

Application Parameter Settings for TMS320DM365 H.264 Encoder

Sejal Barot, Karimulla Shaik, Naveen Srinivasamurthy, and Yashwant Dutt

MMCodecs

ABSTRACT

Texas Instrument's TMS320DM365 H.264 video encoder is a flexible implementation of the H.264 video encoder standard. It supports various profiles and features, each of which can be independently selected, set or controlled. This flexibility enables the TMS320DM365 H.264 video encoder to be used in a variety of different applications. This application report lists the different parameters that can be set to control the operation of the TMS320DM365 H.264 video encoder. Recommended values of the parameters for several applications scenarios, video conferencing, video security, video surveillance, video broadcast, video streaming, storage, local playback and digital video recorder (DVR) are provided. These recommended values enable the optimal usage of the video encoder for these application scenarios.

1 Introduction

DM365 is a digital multi-media system on-chip comprising an ARM926 host processor, two co-processors and various peripherals. The DM365 also has a video front-end hardware block that enables seamless interface to most additional external devices required for video applications. The H.264 video encoder, which is a popular video coding algorithm enabling high-quality multimedia services even on limited bandwidth network, has been implemented on DM365. The programmable architecture of the DM365 H.264 video encoder makes it suitable to be used in a variety of different application scenarios including video conferencing, video surveillance, video security, video broadcast, video streaming, video storage, local playback and DVR, among others.

The DM365 H.264 video encoder has several different parameter settings that can be independently controlled enabling it to achieve the best quality for the desired application scenario; however, it is vital that the parameters be set correctly to leverage the DM365 video codec to the fullest extent. This document briefly describes the common application scenarios and provides recommended settings for the different parameters for each of these applications. These recommended settings enable best use of the DM365 H.264 encoder in the different applications.

[Section 2](#) lists the various parameters used to control the DM365 H.264 video encoder. [Section 3](#) describes the parameters to modify for the different application scenarios and specifies the recommended values for the different parameters of the DM365 H.264 video encoder for the different application scenarios.

2 Definition of Common API Parameters

This section describes all the programmable parameters of DM365 video encoder. All the parameters are categorized into two sets of parameters: base and extended parameters. Both base and extended parameters consist of two subsets: IH264VENC_Params, which can be set only at creation time and IVIDENC1_DynamicParams, which can be changed dynamically at run time before a frame encoding. Base and extended parameters are described in detail in the following sections.

2.1 Base Params

Base params are the minimum parameter set required for running the encoder; the corresponding structures are described below. The default values are specified in the device-specific user's guide.

2.1.1 IH264VENC_Params

This section describes the creation parameters for an algorithm instance object. When this data structure is set to NULL, default parameters are used for the following parameters.

2.1.1.1 Fields

Size— Size of the basic or extended (if being used) data structure in bytes.

encodingPreset— `encodingPreset` is used to select either full-featured high-quality or standard-featured high-speed video encoding. If it is set to high-quality mode, encoder encodes with high quality with a trade off in performance. When standard-quality mode is selected, higher performance is achieved. Standard quality should be selected for 720p resolution and multi-channel operation at lower resolutions. For all other cases, high-quality mode should be selected. Setting this to user defined enables selection of encoder setting through `encQuality` in extended parameters.

rateControlPreset— `rateControlPreset` indicates to the encoder the appropriate rate control to be used. This needs to be set depending on the use cases like low-delay encoding and storage. You do not need to specify the rate control algorithm as the encoder selects the appropriate algorithm depending on the `rateControlPreset` value.

maxHeight— Maximum video height to be supported in pixels.

maxWidth— Maximum video width to be supported in pixels.

maxFrameRate— Maximum frame rate in frames-per-sec (fps) to be supported. This is specified as maximum frame/field rate multiplied by 1000. This enables specifying non-integer fps. For example, if the maximum frame rate is 29.97, set this field to 29970.

maxBitRate— Maximum bit-rate to be supported in bits-per-second.

dataEndianness— Endianness of input data.

maxInterFrameInterval— `maxInterFrameInterval` indicates the distance from I-frame to P-frame. If there are no B-frames, the value is set to 1. Because the encoder does not support B-frames, this is set to 1.

inputChromaFormat— `inputChromaFormat` indicates the supported input YUV format for the encoder. The encoder only supports YUV420 semi-planar format where chroma-U and chroma-V are interleaved.

inputContentType— This flag indicates the input video format and the supported formats are progressive or interlace.

reconChromaFormat— `reconChromaFormat` indicates the supported output YUV format for the encoder reconstructed output. The encoder only supports YUV420 semi-planar format where chroma-U and chroma-V are interleaved.

