Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG
(ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)
24th Meeting: Geneva, CH, 29 June – 5 July, 2007

Document: JVT-X072
Filename: JVT-X072.doc

Title: **H.264/MPEG-4 AVC Reference Software Manual**

Status: Input Document to JVT

Purpose: Proposed Amended Draft

Author(s) or Contact(s):

Alexis Michael Tourapis
Athanasios Leontaris
Dolby Laboratories Inc.
3601 West Alameda Ave.
Burbank, CA, 91505, USA

Tel: +1 (818) 823-2842
+1 (818) 823-2827

Email: alexis.tourapis@dolby.com
athanasios.leontaris@dolby.com

Karsten Sühring
Image Processing Department
Fraunhofer-Institute HHI
Einsteinufer 37
10587 Berlin, Germany

Tel: +49 30 31002-209

Email: Karsten.Suehring@hhi.fraunhofer.de

Gary Sullivan
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052 USA

Tel: +1 (425) 703-5308

Email: garysull@microsoft.com

Source: Dolby Laboratories Inc., Fraunhofer-Institute HHI, Microsoft Corporation

Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG
(ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)

H.264/MPEG-4 AVC REFERENCE SOFTWARE MANUAL

July, 2007

Revision Sheet

Release No.	Date	Author	Revision Description
Rev. 0	10/08/04	AT/KS/GS	Initial version of Reference Software Manual
Rev. 1	01/12/05	AT/KS/GS	Amendment to original document to reflect modified and new parameters
Rev. 2	01/18/05	AT/KS/GS	Amendment based on meeting notes
Rev. 3	02/15/05	TO	Various Document updates
Rev. 4	04/13/05	AT/KS/GS	Addition of new parameters supported in software such as Fast Mode parameters and Adaptive Offset Rounding
Rev. 5	10/12/05	AT	Parameter updates. Addition of new FME parameters
Rev. 6	04/02/06	AT/KS/GS	Parameter updates. Addition of new FME parameters. Addition of limitations section.
Rev. 7	10/16/06	AT/AL	
Rev. 8	4/14/07	AT/AL/KS	Additions of new parameters from JM version 12.2
Rev.9	7/01/07	AT/AL/KS	Additions of new parameters from JM version 12.3

AT : Alexis Michael Tourapis (alexis.tourapis@dolby.com)
GS : Gary Sullivan (garysull@windows.microsoft.com)
KS : Karsten Sühring (Karsten.Suehring@hhi.fraunhofer.de)
TO : Tobias Oelbaum (oelbaum@tum.de)
AL : Athanasios Leontaris (athanasios.leontaris@dolby.com)

H.264/MPEG-4 AVC REFERENCE SOFTWARE MANUAL

TABLE OF CONTENTS

Page #

1. GENERAL INFORMATION

1. GENERAL INFORMATION

1.0 System Overview

This document contains a detailed description of the usage of the H.264/MPEG-4 AVC reference software, and more specifically version 12.2. This includes information about the encoder and decoder input parameters, syntax, compilation issues, and additional information with regards to best usage and configuration of this software.

1.1 Project References

It is recommended that the users of this software obtain a copy of the ITU H.264/ ISO MPEG-4 AVC recommendation for full understanding of the capabilities and specifics of the standard. For further info, users may access the ITU web site at www.itu.int or the ISO web site at www.iso.int. The ITU document can be downloaded for free from http://www.itu.int/rec/T-REC-H.264, while the equivalent ISO one can be purchased at http://tinyurl.com/pnyvo. Some additional public references that could be useful in understanding this new standard and consequently the software are as follows:

- T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *in the IEEE Transactions on Circuits and Systems for Video Technology*, July 2003
- G. Sullivan and T. Wiegand, "Video Compression - From Concepts to the H.264/AVC Standard," *in Proceedings of the IEEE, Special Issue on Advances in Video Coding and Delivery*, December 2004
- D. Marpe, H. Schwarz, and T. Wiegand, "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard," *in the IEEE Transactions on Circuits and Systems for Video Technology*, July 2003
- G.J. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression," *in the IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74-90, Nov. 1998

The reference software described in these pages can be downloaded from the following link:

- http://iphome.hhi.de/suehring/tml

1.2 Authorized Use Permission

The software package contains a text file and source code header comments containing disclaimer text that describes the terms associated with the use of the software and clarifying its copyright and patent rights status.

1.3 Points of Contact

1.3.1 Information

For general inquiries with regards to the H.264/MPEG-4 AVC standard users may contact Dr. Gary Sullivan (garysull@windows.microsoft.com), Dr. Thomas Wiegand (wiegand@hhi.fraunhofer.de), and Dr. Ajay Luthra (aluthra@motorola.com) for further information. Certain information can also be provided through the ITU (www.itu.int) and ISO (www.iso.int) websites.

1.3.2 Coordination

Software coordination is performed by Mr. Karsten Sühring (Karsten.Suehring@hhi.fraunhofer.de). Additional information about the software, and also this document could be provided by Dr. Alexis Michael Tourapis (alexismt@ieee.org). For further information on key contributors to the reference software implementation please check the files "contributors.h" within the reference software package.

1.3.3 Bug Reporting

Any bugs relating to the usage of this package can be reported directly to the software coordinators using the dedicated Mantis bug tracking system at https://ipbt.hhi.de/. Information of how to use this system can be found online. Nevertheless, it is suggested that the users consider the following simple rules before reporting any new bugs:

a) The user should initialy search the database for earlier reports that may relate to the same issue. If the problem has already been reported, however the user would like to report additional information that may help in the resolution of the software, this can be added to the original report.
b) The user should specify if the problem relates to the encoder, decoder or both.
c) The software version should be specified. Note however that it is recommended that the user first examines the latest version of the software and whether the problem to be reported has already been resolved.
d) The bug encountered needs to be described as precisely as possible.
e) The necessary steps to reproduce the problem should be described.
f) The configuration files that were used or any other files that may be relevant to this bug and may help with its resolution should be provided.
g) The users are strongly adviced to use the language followed by the standard when referencing the text description.
h) After a user files a report, he/she should frequently examine whether any additional information is requested relating to this issue.

1.4 Organization of the Manual

In Section 2.0 a brief summary of the reference software will be provided. This is followed by instructions of how to install and compile the reference software under different environments (i.e. Windows and Unix/Linux based platforms) in Section 3.0. The use of the encoder is described in Section 4.0, while all encoder specific parameters are analyzed in Sections 5.0 (runtime-based) and 6.0 (compilation-based). Section 7.0 presents the decoder syntax and parameters, while finally Section 8.0 presents some of the output reports generated by the different modules of this software distribution.

1.5 Acronyms and Abbreviations

1.5.1 **AVC**: Advanced Video Codec

1.5.2 **CABAC**: Context-based Adaptive Binary Arithmetic Coding

1.5.3 **CAVLC**: Context-based Adaptive Variable Length Coding

1.5.4 **CBR**: Constant Bit Rate

1.5.5 **DPB**: Decoded Picture Buffer

1.5.6 **EPZS**: Enhanced Predictive Zonal Search

1.5.7 **FME**: Fast Motion Estimation

1.5.8 **FRExt**: Fidelity Range Extension

1.5.9 **FS**: Full Search

1.5.10 **HRD**: Hypothetical Reference Decoder

1.5.11 **IDR**: Instantaneous Decoding Refresh

1.5.12 **MB**: Macroblock

1.5.13 **MBAFF**: Macroblock-Adaptive Frame-Field Coding

1.5.14 **NAL**: Network Abstraction Layer

1.5.15 **Pel**: Pixel

1.5.16 **RTP**: Rapid Transport Protocol

1.5.17 **SAD**: Sum of Absolute Differences

1.5.18 **SATD**: Sum of Absolute Transformed Differences

1.5.19 **SEI**: Supplemental Enhancement Information

1.5.20 **SSE**: Sum of Square Errors

1.5.21 **UMHex**: Uneven Multi-Hexagon search

1.5.22 **VBR**: Variable Bit Rate

1.5.23 **VUI**: Video Usability Information

Deleted: Installation and Compilation

2. INSTALLATION AND COMPILATION

2. INSTALLATION AND COMPILATION.

2.1 Windows using MS Visual Studio 6

The software package contains a Visual Studio 6 workspace named "`jm.dsw`". This workspace includes three projects:

lencod H.264/AVC reference encoder
ldecod H.264/AVC reference decoder
rtpdump a tool for analyzing contents of RTP packets

Select the desired project and "Debug" or "Release" mode. Compilation will create the binaries "lencod.exe" or "ldecod.exe" in the "bin" directory. "rtpdump.exe" will be created in the rtpdump directory.

For compile time settings and options see section 5.

2.2 Windows using MS Visual Studio .NET

The software package contains a Visual Studio .NET workspace named "`jm_vc7.sln`" for .NET 2003 (v7) and a workspace named "`jm_vc8.sln`" for .NET 2005 (v8). The user should select the appropriate solution according to his/her .NET package. These workspaces include the following three projects:

lencod H.264/AVC reference encoder
ldecod H.264/AVC reference decoder
rtpdump a tool for analyzing contents of RTP packets

Select the desired project and the appropriate compilation mode, i.e. "Debug" or "Release". Compilation will create the binaries "lencod.exe" or "ldecod.exe" in the "bin" directory. "rtpdump.exe" will be created in the rtpdump directory.

For compile time settings and options see section 5.

2.3 UNIX and Windows using gcc (GNU Compiler Collection)

After unpacking the software package run the "unixprep.sh" shell script. This will remove Windows line break characters and create directories necessary for compilation.
In most shell this should work with:

```
. unixprep.sh
```

or

```
chmod u+x unixprep.sh
./unixprep.sh
```

For compiling the encoder change to the "lencod" directory and type:

```
make
```

For compiling the decoder change to the “ldecod” directory and type:

```
make
```

Binaries named “lencod.exe” and “ldecod.exe” are created in the “bin” directory. For debug mode binaries one can compile the software using the following syntax:

```
make DBG=1
```

The above would generate debug binary files named “lencod.dbg.exe” and “ldecod.dbg.exe” in the “bin” directory for the encoder and decoder respectively.

For compile time settings and options see section 5.

Deleted: Using The JM Encoder Module

3. USING THE JM ENCODER MODULE

3. USING THE JM ENCODER MODULE

This section provides a detailed description of the JM encoder's usage.

3.1 Encoder Syntax

```
lencod  [-h] [-d defenc.cfg] {[-f curenc1.cfg]...[-f curencN.cfg]}
        {[-p EncParam1=EncValue1]...[-p EncParamM=EncValueM]}
```

Options:	
-h	Prints parameter usage.
-d	Use <defenc.cfg> as default file for parameter initializations. If not used then file defaults to "encoder.cfg" in local directory.
-f	Read <curencM.cfg> for resetting selected encoder parameters. Multiple files could be used that set different parameters.
-p	Set parameter <EncParamM> to <EncValueM>. The entry for <EncParamM> is case insensitive.

See section 4 for a description of all parameters.

Supported video file formats:
RAW: .yuv.,rgb : YUV 4:0:0
YUV 4:2:0
YUV 4:2:2
YUV 4:4:4
RGB

Examples of usage:

```
lencod.exe

lencod.exe  -h

lencod.exe  -d  default.cfg

lencod.exe  -f  curenc1.cfg

lencod.exe  -f  curenc1.cfg –p InputFile="e:\data\container_qcif_30.yuv"\
            -p  SourceWidth=176 -p SourceHeight=144

lencod.exe  -f curenc1.cfg -p FramesToBeEncoded=30 \
            -p QPFirstFrame=28 -p QPRemainingFrame=28 -p QPBPicture=30
```

3.2 Encoder Output

When running the encoder, the encoder will display on screen rate/distortion statistics for every frame coded. Cumulative results will also be presented. The output information generated may look as follows depending on the setting of the Verbose input parameter:

```
Parsing Configfile
encoder.cfg.....................................................................
................................................................................
...................
................................................................................
...................
........

------------------------------ JM 12.2 (FRExt) -------------------------------
 Input YUV file                   : foreman_part_qcif.yuv
 Output H.264 bitstream           : test.264
 Output YUV file                  : test_rec.yuv
 YUV Format                       : YUV 4:2:0
 Frames to be encoded I-P/B       : 2/1
 PicInterlace / MbInterlace       : 0/0
 Transform8x8Mode                 : 1
--------------------------------------------------------------------------------
  Frame  Bit/pic    QP   SnrY    SnrU    SnrV    Time(ms) MET(ms) Frm/Fld Ref
--------------------------------------------------------------------------------
0000(NVB)     176
0000(IDR)   20672   28  37.505  41.289  42.851       766       0    FRM    1
0002(P)      7888   28  36.789  40.817  42.246      2406    1348    FRM    1
0001(B)      2256   30  36.160  41.043  42.735      6281    4932    FRM    0
--------------------------------------------------------------------------------
 Total Frames:  3 (2)
 Leaky BucketRateFile does not have valid entries.
 Using rate calculated from avg. rate
 Number Leaky Buckets: 8
     Rmin     Bmin     Fmin
   154080    20672    20672
   192600    20672    20672
   231120    20672    20672
   269640    20672    20672
   308160    20672    20672
   346680    20672    20672
   385200    20672    20672
   423720    20672    20672
--------------------------------------------------------------------------------
 Freq. for encoded bitstream       : 15
 ME Metric for Refinement Level 0  : SAD
 ME Metric for Refinement Level 1  : Hadamard SAD
 ME Metric for Refinement Level 2  : Hadamard SAD
 Mode Decision Metric              : Hadamard SAD
 Motion Estimation for components  : Y
 Image format                      : 176x144
 Error robustness                  : Off
 Search range                      : 32
 Total number of references        : 5
 References for P slices           : 5
 List0 references for B slices     : 5
 List1 references for B slices     : 1
 Total encoding time for the seq.  : 9.453 sec (0.32 fps)
 Total ME time for sequence        : 6.280 sec
 Sequence type                     : I-B-P-B-P (QP: I 28, P 28, B 30)
 Entropy coding method             : CABAC
 Profile/Level IDC                 : (100,40)
 Motion Estimation Scheme          : Fast Full Search
```

```
 Search range restrictions         : none
 RD-optimized mode decision        : used
 Data Partitioning Mode            : 1 partition
 Output File Format                : H.264 Bit Stream File Format
------------------ Average data all frames  ---------------------------------
 PSNR Y(dB)                        : 36.82
 PSNR U(dB)                        : 41.05
 PSNR V(dB)                        : 42.61
 cSNR Y(dB)                        : 36.78 (13.64)
 cSNR U(dB)                        : 41.05 ( 5.11)
 cSNR V(dB)                        : 42.60 ( 3.57)
 Total bits                        : 30992 (I 20672, P 7888, B 2256 NVB 176)
 Bit rate (kbit/s)  @ 30.00 Hz     : 309.92
 Bits to avoid Startcode Emulation : 0
 Bits for parameter sets           : 176
-------------------------------------------------------------------------------
Exit JM 12 (FRExt) encoder ver 12.2
```

