

NEXTSOFTWARE: AN ALTERNATIVE IMPLEMENTATION OF THE JOINT EXPLORATION MODEL (JEM)

Input document JVET-H0084

A. Wieckowski	B. Bross
T. Hinz	H. Schwarz
V. George	D. Marpe
J. Brandenburg	T. Wiegand
J. Ma	



Motivation



Photo: fotolia.com/SSilver

- Better code readability
- Runtime configurability
- Encoder optimizations
- Improvement of debugging capabilities
- ▶ Easier and faster integration of new tools
- Used for implementation of perceptually optimized QP adaptation and the WPSNR measure

Comparison to JEM

- NextSoftware is based on JEM → **similar, yet different**
- Elimination of Z-index → **no absPartIdx** (less ambivalence)
 - **Global addressing** is used, which better suits QTBT
- More object-oriented design, mostly **low-level OO** design
- Splitting of TComDataCU into simple types describing basic logical units
 - Prediction, Transform and Coding Units
- Elimination of macros
 - **20% fewer lines of code** (elimination of dead code)
 - Single runtime configurable executable enables **easier test automation**
- Abstraction of low-level memory operations

Comparison to JEM

- NextSoftware can decode JEM-7.0-CrossCheck and vice-versa
 - It can also decode the HEVC conformance data-set (except for MV-clipping)
- Tools not ported
 - Coding mode for 65 intra angles (JVET_B0051_NON_MPM_MODE)
 - Adaptive reference sample smoothing (COM16_C983_RSAF)
 - Enhanced motion cost estimation (JVET_D0123_ME_CTX_LUX_BITS)
 - KLT transforms (VCEG_AZ08_USE_KLT)
- Syntax changes **easily trackable** using the NEXT_HM_BIT_EQUAL macro
 - Possible bugs
 - **Enhanced tool integration**
 - Possibly reflecting intended behavior

Comparison to JEM

JEM – NextSoftware Model Equivalencies

JEM	NextSoftware
Z-index, (CTU)-RS-address, Depth	Position, Size, (Comp)Area, UnitArea
TComDataCU	CodingUnit, PredictionUnit, TransformUnit Operations in CU, PU, TU namespaces CodingStructure
TComTU	Partitioning is governed by Partitioner
TComPicYuv	Picture
TComPic	Picture
TComYuv	AreaBuf, UnitAreaBuf

Architecture

Addressing and data mapping

- All units are globally addressed and coded using their in-picture coordinates
- CodingStructure ensures coherence of RD-search and encode-time contexts
 - Enables easy and **transparent top-down approach**
 - Allows for different forms of abstraction
- A logical unit (e.g. a CU) corresponds to a single instance of that specific type
 - Iterating over units becomes **trivial**
 - An emphasis has been put on **explicit model description**
- Extensive use of C++11 functionality allows more expressive code
 - No compromises with low-level performance
 - More abstraction → **better readability**

Results

NextSoftware (no speed-ups) vs JEM

	All Intra					Random Access				
	Y	U	V	EncT	DecT	Y	U	V	EncT	DecT
Class A1	-0.1%	-0.1%	-0.2%	107%	130%	-0.1%	0.2%	0.0%	97%	117%
Class A2	-0.1%	-0.3%	-0.2%	108%	126%	0.0%	-0.3%	-0.4%	98%	111%
Class B	-0.2%	-0.3%	-0.2%	111%	125%	-0.2%	-0.4%	-0.4%	95%	113%
Class C	-0.3%	0.0%	-0.5%	105%	99%	0.0%	0.0%	0.1%	98%	107%
Class D	-0.8%	-0.6%	-0.5%	103%	71%	0.2%	0.7%	-0.1%	88%	76%
Class E	-0.4%	-0.3%	-0.5%	111%	104%					
Class F (optional)	-0.2%	0.0%	0.1%	109%	111%	-0.3%	2.1%	0.0%	98%	117%
Overall	-0.3%	-0.3%	-0.4%	108%	108%	0.0%	0.0%	-0.2%	95%	104%

Results

NextSoftware (with speed-ups) vs JEM

	All Intra					Random Access				
	Y	U	V	EncT	DecT	Y	U	V	EncT	DecT
Class A1	0.0%	0.2%	0.1%	77%	118%	0.5%	0.6%	1.2%	72%	108%
Class A2	0.2%	0.2%	0.2%	78%	117%	1.0%	1.2%	0.8%	70%	103%
Class B	0.0%	0.3%	0.2%	83%	120%	0.6%	0.7%	0.5%	68%	103%
Class C	0.1%	0.7%	0.5%	78%	98%	0.2%	0.1%	0.7%	80%	102%
Class D	-0.5%	0.1%	-0.3%	77%	68%	0.4%	0.9%	0.2%	81%	77%
Class E	0.1%	0.9%	0.5%	70%	102%					
Class F (optional)	0.4%	0.4%	0.5%	80%	109%	0.0%	2.1%	1.1%	80%	114%
Overall	0.0%	0.4%	0.2%	78%	103%	0.5%	0.7%	0.7%	74%	98%

Fraunhofer-Institut für Nachrichtentechnik, Heinrich-Hertz-Institut, HHI

WE PUT SCIENCE INTO ACTION.

Contact:

Prof. Dr.-Ing. Thomas Wiegand
Thomas.Wiegand@hhi.fraunhofer.de
+49 30 31002-617

Einsteinufer 37
10587 Berlin