2.1.2 IVIDENC1_DynamicParams

This structure defines the run-time parameters for an algorithm instance object. When this data structure is set to NULL, default parameters are used for the following parameters. For more information, see the device-specific user's guide.

2.1.2.1 Fields

Size— Size of the basic or extended (if being used) data structure in bytes.

inputHeight— Height of input frame in pixels. Input height can be changed before start of encoding within the limits of maximum height. `inputHeight` must be in multiples of twos;.minimum height supported is 96. Irrespective of interlaced or progressive content, input height should be given as frame height.

- inputWidth**— Width of input frame in pixels. Input width can be changed before the start of encoding within the limits of maximum. `inputWidth` must be in multiples of twos; minimum width supported is 128.
- refFrameRate**— Reference or input frame rate in frames-per-sec (fps) to be supported. This is specified as input frame/field rate multiplied by 1000. This enables specifying non-integer fps. For example, if the frame rate is 29.97, set this field to 29970. This must be set the same as `targetFrameRate`.
- targetFrameRate**— Target input frame rate in frames-per-sec (fps) to be supported. This is specified as input frame/field rate multiplied by 1000. This enables specifying non-integer fps. For example, if the frame rate is 29.97, set this field to 29970. This must be set the same as `refFrameRate`.
- targetBitRate**— Target bit-rate to be achieved in bits-per-second. For example, if the bit-rate is 2 Mbps, set this field to 2,000,000.
- intraFrameInterval**— Interval between two consecutive intra frames. This is to insert period I-Slices in the GOP. For example, if the value to set to 30, every 30th picture is encoded as I-Slice.
- generateHeader**— When this is set to 1, the encoder only generates headers (SPS, PPS). For normal encoding, this should be set to 0.
- captureWidth**— Capture-width parameter enables the application to provide input buffers with different line width (pitch) alignment than image width. For example, for progressive content if the input capture is 352x288, but the encoder is only required to encode a part of it, say 320x288, then `captureWidth` should be set to 352. For interlaced content, `captureWidth` should be equal to the pitch/stride value needed to move to the next row of pixels in the same field.
- forceFrame**— Force the current (immediate) frame to be encoded as a specific frame type. It is used to force I/IDR slices while encoding. It is not recommended to use this flag when periodic I/IDRs are used, instead `intraFrameInterval` should be used.
- interFrameInterval**— Number of B pictures between two reference frames; i.e., the number of B frames between two P pictures or I/P pictures, as the B-pictures are not supported. This parameter must be set to 0.
- mbDataFlag**— This parameter is not supported. It should be set to 0.

2.2 Extended Params

Extended params are the additional advanced parameters to control the quality/performance of the encoder. These enable more detailed and precise control of the encoder to tailor it more specifically to the desired application. The corresponding structures and default values of the members of the structure are described below. The default values are specified in the device-specific user's guide.

2.2.1 IH264VENC_Params

This structure defines the creation parameters and any other implementation specific parameters for a H.264 encoder instance object. The creation parameters are defined in the XDM data structure, `IVIDENC1_Params`.

2.2.1.1 Fields

- profileIdc**— Profile identification for the encoder. The current version supports up to high profile. The value can be set to 66 (baseline profile), 77 (main profile), 100 (high profile), 100 (default value). The extended profile (88) is not supported.
- levelIdc**— Level identification for the encoder. For details, see H.264: Advanced Video Coding for Generic Audiovisual Services (03/2010) available at <http://www.itu.int/rec/T-REC-H.264-201003-I>.
- aspectRatioX**— X scale for aspect ratio. This value should be greater than 0 and co-prime with `AspectRatioY`.

aspectRatioY— Y scale for aspect ratio. This value should be greater than 0 and co-prime with AspectRatioX.

pixelRange— The range for the luma and chroma pixel values; you can allow the pixels to be full range (from 0 to 255), or you can restrict the range.

meAlgo—

This is to select the type of motion estimation search algorithm; the encoder supports normal search and low-power motion estimation search algorithms. For relevant settings, see the user's guide.

timescale— Time resolution value for picture timing information. This should be greater than or equal to frame rate in fps.

numUnitsInTicks— Units of time resolution constituting the single tick. For details, see device-specific user's guide.

enableVUIparams— Flag to enable video usability information (VUI) parameters.

entropyMode— Flag to select entropy coding algorithm: CAVLC or CABAC. Recommendation is to set it to CAVLC for baseline profile; otherwise, set it to CABAC.

transform8x8FlagIntraFrame— Flag for enabling 8x8 transform in I-frame. Recommendation is to set it to true for resolution greater or equal to D1.

transform8x8FlagInterFrame— Flag for enabling 8x8 transform in P-frame. Recommendation is to set it to true for resolution greater or equal to D1.