The generated statistics in the above list represent the following information. Note that fields which are associated with *Verbose Mode* column set only to *Detailed* will not be shown when verbose is set to Normal (see section 4.1.17):

Name	***Format***	***Purpose***	***Verbose Mode***
Frame	%04d($Type)	Frame Display Order and Type	Normal/Detailed
Bit/pic	%8d	Allocated bits for current frame	Normal/Detailed
WP	%1d	Weighted Prediction method	Normal/Detailed
QP	%2d	Frame Quantization value	Normal/Detailed
QL	%2d	Frame Quantized Lagrangian value	Detailed
SnrY	%7.3f	Luminance Y PSNR	Normal/Detailed
SnrU	%7.3f	Chrominance U PSNR	Normal/Detailed
SnrV	%7.3f	Chrominance V PSNR	Normal/Detailed
Time(ms)	%7d	Total encoding time for frame	Normal/Detailed
MET(ms)	%5d	Total motion estimation time for frame	Normal/Detailed
Frm/Fld	FLD|FRM	Picture coding mode	Normal/Detailed
I	%3d	Intra Coded Macroblocks	Detailed
D	%1d	Direct mode (direct_spatial_mv_pred_flag)	Detailed
L0	%2d	List0 number of references	Detailed
L1	%2d	List1 number of references	Detailed
RDP	%d	Picture Level RD decision	Detailed
Ref	%d	Current Picture Reference Indicator (nal_reference_idc)	Normal/Detailed

3.3 Encoder Limitations

At this point, the encoder is characterized by certain limitations which may limit its usage. In particular, some items that have been identified as not being problematic or not properly supported in the software include:

- The encoder may not perform all level/profile checks as specified in Annex A of the standard which may result in incompatible/non-conforming bitstreams.
- The data partitioning implementation needs checking.
- The currently provided Rate Control has been reported to have problems when encoding bitstreams using slice mode 2, while its slow adaptation speed can result in the encoder not properly achieving the target bit rate for short sequences.
- Picture Level RD Optimization does not currently support interlace coding modes and may require memory optimizations.
- SP/SI code seems to be broken.

Deleted: Encoder Parameters

4. ENCODER PARAMETERS

4. ENCODER PARAMETERS

4.1 File Input/Output Related Parameters

These parameters specify input/output control of the encoder, including input (source)/output (generated bitstreams or reconstructed sequence) file names, and file format.

4.1.1 InputFile

Class: Text
Description: Input sequence name. Name could include file path. Current software only supports concatenated input sources (i.e. all components and frames should be included in a single file)

Note: For Unix/Linux based systems directories should be separated using a backslash "\", while for DOS\Windows systems, directories should be separated using a forward slash "/".

Example 1 (DOS):

```
lencod.exe -p InputFile="f:\seq\420\176x144\foreman_176x144_30.yuv"
```

Example 2 (Unix/Linux):

```
lencod.exe -p InputFile="/vol/seq/420/176x144/foreman_176x144_30.yuv"
```

4.1.2 RGBInput

Class: Numeric (Integer)
Description: Sets YUV or RGB Input

Options:	
0	GRB or YUV input (default)
1	RGB Input

4.1.3 YUVFormat

Class: Numeric (Integer)
Description: YUV format

Options:	
0	4:0:0
1	4:2:0 (default)
2	4:2:2
3	4:4:4

4.1.4 BitDepthLuma

Class: Numeric (Integer)
Description: Specifies bit depth for Luminance component. Allowable values are in the range of 8 (default) through 12.

4.1.5 BitDepthChroma

Class: Numeric (Integer)
Description: Specifies bit depth for Chrominance component. Allowable values are in the range of 8 (default) through 12.

4.1.6 InputHeaderLength

Class: Numeric (Integer)
Description: Specifies inputfile header size in terms of bytes. For RAW data files (i.e. YUV) this is usually 0 (default).

4.1.7 FrameRate

Class: Numeric (Double)
Description: Input File Frame rate. Supports values in the range [0.0, 100.0]. Default value is 30.0.

Note: For interlace material (i.e. 60 or 50 fields), value should be set equal to FieldRate/2 (i.e. 30.0 and 25.0 respectively).

4.1.8 SourceWidth

Class: Numeric (Integer)
Description: Image width in Luminance Samples. If the value is not a multiple of 16 the image is automatically cropped to the next number that is a multiple of 16. Default is 176.

4.1.9 SourceHeight

Class: Numeric (Integer)
Description: Image height in Luminance Samples. If no Interlace tools are used and if the value is not a multiple of 16 the image is automatically cropped to the next number that is a multiple of 16. Otherwise if the value is not a multiple of 32 the image is automatically cropped to the next number that is a multiple of 32. Default is 144.

4.1.10 StartFrame

Class: Numeric (Integer)
Description: Specifies initial frame for encoding. Default value is 0.

4.1.11 FramesToBeEncoded

Class: Numeric (Integer)
Description: Specifies number of frames to be coded excluding B slice coded frames. Default is set to 1. We shall call this as the *primary layer* of the bitstream. If B slices (or Explicit Coding Structure) are to be used (we will call this as the *secondary layer*) then:

FramesToBeEncoded = int((TotalNumberOfFrames-1)/(NumberBFrames + 1)) + 1

Example 1:
Code 10 frames using an IPPPP… assignment and sequential ordering

```
lencod.exe -p FramesToBeEncoded=10
```

Example 2:
Code 10 frames using an IBBPBBPBBP assignment. B slice coded frames are not accounted in this parameter, therefore based on above formula 4 frames need to be coded.

```
lencod.exe -p FramesToBeEncoded=4 –p FrameSkip=2 –p NumberBFrames=2
```

4.1.12 OutputFile

Class: Text
Description: Output bitstream name. Name could include file path.
Example:

```
lencod.exe  –p InputFile="f:\seq\420\176x144\foreman_176x144_30.yuv" \
            –p OutputFile=”foreman.264”
```

4.1.13 ReconFile

Class: Text
Description: Output reconstructed name. Name could include file path. If empty, no output is generated.

4.1.14 TraceFile

Class: Text
Description: Bitstream Tracefile. File is useful for debugging. To enable, code needs to be compiled by setting the define TRACE in defines.h to 1.

Warning!!!
Enabling this option may result in the generation of very large files, while it could also slow down encoding considerably. Enable with caution. Parameter recommended for debugging purposes.

4.1.15 ReportFrameStats

Class: Numeric (Integer)
Description: Allows the generation of a file (*stat_frame.dat*) containing statistical information such as number of intra/inter coded blocks, modes used etc. (0 (default): disabled, 1: enabled)

4.1.16 DisplayEncParams

Class: Numeric (Integer)
Description: If enabled outputs all encoder parameters on screen, therefore capturing a snapshot of the encoder configuration. Default is 0 (disabled).

4.1.17 Verbose

Class: Numeric (Integer)
Description: Controls level of display verboseness.

Options:	
0	Short
1	Normal
2	Full Detail

4.2 Primary Control Parameters

This section described encoder parameters that are common for all profiles and essentially control encoder behavior, available test modes, Motion Estimation and Mode decision etc.

4.2.1 ProfileIDC

Class: Numeric (Integer)
Description: Set bitstream Profile IDC. Default is 88.

Note: Some profiles cannot support certain features. See MPEG-4 AVC for supported features for each profile. Reference software may perform tests for certain features for profile conformance, but it is possible that certain validations are missing. See Annex A of H.264/AVC.

Options:	
66	Baseline
77	Main
88	Extended
100	High (FRExt)
110	High 10 (FRExt)
122	High 4:2:2 (FRExt)
244	High 4:4:4 (FRExt)
44	CAVLC 4:4:4 Intra

4.2.2 IntraProfile

Class: Numeric (Integer)
Description: Specifies usage of Intra only profile for FRExt. Default is 0 (disabled).

4.2.3 LevelIDC

Class: Numeric (Integer)
Description: Set bitstream Level IDC. Default is 21.

Note: Similar with the ProfileIDC, LevelIDC specifies the capabilities a decoder must fulfill to decode a bitstream of a certain level. Most level restrictions are driven by memory restrictions and set restrictions such as resolution supported, maximum number of references, frame rate etc. See Annex A of H.264/AVC.
Note that the level setting does not prevent the encoder from breaking certain level restrictions.

Options:	
10	1 (supports only QCIF format and below with 380160 samples/sec)
11	1.1 (CIF and below. 768000 samples/sec)
12	1.2 (CIF and below. 1536000 samples/sec)
13	1.3 (CIF and below. 3041280 samples/sec)
20	2 (CIF and below. 3041280 samples/sec)
21	2.1 (Supports HHR formats. Enables Interlace support. 5068800 samples/sec)
22	2.2 (Supports SD/4CIF formats. Enables Interlace support. 5184000 samples/sec)
30	3 (Supports SD/4CIF formats. Enables Interlace support. 10368000 samples/sec)
31	3.1 (Supports 720p HD format. Enables Interlace support. 27648000 samples/sec)
32	3.2 (Supports SXGA format. Enables Interlace support. 55296000 samples/sec)
40	4 (Supports 2Kx1K format. Enables Interlace support. 62914560 samples/sec)
41	4.1 (Supports 2Kx1K format. Enables Interlace support. 62914560 samples/sec)
42	4.2 (Supports 2Kx1K format. Frame coding only. 125829120 samples/sec)
50	5 (Supports 3672x1536 format. Frame coding only. 150994944 samples/sec)

51	5.1 (Supports 4096x2304 format. Frame coding only. 251658240 samples/sec)

4.2.4 IntraPeriod

Class: Numeric (Integer)
Description: Period of I-frames (non IDR) compared to FramesToBeEncoded. i.e. frame will be coded using intra slices every IntraPeriod frames. 0 (default) implies that only first frame will be coded as intra.
Note: If field coding is enabled, depending on the value of parameter IntraBottom, only top field will be coded as intra.

4.2.5 IDRPeriod

Class: Numeric (Integer)
Description: Period of IDR frames compared to FramesToBeEncoded. 0 (default) implies that only first frame will be coded as IDR.
Note: If field coding is enabled, depending on the value of parameter IntraBottom, only top field will be coded as IDR.

4.2.6 EnableIDRGop

Class: Numeric (Integer)
Description: Enabled closed IDR GOPs, i.e. IDR_0-P_3-B_1-B_2-P_6-B_4-B_5-IDR_7-P_{10}-B_8-B_9-P_{13}-B_{11}-B_{12}. Can considerably improve coding efficiency when IDR pictures are desired. Default is disabled (0).

4.2.7 IntraDelay

Class: Numeric (Integer)
Description: Enable delayed (in display order) IDRs, i.e. IDR_2-P_1-P_0-P_3-P_4-P_5-P_6-IDR_9-P_8-P_7-P_{10}-P_{11}-P_{12}. Can considerably improve coding efficiency. Default is disabled (0).

4.2.8 EnableOpenGOP

Class: Numeric (Integer)
Description: Enables support for Open GOP encodes. Default is disabled (0).
Note: Parameter currently does not support field coding, while it enforces reference reordering if hierarchical encoding is used.

4.2.9 QPISlice

Class: Numeric (Integer)
Description: Sets Quantization value for intra slices. Allowable values are in the range of 6*(BitDepthLuma - 8) to 51. Default is 24.

4.2.10 QPPSlice

Class: Numeric (Integer)
Description: Sets Quantization value for all P slices. Allowable values are in the range of 6*(BitDepthLuma - 8) to 51. Default is 24.

4.2.11 ChromaQPOffset

Class: Numeric (Integer)
Description: Sets the appropriate QP offset that will be used for coding Chroma components. Value can be both negative and positive (-51..51). Default is 0 (no offset).

4.2.12 CbQPOffset

Class: Numeric (Integer)
Description: Sets the appropriate QP offset that will be used for coding Cb components. Value can be both negative and positive (-51..51). Default is 0 (no offset).

