format have been introduced. The previous (IV-PSNR v2.0 and older) positional comandline arguments format have been preserved for compatibility reasons, howewer it is expected to be removed in future.

2.5 YUV file reader changes

Detection of corrupted YUV files have been implemented. In case of corrupted file the error will be reported and application will exit returning `EXIT_FAILURE`.

2.6 Optimization

All data processing routines have been rewritten to made auto-vectorization possible. Threfore, it is highly encouraged to enable vector extesions at compile time (using `-mtune` and `-march` switch in GCC/LLVM or using `/arch` switch in MSVC)

3 Application parameters

3.1 Commandline parameters

Position independent commandline parameters reference:

General parameters

Cmd	ParamName	Description
-i0	InputFile0	YUV file path – reference
-i1	InputFile1	YUV file path – tested
-w	PicturWidth	Width of sequence
-h	PicturHeight	Height of sequence
-bd	BitDepth	Bit depth (optional, default: 8, up to 14)
-cf	ChromaFormat	Chroma format (optional, default: 420) [420, 444]
-s0	StartFrame0	Start frame (optional, default: 0)
-s1	StartFrame1	Start frame (optional, default: 0)
-l	NumberOfFrames	Number of frames to be processed (optional, default: -1 = all)

Equirectangular parameters

Cmd	ParamName	Description
-erp	Equirectangular	Equirectangular sequence (flag, default disabled)
-lor	LonRangeDeg	Longitudinal range of ERP sequence [°] (optional, default: 360)
-lar	LatRangeDeg	Lateral range of ERP sequence [°] (optional, default: 180)

Application parameters

Cmd	ParamName	Description
-t	NumberOfThreads	Number of worker threads if compiled with OpenMP (optional, default: -1 = all, suggested 4-8)
-v	VerboseLevel	Verbose level (optional, default: 2)

VerboseLevel description

Value	Printed data
0	final PSNR, WS-PSNR, IV-PSNR values only
1	0 + configuration + detected number of frames
2	1 + argc/argv + frame level PSNR, WS-PSNR, IV-PSNR
3	2 + computing time (LOAD, PSNR, WS-PSNR, IV-PSNR) (uses high resolution clock, could slightly slow down computations)
4	3 + IV-PSNR specific debug data (GlobalColorShift, R2T+T2R)

3.2 Compile-time parameters

The IV-PSNR v2.1 include a set of compile time parameters. Those parameters are defined in `CommonDef.h` file:

- **`USE_KBNS`** – Enables the usage of Kahanand-Babuska-Neumaier summation algorithm. (default = enabled)
- **`USE_FIXED_WEIGHTS`** – Enables faster 5×5 block search with fixed components weight (equal to 4:1:1). In case of different components weight are to apllied, this switch has to be disabled. (default = enabled)
- **`WSPSNR_COMPATIBILITY`** – Changes signal peak value from 1023 to 1020 (default = disabled)

4 Compilation requirements

The IV-PSNR v2.1 software uses following external components:

- "Formatting library for C++ " (libfmt) – distributed under BSD licence and included in IV-PSNR source package
- OpenMP API (optional) – expected to be a part of build envirioment

In order to build the software, the ISO C++17 conformant compiler is required.

5 Results

Both versions of IV-PSNR were compared on TMIV6.0 anchor results. Full results are attached in the .xlsx file.

5.1 Measured quality differences

5.1.1 IV-PSNR values for SC

	IV-PSNR difference [dB]
Minimum	-0.8242
Maximum	-0.5039
Average	-0.6680

All the differences are an effect of fixed incorrect WS-weight calculation for ERP sequences (section 2.3).

5.1.2 Chroma WS-PSNR values for omnidirectional content

	WS-PSNR$_U$ difference [dB]			
Sequence	SA	SB	SC	SN
Minimum	0.0012	-0.0044	-0.0001	-0.0005
Maximum	0.0018	0.0033	0.0001	0.0110
Average	0.0016	-0.0007	0.0001	0.0009

	WS-PSNR$_V$ difference [dB]			
Sequence	SA	SB	SC	SN
Minimum	0.0010	-0.0032	0.0000	-0.0002
Maximum	0.0014	0.0045	0.0002	0.0068
Average	0.0012	0.0003	0.0001	0.0008

All the differences are an effect of fixed choma pixel vertical position (section 2.2.2).

5.1.3 WS-PSNR values for all sequences with disabled `WSPSNR_COMPATIBILITY`

	$WS\text{-}PSNR_Y$ difference [dB]
Minimum	0.0255
Maximum	0.0256
Average	0.02551

All the values generated by IV-PSNR v2.1 with disabled `WSPSNR_COMPATIBILITY` (section 3.2) are higher than values generated by WS-PSNR software [N18069] by a constant offset o:

$$o = 10\log_{10} 1023^2 - 10\log_{10} 1020^2 \cong 0.02550924 .$$

When `WSPSNR_COMPATIBILITY` is enabled, all the $WS\text{-}PSNR_Y$ values are **exactly the same** as ones generated using WS-PSNR software. However, regarding the fact, that values from range [1021, 1023] are valid and can be found in content being used (e.g. some reflections in SJ), we recommend to keep this switch disabled.

5.2 Processing time comparison

Calculated using 6-core CPU.

Sequence	Processing time [s]			Time reduction			Speedup		
	IV-PSNR v1.0	IV-PSNR v2.0	IV-PSNR v2.1	v1.0 v2.0	v2.0 v2.1	v1.0 v2.1	v1.0 v2.0	v2.0 v2.1	v1.0 v2.1
SA	257.36	84.64	17.05	67%	80%	93%	3.04	4.96	15.10
SB	127.48	42.01	8.63	67%	79%	93%	3.03	4.87	14.78
SC	235.01	83.77	17.02	64%	80%	93%	2.81	4.92	13.80
SD	66.88	22.30	4.62	67%	79%	93%	3.00	4.83	14.49
SE	59.95	17.69	3.83	70%	78%	94%	3.39	4.61	15.64
SJ	56.67	17.71	3.86	69%	78%	93%	3.20	4.59	14.69
SL	56.53	17.73	3.87	69%	78%	93%	3.19	4.59	14.62
Total	**859.88**	**285.84**	**58.87**	**68%**	**79%**	**93%**	**3.09**	**4.86**	**14.61**

6 Acknowledgement

This work was supported by the Ministry of Science and Higher Education.

7 Recommendations

We recommend:

- to use IV-PSNR v2.1 instead of v2.0,
- to change `Peak_value_of_10bits` parameter for WS-PSNR software from 1020 to 1023,
- to calculate WS-PSNR values using IV-PSNR software.

8 References

[N18069] "WS-PSNR Software Manual"
ISO/IEC JTC1/SC29/WG11 MPEG/N18069, October 2018, Macao, China.

[N19495] "Software manual of IV-PSNR for Immersive Video"
ISO/IEC JTC1/SC29/WG11 MPEG/N19495, July 2020, Online.