SequenceScalingFlag— Flag for selecting sequence scaling matrix. The supported matrices are low, moderate, sharp, and auto. When auto mode is selected, the encoder internally chooses either low, moderate or sharp matrices based on the input resolution and bit rate. Choosing a sharp matrix highly attenuates high-frequency coefficients and provides a high boost to low-frequency coefficients. On the other hand, choosing a low matrix provides a small boost to low-frequency coefficients and a low attenuation to high-frequency coefficients. Moderate matrix provides a moderate boost and attenuation. It is recommended to set this value to auto.

disableHVICPeveryFrame— Reserved field.

encQuality— Flag for selecting high- or standard-quality encoding. For details, see encodingPreset in base parameters. This is effective only when encodingPreset is set to user defined. This is also an alternate way of selecting between standard- and high-quality encoding. The other way is using encodingPreset.

unrestrictedMV— Flag for enabling unrestricted motion vectors, which allows the vectors to point beyond the picture boundaries. Recommendation is to set it to true.

enableARM926Tcm— Flag for enabling/disabling usage of ARM926 TCM for the memory operations internal to the encoder; enabling this flag frees the usage of IMCOP memory.

NOTE: For the default values of the above elements, see the device-specific user's guide.

2.2.2 IH264VENC_DynamicParams

This structure defines the run-time parameters and any other implementation-specific parameters for a H.264 encoder instance object. The run-time parameters are defined in the XDM data structure, IVIDENC1_DynamicParams.

2.2.2.1 Fields

intraFrameQP— Quantization parameter (QP) of I-frames in fixed-QP mode; values are 0 to 51. This is valid only when rate control is off, otherwise it is ignored.

- interPFrameQP**— Quantization parameter (QP) of P-frames in fixed-QP mode; values are 0 to 51. This is valid only when rate control is off, otherwise it is ignored.
- initQ**— Initial quantization (QP) for the first frame; values are 0 to 51. The parameter is applicable only when rate-control is enabled. It should be set based on the target bit-rate. If the appropriate value is not known, the recommended setting is -1. This prompts the encoder to select the appropriate value based on the target bit-rate and video resolution.
- rcQMax**— Maximum value of quantization parameter (QP) used while encoding; values are 0 to 51. The value for rcQMax should not be less than rcQMin. The parameter is valid only when rate-control is enabled, otherwise it is ignored.
- rcQMin**— Minimum value of quantization parameter (QP) used while encoding; values are 0 to 51. The value for rcQMin should not be greater than rcQMax. The parameter is valid only when rate-control is enabled, other wise it is ignored.
- airRate**— Parameter to force intra macro-blocks insertion in P-pictures. This is used to refresh all the macro-blocks adaptively depending on the given airRate. The value specified indicates the number of macro-blocks in a frame to be forced as intra macro-blocks. For example, when encoding 720p@30fps, which has 3600 macroblocks and the air rate is given as 30, then all the macro-blocks will be refreshed in 120 frames, i.e., four second.
- sliceSize**— Size of each slice in bits. This field should be populated when H.241 is used.
- ifDisableIdc**— Option to enable or disable loop filter. Enabling the loop filtering improves the video quality without performance degradation. However, certain decoders implementing the low-complexity version of the H.264 baseline profile requires that the loop filter be disabled. When the encoded stream needs to be decoded by such decoders, the loop filter must be disabled during encoding.
- rcAlgo**— Option to specify the type of rate control algorithm, either CBR or VBR or FixedQP.
- maxDelay**— Maximum acceptable delay in milliseconds for rate control. This value should be greater than 100 ms and should be lesser or equal to 10000 ms. For more details, see [Section 3.6](#).
- intraSliceNum**— Index of the slice to be coded as fast intra-update slice.
- meMultiPart**— Flag to enable multiple partitions of inter macro-blocks. Enabling this can allow up to four motion vectors per macro-block.
- enableBufSEI**— Flag for enabling buffering period SEI message.
- enablePicTimSEI**— Flag for enabling picture timing SEI message.
- intraThrQF**— Reserved.
- perceptualRC**— This enables improving the perceptual video quality by differently distributing bits across high texture and smooth areas of video unevenly based on the human vision system. Enabling this flag improves the perceptual quality at the cost of performance.
- idrFrameInterval**— Interval between two consecutive IDR-frames. This is to insert period IDR-frames in the GOP. For example, if the value to set is 30, every 30th frame is encoded as IDR-Frame.
- mvSADoutFlag**— This flag enables dumping of MV and SAD value of the encoded stream to the application.
- resetHDVICPeveryFrame**— Flag to reset HDVICP at the start of every frame that is encoded. This is useful for multi-channel and multi-format encoding. If this flag is set, the H.264 encoder assumes that the memories of the HDVICP were overwritten by another codec or another instance of the same codec with different quality settings between the process call, so it reloads the code and data. For example, the application sets this flag to 1 if running another instance of a different codec, like