Note: This is a FRExt only option

4.2.13 CrQPOffset

Class: Numeric (Integer)
Description: Sets the appropriate QP offset that will be used for coding Cr components. Value can be both negative and positive (-51..51). Default is 0 (no offset).

Note: This is a FRExt only option

4.2.14 FrameSkip

Class: Numeric (Integer)
Description: Number of frames to be skipped in input when encoding primary layer. This option allows encoding a sequence at a different layer, but also is needed to support B slice coded frames (i.e. if we wish to use 2 B slice coded frames, then this value needs to be set to 2). Default is 0.

Note: Name should be changed to correspond to actual usage of parameter.

Example 1:
Reduce original framerate by half.

```
lencod.exe –p FrameSkip=1
```

Example 2:
Use an IBBPBBP… coding structure

```
lencod.exe –p FrameSkip=2 –p NumberBFrames=2
```

4.2.15 MEDistortionFPel

Class: Numeric (Integer)
Description: Error Metric for Full-Pel (first layer) motion estimation. Default is 0.

Options:	
0	Sum of Absolute Differences (SAD). Default
1	Sum of Square Errors (SSE).
2	Sum of Absolute Transformed/Hadamard Differences (SATD).

4.2.16 MEDistortionHPel

Class: Numeric (Integer)
Description: Error Metric for Half-Pel (second layer) motion estimation. Default is 2.

Options:	
0	Sum of Absolute Differences (SAD).
1	Sum of Square Errors (SSE).
2	Sum of Absolute Transformed/Hadamard Differences (SATD). Default

4.2.17 MEDistortionQPel

Class: Numeric (Integer)
Description: Error Metric for Quarter-Pel (third layer) motion estimation. Default is 2.

Options:	
0	Sum of Absolute Differences (SAD).
1	Sum of Square Errors (SSE).
2	Sum of Absolute Transformed/Hadamard Differences (SATD). Default

4.2.18 MDDistortion

Class: Numeric (Integer)
Description: Error Metric for Mode distortion operations. Default is 2.

Note: If RDOptimization is set to 0, this parameter should be set to exactly the same value as the last subpixel refinement performed. That is, if DisableSubpelME is 0 MDDistortion should be equal to MEDistortionQPel. Otherwise, if DisableSubpelME is 1, MDDistortion should be equal to MEDistortionFPel.

Options:	
0	Sum of Absolute Differences (SAD).
1	Sum of Square Errors (SSE).
2	Sum of Absolute Transformed/Hadamard Differences (SATD). Default

4.2.19 ChromaMCBuffer

Class: Numeric (Integer)
Description: Generates and stores sub pixel values for chroma components. Can improve performance somewhat if multiple references are used at the cost of increased memory usage. Default is 0.

4.2.20 ChromaMEEnable

Class: Numeric (Integer)
Description: Considers Chroma components during motion estimation, potentially improving chroma and even overall quality. Requires *ChromaMCBuffer* to be enabled.

Options:	
0	Disabled (default)
1	Consider Chroma for Motion Compensation only for integer, first level, motion estimation
2	Consider Chroma for Motion Compensation for all motion estimation levels

4.2.21 DisableSubpelME

Class: Numeric (Integer)
Description: Disables (1) subpixel Motion Estimation. Default is 0.

4.2.22 SearchRange

Class: Numeric (Integer)
Description: Sets allowable search range for Motion Estimation.

Note: If Rate Distortion Optimization is enabled, Search window is centered around median predictor, not (0,0). Default is set to 16.

4.2.23 NumberReferenceFrames

Class: Numeric (Integer)
Description: Sets maximum number of references stored in buffer (DPB)for motion estimation compensation. Essentially sets **num_ref_frames** in the sequence parameter sets. Default is set to 1.

Note: Parameter needs to conform to level constrains for number of references allowed. See Annex A.

4.2.24 PList0References

Class: Numeric (Integer)
Description: Override of allowable references used for predicting P slices (basically sets **num_ref_idx_l0_active_minus1)**. 0 (default) sets number to be equal to NumberReferenceFrames. Value needs to be smaller or equal to NumberReferenceFrames.

4.2.25 DisposableP

Class: Numeric (Integer)
Description: Enable Disposable P slices in the primary layer.

Note: Parameter will enable the encoding of a sequence of this form: I0p1P2p3P4p5… where the numeric index corresponds to coding and display order, while uppercase and lowercase imply reference and non reference pictures respectively.

4.2.26 DispPQPOffset

Class: Numeric (Integer)
Description: Specifies QP offset used for Disposable P.

4.2.27 Log2MaxFNumMinus4

Class: Numeric (Integer)
Description: Parameter sets **log2_max_frame_num_minus4** which impacts the value of **frame_num** in each slice. If value set to -1, value is computed based on FramesToBeEncoded and number of B coded frames. Otherwise **log2_max_frame_num_minus4 =** Log2MaxFNumMinus4. Default is 0.

4.2.28 Log2MaxPOCLsbMinus4

Class: Numeric (Integer)
Description: Parameter sets **log2_max_pic_order_cnt_lsb_minus4** which impacts the value of **pic_order_cnt_lsb**._If value set to -1, value is computed based on FramesToBeEncoded and number of B coded frames. Otherwise **log2_max_pic_order_cnt_lsb_minus4 =** Log2MaxPOCLsbMinus4. Default is 2.

Note: Parameter has to be properly set to avoid repetitions of pic_order_cnt.

4.2.29 GenerateMultiplePPS

Class: Numeric (Integer)
Description: Parameter enabled the transmission of 3 different picture parameter sets (PPS). These PPSs allow the combination of weighted and non weighted prediction for P and B slices. Option can be combined with parameter RDPictureDecision to perform an RD optimal decision between picture coding modes. (0: disabled/default, 1: enabled)

4.2.30 ResendPPS

Class: Numeric (Integer)
Description: Enables retransmission of Picture Parameter sets for every picture. This could be useful if the encoder decides for various reasons to update the PPS, i.e for use of a different WP method, different chroma offsets, different weighted matrices/transform, deblocking etc. (0: disabled/default, 1: enabled)

4.2.31 PicOrderCntType

Class: Numeric (Integer)
Description: Parameter sets **pic_order_cnt_type** in SPS.

Options:	
0	POC mode 0. Recommended mode (default).
1	POC mode 1, Not fully supported in software.
2	POC mode 2. Not for use with out of order coding. i.e. all pictures need to be in sequential order.

4.2.32 UseConstrainedIntraPred

Class: Numeric (Integer)
Description: If set, disallows inter pixels from being used for intra prediction. Default is 0.

4.2.33 MbLineIntraUpdate

Class: Numeric (Integer)
Description: Enables error robustness by performing extra intra macro block updates. 0 (default) off, N: One GOB every N frames is intra coded.

4.2.34 RandomIntraMBRefresh

Class: Numeric (Integer)
Description: Forced intra MBs per picture. Default is 0.

4.2.35 Inter/Intra Mode Prediction Control

The following parameters essentially control which inter or intra prediction modes could be used for encoding purposes.

4.2.35.1 InterSearch16x16

Class: Numeric (Integer)
Description: Enable 16x16 Inter Prediction & Motion Compensation (0= disable, 1= enable/default).

4.2.35.2 InterSearch16x8

Class: Numeric (Integer)
Description: Enable 16x8 Inter Prediction & Motion Compensation (0= disable, 1= enable/default).

4.2.35.3 InterSearch8x16

Class: Numeric (Integer)
Description: Enable 8x16 Inter Prediction & Motion Compensation (0= disable, 1= enable/default).

4.2.35.4 InterSearch8x8

Class: Numeric (Integer)
Description: Enable 8x8 Inter Prediction & Motion Compensation (0= disable, 1= enable/default).

4.2.35.5 InterSearch8x4

Class: Numeric (Integer)
Description: Enable 8x4 Inter Prediction & Motion Compensation (0= disable, 1= enable/default).

4.2.35.6 InterSearch4x8

Class: Numeric (Integer)
Description: Enable 4x8 Inter Prediction & Motion Compensation (0= disable, 1= enable/default).

4.2.35.7 InterSearch4x4

Class: Numeric (Integer)
Description: Enable 8x4 Inter Prediction & Motion Compensation (0= disable, 1= enable/default).

4.2.35.8 Intra4x4ParDisable

Class: Numeric (Integer)
Description: Disable I4x4 Vertical and Horizontal prediction modes (0: off/default, 1: on).

4.2.35.9 Intra4x4DiagDisable

Class: Numeric (Integer)
Description: Disable I4x4 Diagonal Down-Left and Diagonal Down-Right prediction modes (0: off/default, 1: on).

4.2.35.10 Intra4x4DirDisable

Class: Numeric (Integer)
Description: Disable I4x4 Vertical Right, Vertical Left, Horizontal Down, and Horizontal Up prediction modes (0: off/default, 1: on).

4.2.35.11 Intra16x16ParDisable

Class: Numeric (Integer)
Description: Disable I16x16 Vertical and Horizontal prediction modes (0: off/default, 1: on).

4.2.35.12 Intra16x16PlaneDisable

Class: Numeric (Integer)
Description: Disable I16x16 plane prediction mode (0: off/default, 1: on).

4.2.35.13 IntraDisableInterOnly

Class: Numeric (Integer)
Description: Disable Intra prediction modes (in sections 4.2.35.8 through 4.2.35.12) only for Inter slices (0: all slice types/default, 1: inter slice types).

Deleted: 4.2.32.8

Deleted: 4.2.32.12

4.2.35.14 ChromaIntraDisable

Class: Numeric (Integer)
Description: Disable all Intra Chroma prediction modes except DC (0: off/default, 1: on).

4.2.35.15 FastCrIntraDecision

Class: Numeric (Integer)
Description: Perform a separate intra chroma mode decision prior to determining final coding mode. Can provide significant encoding speedup (0: off/default, 1: on).

4.2.35.16 EnableIPCM

Class: Numeric (Integer)
Description: Enables usage of I_PCM mode type. (0: off/default, 1: on).

4.2.36 Loop Filter Control

Parameters to set In-loop filter behavior.

4.2.36.1 LoopFilterParametersFlag

Class: Numeric (Integer)
Description: Sets **deblocking_filter_control_present_flag.**

Note: Although currently encoder supports multiple PPS this parameter still sets the same Loop filter for all coded pictures. Default is 0.

4.2.36.2 LoopFilterDisable

Class: Numeric (Integer)
Description: Sets **disable_deblocking_filter_idc.** Requires LoopFilterParametersFlag to be set.

Options:	
0	Default. Additional loopfilter offsets are also encoded and considered during deblocking
1	Disables deblocking for all edges.
2	Disables deblocking at slice boundaries only

4.2.36.3 LoopFilterAlphaC0Offset

Class: Numeric (Integer)
Description: Sets **slice_alpha_c0_offset_div2.** Requires LoopFilterParametersFlag to be set and LoopFilterDisable to be zero. Allowable values are in the range {-6, -5, ... 0, +1, .. +6}**.** Default is 0.

4.2.36.4 LoopFilterBetaOffset

Class: Numeric (Integer)
Description: Sets **slice_beta_offset_div2.** Requires LoopFilterParametersFlag to be set and LoopFilterDisable to be zero. Allowable values are in the range {-6, -5, ... 0, +1, .. +6}. Default is 0.

4.2.37 Weighted Prediction Parameters

The following parameters enable weighted prediction.

4.2.37.1 WeightedPrediction

Class: Numeric (Integer)
Description: Sets **weighted_pred_flag** and enables explicit weighted prediction for P slices. A simple model, based on picture DC values is used for estimating weights. Default is 0.

Note: Parameter ignored when RDPictureDecision is used.

4.2.37.2 WeightedBiprediction

Class: Numeric (Integer)
Description: Sets **weighted_bipred_idc** for weighted prediction in B slices.

Options:	
0	Disabled (default).
1	Explicit Weighted Prediction.
2	Implicit Weighted Prediction. Weights are based on POC distances.

Note: Parameter ignored when RDPictureDecision is used.

4.2.37.3 UseWeightedReferenceME

Class: Numeric (Integer)
Description: Use weighted reference for ME (0=off/default, 1=on).

4.2.38 LastFrameNumber

Class: Numeric (Integer)
Description: Overwrites FramesToBeEncoded parameter. Parameter possibly broken when used with B coded frames. Default is 0.

4.2.39 ChangeQPStart

Class: Numeric (Integer)
Description: Allows the use of a secondary QP set from frame at temporal frame position ChangeQPStart. Default is 0.

4.2.40 ChangeQPI

Class: Numeric (Integer)
Description: Sets secondary Quantization value for intra coded pictures to be used from frame ChangeQPStart and beyond. Allowable values are in the range of 0 to 51. Default is 24.

4.2.41 ChangeQPP

Class: Numeric (Integer)
Description: Sets secondary Quantization value for inter P coded pictures to be used from frame ChangeQPStart and beyond. Allowable values are in the range of 0 to 51. Default is 24.

4.2.42 ChangeQPB

Class: Numeric (Integer)
Description: Sets secondary Quantization value for non reference inter B coded pictures to be used from frame ChangeQPStart and beyond. Allowable values are in the range of 0 to 51. Default is 24.

4.2.43 ChangeQPBSRefOffset

Class: Numeric (Integer)
Description: Sets quantization offset for reference inter B coded pictures to be used from frame ChangeQPStart and beyond. Allowable values are in the range of -51 to 51. Default is 0.

4.3 Secondary Layer Parameters

Parameters for controlling secondary(enhancement) layer such as B slice usage, hierarchical structure, etc.

