Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG
(ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6)
31st Meeting: London, UK, 28 June – 3 July, 2009

Document: JVT-AE010 revised
Filename: JVT-Manual.doc

Title: **H.264/14496-10 AVC Reference Software Manual (revised for JM 18.0)**

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

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

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

Tel:
+1 (818) 823-2842
+1 (818) 823-2827
+49 30 31002-209
+1 (425) 703-5308

Email:
alexis.tourapis@dolby.com
athanasios.leontaris@dolby.com
Karsten.Suehring@hhi.fraunhofer.de
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.264/14496-10 AVC REFERENCE SOFTWARE MANUAL

May, 2010

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
Rev.10	8/01/07	KS	Various document cleanups
Rev.11	12/03/08	AT	Additions of new parameters/cleanups from JM version 15.0
Rev. 12	06/29/09	KS/AT	Revisions according to JM version 16.0
Rev. 13	05/25/10	AT/KS	Revisions according to JM version 17.0/17.1
Rev.14	05/03/11	AT/KS	Revisions according to JM version 18

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.264/14496-10 AVC REFERENCE SOFTWARE MANUAL

TABLE OF CONTENTS

Page #

1. GENERAL INFORMATION

1. GENERAL INFORMATION

1.1 System Overview

This document contains a detailed description of the usage of the H.264/14496-10 AVC reference software, and more specifically version 18.0[1]. 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.
The JM software is the official reference software for the following H.264/14496-10 AVC profiles:

- Basline profile
- Main profile
- Extended profile
- High profile
- High 10 profile
- High 4:2:2 profile
- High 4:4:4 Predictive profile
- High 10 Intra profile
- High 4:2:2 Intra profile
- High 4:4:4 Intra profile
- CAVLC 4:4:4 Intra profile

The reference software for Annex G (Scalable Video Coding) profiles is the JSVM software. The reference software for Annex H (Multi-View Video Coding) is the JMVC software.
In addition the JM software supports encoding end decoding of the follwing profiles:

1.2 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

[1] The software coordinators should point out that considerable effort was done to reorganize the reference software, remove almost all global variables, make various components reentrant and modular, while at the same time keeping or even extending the existing functionality of the software. Nevertheless, the interface of running the software has remained the same and thre should be little impact to the end user.

- 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

1.3 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.

1.4 Points of Contact

1.4.1 Information

For general inquiries with regards to the H.264/MPEG-4 AVC standard users may contact Dr. Gary Sullivan (garysull@microsoft.com), Dr. Thomas Wiegand (wiegand@hhi.fraunhofer.de), and Dr. Ajay Luthra (aluthra@motorola.com). Certain information can also be provided through the ITU (www.itu.int) and ISO (www.iso.int) websites. Information pertinent to the reference software should be directed to the reference software coordinators (see Section 1.4.2).

1.4.2 Coordination

Software coordination is performed by Mr. Karsten Sühring (Karsten.Suehring@hhi.fraunhofer.de) and Dr. Alexis Michael Tourapis (alexismt@ieee.org/atour@dolby.com). For further information on key contributors to the reference software implementation please check the files "contributors.h" within the reference software package.

1.4.3 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.

1.5 Acronyms and Abbreviations

1.5.1 **AUD**: Access Unit Delimiter

1.5.2 **AVC**: Advanced Video Codec

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

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

1.5.5 **CBR**: Constant Bit Rate

1.5.6 **DPB**: Decoded Picture Buffer

1.5.7 **EPZS**: Enhanced Predictive Zonal Search

1.5.8 **FFS**: Fast Full Search

1.5.9 **FME**: Fast Motion Estimation

1.5.10 **FRExt**: Fidelity Range Extension

1.5.11 **FS**: Full Search

1.5.12 **GOP**: Group of Pictures

1.5.13 **HGOP**: Hierarchical Group of Pictures

1.5.14 **HRD**: Hypothetical Reference Decoder

1.5.15 **IDR**: Instantaneous Decoding Refresh

1.5.16 **MB**: Macroblock

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

1.5.18 **MVC**: Multiview Video Coding

1.5.19 **NAL**: Network Abstraction Layer

1.5.20 **Pel**: Pixel

1.5.21 **PSNR**: Peak Signal to Noise Ratio

1.5.22 **RTP**: Rapid Transport Protocol

1.5.23 **SAD**: Sum of Absolute Differences

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

1.5.25 **SEI**: Supplemental Enhancement Information

1.5.26 **SSE**: Sum of Square Errors

1.5.27 **SSIM**: Structural Similarity Index

1.5.28 **SVC**: Scalable Video Coding

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

1.5.30 **VBR**: Variable Bit Rate

1.5.31 **VDRD**: View Dependency and Representation Delimiter

1.5.32 **VUI**: Video Usability Information

2. INSTALLATION AND COMPILATION

2. INSTALLATION AND COMPILATION.

2.1 Windows using MS Visual Studio .NET

The software package contains a Visual Studio .NET workspace named "`jm_vc7.sln`" for .NET 2003 (v7), a workspace named "`jm_vc8.sln`" for .NET 2005 (v8), and a workspace named "`jm_vc9.sln`" for .NET 2008 (v9). 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
rtp_loss a tool for simulating RTP packet losses

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" and "rtp_loss.exe" will be created in the rtpdump and rtp_loss directories respectively.

For compile time settings and options see section 5.

Please note that the software package does not anymore provide support for Visual Studio 6.

2.2 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 for compilation.
In most shell this should work with:

```
. unixprep.sh
```

or

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

For compiling the both encoder and decoder type:

```
make
```

For compiling only the encoder or only the decoder change to the "lencod" or "ldecod" directory and type:

```
make
```

within that directory

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.

Additional options that can be used during compilation include M32=1 for enforcing generation of 32-bit binary executables on 64-bit architectures, OPT=N for controlling the optimization level, and STC=1 for static linking of libraries.

For compile time settings and options see section 5.

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:
The software supports both planar and interleaved/packed raw image data (8 to 14 bit sample inputs). Furthermore, support is provided for both concatenated (all video frames in a single file) and separate (all frames in distinct/separate) video data. See more information about different raw pixel formats at "http://www.fourcc.org/".

RAW: .yuv.,rgb : P444 - Planar, 4:4:4
P422 - Planar, 4:2:2
P420 - Planar, 4:2:0
P400 - Planar, 4:0:0
I444 - Packed, 4:4:4
I422 - Packed, 4:2:2
I420 - Packed, 4:2:0
IYUV/YV12 - Planar, 4:2:0
IYU1 - Packed, 4:2:0 (UYYVYY)
IYU2 - Packed, 4:4:4 (UYV)
YUY2 – Packed, 4:2:2 (YUYV)
YUV – Packed, 4:4:4 (YUV)

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:

```
Setting Default Parameters...
Parsing Configfile encoder.cfg
......................................................................
......................................................................
......................................................................
......................................................................
.....................................................
Parsing Quantization Offset Matrix file q_offset.cfg ...................
------------------------------ JM 17.1 (FRExt) -------------------------------
 Input YUV file                   : /data/foreman_176x144_30p.yuv
 Output H.264 bitstream           : test.264
 Output YUV file                  : test_rec.yuv
 YUV Format                       : YUV 4:2:0
 Frames to be encoded             : 3
 Freq. for encoded bitstream      : 30.00
 PicInterlace / MbInterlace       : 0/0
 Transform8x8Mode                 : 1
 ME Metric for Refinement Level 0 : SAD
 ME Metric for Refinement Level 1 : SAD
 ME Metric for Refinement Level 2 : Hadamard SAD
 Mode Decision Metric             : Hadamard SAD
 Motion Estimation for components : Y
 Image format                     : 176x144 (176x144)
 Error robustness                 : Off
 Search range                     : 32
 Total number of references       : 5
 References for P slices          : 5
 References for B slices (L0, L1) : 5, 1
 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/AVC Annex B Byte Stream Format
-------------------------------------------------------------------------------
Frame     Bit/pic    QP   SnrY    SnrU    SnrV    Time(ms) MET(ms) Frm/Fld Ref
-------------------------------------------------------------------------------
00000(NVB)     176
00000(IDR)   24536   28  37.414  39.765  42.224        65       0    FRM    1
00002( P )    4664   28  36.714  39.563  41.932       376     281    FRM    1
00001( B )     992   30  36.275  39.467  41.829      1162    1011    FRM    0
-------------------------------------------------------------------------------
 Total Frames:  3
 Leaky BucketRateFile does not have valid entries.
 Using rate calculated from avg. rate
 Number Leaky Buckets: 8
     Rmin     Bmin     Fmin
   301920    24536    24536
   377400    24536    24536
```

```
   452880    24536    24536
   528360    24536    24536
   603840    24536    24536
   679320    24536    24536
   754800    24536    24536
   830280    24536    24536
------------------ Average data all frames  ---------------------------------

 Total encoding time for the seq.  :   1.603 sec (1.87 fps)
 Total ME time for sequence        :   1.292 sec

 Y { PSNR (dB), cSNR (dB), MSE }   : { 36.80, 36.78, 13.66 }
 U { PSNR (dB), cSNR (dB), MSE }   : { 39.60, 39.60,  7.14 }
 V { PSNR (dB), cSNR (dB), MSE }   : { 41.99, 41.99,  4.11 }

 Total bits                        : 30368 (I 24536, P 4664, B 992 NVB 176)
 Bit rate (kbit/s)  @ 30.00 Hz     : 303.68
 Bits to avoid Startcode Emulation : 25
 Bits for parameter sets           : 176
 Bits for filler data              : 0

-------------------------------------------------------------------------------
Exit JM 17 (FRExt) encoder ver 17.1
```

The generated statistics in the above list represent the following information. Note that availability of the fields depends on the *Verbose Mode* currently used(see section 4.1.33):

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

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 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 currently provided Rate Control is not a state of the art scheme and 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 fully support interlace coding modes and may require memory optimizations.
- Adaptive coding structures, i.e. creating Hierarchical groups of pictures of different length and with different coding/type arrangements, are not supported
- Reduntant picture encoding works only with some settings

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: First input sequence file 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 forward slash "/", while for DOS/Windows systems, directories should be separated using a backslash "\". The parameter InputFile1 has exactly the same functionality as InputFile and can be used in it's place.

