

**INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 MPEG2020/M54279
June 2020, Online**

Source Poznań University of Technology (PUT), Poznań, Poland
Status Input
Title [MPEG-I Visual] Fast implementation of IV-PSNR software
Author 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. With additional operations like GCD (Global Color Difference) calculating or WS- weighting it is even 2 times slower than VMAF.

The goal of creating IV-PSNR v2.0 was to decrease computational time without changing the output. However, during code analysis we have found some flaws of IV-PSNR v1.0 [N18709], e.g. floating point error accumulation. Therefore, results outputted by IV-PSNR v2.0 are similar to results from v1.0 but not exact.

IV-PSNR v2.0 contains also two functionalities available earlier in “withOffsets” branch:

- support for different frame offsets for both yuv files,
- optional per-frame quality printing.

Source code for IV-PSNR v2.0 is available on MPEG Git repository (v2.0 tag).

2 IV-PSNR v2.0 software features

The IV-PSNR v2.0 software was redesigned and written from scratch.

2.1 IV PSNR metric calculation changes

- Removal of redundant GCD calculations.
- Usage of `uint16_t` data type and 444 chroma format for internal picture storage. Allows for bit depths up to 16 bit/pixel and preserves consistent result for any input chroma format.
- New implementation of pixel level processing steps (5×5 block search, distortion calculation, etc) with the use of integer only processing. The implementation some algorithmic optimizations, memory bandwidth reduction and prefetch-friendly memory access patterns. Abovementioned features result in reduction of computational complexity and avoidance of floating point related inaccuracies.
- The distortion for picture row is also accumulated as integer value.

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

2.2 YUV file reader changes

- Reduction of filesystem burden by read coalescing.
- Detection of read errors – causes application to exit returning `EXIT_FAILURE`.
- Improved conversion of 8bit/pixel input sequences.
- Improved interpolation for input sequences with non-444 chroma format (only 420 implemented).

2.3 Fixed issues

- Fixed possibility of reading from unallocated memory region during 5×5 block search.
- Fixed GCD values rounding and clipping.

2.4 Possible improvements

The IV-PSNR v2.0 was designed to allow prospective improvements i.e. parallelisation by using vector instruction and multiple processing threads.

2.5 Commandline parameters

In order to preserve compatibility with IVPSNR v1.0 software, the commandline parameters syntax and semantics was not changed.

2.6 Compile-time parameters

The IV-PSNR v2.0 include a set of compile time parameters. Those parameters are defined in `CommonDef.h` file:

- **VERBOSE_LEVEL** – Controls number of per-frame printing. (default = 0)
- **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 applied, this switch has to be disabled. (default = enabled)

2.7 Compilation requirements

The IV-PSNR v2.0 does not need any external libraries (except C++ standard library). In order to build the software, the ISO C++11 conformant compiler is required.

3 Results

Both versions of IVPSNR were compared on TMIV5.0 anchor results. Full results are attached in the .xlsx file.

3.1 Measured quality differences

	all data	all data excl. 2 views affected by GCD bug in IVPSNR v1.0
min_diff	-0.0248	-0.0248
max_diff	0.8966	0.0718
avg_diff	0.0156	0.0072

3.2 Processing time comparison

Sequence	Processing time		Time reduction	Speedup
	IVPSNR v1.0	IVPSNR v2.0		
SA	257.357	84.643	67%	3.04
SB	127.475	42.009	67%	3.03
SC	235.013	83.765	64%	2.81
SD	66.884	22.297	67%	3.00
SE	59.950	17.688	70%	3.39
SJ	56.673	17.709	69%	3.20
SL	56.531	17.731	69%	3.19
Total	859.883	285.842	68%	3.09

4 Recommendation

We recommend to use IVPSNR v2.0 instead of v1.0.

5 References

- [N18709] “Software manual of IV-PSNR for Immersive Video”
ISO/IEC JTC1/SC29/WG11 MPEG/N18709, July 2019, Göteborg, Sweden.