4.3.1 NumberBFrames

Class: Numeric (Integer)

Description: Number of B slice coded frames used. Value has to be smaller or equal to FrameSkip (4.2.14). Parameter is overwritten if the HierarchicalCoding (4.3.13) parameter is set to 3. Default is 0.

Deleted: 4.2.11

4.3.2 QPBSlice

Class: Numeric (Integer)
Description: Quantization parameter used for non stored B slices. Should be in the range [0-51]. Usually these quantizer can be set slightly higher than the quantizer for stored pictures. Default is 24.

4.3.3 BRefPicQPOffset

Class: Numeric (Integer)
Description: Quantization offset parameter used for stored B slices. Should be in the range [-51..51]. Default is 0.

4.3.4 DirectModeType

Class: Numeric (Integer)
Description: Sets **direct_spatial_mv_pred_flag** which controls the direct mode type to be used. 0 is temporal direct, while 1 is spatial direct. Default is 0 (temporal).

4.3.5 DirectInferenceFlag

Class: Numeric (Integer)
Description: Sets **direct_8x8_inference_flag** in the SPS which affects semantics of Direct Mode. Value is related to the level used (i.e. for any level above or equal to 3 parameter needs to be set to 1), and should be set appropriately even if no B slices are to be used. Default is 0.

4.3.6 BList0References

Class: Numeric (Integer)
Description: Override of allowable references used for predicting B slices using List0 (basically sets **num_ref_idx_l0_active_minus1)**. 0 (default) sets number to be equal to NumberReferenceFrames. Value needs to be smaller or equal to NumberReferenceFrames.

Note: Under most cases, setting this value to 2 should be sufficient (i.e. in terms of performance), while having a significant reduction in terms of complexity.

4.3.7 BList1References

Class: Numeric (Integer)
Description: Override of allowable references used for predicting B slices using List1 (basically sets **num_ref_idx_l1_active_minus1)**. 0 (default) sets number to be equal to NumberReferenceFrames. Value needs to be smaller or equal to NumberReferenceFrames.

Note: Under most cases, setting this value to 1 should lead to better performance (i.e. since no bits are spend for coding the reference index more bits can be allocated to code mvs or residual). If HierarchicalCoding is used nevertheless, a larger value might be better.

4.3.8 BReferencePictures

Class: Numeric (Integer)
Description: Use B coded pictures as references (overwritten by HierarchicalCoding). Default is 0.

Note: Mainly available for testing purposes.

Options:

0	Disabled (default).
1	Code B coded pictures in secondary layer as references.
2	Code primary layer reference pictures (normally coded as P) with B coded pictures.

4.3.9 BiPredMotionEstimation

Class: Numeric (Integer)
Description: Enables Multihypothesis based Motion Estimation for B slice coding. Option currently only supports 16x16 block sizes and the first list 0 and list 1 references. Option also considers weights if necessary. Default is disabled (0). For further information on such ME algorithms check the following papers.

- S.W. Wu and A. Gersho, ”Joint estimation of forward and backward motion vectors for interpolative prediction of video,” *in IEEE Transactions on Image Processing*, Vol.3, Iss.5 , pp.684=7, Sept.’94.
- Markus Flierl, Thomas Wiegand, and Bernd Girod, "A Locally Optimal Design Algorithm for Block-Based Multi-Hypothesis Motion-Compensated Prediction", *Proceedings of the Data Compression Conference*, Snowbird, USA, April 1998

4.3.10 BiPredMERefinements

Class: Numeric (Integer)
Description: Enables additional ME refinements for Multihypothesis based ME. Only considered if BiPredMotionEstimation is used. Possible values are [0-5]. Default is 0 (only initial step is performed).

4.3.11 BiPredMESearchRange

Class: Numeric (Integer)
Description: Specifies search range for BiPredMotionEstimation. However, if BiPredMERefinements are used then search range is decreased by half for every additional refinement. Default is set to 8.

4.3.12 BiPredMESubPel

Class: Numeric (Integer)
Description: Controls subpixel refinement for BiPredMotionEstimation.

Options:	
0	Disabled. No Subpel refinement is performed (default)
1	Subpel refinement is performed only for first list.
2	Subpel refinement is performed for both lists

4.3.13 HierarchicalCoding

Class: Numeric (Integer)
Description: Enables the use of advanced coding picture structures for the secondary layer. This includes the use of a hierarchical type order, or explicit frame coding types/ordering.

Options:	
0	Disabled (default). Use default coding types.
1	Use double layer approach. More specifically, if N number of B coded frames are used, all B coded frames at odd positions (starting from 0) will be coded first and stored and used as references, while even ones will follow and be coded as non

	reference.
2	Use Hierarchical layer approach with multiple levels. Basically a power of two approach is used, where each level is assigned a different priority.
3	Explicit Coding type & order. Requires presence of ExplicitHierarchyFormat parameter.

Example 1:
We would like to encode video with the following coding order I0-P8-Bs4-Bs2-Bs6-B1-B3-B5-B7-P16… We would also like to assign QP values of 24 to referenced B coded frames, and 26 to non reference frames. Also, although we will like to have 5 total references, only one reference should be used for list0 and list1 for B slices. Note that the above structure looks as follows:

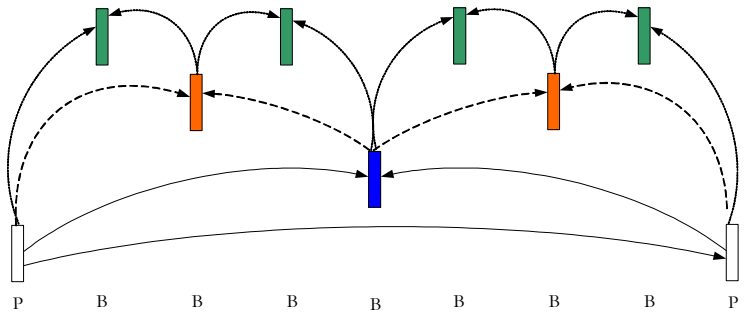

Figure 1. 4 Level Hierarchical structure.

The above could be easily done using HierarchicalCoding mode 2 which automatically generates this hierarchy. An alternative way would be to use HierarchicalCoding mode 3, and to appropriately set the necessary params using the ExplicitHierarchyFormat parameter.

```
lencod.exe  -p NumberReferenceFrames=5 –p FrameSkip=7 \
            –p HierarchicalCoding=2 -p QPBSlice=26 \
            –p BRefPicQPOffset=-2 \
            -p BList0References=1 –p BList1References=1
```

Example 2:
Lets assume that for the previous example we would prefer having only 3 levels, and that each level follows a sequential coding order. More specifically we would like the coding order to be as I0-P8-Bs2-Bs4-Bs6-B1-B3-B5-B7-P16… Note that this structure would now look as follows (i.e. we observe that now references are differently organized than in the previous case):

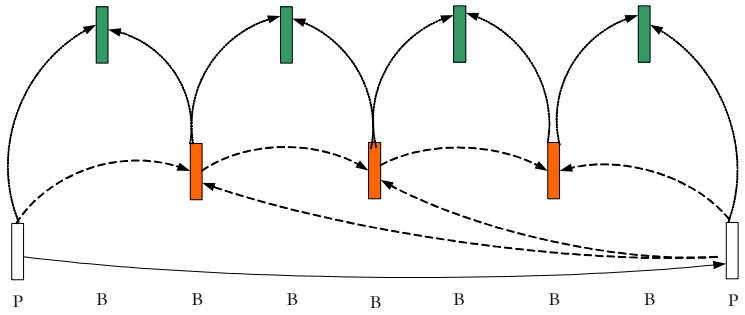

Figure 2. 3 Level Hierarchical structure.

The above could be easily done using HierarchicalCoding mode 1 which automatically generates this hierarchy. HierarchicalCoding mode 3 could also be used.

```
lencod.exe  -p NumberReferenceFrames=5 –p FrameSkip=7 \
            –p HierarchicalCoding=1 -p QPBSlice=26 \
            –p BRefPicQPOffset=-2 \
            -p BList0References=1 –p BList1References=1
```

4.3.14 ExplicitHierarchyFormat

Class: Text
Description: Parameter used with HierarchicalCoding==3 and specifies coding method (i.e. type, quantizer, coding order etc) of a frame. Parameter also overwrites use of NumberBFrames, although frames specified in parameter need to be fewer than FrameSkip.

Syntax:
[TypeFrame0][OrderFrame0][ReferenceFrame0][QPFrame0][TypeFrame1][OrderFrame1][ReferenceFrame1][QPFrame1]… [TypeFrameN][OrderFrameN][ReferenceFrameN][QPFrameN]

Allowed entries:	
[TypeFrameN]	I/i (Intra coded frame)
	P/p (P type coded frame)
	B/b (B type coded frame)
[OrderFrameN]	0-FrameSkip (specifies display order of coded frame. No duplicates are allowed)
[ReferenceFrameN]	R/r (Reference)
	E/e (Non Reference/Disposable)
[QPOffsetN]	Frame QP Offset.Final QP depends on slice type as defined by the QPNSlice parameters

Example 1:

We would like to encode video using 5 references and the following coding order I0-P8-Bs4-Bs2-B1-B3-Bs6-B5-B7-P16… We would also like to assign QP values of 24 to referenced B coded frames, and 26 to non reference frames.

```
lencod.exe  -p NumberReferenceFrames=5 –p FrameSkip=7 –p QPBSlice=24 \
            –p HierarchicalCoding=3 \
            -p ExplicitHierarchyFormat=”B4r0B2r0B1e2B3e2B6r0B5e2B7e2”
```

Example 2:
In the previous example, we would like to replace Bs6 with a P coded frame, while B7 is coded in intra mode with a QP of 22. The original QP for I and P slices was 24. Regardless of the slice type used, note that frame 7 will still not be used as a reference.

```
lencod.exe -p NumberReferenceFrames=5 –p FrameSkip=7 \
        –p HierarchicalCoding=3 \
        –p QPISlice=24 –p QPPSlice=24 –p QPBSlice=24 \
        -p ExplicitHierarchyFormat=”B4r0B2r0B1e2B3e2P6r0B5e2I7e-2”
```

Example 3:
We would like to encode a video sequence using a relatively similar coding structure as in example 1, with the difference that we would like to code all non reference frames last, i.e. I0-P8-Bs4-Bs2-Bs6-B1-B3 -B5-B7-P16… In this case we may use HierarchicalCoding=2 also which would create this structure automatically.

```
lencod.exe  -p NumberReferenceFrames=5 –p FrameSkip=7 \
            –p HierarchicalCoding=2
```

4.3.15 HierarchyLevelQPEnable

Class: Numeric (Integer)
Description: Parameter, if enabled, adjusts QP values for hierarchical structures based on the current level in increments of 1. Ignores the BRefPicQPOffset parameter. Default is 0 (disabled).

4.3.16 ReferenceReorder

Class: Numeric (Integer)
Description: Performs reference reordering for P coded frames based on POC values. This essentially places references according to temporal correlation instead of coding order. *Not supported for interlace coding modes*. Default is 0 (disabled).

Example:
In example 1 of 4.3.14 the default coding order that will be used for coding frame 16 will be {Bs6, Bs2, Bs4, P8, I0}. Nevertheless, temporally frame 8 is much closer to frame 16 and therefore this coding mode may not be as efficient. Instead, we want to use reordering commands to consider references according to their display order.

```
lencod.exe  -p NumberReferenceFrames=5 –p FrameSkip=7 \
     –p HierarchicalCoding=3 –p ReferenceReorder=1 \
     -p ExplicitHierarchyFormat=”B4r24B2r24B1e26B3e26P6r24B5e26I7e40”
```

Note: ReferenceReorder is not supported for interlace coding modes.

4.3.17 PocMemoryManagement

Class: Numeric (Integer)
Description: Performs memory management control based on POC values. Basically allows better memory management for "arbitrary" or hierarchical type coding methods if only a certain number of references are allowed due to level limitations. Parameter also recommended to be used with the EnableOpenGop parameter. Not supported for interlace coding methods. Default is 0 (disabled)

Example:
Lets assume that for the first example in 4.3.14, only a maximum of 4 references can be used. Unfortunately this would result, according to the default memory management behavior, in frame 8 being removed from the reference buffer immediately after adding frame 16, since this has the smallest frame_num in the list. It would be preferable to remove frame 2 instead, since this frame would most likely not be very useful for predicting any future frames.

```
lencod.exe  -p NumberReferenceFrames=5 –p FrameSkip=7 \
     –p HierarchicalCoding=3 –p ReferenceReorder=1 \
     -p PocMemoryManagement=1 \
     -p ExplicitHierarchyFormat="B4r24B2r24B1e26B3e26P6r24B5e26I7e40"
```

Note: PocMemoryManagement is not supported for interlace coding modes.

4.4 Error Resiliency and Slice control

4.4.1 SliceMode

Class: Numeric (Integer)
Description: Sets slice mode.

Options:	
0	Disabled (default)
1	Fixed number of MBs per slice
2	Fixed number of Bytes per slice
3	Use Callback

4.4.2 SliceArgument

Class: Numeric (Integer)
Description: Slice arguments for modes 1 (number of MBs) and 2 (bytes). Default is 0.

4.4.3 num_slice_groups_minus1

Class: Numeric (Integer)
Description: Number of Slice Groups Minus 1, 0 == one slice group (default), 1 == two slice groups, etc.

4.4.4 slice_group_map_type

Class: Numeric (Integer)
Description: Specifies slice group map type.