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 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).

4.1.3 StartFrame

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

4.1.4 FramesToBeEncoded

Class: Numeric (Integer)
Description: Specifies number of frames to be encoded. Unlike earlier versions (14.2 and older) this parameter now considers all frames at all layers (primary and secondary)

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.

```
lencod.exe -p FramesToBeEncoded=10 -p NumberBFrames=2
```

4.1.5 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).

4.1.6 Enable32Pulldown

Class: Numeric (Integer)
Description: Alters the input sequence and converts it into a new interlace sequence using 3:2 pulldown methods.

Options:	
0	Do nothing (default)
1	A, B, Bt|Cb, Ct|Db, D
2	A, B, C, Ct|Db, D

4.1.7 SEIVUI32Pulldown

Class: Numeric (Integer)
Description: Enables 3:2 pulldown using pic_struct signaling in Timing SEI messages.

Options:	
0	Do nothing (default)
1	A, Bt|Bb, Bt|Cb, Ct|Cb, D
2	A, B, C, C, D
3	At|Ab, Bt|Bb, Bt|Cb, Ct|Cb, Dt|Db
4	A, Bt|Bb, Bt|Cb, Ct|Db, Dt|Db
5	At|Ab, Bt|Bb, Bt|Cb, Ct|Db, Dt|Db

4.1.8 SourceWidth

Class: Numeric (Integer)
Description: Input image width in Luma Samples. If the value is not a multiple of 16 the image is automatically extended to the next number that is a multiple of 16 and cropping parameters to the original size are set in the Sequence Parameter Set. Default value is 176.

4.1.9 SourceHeight

Class: Numeric (Integer)
Description: Input image height in Luma Samples. If no Interlace tools are used and if the value is not a multiple of 16 the image is automatically extended to the next number that is a multiple of 16. Otherwise if the value is not a multiple of 32 the image is automatically extended to the next number that is a multiple of 32. If the picture is extended, cropping parameters to the original size are set in the Sequence Parameter Set. Default value is 144.

4.1.10 SourceResize

Class: Numeric (Boolean)

Description: If set to 1, then input sequence is resized according to the parameters OutputWidth and OutputHeight. Currently, resizing is considered through cropping or padding depending on the relationships of the source and output dimensions. The default value is 0 (disabled).

4.1.11 OutputWidth

Class: Numeric (Integer)
Description: Output image width in Luma Samples. Value is ignored if SourceResize is 0. If the value is not a multiple of 16 the image is automatically extended to the next number that is a multiple of 16 and cropping parameters to the original size are set in the Sequence Parameter Set. Default value is 176.

4.1.12 OutputHeight

Class: Numeric (Integer)
Description: Output image height in Luma Samples. Value is ignored if SourceResize is 0. If no Interlace tools are used and if the value is not a multiple of 16 the image is automatically extended to the next number that is a multiple of 16. Otherwise if the value is not a multiple of 32 the image is automatically extended to the next number that is a multiple of 32. If the picture is extended, cropping parameters to the original size are set in the Sequence Parameter Set. Default value is 144.

4.1.13 ProcessInput

Class: Numeric (Integer)
Description: Perform optional preprocessing on the input sequence. Currently only supports YV12 to IYUV conversion, i.e. order inversion of chroma components, (ProcessInput=2), but will be extended in the future to provide further functionality such as simple pre-filtering.

4.1.14 Interleaved

Class: Numeric (Integer)
Description: Enables support for Packed/Interleaved image formats. By default, if disabled (0), then a planar format is assumed. Note that currently this parameter only affects the input video sequence. Output video sequences are always generated using the planar format.

4.1.15 RGBInput

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

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

4.1.16 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

4.1.17 StandardRange

Class: Numeric (Integer)
Description: If RGB input, specifies if content uses standard (0) or full (1) range.

4.1.18 VideoCode

Class: Numeric (Integer)
Description: Specifies Video Codes for RGB => YUV conversions

Options:	
0	NULL
1	ITU_REC709
2	CCIR_601
3	FCC
4	ITU_REC624BG
5	SMPTE_170M
6	SMPTE_240M
7	SMPTE_260M
8	ITU_REC709_EXACT

4.1.19 SourceBitDepthLuma

Class: Numeric (Integer)
Description: Specifies input source bit depth for Luma component. Allowable values are in the range of 8 (default) through 14.

4.1.20 SourceBitDepthChroma

Class: Numeric (Integer)
Description: Specifies input source bit depth for Chroma component. Allowable values are in the range of 8 (default) through 14.

4.1.21 SourceBitDepthRescale

Class: Numeric (Boolean)
Description: If enabled then the bitdepth of the original sequence is altered according to OutputBitDepthLuma and OutputBitDepthChroma. Default is 0 (disabled).

4.1.22 OutputBitDepthLuma

Class: Numeric (Integer)
Description: Specifies output source bit depth for Luma component. Allowable values are in the range of 8 (default) through 14.

4.1.23 OutputBitDepthChroma

Class: Numeric (Integer)
Description: Specifies input source bit depth for Chroma component. Allowable values are in the range of 8 (default) through 14.

4.1.24 OutputFile

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

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

Example:

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

4.1.25 ReconFile

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

4.1.26 TraceFile

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

Note: Trace file generation may fail, if the size of the trace file exceeds 2GB.

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.

4.1.27 StatsFile

Class: Text
Description: Statistics output file. The file contains an overview of different coding statistics for the entire sequence such as number and types of intra and inter macroblocks used , motion vector and redisual bit information, etc.

4.1.28 ReportFrameStats

Class: Boolean
Description: Allows the generation of a file (*stat_frame.dat*) containing statistical information such as number of intra/inter coded blocks, modes used etc. Default value is 0 (disabled).

4.1.29 DisplayEncParams

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

4.1.30 Verbose

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

Options:	
0	Short
1	Normal (default)
2	Full Detail
3	Full Detail with combined Frame/NVB bit statistics (Extended)

4.1.31 GrayScale

Class: Boolean
Description: Enforce GrayScale encoding of video data by resetting color components to the value (1<<(bitdepth – 1)).. Default value is 0 (disabled). Requires that code is compiled with the #define flag `ALLOW_GRAYSCALE` set to 1 .

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: Value of the **profile_idc** syntax element. For switching between predictive and intra only profiles, see the IntraProfile parameter. Default value is 88.

Note:Profiles restrict the usagecertain features and encoding modes. See Annex A of H.264/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.

Options:	
44	CAVLC 4:4:4 Intra
66	Baseline
77	Main
88	Extended
100	High
110	High 10 or High 10 Intra
122	High 4:2:2Predictive or High 4:2:2 Intra
244	High 4:4:4 or High 4:4:4 Intra
118	Multiview High
128	Stereo High

4.2.2 IntraProfile

Class: Boolean
Description: Specifies usage of Intra only profiles for ProfileIDC equal to 110, 122 and 244. This results in setting **constraint_set_3_flag** equal to 1 in the bitstream. Default value is 0 (disabled).

4.2.3 LevelIDC

Class: Numeric (Integer)
Description: Value of the **level_idc** syntax element. Default value 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:	
9	1b (Supports QCIF format and below with 380160 samples/sec)
10	1 (Supports QCIF format and below with 380160 samples/sec)
11	1.1 (Supports CIF and below. 768000 samples/sec)
12	1.2 (Supports CIF and below. 1536000 samples/sec)
13	1.3 (Supports CIF and below. 3041280 samples/sec)
20	2 (Supports 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: Max period of I-coded frames (non IDR) in the encoded sequence. Note that if the position corresponds to a non "primary" level, e.g. an intermediate picture belonging to a hierarchical structure, the nearest picture belonging to the primary level with a position smaller than IntraPeriod will be encoded as intra instead. A value of 0 (default) implies that only the first frame will be coded as intra.
Note: If field coding is enabled, depending on the value of parameter IntraBottom, only the top field will be coded as intra.

4.2.5 IDRPeriod

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

4.2.6 EnableIDRGop

Class: Boolean
Description: Enables 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 value is 0 (disabled).

4.2.7 IntraDelay

Class: Numeric (Integer)
Description: Enable delayed (in display order) IDR pictures by IntraDelay pictures, e.g. if IntraDelay=2 then the resulting coding structure could look like this:
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 value is 0 (disabled).

4.2.8 AdaptiveIntraPeriod

Class: Boolean
Description: Currently ignored.

4.2.9 AdaptiveIDRPeriod

Class: Boolean
Description: Currently ignored.

4.2.10 EnableOpenGOP

Class: Boolean
Description: Enables support for Open GOP encoding. Default value is 0 (disabled). OpenGOP in this context restricts all pictures after an I coded picture in display order from referencing a picture prior to the I coded picture in display order. This is done through appropriate usage of reordering operations and setting of num_ref_idx_lX_active_minus1 for each available reference list.
Note: Parameter currently does not support field coding, while it enforces reference reordering if hierarchical encoding is used.

