

JMVC 使用手册

目录

1.	JMVC 的获取方法.....	1
2.	JMVC 的运行平台.....	1
2.1.	windows 平台.....	1
2.2.	Linux 平台.....	1
3.	JMVC 组织框架.....	2
4.	JMVC 的使用方法.....	2
4.1.	编码器.....	2
4.1.1.	配置文件.....	3
4.2.	码流整合工具.....	6
4.3.	解码器.....	6
5.	测试序列.....	6

JMVC (Joint Multiview Video Coding)是多视点视频编解码 MVC 的参考软件，由联合视频小组 JVT(Joint Video Team)负责实现和改进更新。

1. JMVC 的获取方法

使用 CVS 软件来获取最新版本的 JMVC 软件。相关配置说明如下：

表. 1. CVS 参数设置

authentication(协议):	pserver
host address(服务器):	garcon.ient.rwth-aachen.de
path(仓库文件夹):	/cvs/jvt
user name(用户名):	jvtuser
password:	jvt.Amd.2

输入配置后，点击获取列表，获得模块列表，从中选择“JMVC”模块。

2. JMVC 的运行平台

JMVC 可在 windows 和 linux 两种平台上运行，相应的工程文件在目录 JMVC\H264Extension\build 中。

2.1. windows 平台

在目录 JMVC/H264Extension/build/windows 中包含一个文件 H264AVCVideoEncDec.sln，该文件即为 JMVC 的工程文件，使用 Microsoft Visual Studio .NET 软件打开。该目录下也包含一个 VC6 的工程文件 H264AVCVideoEncDec.dsw, 不过后期版本没有对该工程文件进行及时更新，可能会出错，所以建议最好不用该版本。

2.2. Linux 平台

JMVC 提供了在 Linux 下使用 gcc 编译器运行的 makefile 文件，在目录

JMVC/H264Extension/build/Linux 下，其运行指令为：

```
cd JMVC/H264AVCEExtension/build/linux
make
```

其运行后的结果相应地存放在主目录的 bin 和 lib 目录下。

3. JMVC 组织框架

JMVC 整个工程包含 4 个库和 6 个可执行的工具。四个库的工程执行后，相应的 .lib 文件存放在主目录的 lib 目录下：

1. H264AVCCommonLibStatic

该库是编码器和解码器的公用库，其中包含了很多类的定义，例如宏块的数据结构，序列参数集，图片参数集，图像数据的存取与访问，以及 DCT 变换、量化、去方块滤波等算法。

2. H264AVCDecoderLibStatic

该库主要提供了解码器所需的类，例如码流解析，熵解码等

3. H264AVCEncoderLibStatic

该库主要提供了编码器所需的类，例如运动估计，模式选择，熵编码等。

4. H264AVCVideoIoLibStatic

该库主要提供视频数据的读写，包括原始视频数据 YUV 文件的读写，以及码流中 NAL 单元数据的读写。

六个工具的工程执行完后，相应的可执行文件 .exe 存放在主目录下的 bin 目录下：

1. H264AVCEncoderLibTestStatic

JMVC 的编码器，用来对单个视点进行编码。

2. H264AVCDecoderLibTestStatic

JMVC 的解码器，用来对 MVC 的视频流进行解码。

3. MVCBitStreamAssembler

JMVC 的码流整合工具，将多个单独的已编码视频流整合到一个 MVC 码流中。

4. MVCBitStreamExtractorStatic

JMVC 的码流提取工具，从 MVC 码流中提取出单个视点的码流信息。

5. PSNRStatic

PSNR 测试工具，用来测试两个原始视频数据或的 PSNR。

6. DownConvertStatic

重采样工具，用来对视频序列进行时间/空间的重采样。

4. JMVC 的使用方法

本章主要介绍 JMVC 在 windows 下的使用方法。JMVC 的使用过程如下所示：

首先使用编码器 **H264AVCEncoderLibTestStatic** 对多视点视频原始序列（YUV 文件，以 8 个视点为例）中的每个序列按照参考关系，分别依次进行编码压缩，得到 8 个 “.264” 文件；将这八个 264 文件利用码流整合工具 **MVCBitStreamAssembler** 进行合并，合并成一个 “.264” 文件；然后使用解码器 **H264AVCDecoderLibTestStatic** 对整合后的 264 文件进行解码，解码后得到 8 个 YUV 原始视频。