Options:	
0	Interleave mode (default)
1	Dispersed Mode
2	Foreground with left-over
3	Box-out
4	Raster Scan
5	Wipe
6	Explicit, slice_group_id read from SliceGroupConfigFileName

4.4.5 slice_group_change_direction_flag

Class: Numeric (Integer)
Description: sets **slice_group_change_direction_flag**.

Options:	
0	box-out clockwise, raster scan or wipe right (default)
1	box-out counter clockwise, reverse raster scan or wipe left

4.4.6 slice_group_change_rate_minus1

Class: Numeric (Integer)
Description: Sets **slice_group_change_rate_minus1**. Default is 0.

4.4.7 SliceGroupConfigFileName

Class: Text
Description: Slice configuration file used for slice group map types 0, 2, and 6.

4.4.8 UseRedundantPicture

Class: Numeric (Integer)
Description: Enables the use of redundant pictures. Default is 0 (disabled)

4.4.9 NumRedundantHierarchy

Class: Numeric (Integer)
Description: Hierarchy mode of redundant pictures. Allowed values 0-4

4.4.10 PrimaryGOPLength

Class: Numeric (Integer)
Description: GOP length for redundant allocation (1-16). NumberReferenceFrames must be no less than PrimaryGOPLength when redundant slice is enabled.

4.4.11 NumRefPrimary

Class: Numeric (Integer)
Description: Actually used number of references for primary slices (1-16).

4.5 SP coding support

4.5.1 SPPicturePeriodicity

Class: Numeric (Integer)

Description: Sets period of SP coded frames compared to FramesToBeEncoded. 0: no SP used (default), N>0: SP coded frames inserted every N frames.

Note: SP coding might be broken in current implementation

4.5.2 QPSPPicture

Class: Numeric (Integer)
Description: Quantization parameter of SP coded pictures for prediction Error (0-51). Default is 24.

4.5.3 QPSP2Picture

Class: Numeric (Integer)
Description: Quantization parameter of SP coded pictures for Predicted Blocks (0-51). Default is 24.

4.5.4 SI_Frames

Class: Numeric (Integer)
Description: SI frame encoding flag (0=not used, 1=used)

Note: Currently this parameters needs to be enabled if SP slices are to be generated.

4.5.5 SP_output

Class: Numeric (Integer)
Description: Controls whether coefficients will be output to encode switching SP frames (0=no, 1=yes).

4.5.6 SP_output_name

Class: Text
Description: Filename for SP output coefficients

4.5.7 SP2_Frames

Class: Numeric (Integer)
Description: Switching SP frame encoding flag (0=not used, 1=used).

4.5.8 SP2_input_name1

Class: Text
Description: Filename for the first swithed bitstream coefficients

4.5.9 SP2_input_name2

Class: Text
Description: Filename for the second swithed bitstream coefficients

4.6 Output Control/Entropy Coding, NALs

The following parameters control the entropy coding method that is to be used, and other output related control options.

4.6.1 SymbolMode

Class: Numeric (Integer)
Description: Entropy Coding method (0: CAVLC, 1: CABAC). Default is 0.

4.6.2 ContextInitMethod

Class: Numeric (Integer)
Description: Cabac Context Initialization (0: fixed, 1: enabled). Default is 0.

4.6.3 FixedModelNumber

Class: Numeric (Integer)
Description: Model number for fixed decision in inter slices of Cabac Context Initialization (0-default, 1, or 2) .

4.6.4 OutFileMode

Class: Numeric (Integer)
Description: Output File mode. Supported formats are 0: Default based on Annex B, 1: RTP.

4.6.5 PartitionMode

Class: Numeric (Integer)
Description: Enable partitioning. Possible values 0: 4. (0: No Data Partitioning/default, 1:3 number of partitions per slice).

4.7 Interlace Format Handling

Options enable interlace coding modes such as field coding, Picture and Macroblock adaptive Field/Frame coding etc.

4.7.1 PicInterlace

Class: Numeric (Integer)
Description: Enables adaptive field/frame coding support at the frame level.

Options:	
0	Use Frame picture coding mode only. Default.
1	Use field picture coding mode only
2	Use adaptive frame/field picture coding mode. Decision is based on lagrangian RDO of the form $J = Distortion + \lambda \times Rate$ where *Distortion* is the SSE distortion of the entire reconstructed frame (or both fields), λ is the lagrangian parameter, and *Rate* is the allotted bits for coding the frame (or fields respectively).

Note: Decision is suboptimal, but works well under certain conditions.

4.7.2 MBInterlace

Class: Numeric (Integer)
Description: Enables adaptive field/frame coding support at the macroblock level.

Options:	
0	Use Frame coding mode only (**mb_adaptive_frame_field_flag=0)**. Default.
1	Set **mb_adaptive_frame_field_flag** =1 but code all macroblocks in frame in field mode. Mainly useful for testing purposes
2	Performs RD optimal decision between frame coded super macroblocks and field coded supermacroblocks.

Note: Decision is suboptimal, but works well under certain conditions.

Example 1:

To encode a sequence using field/frame adaptive coding at both frame and macroblock level encoder should be set as follows:

```
lencod.exe –p PicInterlace=2 –p MBInterlace=2
```

Example 2:
Use only field/frame adaptive coding at the frame level:

```
lencod.exe –p PicInterlace=2 –p MBInterlace=0
```

4.7.3 IntraBottom

Class: Numeric (Integer)
Description: Forces Intra slice coding for bottom fields at intra periods. By default (0), if field coding, bottom field is coded always as inter.

4.8 Non Normative Encoder Decisions

4.8.1 RDOptimization

Class: Numeric (Integer)
Description: Enable Lagrangian based Rate distortion optimized mode decision.

Options:	
0	Enable Low Complexity mode (default)
1	Enable High Complexity mode
2	Enable Fast High Complexity mode (does not support FRExt profiles)
3	RDO consideration with losses

Note: According to common condition, option should be set to 1 when evaluating algorithmic performance.

4.8.2 CtxAdptLagrangeMult

Class: Numeric (Integer)
Description: Flag enabled the Context Adaptive Lagrange Multiplier technique. Technique works best for RDOptimization set to 0. Default is 0 (disabled).

4.8.3 RDPictureDecision

Class: Numeric (Integer)
Description: If parameter is enabled the same picture is coded in up to 3 different modes and the one yielding the best Lagrangian cost is selected as the final coding mode for this picture. Default is 0 (disabled).

Note: If GenerateMultiplePPS is enabled, then coding mode considers all different WP methods supported by a slice. This includes normal, weights, offsets for P slices, and normal, implicit, and explicit modes for B slices. If RDPictureIntra intra slices are also coded multiple times by considering different Quantizers. If the GenerateMultiplePPS parameter is not set then all slice types are considered using 3 different Quantizers. Concept also can perform a “switch to I slice) decision for P slices if number of Intra MBs in a P slice is too high, or consideration of different QPs if Weighted Prediction is not recommended (i.e. weights are identical to default values). Currently tends to increase complexity

significantly but will be improved through the consideration of Fast Motion Estimation and decision schemes.

4.8.4 RDPictureIntra

Class: Numeric (Integer)
Description: Enables RDPictureDecision for Intra slices based on different Quantizers. Default is 0 (disabled).

4.8.5 RDPSliceWeightOnly

Class: Numeric (Integer)
Description: Performs RD Picture Decision for P slices only if explicit weights are available, or if number of Intra macroblocks is high. Default is 1 (enabled).

4.8.6 RDBSliceWeightOnly

Class: Numeric (Integer)
Description: Skips RD Picture Decision for B slices for explicit weighted prediction if explicit weights are not available without testing an alternative QP. Otherwise (if flag 0 and explicit WP is not available) a QP + 1 for non reference B, and QP – 1 for reference B will be tested as well. Default is 0 (disabled).

4.8.7 UseExplicitLambdaParams

Class: Numeric (Integer)
Description: Enables the user to explicitly set the Lagrangian parameters, instead of using the equation based approach within the reference software. Default is 0 (disabled).

Options:	
0	Default (disabled)
1	Use multiplier based lambda computation (i.e. λ = LambdaWeight $\times 2^{(QP-12)/3}$)
2	Use constant lambda values (i.e. λ = FixedLambda)

4.8.8 FixedLambdaIslice

Class: Numeric (Double)
Description: Sets value of constant Lagrangian multiplier for I slices if UseExplicitLambdaParams is set to 2. Default is 0.10.

4.8.9 FixedLambdaPslice

Class: Numeric (Double)
Description: Sets value of constant Lagrangian multiplier for P slices if UseExplicitLambdaParams is set to 2. Default is 0.10.

4.8.10 FixedLambdaBslice

Class: Numeric (Double)
Description: Sets value of constant Lagrangian multiplier for B slices if UseExplicitLambdaParams is set to 2. Default is 0.10.

4.8.11 FixedLambdaRefBslice

Class: Numeric (Double)
Description: Sets value of constant Lagrangian multiplier for referenced B slices if UseExplicitLambdaParams is set to 2. Default is 0.10.

4.8.12 FixedLambdaSPslice

Class: Numeric (Double)
Description: Sets value of constant Lagrangian multiplier for SP slices if UseExplicitLambdaParams is set to 2. Default is 0.10.

4.8.13 FixedLambdaSIslice

Class: Numeric (Double)
Description: Sets value of constant Lagrangian multiplier for SI slices if UseExplicitLambdaParams is set to 2. Default is 0.10.

4.8.14 LambdaWeightIslice

Class: Numeric (Double)
Description: Sets value of Lagrangian multiplier for I slices if UseExplicitLambdaParams is set to 1. Default is 0.65

4.8.15 LambdaWeightPslice

Class: Numeric (Double)
Description: Sets value of Lagrangian multiplier for P slices if UseExplicitLambdaParams is set to 1. Default is 0.68.

4.8.16 LambdaWeightBslice

Class: Numeric (Double)
Description: Sets value of Lagrangian multiplier for B slices if UseExplicitLambdaParams is set to 1. Default is 2.00.

4.8.17 LambdaWeightRefBslice

Class: Numeric (Double)
Description: Sets value of Lagrangian multiplier for Referenced B slices if UseExplicitLambdaParams is set to 1. Default is 1.50.

4.8.18 LambdaWeightSPslice

Class: Numeric (Double)
Description: Sets value of Lagrangian multiplier for SP slices if UseExplicitLambdaParams is set to 1. Default is 1.50.

4.8.19 LambdaWeightSIslice

Class: Numeric (Double)
Description: Sets value of Lagrangian multiplier for SI slices if UseExplicitLambdaParams is set to 1. Default is 1.50.

4.8.20 OffsetMatrixPresentFlag

Class: Numeric (Integer)
Description: Enable explicit Quantization offset support. Default is 0 (disabled).

4.8.20.1 QOffsetMatrixFile

Class: Text
Description: File specifying the values of the explicit quantization offset matrices.

Example: Specify specific Q offset matrices for all blocks from file q_offset_matrix.cfg

```
lencod.exe  –p OffsetMatrixPresentFlag=1 \
            -p QOffsetNatrixFile="q_offset_matrix.cfg"
```

4.8.21 AdaptiveRounding

Class: Numeric (Integer)
Description: Enables adaptive rounding based on JVT_N011. Default is disabled (0)

4.8.21.1 AdaptRoundingFixed

Class: Numeric (Integer)
Description: Consider adaptive rounding separately for different quantization parameters. 0 (separate QPs), 1 (joint - default).

4.8.21.2 AdaptRndPeriod

Class: Numeric (Integer)
Description: Sets the Macroblock period of when to use updated rounding parameters. Default is set to 16. In JVT_N011 value of 1 was used.

4.8.21.3 AdaptRndChroma

Class: Numeric (Integer)
Description: Performs adaptive rounding for chroma. By default (0) only luma is considered.

4.8.21.4 AdaptRndWFactorIRef

Class: Numeric (Integer)
Description: Adaptive Rounding Weighting factor for luma in I and SI slices belonging to a stored picture (divided by 4096). Default is 4.

4.8.21.5 AdaptRndWFactorPRef

Class: Numeric (Integer)
Description: Adaptive Rounding Weighting factor for luma in P and SP slices belonging to a stored picture (divided by 4096). Default is 4.

4.8.21.6 AdaptRndWFactorBRef

Class: Numeric (Integer)
Description: Adaptive Rounding Weighting factor for luma in B slices belonging to a stored picture (divided by 4096). Default is 4.

4.8.21.7 AdaptRndWFactorINRef

Class: Numeric (Integer)
Description: Adaptive Rounding Weighting factor for luma in I and SI slices belonging to a disposable picture (divided by 4096). Default is 4.

4.8.21.8 AdaptRndWFactorPNRef

Class: Numeric (Integer)
Description: Adaptive Rounding Weighting factor for luma in P and SP slices belonging to a disposable picture (divided by 4096). Default is 4.

4.8.21.9 AdaptRndWFactorBNRef

Class: Numeric (Integer)
Description: Adaptive Rounding Weighting factor for luma in B slices belonging to a disposable picture (divided by 4096). Default is 4.

4.8.21.10 AdaptRndCrWFactorIRef

Class: Numeric (Integer)
Description: Adaptive Rounding Weighting factor for chroma in I and SI slices belonging to a stored picture (divided by 4096). Default is 4.

4.8.21.11 AdaptRndCrWFactorPRef

Class: Numeric (Integer)
Description: Adaptive Rounding Weighting factor for chroma in P and SP slices belonging to a stored picture (divided by 4096). Default is 4.