4.2.11 NumberBFrames

Class: Numeric (Integer)
Description: Number of B slice coded frames used. Parameter is overwritten if the HierarchicalCoding (4.7.4) parameter is set to 3. Default value is 0.

4.2.12 QPISlice

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

4.2.13 QPPSlice

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

4.2.14 QPBSlice

Class: Numeric (Integer)
Description: Quantization parameter used for non stored B slices. Should be in the range of 6*(BitDepthLuma - 8) to -51. Usually these quantizer can be set slightly higher than the quantizer for stored pictures. Default value is 24.

4.2.15 ChangeQPFrame

Class: Numeric (Integer)
Description: For fixed QP encodings, specifies a frame for which to start a new coding group while considering a different set of QP parameters (based on 4.2.16 through 4.2.20). Default value is 0 (no QP changes).

4.2.16 ChangeQPI

Class: Numeric (Integer)
Description: Sets quantization parameter (QP) value for all I slices in the second QP Group. Allowable values are in the range of 6*(BitDepthLuma - 8) to 51. Default value is 24.

4.2.17 ChangeQPP

Class: Numeric (Integer)

Description: Sets quantization parameter (QP) value for all P slices in the second QP Group. Allowable values are in the range of 6*(BitDepthLuma - 8) to 51. Default value is 24.

4.2.18 ChangeQPB

Class: Numeric (Integer)
Description: Sets quantization parameter (QP) value for all B slices in the second QP Group. Allowable values are in the range of 6*(BitDepthLuma - 8) to 51. Default value is 24.

4.2.19 ChangeQPSP

Class: Numeric (Integer)
Description: Sets quantization parameter (QP) value for all SP slices in the second QP Group. Allowable values are in the range of 6*(BitDepthLuma - 8) to 51. Default value is 24.

4.2.20 ChangeQPSI

Class: Numeric (Integer)
Description: Sets quantization parameter (QP) value for all SI slices in the second QP Group. Allowable values are in the range of 6*(BitDepthLuma - 8) to 51. Default value is 24.

4.2.21 ChromaQPOffset

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

Note: This option will be used only for Baseline, Main and Extended profiles. For FRExt profiles see CbQPOffset and CrQPOffset.

4.2.22 CbQPOffset

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

Note: This is a FRExt profile only option. For other profiles see ChromaQPOffset.

4.2.23 CrQPOffset

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

Note: This is a FRExt profile only option. For other profiles see ChromaQPOffset.

4.2.24 FrameSkip

Class: Numeric (Integer)
Description: Number of frames to be skipped when encoding the input sequence, essentially altering the frame rate of the output video sequence. This has no relationship to and is completely independent from the number of intermediate, e.g. B coded pictures, that can be used in the encoding. Default value is 0.

Example 1:
Reduce original framerate by half.

```
lencod.exe -p FrameSkip=1
```

4.2.25 MEDistortionFPel

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

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

4.2.26 MEDistortionHPel

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

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

4.2.27 MEDistortionQPel

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

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

4.2.28 MDDistortion

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

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)

4.2.29 ChromaMCBuffer

Class: Boolean

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 value is 0 (disabled).

4.2.30 ChromaMEEnable

Class: Boolean
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

4.2.31 ChromaMEWeight

Class: Numeric (Integer)
Description: Specifies weighting factor of chroma component distortion during motion estimation. Default value is 1.

4.2.32 DisableSubpelME

Class: Boolean
Description: Disables subpixel Motion Estimation. Default value is 0 (enabled).

4.2.33 SearchRange

Class: Numeric (Integer)
Description: Sets allowable search range for Motion Estimation. Default value is 16.

Note: If Rate Distortion Optimization is enabled, the search window is centered around median predictor, not (0,0).

4.2.34 UseMVLimits

Class: Boolean
Description: Constrain maximum absolute motion vector values according to SetMVXLimit and SetMVYLimit values. Default value is 0 (disabled).

4.2.35 SetMVXLimit

Class: Numeric (Integer)
Description: Sets maximum absolute horizontal motion vector value in integer pixel units. Default value is 2048.

4.2.36 SetMVYLimit

Class: Numeric (Integer)
Description: Sets maximum absolute vertical motion vector value in integer pixel units. Default value is 512. Value is further constrained according to LevelIDC limits.

4.2.37 NumberReferenceFrames

Class: Numeric (Integer)

Description: Sets the maximum number of references stored in the Decoded Picture Buffer (DPB) for motion estimation and compensation. Essentially sets the syntax element **num_ref_frames** in the sequence parameter sets. Default value is 1.

Note: This parameter needs to conform to level constrains. See Annex A.

4.2.38 PList0References

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

4.2.39 DisposableP

Class: Boolean
Description: Enable non-reference P slices in the primary layer. Default value is 0 (disabled).

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

4.2.40 DispPQPOffset

Class: Numeric (Integer)
Description: Specifies quantization parameter (QP) offset used for non-reference P slices.

4.2.41 PreferDispOrder

Class: Boolean
Description: Prefer display order when building the prediction structure as opposed to coding order (affects intra and IDR periodic insertion, among others). Default is 0.

4.2.42 PreferPowerOfTwo

Class: Boolean
Description: Prefer prediction structures that have lengths expressed as powers of two. Default is 0.

4.2.43 FrmStructBufferLength

Class: Numeric (Integer)
Description: Length of the frame structure unit buffer; it can be overriden for certain cases. Default is 1.

4.2.44 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.

4.2.45 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.

4.2.46 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.

4.2.47 BListSingleDirRefFlag

Class: Boolean
Description: Remove references from list0 and list1 that come from a different direction. Default is 0 (disabled).

4.2.48 LowDelay

Class: Boolean
Description: Enable Low Delay hierarchical coding structure generation. Default is 0 (disabled).

4.2.49 Log2MaxFNumMinus4

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

Note: With sliding window DPB operation the combination of **log2_max_frame_num_minus4** equal to 0 and **num_ref_frames** equal to 16 could lead to inserting two reference frames with the same value of **frame_num** into the DPB. To avoid this, the JM encoder prohibits this combination.

4.2.50 Log2MaxPOCLsbMinus4

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

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

4.2.51 GenerateMultiplePPS

Class: Boolean
Description: When enabled, three different Picture Parameter Sets (PPS) are generated and included into the bitstream. These Picture Parameter Sets allow the combination of weighted and non weighted prediction for P and B slices. This option can be combined with the parameter RDPictureDecision to perform an RD optimal decision between picture coding modes. Value should be disabled when generating baseline profile bitstreams. Default value is 0 (disabled).

4.2.52 SendAUD

Class: Boolean
Description: Transmit Access Delimiter Unit NALU for every Access Unit. Default value is 0 (disabled).

4.2.53 ResendSPS

Class: Numeric (integer)
Description: Enables repetition of Sequence (SPS) and Picture Parameter Sets (PPS) at various intervals. This can be useful for random access/trick modes, error resilience, etc.

Options:	
0	Disabled (default)
1	Repeat for all Intra coded pictures
2	Repeat for all IDR pictures
3	Repeat for all IDR and Open GOP intra pictures

4.2.54 ResendPPS

Class: Boolean
Description: Enables repetition of Picture Parameter Sets (PPS)before every primary coded picture. This could be useful for error resilience or if the encoder decides to update the PPS, i.e for use of a different WP method, different chroma offsets, different weighted matrices/transform, deblocking, etc. Default value is 0 (disabled).

4.2.55 PicOrderCntType

Class: Numeric (Integer)
Description: Parameter sets the value of the syntax element **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.

4.2.56 UseConstrainedIntraPred

Class: Boolean
Description: If set, disallows inter pixels from being used for intra prediction (sets the syntax element **constrained_intra_pred_flag** in the PPS). Default value is 0.

4.2.57 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.

4.2.58 RandomIntraMBRefresh

Class: Numeric (Integer)
Description: Number of macroblocks per picture that are forced to be intra coded. If non-zero, the intra macroblocks are selected randomly. Default value is 0.

4.3 Inter/Intra Mode Prediction Control

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

4.3.1 PSliceSkip

Class: Boolean
Description: Enables Skip Inter modes in P Slices. Default value is 1 (enabled).

4.3.2 PSliceSearch16x16

Class: Boolean
Description: Enables 16x16 Inter Prediction & Motion Compensation in P Slices. Default value is 1 (enabled).

4.3.3 PSliceSearch16x8

Class: Boolean
Description: Enables 16x8 Inter Prediction & Motion Compensation in P Slices. Default value is 1 (enabled).

4.3.4 PSliceSearch8x16

Class: Boolean
Description: Enables 8x16 Inter Prediction & Motion Compensation in P Slices. Default value is 1 (enabled).

4.3.5 PSliceSearch8x8

Class: Boolean
Description: Enables 8x8 Inter Prediction & Motion Compensation in P Slices. Default value is 1 (enabled).

4.3.6 PSliceSearch8x4

Class: Boolean
Description: Enables 8x4 Inter Prediction & Motion Compensation in P Slices. Default value is 1 (enabled).

4.3.7 PSliceSearch4x8

Class: Boolean
Description: Enables 4x8 Inter Prediction & Motion Compensation in P Slices. Default value is 1 (enabled).