4.1. 编码器

JMVC 的编码器可以对用户指定的单个视点进行编码，但是在预测结构中该视点所参考的视点必须已编码，该视点才能够利用该编码器进行编码，否则会出错，所以一般按照**配置**

文件中所写的视点顺序 ViewOrder 来对多个视点依次进行编码。

其运行指令为：

H264AVCEncoderLibTestStatic.exe -vf <file.cfg> <view_id>

其中：

-vf：运行编码的选项参数，这个是从 JSVM 中继承过来的，对于 JMVC 来说，只用到 vf 选项，表明后面为输入的配置文件。

<file.cfg>：配置文件，文件名可为 xxx.cfg。该配置文件用来方便用户对相关的编码信息进行定义，具体介绍见 4.1.1。

<view_id >：要进行编码的视频的视点号。

编码器输出一个 264 的文件，其文件名为配置文件中定义的输出文件名 OutputFile_<view_id>.264。

4.1.1. 配置文件

编码器的主要输入为配置文件，用户在配置文件中对相关信息进行设置，包括输入和输出视频流的信息，编码相关算法，以及 MVC 的预测结构等。具体配置信息和相应格式如下表 2 所示。

在该配置文件中，常用到的参数主要包括如下几个：（表 2 中加粗）

- 1. InputFile：输入视频序列的文件名。多视点视频是由多个视频组成视频序列，该视频序列中每个文件的文件名为“视频序列名_视点号.yuv”，例如“golf1_1.yuv”
- 2. OutputFile：编码器输出视频的文件名
- 3. SourceWidth和SourceHeight：输入视频的大小信息，由于输入为yuv格式的视频，JMVC 不提供分辨率检测的功能，所以需要用户指定。
- 4. FrameRate：输入视频的帧率。
- 5. FrameToBeEncoded：进行编码的帧数。
- 6. BasisQP：初始 QP 值。
- 7. GOPSize：GOP 的大小，常用值有 8、12、15 等。
- 8. NumViewsMinusOne：视频序列中的视点数
- 9. ViewOrder：视频序列中各视点之间的编码顺序
- 10. 表中绿色部分：用来描述预测结构信息，其中定义了各个视点间的参考关系。