H264 decoder, or if running another H264 encoder instance with different quality setting in `encQuality` or `encodingPreset`. This is because H264 encoder uses different HDVICP configuration and data for standard quality and high quality, however, the application can set this flag to 0 for better performance if it runs multiple instances of the H264 encoder with the same quality settings in both `encQuality` and `encodingPreset` parameter fields.

3 Recommended Setting of the Encoder Parameters for Different application

To utilize the encoder to its fullest extent, in different applications, requires different settings for the encoder parameters. The required parameters that should be fine tuned for different applications are described in detail below. For details of all the programmable parameters, see [Section 2](#).

3.1 *InitQp*

This parameter selects the QP to be used for the very first frame of the video sequence. It is important to select an appropriate value for this parameter that is inline with both the video resolution and the target bit-rate. Both the video resolution and the target bit-rate dependencies are taken care of internally by the H.264 video encoder. To make use of this, it is recommended that the auto setting of the init QP is used by the application by setting this value to -1.

The higher the bit-rate, the lower the value of the initial QP and vice-versa. The choice of the initial QP is also dependent on the texture complexity of the video sequence. If the application has specific knowledge about the texture complexity of the video sequence to be encoded, the initial QP should be set by the application. For high texture complexity video sequences, the initial QP should be set to a high value and for low texture complexity video sequences the initial QP should be set to a low value. The exact values to be set are outside the scope of this application report.

3.2 *rcQMax and rcQMin*

To achieve the target bit-rate, the encoder uses rate control to decide the QP for every picture in the sequence.

The minimum and maximum value of QP is the operating zone for the rate control algorithm. For every picture it selects, a QP value is set in between (and inclusive) of the minimum and maximum QP. By setting the minimum QP, you control the minimum compression and maximum bit consumption of a picture. Setting the lowest value for minimum QP implies higher fidelity, however, if the lowest value of QP is chosen, which is equal to 0 as allowed by the standard, then this can in some cases lead to some frames consuming too many bits. This indirectly reduces the bit available for subsequent pictures and can lead to overall quality degradation.

By setting the maximum QP value, you control the minimum quality for every picture. Setting it to the maximum allowed by the standard, equal to 51, can lead to some pictures having very poor quality. However, the value should not be set too small as this would constrain the rate-control algorithm and the target bit-rate might not be achieved.

The recommended value of the minimum QP is 8 and the maximum QP is 44.

If the target bit-rate is very low and rate control is unable to achieve it, the maximum QP should be increased to 48.

3.3 *Perceptual Rate Control (PRC)*

The texture masking property of the Human Visual System (HVS) states that more quantization distortion should be introduced in areas of high texture and less quantization distortion should be introduced in smoother areas. This will result in better subjective quality of the video frame. Such a quantization noise shaping is performed by using a fine quantization (less quantization noise) in smooth areas and coarse quantization (more quantization noise) in high texture areas. Humans will perceive such a *noise-shaped* video frame as having better subjective quality than a video frame that has the same amount of noise evenly distributed throughout.

This is achieved by using PRC. PRC calculates the texture measure of every macroblock and assigns a suitable QP that takes into account the texture masking property of the HVS. This ensures that you perceive that the bits allocated in every frame maximize the perceptual quality.

The recommendation is that PRC should be switched ON for all resolutions up to and including 720p.

3.4 EncQuality

The two defined settings below are recommended for the usage of quality settings.