4.3.8 PSliceSearch4x4

Class: Boolean
Description: Enables 8x4 Inter Prediction & Motion Compensation in P Slices. Default value is 1 (enabled).

4.3.9 BSliceDirect

Class: Boolean
Description: Enables Direct Mode Inter Prediction & Motion Compensation in B Slices. Default value is 1 (enabled).

4.3.10 BSliceSearch16x16

Class: Boolean
Description: Enables 16x16 Inter Prediction & Motion Compensation in B Slices. Default value is 1 (enabled).

4.3.11 BSliceSearch16x8

Class: Boolean
Description: Enables 16x8 Inter Prediction & Motion Compensation in B Slices. Default value is 1 (enabled).

4.3.12 BSliceSearch8x16

Class: Boolean
Description: Enables 8x16 Inter Prediction & Motion Compensation in B Slices. Default value is 1 (enabled).

4.3.13 BSliceSearch8x8

Class: Boolean
Description: Enables 8x8 Inter Prediction & Motion Compensation in B Slices. Default value is 1 (enabled).

4.3.14 BSliceSearch8x4

Class: Boolean
Description: Enables 8x4 Inter Prediction & Motion Compensation in B Slices. Default value is 1 (enabled).

4.3.15 BSliceSearch4x8

Class: Boolean
Description: Enables 4x8 Inter Prediction & Motion Compensation in B Slices. Default value is 1 (enabled).

4.3.16 BSliceSearch4x4

Class: Boolean
Description: Enables 8x4 Inter Prediction & Motion Compensation in B Slices. Default value is 1 (enabled).

4.3.17 BiPredMotionEstimation

Class: Boolean
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 value is 0 (disabled).

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

4.3.18 BiPredMERefinements

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

4.3.19 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 value is 8.

4.3.20 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

4.3.21 BiPredSearch16x16

Class: Boolean
Description: Enables bipredictive motion estimation for 16x16 partitions in B Slices. Default value is 1 (enabled).

4.3.22 BiPredSearch16x8

Class: Boolean
Description: Enables bipredictive motion estimation for 16x8 partitions in B Slices. Default value is 0 (disabled).

4.3.23 BiPredSearch8x16

Class: Boolean
Description: Enables bipredictive motion estimation for 8x16 partitions in B Slices. Default value is 0 (disabled).

4.3.24 BiPredSearch8x8

Class: Boolean
Description: Enables bipredictive motion estimation for 8x8 partitions in B Slices. Default value is 0 (disabled).

4.3.25 DisableIntraInInter

Class: Boolean
Description: Disable all intra prediction modes for Inter (P or B) slices. Default value is 0 (Intra prediction modes enabled in all slice types).

4.3.26 IntraDisableInterOnly

Class: Boolean
Description: Disable Intra prediction modes (in sections **Error! Reference source not found.** through **Error! Reference source not found.**) for Inter (P or B) slices. Default value is 0.

4.3.27 DisableIntra4x4

Class: Boolean
Description: Disables all intra 4x4 modes. Default value is 0 (enabled).

4.3.28 DisableIntra16x16

Class: Boolean
Description: Disables all intra 16x16 modes. Default value is 0 (enabled).

4.3.29 DisableIntraInInter

Class: Boolean
Description: Disable Intra prediction modes (in sections 4.3.10 through 4.3.34) for Inter (P or B) slices. Default value is 0 (Intra prediction modes enabled in all slice types).

4.3.30 Intra4x4ParDisable

Class: Boolean
Description: Disables I4x4 Vertical and Horizontal prediction modes. Default value is 0 (enabled).

4.3.31 Intra4x4DiagDisable

Class: Boolean
Description: Disables I4x4 Diagonal Down-Left and Diagonal Down-Right prediction modes. Default value is 0 (enabled).

4.3.32 Intra4x4DirDisable

Class: Boolean
Description: Disables I4x4 Vertical Right, Vertical Left, Horizontal Down, and Horizontal Up prediction modes . Default value is 0 (enabled).

4.3.33 Intra16x16ParDisable

Class: Boolean
Description: Disables I16x16 Vertical and Horizontal prediction modes. Default value is 0 (enabled).

4.3.34 Intra16x16PlaneDisable

Class: Boolean
Description: Disables I16x16 plane prediction mode. Default value is 0 (enabled).

4.3.35 ChromaIntraDisable

Class: Boolean
Description: Disable all Intra Chroma prediction modes except DC. Default value is 0 (enabled).

4.3.36 FastCrIntraDecision

Class: Boolean
Description: Perform a separate intra chroma mode decision prior to determining final coding mode. Can provide significant encoding speedup. Default value is 1 (enabled).

4.3.37 EnableIPCM

Class: Numeric (Integer)

Description: Enables usage of I_PCM macroblock mode.

Options:	
0	Disabled (default)
1	Enabled
2	Enabled as needed, i.e. based on performance of other inter/intra modes (only available for RDOptimization = 9)

4.4 MVC coding parameters (main cofiguration file)

4.4.1 NumberOfViews

Class: Numeric (Integer)
Description: Set the number of views/layers to encoder for MVC, up to a maxumum of 2. Default value is 1 (single view/layer).

4.4.2 View1ConfigFile

Class: Text
Description:Specifies the configuration file name from which the options for the second view are read.

4.5 MVC coding parameters (View 1 cofiguration file)

4.5.1 MVCEnableInterViewFlag

Class: Boolean
Description: Enable inter layer/view prediction for MVC. Default is 1 (enabled)

4.5.2 MVCInterViewReorder

Class: Numeric (Integer)
Description: Enable reference reordering for interview prediction.

Options:	
0	Disabled (default)
1	Enabled (forced)

4.5.3 QPOffset

Class: Numeric (Integer)
Description: Sets the quantization parameter (QP) offset that will be used for coding the enhancement layer for MVC. Value can be either negative or positive (-51..51). Default value is 0 (no offset).

4.5.4 MVCInterViewForceB

Class: Boolean
Description: Force B coded pictures for the enhancement layer coding of MVC. Default is 0 (disabled).

4.5.5 MVCInterViewReorder

Class: Boolean
Description: Reorder References according to interview pictures. Default is 0 (disabled).

4.5.6 SepViewInterSearch

Class: Boolean
Description: If set, allows different InterSearch modes to be set for each view. Default is 0 (disabled).

4.5.7 NoResidueRDO

Class: Boolean
Description: Test no residue case for View 1 during RDO. Default is 1 (enabled).

4.5.8 MVCFlipViews

Class: Boolean
Description: Reverses the order of the views in the bitstream (view 1 has VOIdx 0 and view 1 has VOIdx 0)

4.5.9 Other View 1 parameters

In addition the following main configuration file parameters are available in the View 1 configuration file with the same meaning:

- InputFile
- ReconFile
- SearchRange
- DisableSubpelME
- DisableIntraInInter
- PList0Refs
- BList0Refs
- BList1Refs
- LambdaMultiplier
- MELambdaMultiplier
- PSliceSkip
- PSliceSearch16x16
- PSliceSearch16x8
- PSliceSearch8x16
- PSliceSearch8x8
- PSliceSearch8x4
- PSliceSearch4x8
- PSliceSearch4x4
- BSliceDirect
- BSliceSearch16x16
- BSliceSearch16x8
- BSliceSearch8x16
- BSliceSearch8x8
- BSliceSearch8x4
- BSliceSearch4x8
- BSliceSearch4x4
- BiPredMESearchRange
- MDReference
- DFDisableRefISlice
- DFDisableNRefISlice
- DFDisableRefPSlice
- DFDisableNRefPSlice

- DFDisableRefBSlice
- DFDisableNRefBSlice
- DFDisableRefSPSlice
- DFDisableNRefSPSlice
- DFDisableRefSISlice
- DFDisableNRefSISlice
- DFAlphaRefISlice
- DFAlphaNRefISlice
- DFAlphaRefPSlice
- DFAlphaNRefPSlice
- DFAlphaRefBSlice
- DFAlphaNRefBSlice
- DFAlphaRefSPSlice
- DFAlphaNRefSPSlice
- DFAlphaRefSISlice
- DFAlphaNRefSISlice
- DFBetaRefISlice
- DFBetaNRefISlice
- DFBetaRefPSlice
- DFBetaNRefPSlice
- DFBetaRefBSlice
- DFBetaNRefBSlice
- DFBetaRefSPSlice
- DFBetaNRefSPSlice
- DFBetaRefSISlice
- DFBetaNRefSISlice
- SearchMode
- EPZSTemporal
- EnableEPZSScalers
- EPZSMinThresScale
- EPZSMaxThresScale
- EPZSMedThresScale
- EPZSSubPelThresScale

4.6 Deblocking Filter Control

Parameters to control in-loop deblocking filter behavior.

4.6.1 DFParametersFlag

Class: Boolean
Description: Sets the value of the syntax element **deblocking_filter_control_present_flag**. Default value is 0 (disabled).

Note: Although currently the encoder supports multiple PPS this parameter still sets the same deblocking filter parameters for all coded pictures.

4.6.2 DFDisableRefISlice

Class: Numeric (Integer)
Description: Sets the value of the syntax element **disable_deblocking_filter_idc** for I slices belonging to a reference picture**.** Requires DFParametersFlag 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

4.6.3 DFAlphaRefISlice

Class: Numeric (Integer)
Description: Sets the value of the syntax element **slice_alpha_c0_offset_div2** for I slices belonging to a reference picture. Requires DFParametersFlag to be set. Allowable values are in the range {-6, -5, ... 0, +1, .. +6}**.** Default value is 0.

