

**INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG04
MPEG VIDEO CODING**

**ISO/IEC JTC1/SC29/WG04 MPEG VC/M64727
October 2023, Hannover, DE**

Source Poznań University of Technology, Poznań, Poland
Status Input document
Title Optimized IV-PSNR software with invalid pixel detection
Author Jakub Stankowski, Adrian Dziembowski

Abstract

The document presents an improved version of IVPSNR software. The output of the new version is the same as for IVPSNR v4.0, so they can be used interchangeably. Recommendations: * create IVPSNR 5.0 based on this proposal, * issue an output document for the IV-PSNR 5.0 manual.

1 IV-PSNR v5.0 software changes

The IV-PSNR software has undergone a general overhaul, including a change of code structure and build system. The goal of the work was to improve performance and allow for detecting some source data-related errors. The source code is available on MPEG Git repository (*dev* tag).

1.1 Application parameters available at runtime

- **InvalidPelActn** = Select action taken if invalid pixel value is detected (optional, default STOP) [SKIP - disable pixel value checking, WARN - print warning and ignore, STOP - stop execution, CNCL - try to conceal by clipping to bit depth range].
- **NameMismatchActn** = Select action taken if parameters derived from filename are different than provided as input parameters. Checks resolution, bit depth and chroma format. (optional, default WARN) [SKIP - disable checking, WARN - print warning and ignore, STOP - stop execution].

1.2 Build system

Building the IV-PSNR software requires using CMake (<https://cmake.org/>) and C++17 conformant compiler (e.g., GCC >= 8.0, clang >= 5.0, MSVC >= 19.15). For user convenience, we prepared a set of scripts for easy "one click" configure and build:

- `configure_and_build.bat` - for Windows users
- `configure_and_build.sh` - for Unix/Linux users

The IV-PSNR application and its build system is designed to create fastest possible binary. On x86-64 microarchitectures the build system can create four version of compiled application, each optimized for one predefined x86-64 Microarchitecture Feature Levels [x86-64, x86-64-v2, x86-64-v3, x86-64-v4] (defined in <https://gitlab.com/x86-psABIs/x86-64-ABI>). The final binary consists of these four optimized variants and a runtime dynamic dispatcher. The dispatcher uses

CPUID instruction to detect available instruction set extensions and selects the fastest possible code path.

The IV-PSNR CMake project defines the following parameters:

- **PMBB_GENERATE_MULTI_MICROARCH_LEVEL_BINARIES** = Enables generation of multiple code paths, optimized for each variant of x86-64 Microarchitecture Feature Levels.
- **PMBB_GENERATE_SINGLE_APP_WITH_RUNTIME_DISPATCH** = Enables building single application with runtime dynamic dispatch. Requires `PMBB_GENERATE_MULTI_MICROARCH_LEVEL_BINARIES=True`.
- **PMBB_GENERATE_DEDICATED_APPS_FOR_EVERY_MFL** = Enables building multiple applications, each optimized for selected x86-64 Microarchitecture Feature Level. Requires `PMBB_GENERATE_MULTI_MICROARCH_LEVEL_BINARIES=True`.
- **PMBB_BUILD_WITH_MARCH_NATIVE** = Enable option to force compiler to tune generated code for the micro-architecture and ISA extensions of the host CPU. Conflicts with `PMBB_GENERATE_MULTI_MICROARCH_LEVEL_BINARIES``. Generated binary is not portable across different microarchitectures.

1.3 Testing

- IV-PSNR gained support for unit testing. In this release unit tests for basic data processing routines were added. Moreover, testing includes consistency between all implementations (portable C++, SSE4.2, AVX2 and AVX512).

1.4 Pixel values checking notes (*InvalidPelActn*)

Before calculating the IV-PSNR metric the software scans the content of YUV file in order to evaluate if all pixel values are in range $[0, \text{MaxVal}]$ where $\text{MaxVal} = (1 \ll \text{BitDepth}) - 1$. If invalid pel is detected the software can take following actions based on `InvalidPelActn` parameter value:

- **SKIP** = disable pixel value checking
- **WARN** = print warning and ignore invalid pel values (may lead to unreliable IV-PSNR, WS-PSNR and PSNR metrics value)
- **STOP** = print warning and stop execution
- **CNCL** = print warning and try to conceal the pel value by clipping to highest value within bit depth range

This mode was added because in some cases RVS (and possibly other software) may produce invalid sequence (with values above `MaxVal`).

1.5 Filename (*filepath*) parameters mismatch checking notes (*InvalidPelAction*)

Before opening the YUV file the software tries to derive important video parameters (resolution, bit depth and chroma format) from file name and file path. If mismatch between parameters provided from commandline (or config file) and derived values is detected, the software can take following actions based on `NameMismatchActn` parameter value:

- **SKIP** = disable pixel value checking

- **WARN** = print warning and ignore invalid pel values
- **STOP** = print warning and stop execution

1.6 Metrics calculation optimizations

- more data processing functions implemented using AVX2
- wider SIMD (AVX512) implementation for some data processing functions

1.7 Other changes

- added usage of hugepages on Linux-based systems (using madvise)
- added support for chroma format 4:2:2
- added warning for settings influencing performance or breaking conformance with IV-PSNR metric defined in [M54279]

2 Compilation requirements

The IVPSNR v5.0 software uses following external components:

- “Formatting library for C++“ (libfmt) – distributed under BSD licence and automatically fetched by Cmake build system.

Building the IVPSNR tests requires:

- “doctest: The fastest feature-rich C++11/14/17/20/23 single-header testing framework” – distributed under MIT License and automatically fetched by Cmake build system.

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

3 Results

3.1 Outputted quality

The results obtained by the proposed version are exactly the same as for IVPSNR v3.0.

3.2 Performance

The IVPSNR 5.0 is once again slightly faster than IVPSNR 4.0, but changes are not significant enough to report some big speedups. Newer CPUs (capable of executing AVX512 code path) should gain better performance.

4 Recommendations

We recommend to:

- create IVPSNR 5.0 based on this proposal,
- issue an output document for the IVPSNR v5.0 manual.

5 Acknowledgement

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