4.8.21.12 AdaptRndCrWFactorBRef

Class: Numeric (Integer)
Description: Adaptive Rounding Weighting factor for chroma in B slices belonging to a stored picture (divided by 4096). Default is 4.

4.8.21.13 AdaptRndCrWFactorINRef

Class: Numeric (Integer)
Description: Adaptive Rounding Weighting factor for chroma in I and SI slices belonging to a disposable picture (divided by 4096). Default is 4.

4.8.21.14 AdaptRndCrWFactorPNRef

Class: Numeric (Integer)
Description: Adaptive Rounding Weighting factor for chroma in P and SP slices belonging to a disposable picture (divided by 4096). Default is 4.

4.8.21.15 AdaptRndCrWFactorBNRef

Class: Numeric (Integer)
Description: Adaptive Rounding Weighting factor for chroma in B slices belonging to a disposable picture (divided by 4096). Default is 4.

4.8.22 LossRateA

Class: Numeric (Integer)
Description: Expected packet loss rate of the channel for the first partition. Only valid if RDOptimization is set to 2. Default is 0.

4.8.23 LossRateB

Class: Numeric (Integer)
Description: Expected packet loss rate of the channel for the second partition. Only valid if RDOptimization is set to 2. Default is 0.

4.8.24 LossRateC

Class: Numeric (Integer)

Description: Expected packet loss rate of the channel for the third partition. Only valid if RDOptimization is set to 2. Default is 0.

4.8.25 NumberOfDecoders

Class: Numeric (Integer)
Description: Numbers of decoders used to simulate the channel. Only valid if RDOptimization is set to 2. Default is 0.

4.8.26 RestrictRefFrames

Class: Numeric (Integer)
Description: Doesnt allow reference to areas that have been intra updated in a later frame. Default is 0.

4.8.27 RestrictSearchRange

Class: Numeric (Integer)
Description: Reduces Search range for motion estimation based on references and/or block types.

Options:	
0	Based on Block Type and Reference. Default.
1	Based on reference (i.e. divide by (1<<reference_index))
2	No restrictions (should be used for common conditions)

4.8.28 DisableThresholding

Class: Numeric (Integer)
Description: Disable Thresholding of Transform Coefficients (0:off/default, 1: on)

Note: Thresholding is usually more appropriate for low to medium bitrates, while this could result in loss of details under certain situations.

4.8.29 DisableBSkipRDO

Class: Numeric (Integer)
Description: Disable B Skip Mode consideration from the RDO based mode decision (0:off/default, 1: on)

4.8.30 SkipIntraInInterSlices

Class: Numeric (Integer)
Description: Avoids testing Intra modes in Inter slices if best mode is P_SKIP or B_SKIP. (0:off/default, 1: on)

4.8.31 SearchMode

Class: Numeric (Integer)
Description: Enables Usage of Fast Motion Estimation..

Options:	
-1	Full Search
0	Fast Full Search. Default.
1	Uneven Multi-Hexagon Search (UMHex)
2	Simplified Hexagon Search
3	Enhanced Predictive Zonal Search (EPZS)

Note: Currently common conditions specify that Fast Full Search should be used. Options 1 and 2 are joint integer and fractional ME implementations. EPZS, on the other hand, can operate simultaneously on both integer and fractional positions if desired.

4.8.32 EPZS Options

EPZS is a very generic FME scheme which can achieve very high performance. For educational purposes but to also allow a user to refine the algorithm based on the target application additional parameters have been added to control the behavior of this scheme. The scheme could be further extended as is described in the original contribution as to support more patterns and additional adaptation. EPZS currently

4.8.32.1 EPZSPattern

Class: Numeric (Integer)
Description: Specifies primary refinement pattern for EPZS (around best predictor)

Options:	
0	Diamond
1	Square
2	Extended Diamond (default)
3	Large Diamond
4	Subpixel Diamond
5	PMVFAST (switching large/small diamond)

4.8.32.2 EPZSDualRefinement

Class: Numeric (Integer)
Description: Specifies usage of Dual Refinement around second best predictor

Options:	
0	Disabled
1	Diamond
2	Square
3	Extended Diamond (default)
4	Large Diamond
5	Subpixel Diamond
6	PMVFAST (switching large/small diamond)

4.8.32.3 EPZSFixedPredictors

Class: Numeric (Integer)
Description: Specifies usage Window based predictors that can improve performance for encodings requiring large search windows.

Options:	
0	Disabled
1	P only

2	P and B (default)

4.8.32.4 EPZSTemporal

Class: Numeric (Integer)
Description: Enables usage of Temporal Predictors through the consideration of co-located partitions (i.e. similar to temporal direct). Default is enabled.

4.8.32.5 EPZSSpatialMem

Class: Numeric (Integer)
Description: Enables usage of Spatial Predictors through the consideration of all block type MVs from surrounding MBs. Implementation is optimized as to require only a single row of MB Motion Vectors. Default is enabled.

4.8.32.6 EPZSMinThresScale

Class: Numeric (Integer)
Description: Lower limit for threshold used for early termination. Value depends on block type and is essentially multiplied with the base value MinBaseT in Table 1. Default is 0.

Deleted: Table 1

4.8.32.7 EPZSMedThresScale

Class: Numeric (Integer)
Description: Control multiplier parameter for the Median threshold. Value depends on block type and is essentially multiplied with the base value MedBaseT in Table 1. Default is 1.

Deleted: Table 1

4.8.32.8 EPZSMaxThresScale

Class: Numeric (Integer)
Description: Upper limit for threshold used for early termination. Value depends on block type and is essentially multiplied with the base value MaxBaseT in Table 1. Default is 1.

Deleted: Table 1

Blocktype	***16x16***	***16x8***	***8x16***	***8x8***	***8x4***	***4x8***	***4x4***
MinBaseT	64	32	32	16	8	8	4
MedBaseT	256	128	128	64	32	32	16
MaxBaseT	768	384	384	192	96	96	48

Table 1. EPZS threshold control multipliers

4.8.32.9 EPZSSubPelME

Class: Numeric (Integer)
Description: EPZS Subpel ME consideration for single prediction motion estimation. Default is 1 (enabled).

4.8.32.10 EPZSSubPelMEBipred

Class: Numeric (Integer)
Description: EPZS Subpel ME consideration for Bi-predictive motion estimation. Default is 1 (enabled).

4.8.32.11 EPZSSubPelThresScale

Class: Numeric (Integer)
Description: EPZS Subpel ME threshold scaler. Default is 2.

4.8.32.12 EPZSSubPelGrid

Class: Numeric (Integer)
Description: Perform EPZS Motion estimation using a combined integer/subpel grid. Default is 0 (disabled).

4.8.33 UMHex Options

Recently, two new parameters were added to UMHex to improve its performance mainly in terms of speed. These options could probably be used with any other ME scheme as well.

4.8.33.1 UMHexDSR

Class: Numeric (Integer)
Description: Use an adaptive method to predict the maximum search range. Default is 1 (enabled).

4.8.33.2 UMHexScale

Class: Numeric (Integer)
Description: Distortion based Threshold Scaling factor relevant to picture size. Selecting a larger value should increase speed somewhat for larger resolutions. 0:Disabled. Default is set to 3.

4.8.34 EarlySkipEnable

Class: Numeric (Integer)
Description: Early skip mode detection when RDOptimization is set to 2 based on document JVT-xxxx,doc (0: disabled/default, 1: enabled).

Note: Common conditions specify that High complexity RDO mode should be used

4.8.35 SelectiveIntraEnable

Class: Numeric (Integer)
Description: Enables Selective Intra mode decision when RDOptimization is set to 2 based on document JVT-xxxx,doc (0: disabled/default, 1: enabled).

Note: Common conditions specify that High complexity RDO mode should be used

4.8.36 Rate Control & HRD support

Parameters for rate control support.

4.8.36.1 RateControlEnable

Class: Numeric (Integer)
Description: Enable simple CBR Rate Control that conforms to the HRD. See JVT-F0xx,doc for more details. (0: disabled/default, 1: enabled). No VBR rate control is currently supported.

Example: Encode a sequence at 100kbps, with an initial QP of 32, while performing adaptation at the frame level.

```
lencod.exe  –p RateControlEnable=1 –p Bitrate=100000 \
            –p InitialQP=32 –p BasicUnit=99
```

Note: Algorithm should be used as a reference only. Has not been verified when coding field pictures or with the use of Macroblock Adaptive Frame/Field Coding.

4.8.36.2 RCUpdateMode

Class: Numeric (Integer)
Description: Specifies the Rate Control type. Could be used in the future to provide support for other rate control algorithms/modes.

Options:	
0	Original quadratic rate control scheme based on JVT-G012r1 (default)
1	Extension of quadratic scheme for all Intra and IBsBsBs… coding.
2	Basic extension of quadratic scheme to better support hierarchical coding structures
3	Extension of quadratic scheme with slice type separation

4.8.36.3 Bitrate

Class: Numeric (Integer)
Description: Set bitrate target in bits per second for HRD conforming Rate Control. Default is 0.

4.8.36.4 InitialQP

Class: Numeric (Integer)
Description: Set the initial quantization parameter for the HRD conforming Rate Control. Parameter should be selected based on bitrate goal, GOP length/type, and image spatiotemporal characteristics. If 0, the encoder tries to automatically select the best quantizer for the first picture. Default is 0.

4.8.36.5 BasicUnit

Class: Numeric (Integer)
Description: Number of Macroblocks in rate control basic unit. Value needs to be a factor of the total number of MBs in a frame. Default is 1.

4.8.36.6 ChannelType

Class: Numeric (Integer)
Description: Type of Channel. 0 (default) assumes a constant channel, 1 is a time varying channel.

4.8.36.7 NumberofLeakyBuckets

Class: Numeric (Integer)
Description: Number of Leaky Bucket values. Default is set to 2.

4.8.36.8 LeakyBucketRateFile

Class: Text
Description: File from which encoder derives rate values

4.8.36.9 LeakyBucketParamFile

Class: Text
Description: File where encoder stores leakybucketparams

4.8.36.10 RCISliceBitRatio

Class: Numeric (Double)

Description: Sets the bitrate target ratio between I and P coded slices when RCUpdateMode is set to 3. Default is 1.00.

4.8.36.11 RCBSliceBitRatio0

Class: Numeric (Double)
Description: Sets the bitrate target ratio between B and P coded slices for hierarchical level 0 when RCUpdateMode is set to 3. Default is 0.5.

4.8.36.12 RCBSliceBitRatio1

Class: Numeric (Double)
Description: Sets the bitrate target ratio between B and P coded slices for hierarchical level 1 when RCUpdateMode is set to 3. Default is 0.25.

4.8.36.13 RCBSliceBitRatio2

Class: Numeric (Double)
Description: Sets the bitrate target ratio between B and P coded slices for hierarchical level 2 when RCUpdateMode is set to 3. Default is 0.25.

4.8.36.14 RCBSliceBitRatio3

Class: Numeric (Double)
Description: Sets the bitrate target ratio between B and P coded slices for hierarchical level 3 when RCUpdateMode is set to 3. Default is 0.25.

4.8.36.15 RCBSliceBitRatio4

Class: Numeric (Double)
Description: Sets the bitrate target ratio between B and P coded slices for hierarchical level 4 when RCUpdateMode is set to 3. Default is 0.25.

4.8.36.16 RCIoverPRatio

Class: Numeric (Double)
Description: Sets the “predicted” bit ratio relationship/complexity between I and P coded slices given the same QP. Used only when RCUpdateMode is set to 3. Default is 3.8.

4.8.36.17 RCBoverPRatio

Class: Numeric (Double)
Description: Sets the “predicted” bit ratio relationship/complexity between I and P coded slices given the same QP. Used only when RCUpdateMode is set to 3. Default is 0.45.

4.8.36.18 RCMinQP

Class: Numeric (Integer)
Description: Sets the minimum allowable QP value for the rate control. Default is 0.

4.8.36.19 RCMaxQP

Class: Numeric (Integer)
Description: Sets the maximum allowable QP value for the rate control. Default is 51.

4.8.37 SEI Parameters

4.8.37.1 GenerateSEIMessage

Class: Numeric (Integer)
Description: Adds data unregistered SEI message (payload type 5) in the video. Default is 0 (disabled).

4.8.37.2 SEIMessageText

Class: Text
Description: Text message added as unregistered SEI.

4.8.37.3 ToneMappingSEIPresentFlag

Class: Numeric (Integer)
Description: Enable Tone mapping SEI. Default is 0 (Not present).

4.8.37.4 ToneMappingFile

Class: Text
Description: Tone mapping parameter file.

4.8.38 VUI Parameters

4.8.38.1 VUI_aspect_ratio_info_present_flag

Class: Numeric (Integer)
Description: If enabled (1) specifies that aspect_ratio_idc is present. Default is 0 (disabled).

4.8.38.2 VUI_aspect_ratio_idc

Class: Numeric (Integer)
Description: Specifies the value of the sample aspect ratio of the luma samples Default is 0 (unspecified). See Annex E, Table E-1 of the AVC text for more info.

Options:	
0	Unspecified
1	1:1 (“square”)
2	12:11
3	10:11
4	16:11
5	40:33
6	24:11
7	20:11
8	32:11
9	80:33
10	18:11
11	15:11
12	64:33
13	160:99

14	4:3
15	3:2
16	2:1
17..254	Reserved
255	Extended_SAR

4.8.38.3 VUI_sar_width

Class: Numeric (Integer)
Description: indicates the horizontal size of the sample aspect ratio (in arbitrary units).