4.6.4 DFBetaRefISlice

Class: Numeric (Integer)
Description: Sets the value of the syntax element **slice_beta_offset_div2** for I slices belonging to a reference picture**.** Requires DFParametersFlag to be set. Allowable values are in the range {-6, -5, ... 0, +1, .. +6}. Default value is 0.

4.6.5 DFDisableNRefISlice

Class: Numeric (Integer)
Description: Sets the value of the syntax element **disable_deblocking_filter_idc** for I slices belonging to a non reference picture**.** Requires DFParametersFlag 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

4.6.6 DFAlphaNRefISlice

Class: Numeric (Integer)
Description: Sets the value of the syntax element **slice_alpha_c0_offset_div2** for I slices belonging to a non reference picture**.** Requires DFParametersFlag to be set. Allowable values are in the range {-6, -5, ... 0, +1, .. +6}**.** Default value is 0.

4.6.7 DFDisableRefISlice

Class: Numeric (Integer)
Description: Sets the value of the syntax element **disable_deblocking_filter_idc** for I slices belonging to a reference picture**.** Requires DFParametersFlag 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

4.6.8 DFAlphaRefPSlice

Class: Numeric (Integer)
Description: Sets the value of the syntax element **slice_alpha_c0_offset_div2** for P slices belonging to a reference picture. Requires DFParametersFlag to be set. Allowable values are in the range {-6, -5, ... 0, +1, .. +6}**.** Default value is 0.

4.6.9 DFBetaRefPSlice

Class: Numeric (Integer)
Description: Sets the value of the syntax element **slice_beta_offset_div2** for P slices belonging to a reference picture**.** Requires DFParametersFlag to be set. Allowable values are in the range {-6, -5, ... 0, +1, .. +6}. Default value is 0.

4.6.10 DFDisableNRefPSlice

Class: Numeric (Integer)
Description: Sets the value of the syntax element **disable_deblocking_filter_idc** for P slices belonging to a non reference picture**.** Requires DFParametersFlag 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

4.6.11 DFAlphaNRefPSlice

Class: Numeric (Integer)
Description: Sets the value of the syntax element **slice_alpha_c0_offset_div2** for P slices belonging to a non reference picture**.** Requires DFParametersFlag to be set. Allowable values are in the range {-6, -5, ... 0, +1, .. +6}**.** Default value is 0.

4.6.12 DFDisableRefBSlice

Class: Numeric (Integer)
Description: Sets the value of the syntax element **disable_deblocking_filter_idc** for B slices belonging to a reference picture**.** Requires DFParametersFlag 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

4.6.13 DFAlphaRefBSlice

Class: Numeric (Integer)

Description: Sets the value of the syntax element **slice_alpha_c0_offset_div2** for B slices belonging to a reference picture. Requires DFParametersFlag to be set. Allowable values are in the range {-6, -5, ... 0, +1, .. +6}**.** Default value is 0.

4.6.14 DFBetaRefBSlice

Class: Numeric (Integer)
Description: Sets the value of the syntax element **slice_beta_offset_div2** for B slices belonging to a reference picture**.** Requires DFParametersFlag to be set. Allowable values are in the range {-6, -5, ... 0, +1, .. +6}. Default value is 0.

4.6.15 DFDisableNRefBSlice

Class: Numeric (Integer)
Description: Sets the value of the syntax element **disable_deblocking_filter_idc** for B slices belonging to a non reference picture**.** Requires DFParametersFlag 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

4.6.16 DFAlphaNRefBSlice

Class: Numeric (Integer)
Description: Sets the value of the syntax element **slice_alpha_c0_offset_div2** for B slices belonging to a non reference picture**.** Requires DFParametersFlag to be set. Allowable values are in the range {-6, -5, ... 0, +1, .. +6}**.** Default value is 0.

4.7 Weighted Prediction Parameters

The following parameters enable weighted prediction.

4.7.1 WeightedPrediction

Class: Boolean
Description: Sets the value of the syntax element **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 value is 0 (disabled).

Note: This parameter is ignored when RDPictureDecision is used.

4.7.2 WeightedBiprediction

Class: Numeric (Integer)
Description: Sets the value of the syntax element **weighted_bipred_idc** and enables weighted prediction for B slices.

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

Note: This parameter is ignored when RDPictureDecision is used.

4.7.3 UseWeightedReferenceME

Class: Boolean
Description: Use weighted references for motion estimation. Default value is 0 (disabled).

4.7.4 WPMethod

Class: Boolean
Description: Use DC based (0: default) or Least Mean Square (LMS) method (1) for weighted prediction.

4.7.5 WPIterMC

Class: Boolean
Description: Use DC Iterative Motion compensated based weighted prediction method. Default value is 0 (disabled)

4.7.6 EnhancedBWeightSupport

Class: Boolean
Description: Use LMS method for B slice weighted prediction. Default value is 0 (disabled).

4.7.7 WPMCPrecision

Class: Numeric (Integer)
Description: Improved Motion Compensation Precision using WP based methods. Clones WP references with slightly modified rounding offsets (Requires RDPictureDecision and GenerateMultiplePPS) :

Options:	
0	Disabled (default).
1	Up to one additional coding pass. Ref0 is 0, ref1 is 0 with a -1 offset
2	Up to two additional coding passes. (1) Ref0 is 0, ref1 is 0 with a -1 offset, (1) Ref0 is 0 with a -1 offset, ref1 is 0

4.7.8 WPMCPrecFullRef

Class: Numeric (Integer)
Description: Increases the number of references in the reference picture lists to account for the lost reference slot when reordering is used during a coding pass in WPMCPrecision for reference replication. The number of references in non-reordered passes stays unchanged. Default value is 0 (keep the same number of references).

4.7.9 WPMCPrecBSlice

Class: Numeric (Integer)
Description: Applies different methods when considering the improved motion compensation precision process in B slices.

Options:	
0	Disable rounding for B slices.
1	Disable rounding for non-reference B slices. Non-reference B slices are evaluated for alternative QPs during RDPictureDecision.(default)
2	Apply rounding on every B slice. This efectively disables the evaluation of

	alternative QPs during RDPictureDecision.

4.8 B Pictures and CodingStructure

4.8.1 BRefPicQPOffset

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

4.8.2 DirectModeType

Class: Boolean
Description: Sets the value of the syntax element **direct_spatial_mv_pred_flag** which controls the direct mode type to be used. 0 means temporal direct, while 1 means spatial direct. Default value is 0 (temporal).

4.8.3 DirectInferenceFlag

Class: Boolean
Description: Sets the value of the syntax element **direct_8x8_inference_flag** in the SPS which affects semantics of Direct Mode. The value is constrained by level restriction in Annex A of H.264/AVC (i.e. for any level above or equal to 3 the parameter needs to be equal to 1), and should be set appropriately even if no B slices are to be used. Default value is 0.

4.8.4 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:

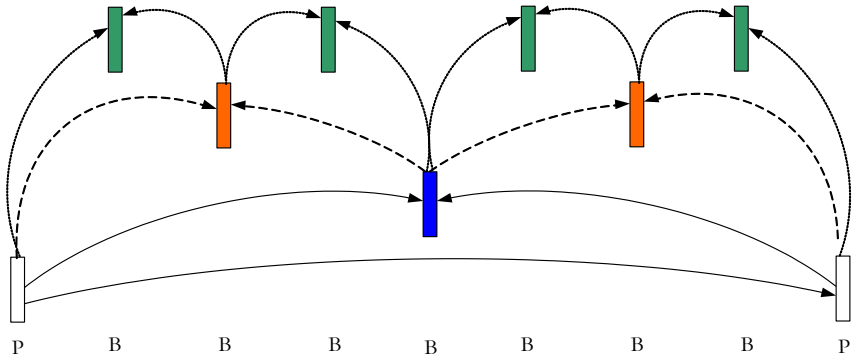

Figure 1. 4 Level Hierarchical structure.

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 NumberBFrames=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):

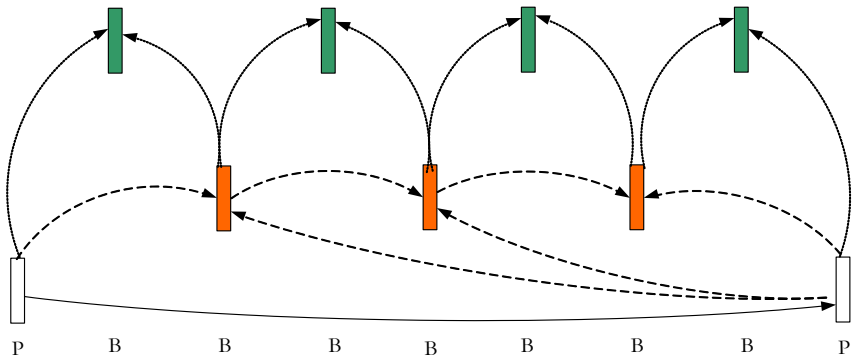

Figure 2. 3 Level Hierarchical structure.

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 NumberBFrames=7 \
            -p HierarchicalCoding=1 -p QPBSlice=26 \
            -p BRefPicQPOffset=-2 \
            -p BList0References=1 -p BList1References=1
```

4.8.5 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

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 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 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 NumberBFrames=7 \
            -p HierarchicalCoding=2
```

4.8.6 HierarchyLevelQPEnable

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

4.8.7 ExplicitSeqCoding

Class: Boolean
Description: Encode video sequence using the Explicit Seq Coding metadata file provided by ExplicitSeqFile. Default value is 0 (disabled).

