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

ISO/IEC JTC1/SC29/WG04 MPEG VC/M68222
July 2024, Sapporo, JP

Source	Poznan University of Technology, Poznań, Poland
Status	Input document
Title	[MIV] The final version of the IV-PSNR software
Author	Jakub Stankowski, Adrian Dziembowski

Abstract

The document presents an improved version of the IV-PSNR software. The output of the new version is the same as for IV-PSNR v5.0, so they can be used interchangeably. Recommendation: * create IV-PSNR 6.0 based on this proposal.

1 IV-PSNR v6.0 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 (*v6.0-rc* tag).

1.1 Application parameters available at runtime

- **CalcMetricInRGB** = Calculate metrics in RGB color space (flag, default disabled). Enables on-demand convesion from YCbCr to RGB.
- **ColorSpace** = Color space of input YUV file (optional, required by CalcMetricInRGB) [BT601, SMPTE170M, BT709, SMPTE240M, BT2020].
- **CmpWeightsSearch** = IV-metric component weights used during search ("Lm:Cb:Cr:0" or "R:G:B:0" - per component integer weights, default="4:1:1:0", quotes are mandatory, requires USE_RUNTIME_CMPWEIGHTS=1)
- **CmpWeightsAverage** = IV-metric component weights used during averaging ("Lm:Cb:Cr:0" or "R:G:B:0" - per component integer weights, default="4:1:1:0", quotes are mandatory)
- **ComponentWeights** – REMOVED

1.2 Other changes

- added fast SIMD implementation of Kahan-Babuška-Neumaier accumulation
- added "reasonable auto" mode for number of threads selection and set it as default behavior (since using all available threads on 128 core machine was slightly unwise)
- added possibility to overwrite dynamic dispatcher decision and manualy select code path
- added weighted PSNR-YCbCr, WSPSNR-YCbCr, PSNR-RGB, and WSPSNR-RGB output

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).

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_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.
- **PMBB_GENERATE_TESTING** = Enables generation of unit tests for critical data processing routines.

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

For developers convenience, we prepared a set of scripts for easy "one click" configure in development mode - without multi-architecture build and dynamic dispatch:

- configure_for_developement.bat - for Windows users
- configure_for_developement.sh - for Unix/Linux users

3 Compilation requirements

The IVPSNR v6.0 software uses following external components:

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

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

4 Results

4.1 Outputted quality

The results obtained by the proposed version are exactly the same as for IVPSNR v3.0, IVPSNR v4.0, and IVPSNR v5.0.

4.2 Performance

The IVPSNR 6.0 is once again slightly faster than IVPSNR 5.0, but changes once again are not significant enough to report some big speedups. Newer CPUs (capable of executing AVX512 code path) should gain better performance.

5 Recommendation

We recommend creating IV-PSNR 6.0 based on this proposal.

6 Acknowledgement

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