表.2. 、编码器配置文件

# JMVC Configuration File in MVC mode		
#===== GENERAL =====		
InputFile	input	# input file
OutputFile	stream	# bitstream file
ReconFile	rec	# reconstructed file
MotionFile	motion	# motion information file
SourceWidth	320	# input frame width
SourceHeight	240	# input frame height
FrameRate	30.0	# frame rate [Hz]
FramesToBeEncoded	250	# number of frames
#===== CODING =====		
SymbolMode	1	# 0=CAVLC, 1=CABAC
PRExt	1	# 8x8 transform (0:off, 1:on)
BasisQP	31	# Quantization parameters
#===== STRUCTURE =====		
GOPSize	8	# GOP Size (at maximum frame rate)
IntraPeriod	8	# Anchor Period
NumberReferenceFrames	2	# 参考帧数，对于当前 MVC 的预测结构来说，其为 2

```

InterPredPicsFirst      1          # 1 Inter Pics; 0 Inter-view Pics
DeltaLayer0Quant        0          # differential QP for layer 0
DeltaLayer1Quant        3          # differential QP for layer 1
DeltaLayer2Quant        4          # differential QP for layer 2
DeltaLayer3Quant        5          # differential QP for layer 3
DeltaLayer4Quant        6          # differential QP for layer 4
DeltaLayer5Quant        7          # differential QP for layer 5
PicOrderCntType         0          # Picture order count type (0 or 2)
#===== MOTION SEARCH =====
SearchMode              4          # Search mode (0:BlockSearch, 4:FastSearch)
SearchFuncFullPel       3          # Search function full pel
                                # (0:SAD, 1:SSE, 2:HADAMARD, 3:SAD-YUV)
SearchFuncSubPel        2          # Search function sub pel
                                # (0:SAD, 1:SSE, 2:HADAMARD)
SearchRange             32         # Search range (Full Pel)
BiPredIter              4          # Max iterations for bi-pred search
IterSearchRange         8          # Search range for iterations (0: normal)

#===== LOOP FILTER =====
LoopFilterDisable       0          # Loop filter idc (0: on, 1: off,
                                # 2:on except for slice boundaries)
LoopFilterAlphaC0offset 0          # AlphaOffset(-6..+6): valid range
LoopFilterBetaOffset    0          # BetaOffset (-6..+6): valid range

#===== WEIGHTED PREDICTION =====
WeightedPrediction      0          # Weighting IP Slice (0:disable, 1:enable)
WeightedBiprediction    0          # Weighting B Slice (0:disable, 1:explicit,
                                # 2:implicit)

#===== NESTING SEI (Supplemental Enhancement Information) MESSAGE =====
NestingSEI              0          # (0: NestingSEI off, 1: NestingSEI on)
SnapShot                0          # (0: SnapShot off, 1: SnapShot on)

#===== ACTIVE VIEW INFO SEI MESSAGE =====
ActiveViewSEI           0          # (0: ActiveViewSEI off, 1: ActiveViewSEI on)

#===== VIEW SCALABILITY INFOMATION SEI MESSAGE =====
ViewScalInfoSEI        0          # (0: ViewScalSEI off, 1: ViewScalSEI on)

#===== MULTIVIEW SCENE INFORMATION SEI MESSAGE =====
MultiviewSceneInfoSEI   1          # (0: off, 1: on)
MaxDisparity            12

#===== MULTIVIEW ACQUISITION INFOMATION SEI MESSAGE =====
MultiviewAcquisitionInfoSEI 0 # (0: off, 1: on), 如果有, 则需填写视频采集信息的配置文件, 见下面
AcquisitionInfoFile     Camera_ballroom.cfg

#===== PARALLEL DECODING INFORMATION SEI Message =====
PDISEIMessage          0          # PDI SEI message enable (0: disable, 1:enable)
PDIInitialDelayAnc      2          # PDI initial delay for anchor pictures
PDIInitialDelayNonAnc   2          # PDI initial delay for non-anchor pictures

#====下面用来描述预测结构的信息=====
NumViewsMinusOne        3          # (Number of view to be coded minus 1)
ViewOrder               0-2-1      # (Order in which view_ids are coded)

#====第 0 个视点的依赖关系
View_ID                 0          # (view_id of a view 0 - 1024)
Fwd_NumAnchorRefs       0          # (number of list_0 references for anchor)
Bwd_NumAnchorRefs       0          # (number of list 1 references for anchor)
Fwd_NumNonAnchorRefs    0          # (number of list 1 references for non-anchor)
Bwd_NumNonAnchorRefs    0          # (number of list 1 references for non-anchor)
#====第 1 个视点的依赖关系
View_ID                 1

```

```

Fwd_NumAnchorRefs      1
Bwd_NumAnchorRefs      1
Fwd_NumNonAnchorRef    1
Bwd_NumNonAnchorRefs   1
Fwd_AnchorRefs         0 0      # (ref_idx view_id)
Bwd_AnchorRefs         0 2
Fwd_NonAnchorRefs      0 0
Bwd_NonAnchorRefs      0 2
#===第 2 个视点的依赖关系
View_ID                2
Fwd_NumAnchorRefs      1
Bwd_NumAnchorRefs      0
Fwd_NumNonAnchorRefs   0
Bwd_NumNonAnchorRefs   0
Fwd_AnchorRefs         0 0