Note: This option is still experimental.

4.8.8 ExplicitSeqFile

Class: Text
Description: Name of Explicit Seq Coding metadata file used when ExplicitSeqCoding is enabled.

4.8.9 ReferenceReorder

Class: Boolean
Description: Performs reference picture list reordering for P coded frames based on POC values. This essentially places references according to temporal correlation instead of coding order. Default value is 0 (disabled).

Note: ReferenceReorder is not supported for interlace coding modes.

Example:
In example 1 of 4.7.5 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"
```

4.8.10 PocMemoryManagement

Class: Boolean
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. Default value is 0 (disabled)

Note: PocMemoryManagement is not supported for interlace coding modes.

Example:
Lets assume that for the first example in 4.7.5, 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 NumberBFrames=7 \
     -p HierarchicalCoding=3 -p ReferenceReorder=1 \
     -p PocMemoryManagement=1 \
     -p ExplicitHierarchyFormat="B4r24B2r24B1e26B3e26P6r24B5e26I7e40"
```

4.9 Error Resiliency and Slice control

4.9.1 SliceMode

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

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

4.9.2 SliceArgument

Class: Numeric (Integer)
Description: For SliceMode equal to 1: number of macroblocks per slice. For SliceMode equal to two: number of bytes per slice. Default value is 0 (invalid).

4.9.3 num_slice_groups_minus1

Class: Numeric (Integer)
Description: Number of slice groups decremented by 1 (i.e.0 == one slice group, 1 == two slice groups, etc.). Default value is 0.

4.9.4 slice_group_map_type

Class: Numeric (Integer)
Description: Specifies slice group map type if num_slice_groups_minus1 is lager than 0 (sets the value of the syntax element **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

*Note:*For slice_group_map_type equal to 0, 2 or 6, additional parameters are read from a file specified in the SliceGroupConfigFileName parameter.

4.9.5 slice_group_change_direction_flag

Class: Numeric (Integer)
Description: Sets the value of the syntax element **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

4.9.6 slice_group_change_rate_minus1

Class: Numeric (Integer)
Description: Sets the value of the syntax element **slice_group_change_rate_minus1**. Default value is 0.

4.9.7 SliceGroupConfigFileName

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

For slice_group_map_type equal to 0 the file consist of one **run_length_minus1** syntax elementvalue per line. For slice_group_map_type equal to 2 the file contains in rows with odd line numbers the values for **top_left** syntax elements and rows with even line numbers the values **bottom_right** syntax elements. For slice_group_map_type equal to 6, each line contains a value of a **slice_group_id** syntax element.

4.9.8 UseRedundantPicture

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

4.9.9 NumRedundantHierarchy

Class: Numeric (Integer)
Description: Hierarchy mode of redundant pictures. Allowed values are in the range of 0 to 4.

4.9.10 PrimaryGOPLength

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

4.9.11 NumRefPrimary

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

4.10 SP coding support

4.10.1 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

4.10.2 QPSPPicture

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

4.10.3 QPSP2Picture

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

4.10.4 SI_Frames

Class: Boolean
Description: Eanbels SI frame encoding. Default value is 0 (disabled).

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

4.10.5 SP_output

Class: Boolean
Description: Controls whether coefficients will be output to encode switching SP frames. Default value is 0 (disabled).

4.10.6 SP_output_name

Class: Text
Description: Filename for SP coefficients output.

4.10.7 SP2_Frames

Class: Boolean
Description: Enables switching SP frame encoding. Default value is 0 (diabled).

4.10.8 SP2_input_name1

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

4.10.9 SP2_input_name2

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

4.11 Output Control/Entropy Coding, NALs

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

4.11.1 SymbolMode

Class: Numeric (Integer)
Description: Entropy Coding method.

Options:	
0	CAVLC (default)

1	CABAC

4.11.2 ContextInitMethod

Class: Numeric (Integer)
Description: CABAC context initialization method

Options:	
0	Fixed (default)
1	Adaptive

4.11.3 FixedModelNumber

Class: Numeric (Integer)
Description: CABAC initialization model number for fixed initialization (ContextInitMethod equal to 0). Allowed model numbers are 0,1 and 2. Default value is 0.

4.11.4 OutFileMode

Class: Numeric (Integer)
Description: Output File mode.

Options:	
0	Annex B Byte Stream format (default)
1	RTP packets

4.11.5 PartitionMode

Class: Numeric (Integer)
Description: Enables Data Partitioning.

Options:	
0	No Data Partitioning (default)
1	Three partitions per slice

4.12 Interlace Format Handling

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

4.12.1 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.

4.12.2 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.
3	Like option 1, but all macroblocks coded as frame macroblocks. Mainly useful for testing purposes

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
```

4.12.3 IntraBottom

Class: Boolean
Description:Forces both fields of a complementary field pair to be coded using I slices at intra periods. If disabled, the bottom field will be coded using P slices. Default value is 0 (disabled).

Note: Has only effect, if PicInterlace is not equal to zero.

4.13 Non Normative Encoder Decisions

4.13.1 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.

4.13.2 I16RDOpt

Class: Boolean
Description: Enable high complexity rate distortion optimization for Intra 16x16 modes. Default value is 0 (disabled).

4.13.3 SubMBCodingState

Class: Integer
Description: Parameter provides performance/speed tradeoffs for submacroblock mode decision when high complexity RDO is enabled. Essentially avoids storing/restoring state values resulting in somewhat suboptimal RD decision.

Options:	
0	Disable store/restore states for sub-macroblock mode decision (low complexity)
1	Reset coding state for every mode at the macroblock level (medium complexity)
2	Store/Reset coding state based on optimal decisions (high complexity/default)

4.13.4 ForceTrueRateRDO

Class: Integer
Description: Mode bias for skip or intra modes during high complexity mode decision. Can provide benefits in some scenarios.

Options:	
0	Penalize skip modes by 1 bit if mode cost is 0 (default)
1	No penalty
2	Penalize intra modes by 1 bit.

4.13.5 RDPictureDecision

Class: Boolean
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 value 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.

4.13.6 RDPSliceBTest

Class: Boolean
Description: Perform Slice level RD decision between P and B slices.

4.13.7 RDPictureMaxPassISlice

Class: Numeric (Integer)
Description: Set maximum number of coding passes for I slices. The valid range for this parameter is from 1 (default) through 3.

4.13.8 RDPictureMaxPassPSlice

Class: Numeric (Integer)
Description: Set the maximum number of coding passes for P slices. The valid range for this parameter is from 1 through 6. The default value is 2.

4.13.9 RDPictureMaxPassBSlice

Class: Numeric (Integer)
Description: Set the maximum number of coding passes for B slices. The valid range for this parameter is from 1 through 6. The default value is 3.

4.13.10 RDPictureFrameQPPSlice

Class: Boolean
Description: Perform an additional frame level QP check (QP+/-1) for P slices. The default value is 0 (disabled).

4.13.11 RDPictureFrameQPBSlice

Class: Boolean
Description: Perform an additional frame level QP check (QP+/-1) for B slices. , The default value is 0 (disabled).

4.13.12 RDPictureDeblocking

Class: Boolean
Description: Perform another coding pass to check deblocking performance, The default value is 0 (disabled).

4.13.13 RDPictureDirectMode

Class: Boolean
Description: Perform another coding pass for B slices that compares performance of the two direct modes. The default value is 0 (disabled).

4.13.14 DistortionSSIM

Class: Boolean
Description: Enable SSIM distortion computation for video analysis. Default value is 0 (disabled).

4.13.15 DistortionSSIM

Class: Boolean
Description: Enable SSIM distortion computation for video analysis. Default value is 0 (disabled).

4.13.16 DistortionYUVtoRGB

Class: Boolean
Description: Compute Distortion in both YUV and RGB color spaces. Default value is 0 (disabled).

4.13.17 RDPictureIntra

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

4.13.18 RDPSliceWeightOnly

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

4.13.19 RDBSliceWeightOnly

Class: Boolean
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 value is 0 (disabled).

4.14 Lambda parameters for Lagrangian based optimization

4.14.1 CtxAdptLagrangeMult

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

4.14.2 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 value 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)

4.14.3 UpdateLambdaChromaME

Class: Boolean
Description: Update Lambda for motion estimation to account for chroma consideration during this process. Default value is 0 (disabled).

4.14.4 FixedLambdaIslice

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

4.14.5 FixedLambdaPslice

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

4.14.6 FixedLambdaBslice

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

4.14.7 FixedLambdaRefBslice

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

4.14.8 FixedLambdaSPslice

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

4.14.9 FixedLambdaSIslice

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

4.14.10 LambdaWeightIslice

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

4.14.11 LambdaWeightPslice

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

4.14.12 LambdaWeightBslice

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

4.14.13 LambdaWeightRefBslice

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

4.14.14 LambdaWeightSPslice

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

4.14.15 LambdaWeightSIslice

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

4.14.16 OffsetMatrixPresentFlag

Class: Boolean
Description: Enable explicit Quantization offset support. Default value is 0 (disabled).

4.14.17 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"
```

4.14.18 AdaptiveRounding

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

4.14.19 AdaptRoundingFixed

Class: Numeric (Integer)
Description: Consider adaptive rounding separately for different quantization parameters.

Options:	
0	Separate QPs
1	Joint (default)

4.14.20 AdaptRndPeriod

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

4.14.21 AdaptRndChroma

Class: Numeric (Integer)
Description: Performs adaptive rounding for chroma. If disabled, only luma is considered. Default value is 0 (disabled).