4.8.38.4 VUI_sar_height

Class: Numeric (Integer)
Description: indicates the vertical size of the sample aspect ratio (in the same arbitrary units as VUI_sar_width).

4.8.38.5 VUI_overscan_info_present_flag

Class: Numeric (Integer)
Description: If equal to 1, it specifies that the overscan_appropriate_flag is present. Default is 0 (not present).

4.8.38.6 VUI_overscan_appropriate_flag

Class: Numeric (Integer)
Description: If equal to 1, this flag indicates that the cropped decoded pictures output are suitable for display using overscan. If equal to 0, it indicates that the cropped decoded pictures output contain visually important information in the entire region out to the edges of the cropping rectangle of the picture, such that the cropped decoded pictures output should not be displayed using overscan. Instead, they should be displayed using either an exact match between the display area and the cropping rectangle, or using underscan.

4.8.38.7 VUI_video_signal_type_present_flag

Class: Numeric (Integer)
Description: If equal to 1, this flag specifies that the video_format, video_full_range_flag and colour_description_present_flag flags are present. Default is 0 (not present).

4.8.38.8 VUI_video_format

Class: Numeric (Integer)
Description: This parameter indicates the video format of the pictures. When this flag is not present then the format is inferred as 5 (unspecified). Default is 0.

Options:	
0	Component
1	PAL
2	NTSC
3	SECAM

4	MAC
5	Unspecified video format
6	Reserved
7	Reserved

4.8.38.9 VUI_video_full_range_flag

Class: Numeric (Integer)
Description: This parameter indicates the black level and range of the luma and chroma signals. When not present, the value shall be inferred to be equal to 0 (default).

4.8.38.10 VUI_colour_description_present_flag

Class: Numeric (Integer)
Description: When equal to 1, it specifies that colour_primaries, transfer_characteristics and matrix_coefficients are present. When, equal to 0 (default), it specifies that colour_primaries, transfer_characteristics and matrix_coefficients are not present.

4.8.38.11 VUI_colour_primaries

Class: Numeric (Integer)
Description: This parameter indicates the chromaticity coordinates of the source primaries.
When this flag is not present, its value shall be inferred to be equal to 2 (the chromaticity is unspecified or is determined by the application). Default is 2.

4.8.38.12 VUI_transfer_characteristics

Class: Numeric (Integer)
Description: This parameter indicates the opto-electronic transfer characteristic of the source picture. When this syntax element is not present, its the value shall be inferred to be equal to 2 (the transfer characteristics are unspecified or are determined by the application). Default is 2.

4.8.38.13 VUI_matrix_coefficients

Class: Numeric (Integer)
Description: This parameter describes the matrix coefficients used in deriving luma and chroma signals from the green, blue, and red primaries. When this syntax element is not present, its value shall be inferred to be equal to 2 (default).

4.8.38.14 VUI_chroma_loc_info_present_flag

Class: Numeric (Integer)
Description: If flag is set to 1, it specifies that chroma_sample_loc_type_top_field and chroma_sample_loc_type_bottom_field are present. If set equal to 0 (default), it specifies that these parameters are not present.

4.8.38.15 VUI_chroma_sample_loc_type_top_field

Class: Numeric (Integer)
Description: This parameter specifies the location of chroma samples for the top field. If not present, the value is inferred to be equal to 0.

4.8.38.16 VUI_chroma_sample_loc_type_bottom_field

Class: Numeric (Integer)
Description: This parameter specifies the location of chroma samples for the bottom field. If not present, the value is inferred to be equal to 0.

4.8.38.17 VUI_timing_info_present_flag

Class: Numeric (Integer)
Description: If this flag is set equal to 1, it specifies that parameters num_units_in_tick, time_scale and fixed_frame_rate_flag are present in the bitstream. If 0 (default) the above parameters are not present.

4.8.38.18 VUI_num_units_in_tick

Class: Numeric (Integer)
Description: This parameter is the number of time units of a clock operating at the frequency time_scale Hz that corresponds to one increment of a clock tick counter. The default value is 1000.

4.8.38.19 VUI_time_scale

Class: Numeric (Integer)
Description: This parameter is the number of time units that pass in one second. The default value is 60000.

4.8.38.20 VUI_fixed_frame_rate_flag

Class: Numeric (Integer)
Description: If set to 1, this flag indicates that the temporal distance between the HRD output times of any two consecutive pictures in output order is constrained according to Annex E. Default is 0 (disabled).
Note: This flag has currently no real impact within the encoder and its presence may not indicate that the proper constraints are imposed.

4.8.38.21 VUI_nal_hrd_parameters_present_flag

Class: Numeric (Integer)
Description: If set to 1, this flag specifies that NAL HRD parameters (pertaining to Type II bitstream conformance) are present. Default is 0 (not present).

4.8.38.22 VUI_nal_vcl_parameters_present_flag

Class: Numeric (Integer)
Description: If set to 1, this flag specifies that VCL HRD parameters (pertaining to all bitstream conformance) are present. Default is 0 (not present).

4.8.38.23 VUI_low_delay_hrd_flag

Class: Numeric (Integer)
Description: This flag specifies the HRD operational mode as specified in Annex C of the text. When VUI_fixed_frame_rate_flag is equal to 1, this flag shall be equal to 0.

4.8.38.24 VUI_pic_struct_present_flag

Class: Numeric (Integer)
Description: If this flag is equal to 1, it specifies that picture timing SEI messages are present that include the pic_struct syntax element. Default is 0 (not present).

Note: This flag has currently no real impact within the encoder and its presence may not indicate that the proper constraints are imposed.

4.8.38.25 VUI_bitstream_restriction_flag

Class: Numeric (Integer)
Description: If this flag is equal to 1, it specifies that several sequence bitstream restriction parameters are present within the bitstream. Default is 0 (not present).
Note: This flag has currently no real impact within the encoder and its presence may not indicate that the proper constraints are imposed.

4.8.38.26 VUI_motion_vectors_over_pic_boundaries_flag

Class: Numeric (Integer)
Description: If this flag is equal to 0, it indicates that no sample outside the picture boundaries and no sample at a fractional sample position whose value is derived using one or more samples outside the picture boundaries is used to inter predict any sample. If equal to 1, it then indicates that one or more samples outside picture boundaries may be used in inter prediction. When not present, its value is inferred to be equal to 1 (default).

4.8.38.27 VUI_max_bytes_per_pic_denom

Class: Numeric (Integer)
Description: This parameter indicates a number of bytes not exceeded by the sum of the sizes of the VCL NAL units associated with any coded picture in the sequence. When not present, its value is inferred to be equal to 2 (default).

4.8.38.28 VUI_max_bits_per_mb_denom

Class: Numeric (Integer)
Description: This parameter indicates the maximum number of coded bits of macroblock_layer( ) data for any macroblock in any picture of the sequence. The value of max_bits_per_mb_denom shall be in the range of 0 to 16, inclusive. When this parameter is not present, its value is inferred to be equal to 1.

4.8.38.29 VUI_log2_max_mv_length_horizontal

Class: Numeric (Integer)
Description: This parameter indicates the maximum absolute value of a decoded horizontal motion vector component, respectively, in ¼ luma sample units, for all pictures in the sequence. When not present, its value is inferred to be equal to 16.

4.8.38.30 VUI_log2_max_mv_length_vertical

Class: Numeric (Integer)
Description: This parameter indicates the maximum absolute value of a decoded vertical motion vector component, respectively, in ¼ luma sample units, for all pictures in the sequence. When not present, its value is inferred to be equal to 16.

4.8.38.31 VUI_num_reorder_frames

Class: Numeric (Integer)
Description: This parameter indicates the maximum number of frames, complementary field pairs, or non-paired fields that precede any frame, complementary field pair, or non-paired field in the sequence in decoding order and follow it in output order. When this flag is not present, its valueis inferred to be equal to max_dec_frame_buffering.

4.8.38.32 VUI_max_dec_frame_buffering

Class: Numeric (Integer)
Description: This parameter specifies the required size of the HRD decoded picture buffer (DPB) in units of frame buffers. When this parameter is not present, its value is inferred to be equal to MaxDpbSize (see AVC text).

4.9 Other settings

4.9.1 NumberFramesInEnhancementLayerSubSequence

Class: Numeric (Integer)
Description: number of frames in the Enhanced Scalability Layer. 0 (default) means that no Enhancement Layer is used.

4.9.2 SparePictureOption

Class: Numeric (Integer)
Description: 0: no spare picture info/default, 1: spare picture available

4.9.3 SparePictureDetectionThr

Class: Numeric (Integer)
Description: Threshold for spare reference pictures detection. Default is 0.

4.9.4 SparePicturePercentageThr

Class: Numeric (Integer)
Description: Threshold for the spare macroblock percentage. Default is 0.

4.10 FRExt profile parameters

In this section all FRExt specific parameters are described, including scaling matrices, 8x8 transform usage, lossless coding etc.

4.10.1 Transform8x8Mode

Class: Numeric (Integer)
Description: Enables 8x8 Transforms

Options:	
0	Disabled. Only 4x4 transforms are used (default).
1	Allows the additional use of 8x8 transform. Results in *optimal* RD performance since it considers all possible modes
2	Consider only 8x8 transform modes (i.e. disables 4x4 transform)

4.10.2 SeparateColourPlane

Class: Numeric (Integer)
Description: Enables use of separate colour plane coding (0: disabled/default 1: enabled)

4.10.3 ScalingMatrixPresentFlag

Class: Numeric (Integer)
Description: Enable Quantization matrix support.

Options:	
0	Not Present – Disabled (Default)
1	Present only in SPS
2	Present only in PPS
3	Present in both SPS and PPS

4.10.3.1 QmatrixFile

Class: Text
Description: File specifying the values of the quantization scaling matrices. Used only if values are explicitly transmitted either at the SPS or PPS level. Otherwise default values are used.

Example: Specify specific Qmatrix for intra4x4 luma blocks. Use default for all other modes.

```
lencod.exe  –p ScalingMatrixPresentFlag=1-p QmatrixFile="q_matrix.cfg" \
            –p ScalingListPresentFlag0=1
```

4.10.3.2 ScalingListPresentFlag0

Class: Numeric (Integer)
Description: Select scaling matrix for Intra4x4 Luminance Component

Options:	
0	Not Present - Use default values if ScalingMatrixPresentFlag is not 0
1	Present only in SPS
2	Present only in PPS
3	Present in both SPS and PPS

4.10.3.3 ScalingListPresentFlag1

Class: Numeric (Integer)
Description: Select scaling matrix for Intra4x4 Chrominance U component

Options:	
0	Not Present - Use default values if ScalingMatrixPresentFlag is not 0
1	Present only in SPS
2	Present only in PPS
3	Present in both SPS and PPS

4.10.3.4 ScalingListPresentFlag2

Class: Numeric (Integer)
Description: Select scaling matrix for Intra4x4 Chrominance V component

Options:	
0	Not Present - Use default values if ScalingMatrixPresentFlag is not 0
1	Present only in SPS
2	Present only in PPS

3	Present in both SPS and PPS

4.10.3.5 ScalingListPresentFlag3

Class: Numeric (Integer)
Description: Select scaling matrix for Inter4x4 Luminance component

Options:	
0	Not Present - Use default values if ScalingMatrixPresentFlag is not 0
1	Present only in SPS
2	Present only in PPS
3	Present in both SPS and PPS

4.10.3.6 ScalingListPresentFlag4

Class: Numeric (Integer)
Description: Select scaling matrix for Inter4x4 Chrominance U component

Options:	
0	Not Present - Use default values if ScalingMatrixPresentFlag is not 0
1	Present only in SPS
2	Present only in PPS
3	Present in both SPS and PPS

4.10.3.7 ScalingListPresentFlag5

Class: Numeric (Integer)
Description: Select scaling matrix for Intrer4x4 Chrominance V component

Options:	
0	Not Present - Use default values if ScalingMatrixPresentFlag is not 0
1	Present only in SPS
2	Present only in PPS
3	Present in both SPS and PPS

4.10.3.8 ScalingListPresentFlag6

Class: Numeric (Integer)
Description: Select scaling matrix for Intra8x8 Luminance component

Options:	
0	Not Present - Use default values if ScalingMatrixPresentFlag is not 0
1	Present only in SPS
2	Present only in PPS
3	Present in both SPS and PPS

4.10.3.9 ScalingListPresentFlag7

Class: Numeric (Integer)
Description: Select scaling matrix for Inter8x8 Luminance component

Options:	
0	Not Present - Use default values if ScalingMatrixPresentFlag is not 0
1	Present only in SPS
2	Present only in PPS
3	Present in both SPS and PPS

4.10.4 QPPrimeYZeroTransformBypassFlag

Class: Numeric (Integer)
Description: Enable lossless coding when qpprime_y is zero (0: disabled, 1: enabled)

Note: Better explanation is needed for this parameter

Deleted: Hardcoded Encoder Parameters

5. HARDCODED ENCODER PARAMETERS

5. HARDCODED ENCODER PARAMETERS

Although encoder behavior is mainly controlled through the parameters provided in section 4, additional hardcoded parameters within the reference software could also modify its behavior. This includes the generation of tracing and output information, and algorithmic considerations.