```

```
NumLevelValuesSignalledMinus1 0
```

```

Level_IDC              1
NumApplicableOpsMinus1 0
ApplicableOpTemporalId 0 0
ApplicableOpNumTargetViewsMinus1 0 2
ApplicableOpNumViewsMinus1 0 2
ApplicableOpTargetViewId 0 0 0
ApplicableOpTargetViewId 0 1 2
ApplicableOpTargetViewId 0 2 1

```

如果主配置文件中设置视频采集信息的配置文件可用，则需要同时输入视频采集信息配置文件，其格式如下：

表. 3. 视频采集信息的配置文件

```

NumViewsMinus1        1 # number of views - 1
IntrinsicParameterFlag 1  # intrinsic_param_flag
IntrinsicParametersEqual 0  # intrinsic_params_equal
ExtrinsicParameterFlag 1  # extrinsic_param_flag
Precision_FocalLength  8
Precision_PrincipalPoint 8
Precision_RadialDistortion 8
Precision_RotationParam 9
Precision_TranslationParam 10

View_ID 0
FocalLengthX 5  # focal_length_x (unit: pixels)
FocalLengthY 6  # focal_length_y (unit: pixels)

PrincipalPointX 4  # principal_point_x (unit: pixels)
PrincipalPointY 3  # principal_point_y (unit: pixels)
RadialDistortion 2  # radial_distortion
R_1 10 11 23  # r_11 r_12 r_13
R_2 32 33 22  # r_21 r_22 r_23
R_3 10 1 3  # r_31 r_32 r_33
Translation 1 5 4  # t_1 t_2 t_3

View_ID 1
FocalLengthX 51
FocalLengthY 61
PrincipalPointX 41

```

```
PrincipalPointY 31
RadialDistortion 21
R_1 101 111 231
R_2 321 331 221
R_3 10 11 13
Translation 11 51 41
```

4.2. 码流整合工具

在编码器对所有视点的视频进行编码后, 利用码流整合工具将编码的结果整合到一个文件中。其运行指令为:

```
MVCBitStreamAssemblerStatic -vf assembler.cfg
```

其中:

-vf: 选项参数, 和编码器中相同。

assembler.cfg: 配置文件名。该文件中指定了码流输出的信息, 以及整合方式等, 具体内容见下表所示:

表. 4. 码流整合工具的配置文件

#===== Assembler: View Encode order =====	
OutputFile	golf1.264
NumberOfViews	8
InputFile0	stream_0.264
InputFile1	stream_2.264
InputFile2	stream_1.264
InputFile3	stream_4.264
InputFile4	stream_3.264
InputFile5	stream_6.264
InputFile6	stream_5.264
InputFile7	stream_7.264

其中用户通过 OutputFile 来制定输出的文件名; NumberOfViews 为该序列总共有多少个视点视频; InputFileX(X=0,1...), 指定了各个视点的整合顺序。

4.3. 解码器

JMVC 的解码器对整合后的码流进行解码, 其运行指令如下所示:

```
H264AVCDecoderLibTestStatic <str> <rec> <numViews>
```

其中:

<str>: 解码器的输入, MVC 码流的文件名。

<rec>: 解码器的输出, 解码后重构得到的多视点视频序列的文件名。

<numViews>: 该序列中的视点数

解码器的输出为<numViews>个 yuv 视频, 每个视频代表一个视点的视频数据, 其文件名的格式为 “<rec>_视点号.yuv”

5. 测试序列

多视点视频编解码的测试序列根据分辨率主要有以下三种:

- 1、 分辨率为 320*240

此类多视点视频序列比较多, 包括 KDDI 提供的 golf、flamenco、objects、race 等序列; CMU 提供的 capimages 动画序列。

- 2、 分辨率为 640*480

该分辨率的视频有名古屋大学提供的 rena 和 Akko&Kayo 两个视频。

3、 分辨率为 1024*768

高分辨率的多视点视频序列目前有 HHI 提供的 Janine、Jungle 和 Uli 视频序列，以及 MSR 提供的 Ballet 和 Breakdancers。