4.14.22 AdaptRndWFactorIRef

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

4.14.23 AdaptRndWFactorPRef

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

4.14.24 AdaptRndWFactorBRef

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

4.14.25 AdaptRndWFactorINRef

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

4.14.26 AdaptRndWFactorPNRef

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

4.14.27 AdaptRndWFactorBNRef

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

4.14.28 AdaptRndCrWFactorIRef

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

4.14.29 AdaptRndCrWFactorPRef

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

4.14.30 AdaptRndCrWFactorBRef

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

4.14.31 AdaptRndCrWFactorINRef

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

4.14.32 AdaptRndCrWFactorPNRef

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

4.14.33 AdaptRndCrWFactorBNRef

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

4.15 Error Resilient Optimized Rate Distortion Optimization

4.15.1 LossRateA

Class: Numeric (Integer)

Description: Expected packet loss rate of the channel for the first partition. Only valid if RDOptimization is set equal to 3. Default value is 0.

4.15.2 LossRateB

Class: Numeric (Integer)
Description: Expected packet loss rate of the channel for the second partition. Only valid if RDOptimization is set equal to 3. Default value is 0.

4.15.3 LossRateC

Class: Numeric (Integer)
Description: Expected packet loss rate of the channel for the third partition. Only valid if RDOptimization is set equal to 3. Default value is 0.

4.15.4 NumberOfDecoders

Class: Numeric (Integer)
Description: Numbers of decoders used to simulate the channel. Only valid if RDOptimization is set equal to 3. Default value is 0.

4.15.5 RestrictRefFrames

Class: Boolean
Description: Does not allow reference to areas that have been intra updated in a later frame. Default value is 0 (disabled).

4.15.6 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)

4.15.7 DisableThresholding

Class: Boolean
Description: Disable Thresholding of Transform Coefficients. Default value is 0 (enabled).

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

4.15.8 BiasSkipRDO

Class: Boolean
Description: Force a negative bias for skip/direct modes during the mode decision process. Default value is 0 (disabled)

4.15.9 ForceTrueRateRDO

Class: Numeric (Integer)
Description: Affects the consideration of the bitrate during the rate distortion optimized mode decision process.

Options:	
0	Bound rate to a minimum of ½ bit. Essentially affects skip/direct modes.
1	Consider true rate during the mode decision process (unbiased)
2	Negatively bias intra modes

4.15.10 DisableBSkipRDO

Class: Boolean
Description: Disable B Skip Mode consideration from the RDO based mode decision. Default value is 0 (enabled)

4.15.11 SkipIntraInInterSlices

Class: Numeric (Integer)
Description: Avoids testing Intra modes in Inter slices if best mode is P_SKIP or B_SKIP. Default value is 0 (disabled)

4.15.12 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.

4.16 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

4.16.1 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)

4.16.2 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)

4.16.3 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)

4.16.4 EPZSTemporal

Class: Boolean
Description: Enables usage of Temporal Predictors through the consideration of co-located partitions (i.e. similar to temporal direct). Default value is 1 (enabled).

4.16.5 EPZSSpatialMem

Class: Boolean
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 value is 1 (enabled).

4.16.6 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 value is 0.

4.16.7 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 value is 1.

4.16.8 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 value is 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

4.16.9 EPZSSubPelME

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

4.16.10 EPZSSubPelMEBipred

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

4.16.11 EPZSSubPelThresScale

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

4.16.12 EPZSSubPelGrid

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

4.17 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.

4.17.1 UMHexDSR

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

4.17.2 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 value is 3.

4.18 EarlySkipEnable

Class: Boolean
Description: Early skip mode detection when RDOptimization is set to 2. Default value is 0 disabled).

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

4.19 SelectiveIntraEnable

Class: Boolean
Description: Enables Selective Intra mode decision when RDOptimization is set to 2. Default value is 0 (disabled).

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

4.20 Rate Control & HRD support

Parameters for rate control support.

4.20.1 RateControlEnable

Class: Boolean
Description: Enable simple Rate Control support. Default value is 0 (disabled).

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: Existing algorithms should be used as a reference only.

4.20.2 RCUpdateMode

Class: Numeric (Integer)
Description: Specifies the Rate Control algorithm used, when RateControlEnable is enabled.

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

4.20.3 Bitrate

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

4.20.4 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 value is 0.

4.20.5 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. If 0, then Basic Unit is equal to the number of macroblocks in a slice. Default value is 0.

4.20.6 ChannelType

Class: Numeric (Integer)
Description: Type of Channel.

Options:	
0	Constant channel (default)
1	Time varying channel

4.20.7 NumberofLeakyBuckets

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

4.20.8 LeakyBucketRateFile

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

4.20.9 LeakyBucketParamFile

Class: Text
Description: File where encoder stores leakybucketparams.

4.20.10 RCISliceBitRatio

Class: Numeric (Double)
Description: Sets the bitrate target ratio between I and P coded slices when RCUpdateMode is set to 3. Default value is 1.00.

4.20.11 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 value is 0.5.

4.20.12 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 value is 0.25.

4.20.13 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 value is 0.25.

4.20.14 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 value is 0.25.

4.20.15 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 value is 0.25.

4.20.16 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 value is 3.8.

4.20.17 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 value is 0.45.

4.20.18 RCMinQPPSlice

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

4.20.19 RCMaxQPPSlice

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

4.20.20 RCMinQPISlice

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

4.20.21 RCMaxQPISlice

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

4.20.22 RCMinQPBSlice

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

4.20.23 RCMaxQPBSlice

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

4.20.24 RCMinQPSPSlice

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

4.20.25 RCMaxQPSPSlice

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

4.20.26 RCMinQPSISlice

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

4.20.27 RCMaxQPSISlice

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

4.21 Rate Distortion Optimized Quantization Parametetrs

4.21.1 UseRDOQuant

Class: Boolean
Description: Enable Rate Distortion Optimized Quantization. Default value is 0 (disabled).

4.21.2 RDOQ_DC

Class: Boolean
Description: Enable Rate Distortion Optimized Quantization for DC components. Default value is 0 (disabled).

4.21.3 RDOQ_CR

Class: Boolean
Description: Enable Rate Distortion Optimized Quantization for Chroma components. Default value is 0 (disabled).

4.21.4 RDOQ_DC_CR

Class: Boolean
Description: Enable Rate Distortion Optimized Quantization for Chroma DC components Default value is 0 (disabled).

4.21.5 RDOQ_QP_Num

Class: Numeric (Integer)
Description: Number of QP values tested in RDO_Q (I/P/B slice). Allowable values are from 1 (default) to 9.

4.21.6 RDOQ_CP_Mode

Class: Boolean
Description: Fast mode decision for RDOQ by copying the mode of the first QP tested for all other QP values. Default value is 0 (disabled).

4.21.7 RDOQ_CP_MV

Class: Boolean
Description: Fast motion estimation for RDOQ by copying the best motion vectors for each mode of the first QP tested for all other QP values. Default value is 0 (disabled).

4.21.8 RDOQ_Fast

Class: Boolean
Description: Fast RDOQ decision method for multiple QPs based on CBP behavior.

4.22 SEI Parameters

4.22.1 GenerateSEIMessage

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

4.22.2 SEIMessageText

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

4.22.3 ToneMappingSEIPresentFlag

Class: Boolean
Description: Enable Tone mapping SEI. Default value is 0 (disabled).

4.22.4 ToneMappingFile

Class: Text
Description: Tone mapping parameter file.

4.23 VUI Parameters

VUI Parameters specify directly the values of the corresponding VUI syntax elements.

4.23.1 VUI_aspect_ratio_info_present_flag

Class: Boolean
Description: If enabled specifies that aspect_ratio_idc is present. Default value is 0 (disabled).

4.23.2 VUI_aspect_ratio_idc

Class: Numeric (Integer)
Description: Specifies the value of the sample aspect ratio of the luma samples Default value 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

4.23.3 VUI_sar_width

Class: Numeric (Integer)

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

4.23.4 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).

4.23.5 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).

4.23.6 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.

4.23.7 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).

4.23.8 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

4.23.9 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).

4.23.10 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.

4.23.11 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.

4.23.12 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.

4.23.13 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).

4.23.14 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.

4.23.15 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.

4.23.16 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.

4.23.17 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.

4.23.18 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.

4.23.19 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.

4.23.20 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.

4.23.21 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).

4.23.22 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).

4.23.23 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.

4.23.24 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.

4.23.25 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.

4.23.26 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).

4.23.27 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).

4.23.28 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.

4.23.29 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.

4.23.30 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.

4.23.31 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.

4.23.32 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).