5.1 defines.h

DUMP_DPB	:	Dumps DPB for debuging purposes
GET_METIME	:	Enabled ME Computation time
IMGTYPE	:	Defines data size type. 0 implies byte (i.e. best for profiles with 8 bit support), where as 1 implies unsigned short which is suitable for all types including 10-12 bit content. When set to 0, this option can provide considerable memory savings and some speed advantages when encoding 8 bit content.
LAMBDA_ACCURACY_BITS	:	Accuracy bits for the motion estimation lambda value.
TRACE	:	Enables tracefile generation.
ZEROSNR	:	Definition avoids generation of infinite SNR by always forcing at least one difference sample
*_FULL_SEARCH_RANGE_*	:	Enables optional search range restrictions depending on the RestrictSearchRange flag. Define could be removed alltogether.
*_LUMA_COEFF_COST_*	:	8x8 block Luminance coefficient threshold cost.
*_CHROMA_COEFF_COST_*	:	Chrominance coefficient threshold cost.
*_LUMA_MB_COEFF_COST_*	:	Macroblock luminance coefficient threshold cost.
*_LUMA_8x8_COEFF_COST_*	:	Threshold for P8x8 sub-macroblocks.

5.2 configfile.h

DEFAULTCONFIGFILENAME : Sets default encoder configuration file.

5.3 block.h

COEFF_COST : Array used for expensive coefficient thresholding. Currently supports two possible modes, 0 (default) enables thresholding, while 1 disables it by setting cost of all coefficients to 9. Selection is based on the value of **DisableThresholding** parameter. Values could be further modified by modifying the values in the array in block.h

5.4 mv_search.h

QP2QUANT: Sets cost for low complexity encoder mode.

6. USING THE JM DECODER MODULE

6. USING THE JM DECODER MODULE

6.1 Decoder Syntax

```
ldecod  [-h] {[defdec.cfg] | {[-p pocScale][-i bitstream.264]...
        [-o output.yuv][-r reference.yuv] [-uv]}}
```

Options:	
-h	Prints parameter usage.
[defdec.cfg]	Optional decoder config file containing all decoder information.
-s	Silent decoding
-i	Decode file <bitstream.264>. Default is set to test.264.
-o	Reconstructed file name is set to <output.yuv>. Default is test_dec.yuv
-r	Reference sequence file for PSNR computation is set to <reference.yuv>. Default is test_rec.yuv
-p	Set Poc Scale to the value pocScale. Default is 2.
-uv	Output 400 content with gray chroma components (i.e. values 128), to allow viewing of output on 420 YUV players.

Examples of usage:

```
ldecod.exe

ldecod.exe -h

ldecod.exe default.cfg

ldecod.exe –s –i bitstream.264

ldecod.exe –i bitstream.264 –o output.yuv –r reference.yuv

ldecod.exe –i bitstream420.264 -uv
```

6.2 Decoder Configuration File Format

Decoder parameters need to be placed in a specific order for the decoder to work correctly. Parameters allowed are as follows:

Decoder Parameters:	
bistream.264	H.26L coded bitstream
output.yuv	Output file in RAW format. Format is based on appropriate parameters in Sequence bitstream SPS.
input.yuv	Ref sequence (for SNR)
1	Write 4:2:0 chroma components for monochrome streams (all chroma samples are set to value 128)

0	NAL mode (0=Annex B, 1: RTP packets)
3	SNR computation offset (parameter useful for computing PSNR compared to reference if encoding does not start from frame 0.
1	Poc Scale (allowable values > 0) . Scales poc for SNR purposes. System does not compute SNR correctly currently if poc resets to zero (this could happen in current encoder if IDRs are used).
500000	Rate Decoder (HRD conformance)
104000	B decoder
73000	F decoder
leakybucketparam.cfg	LeakyBucket Params
0	Error Concealment option. Allowable values are 0 (disabled/default), 1 (frame copy), and 2 (motion copy)
2	Reference POC gap. Default is 2.
2	POC gap. Default is 2.
0	Enable silent decoding. Default is 0 (disabled).

6.3 Decoder Output

When running the decoder, the decoder will display on screen rate/distortion statistics for every frame coded. Cumulative results will also be presented. The output information generated may look as follows:

```
---------------------------- JM 12.2 (FREXt) ----------------------------
 Decoder config file                    : decoder.cfg
-------------------------------------------------------------------------
 Input H.264 bitstream                  : test.264
 Output decoded YUV                     : test_dec.yuv
 Output status file                     : log.dec
 Input reference file                   : test_rec.yuv
-------------------------------------------------------------------------
POC must = frame# or field# for SNRs to be correct
-------------------------------------------------------------------------
  Frame       POC   Pic#   QP   SnrY    SnrU    SnrV   Y:U:V  Time(ms)
-------------------------------------------------------------------------
0000(I)         0     0    28  0.0000  0.0000  0.0000  4:2:0    16
0006(P)        12     1    28  0.0000  0.0000  0.0000  4:2:0     0
0004(b)         8     2    28  0.0000  0.0000  0.0000  4:2:0    15
0002(b)         4     3    28  0.0000  0.0000  0.0000  4:2:0    16
-------------------- Average SNR all frames -----------------------------
 SNR Y(dB)           :  0.00
 SNR U(dB)           :  0.00
 SNR V(dB)           :  0.00
 Total decoding time : 0.062 sec (64.516 fps)
-------------------------------------------------------------------------
 Exit JM 12 (FREXt) decoder, ver 12.2
```

The generated statistics in the above list represent the following information:

Name	Format	Purpose
Frame	%3d($Type)	Frame Display Order and Type
POC	%3d	Frame/Field POC number
Pic#	%3d	Frame_num associated with current frame

QP	%5d	Frame Quantization value
SnrY	%7.4f	Luminance Y PSNR. If value is equal to 0.000 then reference is either not available or is identical to reconstructed.
SnrU	%7.4f	Chrominance U PSNR. If value is equal to 0.000 then reference is either not available or is identical to reconstructed.
SnrV	%7.4f	Chrominance V PSNR. If value is equal to 0.000 then reference is either not available or is identical to reconstructed.
Y:U:V	X:Y:Z	Color format
Time(ms)	%5d	Total decoding time for frame

Deleted: System Generated Reports/Output

7. HARDCODED DECODER PARAMETERS

Deleted: System Generated Reports/Output

7. HARDCODED DECODER PARAMETERS

Although encoder behavior is mainly controlled through the parameters provided in section 4, additional hardcoded parameters within the reference software could also modify its behavior. This includes the generation of tracing and output information, and algorithmic considerations.

7.1 defines.h

DUMP_DPB	:	Dump DPB for debugging purposes
IMGTYPE	:	Defines data size type. 0 implies byte (i.e. best for profiles with 8 bit support), where as 1 implies unsigned short which is suitable for all types including 10-12 bit content. When set to 0, this option can provide considerable memory savings and some speed advantages when encoding 8 bit content.
ZEROSNR	:	Definition avoids generation of infinite SNR by always forcing at least one difference sample
MAX_NUM_SLICES	:	Maximum number of slices supported per picture (increasing the value results in higher memory requirement)
PAIR_FIELDS_IN_OUTPUT	:	always pair consecutive complementary fields in file output independent of their pairing in the DPB (e.g. if second decoded field is IDR)

Deleted: System Generated Reports/Output

8. SYSTEM GENERATED REPORTS/OUTPUT

8. SYSTEM GENERATED REPORTS/OUTPUT

The Encoder and Decoder modules generate various reports that could be used for analysis purposes.

8.1 log.dat

File provides summary statistics for all simulations initiated within the current directory. This includes certain input parameters, PSNR values, bitrate, encoding duration etc. In more detail, the parameters shown in this file are:

Name	***Format***	***Purpose***
Ver	W.X/Y.Z	Encoder Version (W.X main branch, Y.Z FRExt)
Date	MM/DD	Simulation End Date
Time	HH:MM	Simulation End Time
Sequence	%30.30s	Sequence Name
#Img	%5d	Coded Primary Frames (excluding B or Hierarchical Structure)
P/MbInt	%d/%d	Picture level AFF/ Macroblock level AFF
QPI	%-3d	I slice Quantizer
QPP	%-3d	P slice Quantizer
QPB	%-3d	B slice Quantizer
Format	%4dx%4d	Width x Height
Iperiod	%3d	Intra Period
#B	%3d	Number of B coded frames
FMES	FS|FFS|HEX|SHEX|EPZS	Fast Motion Estimation usage
Hdmd	%1d%1d%1d	Distortion functions for Motion estimation
S.R	%3d	Maximum Search Range (around predictor for RDOPT ON)
#Ref	%2d	Maximum number of references (num_ref_frames)
Freq	%3d	Coded Video Frame Rate
Coding	CABAC|CAVLC	Entropy Mode Used
RD-opt	%d	Rate Distortion Optimization Option
Intra upd	ON|OFF	Use of MbLineIntraUpdate. Note that this incorrectly reports that this is off if MbLineIntraUpdate is larger than 1.
8x8Tr	%d	Mode usage of 8x8 transform
SNRY 1	%-5.3f	luminance PSNR for first frame in sequence Note: How useful is this? Should it be maybe PSNR of I coded frames? Note that such is reported in the stat file
SNRU 1	%-5.3f	Chrominance U PSNR for first frame in sequence Note: Same issue as with luma.
SNRV 1	%-5.3f	Chrominance V PSNR for first frame in sequence Note: Same issue as with luma.
SNRY N	%-5.3f	Luminance PSNR for entire sequence

SNRU N	%-5.3f	Chrominance U PSNR for entire sequence
SNRV N	%-5.3f	Chrominance V PSNR for entire sequence
#Bitr I	%6.0f	Bitrate (not bits) assigned to I coded frames
#Bitr P	%6.0f	Bitrate (not bits) assigned to P coded frames
#Bitr B	%6.0f	Bitrate (not bits) assigned to B coded frames
#Bitr IPB	%6.0f	Sequence Bitrate including overheads
Total Time	%12d	Encoding Time in ms
Me Time	%12d	Motion Estimation only time in ms

8.2 stats.dat

This file contains information about the encoded sequence, such as statistics about the macroblock types used for each different slice type, distortion information, the last encoded sequence. An example stat.dat file could look as follows:

```
-------------------------------------------------------------
  This file contains statistics for the last encoded sequence
 -------------------------------------------------------------
 Sequence                      : e:\data\foreman_176x144_30p.yuv
 No.of coded pictures          :   19
 Freq. for encoded bitstream   :   30
 I Slice Bitrate(kb/s)         :  38.98
 P Slice Bitrate(kb/s)         :  58.69
 B Slice Bitrate(kb/s)         :   8.97
 Total Bitrate(kb/s)           : 106.91
 ME Metric for Refinement Level 0 : SAD
 ME Metric for Refinement Level 1 : Hadamard SAD
 ME Metric for Refinement Level 2 : Hadamard SAD
 Mode Decision Metric             : Hadamard SAD
 Motion Estimation for components : Y
 Image format                  : 176x144
 Error robustness              : Off
 Search range                  : 32
 Total number of references    : 10
 References for P slices       : 2
 List0 refs for B slices       : 2
 List1 refs for B slices       : 2
 Entropy coding method         : CABAC
 Profile/Level IDC             : (100,40)
 EPZS Pattern                  : Extended Diamond
 EPZS Dual Pattern             : Extended Diamond
 EPZS Fixed Predictors         : All P + B
 EPZS Temporal Predictors      : Enabled
 EPZS Spatial Predictors       : Enabled
 EPZS Thresholds (16x16)       : (256 0 768)
 EPZS Subpel ME                : Enabled
 Search range restrictions     : none
 RD-optimized mode decision    : used
 ----------------------|----------------|---------------|
     Item              |     Intra      |   All frames  |
 ----------------------|----------------|---------------|
 SNR Y(dB)             |  0.00          |  0.00         |
 SNR U/V (dB)          |  0.00/ 0.00    |  0.00/ 0.00   |
 Average quant         |    28          | 28.00         |

 ----------------------|----------------|---------------|---------------|
     SNR               |        I       |       P       |       B       |
 ----------------------|----------------|---------------|---------------|
 SNR Y(dB)             |      0.000     |     0.000     |     0.000     |
```

SNR U(dB)	0.000	0.000	0.000
SNR V(dB)	0.000	0.000	0.000

Intra	Mode used
Mode 0 intra 4x4	91
Mode 1 intra 8x8	0
Mode 2+ intra 16x16	8
Mode intra IPCM	0

Inter	Mode used	MotionInfo bits
Mode 0 (copy)	129	0.00
Mode 1 (16x16)	203	139.33
Mode 2 (16x8)	108	128.00
Mode 3 (8x16)	191	256.33
Mode 4 (8x8)	257	948.44
Mode 5 intra 4x4	0	
Mode 6 intra 8x8	0	
Mode 7+ intra 16x16	3	
Mode intra IPCM	0	

B frame	Mode used	MotionInfo bits
Mode 0 (copy)	561	0.00
Mode 1 (16x16)	295	210.67
Mode 2 (16x8)	9	10.44
Mode 3 (8x16)	20	21.56
Mode 4 (8x8)	6	22.89
Mode 5 intra 4x4	0	
Mode 6 intra 8x8	0	
Mode 7+ intra 16x16	0	
Mode intra IPCM	0	

Bit usage:	Intra	Inter	B frame
Header	32.00	32.00	40.00
Mode	71.00	416.44	186.00
Motion Info	./.	1472.11	265.56
CBP Y/C	284.00	240.44	27.44
Coeffs. Y	22094.00	1657.78	38.89
Coeffs. C	2141.00	250.89	16.33
Delta quant	7.00	5.22	0.67
Stuffing Bits	7.00	8.00	8.00
average bits/frame	24636.00	4082.89	582.89

NOTE

Statistics are not collected correctly when Picture or Macroblock Level Field/Frame coding is enabled.