4.24 Other settings

4.24.1 NumFramesInELayerSubSeq

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

4.24.2 SparePictureOption

Class: Numeric (Integer)
Description: Creation of spare reference pictures

Options:	
0	No spare picture info (default)
1	Spare picture available

4.24.3 SparePictureDetectionThr

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

4.24.4 SparePicturePercentageThr

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

4.25 FRExt profile parameters

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

4.25.1 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)

4.25.2 SeparateColourPlane

Class: Boolean
Description: Enables use of separate colour plane coding. Default value is 0 (disabled)

4.25.3 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

4.25.4 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
```

4.25.5 ScalingListPresentFlag0

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

4.25.6 ScalingListPresentFlag1

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

4.25.7 ScalingListPresentFlag2

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

4.25.8 ScalingListPresentFlag3

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

4.25.9 ScalingListPresentFlag4

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

4.25.10 ScalingListPresentFlag5

Class: Numeric (Integer)
Description: Select scaling matrix for Intrer4x4 Chroma 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

4.25.11 ScalingListPresentFlag6

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

4.25.12 ScalingListPresentFlag7

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

4.25.13 LosslessCoding

Class: Boolean
Description: Enable lossless coding when qpprime_y is zero. Default value is 0 (disabled).

Note: Better explanation is needed for this parameter

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.
ENABLE_FIELD_CTX	:	Enables field context types for CABAC. Required for interlaced coding. If coding only progressive content, disabling flag can provide some encoding speed up.
ENABLE_HIGH444_CTX	:	Enables High 444 context types for CABAC. If disabled, results in speedup of non High444 profile encodings.
DEBUG_BITDEPTH	:	Ensures that > 8 bit content have no values that would result in out of range results
ALLOW_GRAYSCALE	:	Allows encoding in grayscale of full colored image data
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
*_LUMA_COEFF_COST_*	:	8x8 block Luma coefficient threshold cost.
*_CHROMA_COEFF_COST_*	:	Chroma coefficient threshold cost.
*_LUMA_MB_COEFF_COST_*	:	Macroblock Luma coefficient threshold cost.
*_LUMA_8x8_COEFF_COST_*	:	Threshold for P8x8 sub-macroblocks.
JM_INT_DIVIDE	:	Perform integer divides (shifts) during ME
JM_MEM_DISTORTION	:	Use table lookup for distortion computations
USE_RND_COST	:	Use rounding for distortion computation instead of truncation
JM_INT_DIVIDE	:	Utilize integer divides for distortion computation
JM_MEM_DISTORTION	:	Use table lookup method for distortion computation
JCOST_CALC_SCALEUP	:	Compute RDcost as either J=(D<<λ_bits)+λ*R or J=D+ ((λ*R+(1<<(λ_bits – 1))>> λ_bits)
INTRA_RDCOSTCALC_EARLY_TERMINATE	:	Perform early rdcost termination for intra modes
INTRA_RDCOSTCALC_NNZ	:	Enable to recover block's non-zero coefficients after rd cost computation
JCOST_OVERFLOWCHECK	:	Enable check for distortion cost overflows
JM_PARALLEL_DEBLOCK	:	Enable parallel deblocking (OpenMP based)
MVC_EXTENSION_ENABLE	:	Enables MVC extension support (Stereo High Profile)

5.2 configfile.h

DEFAULTCONFIGFILENAME : Sets default encoder configuration file.

6. EXPLICIT SEQUENCE INFORMATION FILE

6. EXPLICIT SEQUENCE INFORMATION FILE

The explicit sequence information file (ExplicitSeqFile) enables the JM encoder to encode a video sequence using arbitrary coding orders and coding types. In this section we will describe how one may use this file for encoding a video sequence. It should be noted that this option is still experimental, will be further extended in the future, and should be used with care.

6.1 File Format

The explicit sequence information file needs to adhere to a strict file format. The file needs to start with the heading "Sequence". Sequence information are then contained within curly brackets/braces. The first sequence level entry represents the number of frames that are present in the file (FrameCount). The number is separated from FrameCount using a colon ":" symbol.

FrameCount is followed by multiple "Frame" entries, each one again contained within curly brackets/braces. Variables can be present in any order and can be repeated within the Frame structure (only the last entry is considered). Frame entries include the following parameters:

Supported Parameter		**Description**
SeqNumber	:	Sequence order of frame in input video file. Each frame is required to have a distinct number.
SliceType	:	Slice type to be used when encoding this frame
Reference	:	Set whether this frame will be used as a reference or not
IDRPicture	:	Sets picture as IDR assuming picture is using I slices. Currently ignored.

Example:

```
Sequence
{
  FrameCount : 2
  Frame
  {
     SeqNumber : 0
     SliceType : I
     Reference : 1
  }
  Frame
  {
     SeqNumber : 1
     SliceType : I
     Reference : 1
  }
}
```

7. USING THE JM DECODER MODULE

7. USING THE JM DECODER MODULE

7.1 Decoder Syntax

```
ldecod   [-s] [-h] [-d] [defdec.cfg]
         {[-f curenc1.cfg]...[-f curencN.cfg]}
         [-i bitstream.264] [-o output.yuv] [-r reference.yuv]
         {[-p DecParam1=DecValue1]...[-p DecParamM=DecValueM]}
         [-n] Nframes [-mpr] LValue
```

Options:	
-s	Silent decoding
-h	Prints parameter usage.
[defdec.cfg]	Optional decoder config file containing all decoder information.
-f	Read <curencM.cfg> for resetting selected decoder parameters. Multiple files could be used that set different parameters.
-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 parameter <DecParamM> to <DecValueM>. The entry for <DecParamM> is case insensitive.
-mpr	Set Number of layers (LValue) to decode
-n	Number of frames to be decoded

Examples of usage:

```
ldecod.exe

ldecod.exe -h

ldecod.exe –d 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
```

7.2 Decoder Configuration File Format

The decoder format was recently modified to enable higher flexibility and to basically resemble the format of that of the encoder. The parameters allowed now are as follows:

7.2.1 InputFile

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

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):

```
ldecod.exe -p InputFile="f:\seq\420\176x144\foreman_176x144_30.264"
```

Example 2 (Unix/Linux):

```
ldecod.exe -p InputFile="/vol/seq/420/176x144/foreman_176x144_30.264"
```

7.2.2 OutputFile

Class: Text
Description: Decoder output filename. The output will be a raw video file in either YUV or RGB format.

7.2.3 RefFile

Class: Text
Description: Reference input file for distortion (PSNR) computations.

7.2.4 WriteUV

Class: Boolean
Description: Write 4:2:0 chroma components for monochrome streams

7.2.5 FileFormat

Class: Boolean
Description: Input file format (0=Annex B, 1: RTP packets)

7.2.6 RefOffset

Class: Numeric (Integer)
Description: PSNR computation offset

7.2.7 POCScale

Class: Numeric (Integer)
Description: Poc Scale (1 or 2) to be utilized for PSNR computations

7.2.8 DisplayDecParams

Class: Boolean
Description: Display decoder configuration parameters.

7.2.9 ConcealMode

Class: Numeric (Integer)
Description: Perform Error Concealment

Options:	
0	Disabled (default)
1	Frame Copy method
2	Motion Copy method

7.2.10 RefPOCGap

Class: Numeric (Integer)
Description: Reference POC gap (2: IPP (Default), 4: IbP / IpP)

7.2.11 POCGap

Class: Numeric (Integer)
Description: POC gap (2: IPP /IbP/IpP (Default), 4: IPP with frame skip = 1 etc.)

7.2.12 Silent

Class: Boolean
Description: Perform silent decoding

7.2.13 IntraProfileDeblocking

Class: Boolean
Description: Enable Deblocking filter in intra only profiles (0=disable, 1=filter according to SPS parameters). This method essentially works as a post-process.

7.2.14 DecFrmNum

Class: Numeric (Integer)
Description: Number of frames to be decoded (-n)

7.2.15 DecodeAllLayers

Class: Boolean
Description: Enables decoding of all layers for MVC. This is equivalent to –mpr.

7.3 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 17.1 (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)
-------------------------------------------------------------------------
00000(IDR)         0     0    28  0.0000  0.0000  0.0000  4:2:0    16
00006( P )        12     1    28  0.0000  0.0000  0.0000  4:2:0     0
00004( b )         8     2    28  0.0000  0.0000  0.0000  4:2:0    15
00002( 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 17 (FREXt) decoder, ver 17.1
```

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

Name	***Format***	***Purpose***
Frame	%05d($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	Luma Y PSNR. If value is equal to 0.000 then reference is either not available or is identical to reconstructed.
SnrU	%7.4f	Chroma U PSNR. If value is equal to 0.000 then reference is either not available or is identical to reconstructed.
SnrV	%7.4f	Chroma 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

l

8. HARDCODED DECODER PARAMETERS

8. 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.

8.1 defines.h

DUMP_DPB	:	Dump DPB for debugging purposes
PRINTREFLIST	:	Print reference list information 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.
ENABLE_FIELD_CTX	:	Enables field context types for CABAC. Required for interlaced coding.
ENABLE_HIGH444_CTX	:	Enables High 444 context types for CABAC.
ZEROSNR	:	Definition avoids generation of infinite SNR by always forcing at least one difference sample
ENABLE_OUTPUT_TONEMAPPING	:	Enable tone mapping the output if the tone mapping SEI message is found in the bitstream
DISABLE_ERC	:	Disable some error concealment code for speed purposes
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)
JM_PARALLEL_DEBLOCKING	:	Enable parallel deblocking implementation (OpenMP based)
MVC_EXTENSION_ENABLE	:	Enable MVC decoding support
MAX_NUM_DECSLICES	:	Maximum number of allocated decoded slices that can be decoded in parallel

9. SYSTEM GENERATED REPORTS/OUTPUT

9. SYSTEM GENERATED REPORTS/OUTPUT

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

9.1 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	Luma PSNR for first frame in sequence
SNRU 1	%-5.3f	Chroma U PSNR for first frame in sequence
SNRV 1	%-5.3f	Chroma V PSNR for first frame in sequence
SNRY N	%-5.3f	Luma PSNR for entire sequence
SNRU N	%-5.3f	Chroma U PSNR for entire sequence
SNRV N	%-5.3f	Chroma 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

9.2 StatsFile description (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.