- For single channel encode with D1 and smaller resolutions, it is recommended that you set the EncQuality to HIGH_QUALITY. This enables 4MV, UMV and slice-based encoding (including H.241).
- For multi-channel encode or for resolutions greater than D1, it is recommended to set the EncQuality to STD_QUALITY. This enables UMV to be enabled, however, 4MV and slices are not enabled.

3.5 rcAlgo

3.5.1 Constant Bit Rate (CBR)

CBR allows you to maintain the end-to-end delay within the specified limit by having the following properties:

- Bits are allocated to frames to ensure the target bit-rate is not exceeded.
- At the start of a frame, QP is adjusted based on whether the number of bits for the previous frame were above or below the target. When the number of bits is above the target, the QP is increased; when the number of bits is below the target, the QP is reduced.
- If the bits being consumed indicate that bit-rate cannot be achieved, the input frame is dropped or skipped, i.e., it is not encoded.
- Row level-rate control - once per row of macroblocks, an additional adjustment is made based on whether the current frame is running above or below the target. If it is running above the target, the QP is increased; if it is running below the target, the QP is reduced.
- At end of frame, the average QP of the frame is computed to use as baseline for the next frame.

The above setting can be achieved by using RateControlPreset set to IVIDEO_LOW_DELAY in the base parameters.

For video conferencing, the selection-of-rate control is very important. CBR is suitable for the video conferencing applications because it allows you to maintain a constant flow of bit-rate and maintain the end-to-end delay within the specified limit required by this interactive application.

3.5.2 Variable Bit Rate (VBR)

VBR has flexibility to average the bit-rate over a larger time interval when compared to CBR. The bits allocation depends on the complexity of the sequence (motion/texture) rather than short term constant bit-rate averaging as discussed in CBR. Larger bit-rate deviations from target bit-rate are allowed more often on the CBR. The VBV buffer size is larger than that of the CBR because higher end-to-end delay can be tolerated, which enables the encoder to use larger instantaneous bit-rates if required, i.e., larger maximum to average picture size is allowed than compared to CBR. Frame skips are not allowed for VBR rate control. This implies all frames will be encoded.

- Bits are allocated to frames to ensure the target bit-rate is not exceeded.
- At the start of a frame, QP is adjusted based on whether the number of bits for the previous frame were above or below the target. When the number of bits is above the target, the QP is increased; when the number of bits is below the target, the QP is reduced.
- Row level-rate control - once per row of macroblocks, an additional adjustment is made based on whether the current frame is running above or below the target. If it is running above the target, the QP is increased; if it is running below the target, the QP is reduced.
- At end of frame, the average QP of the frame is computed to use as baseline for the next frame. The baseline QP is based on virtual buffer fullness and the difference from the target for the current frame.

The above setting can be achieved by using RateControlPreset set to IVIDEO_STORAGE in the base parameters.

VBR is appropriate for storage, broadcast, streaming and surveillance applications as the video sequences in these applications consist of non-homogenous video content. For example, broadcast/streaming content can be movies, and in surveillance, there will be large inactive periods with sporadic activity followed by frames with high/moderate activity that must be captured with high fidelity. Generally, any bits not used by a given frame may be allocated to future frames. Conversely, a given frame may consume significantly more bits than its target bit allocation, effectively borrowing bits from future frames.

3.6 *maxDelay*

The value of Max Delay controls the size of the VBV buffer size. The VBV buffer size is set to the product of the target bit-rate and the Max Delay. This ensures that the video encoder performs compression in order to achieve the target bit-rate over every Max Delay time duration. The selection of the Max Delay value directly impacts the video quality. For example, if the Max Delay is set to 100 ms, the VBV buffer size sets to 10% of the target bit-rate. Therefore, the rate control needs to achieve the target bit-rate over a 3 frame period, assuming the input video is 30 fps. Smaller Max Delay value implies that the video encoder has to achieve the target period over a small Max Delay interval. This can result in more skipped frames and/or more aggressive frame level rate control. Both of which can lead to quality degradations. Therefore, larger Max Delay is preferable since this implies more flexibility in bit allocation by the rate control algorithm.

Video conferencing applications require a guaranteed bandwidth from the network, small loss probability, and bounded delay. For the participants in a videoconference call to interact naturally, the end-to-end delay should be below human perception: about 300 ms.

Therefore, the Max Delay value must be set so that the end-to-end delay is below the human perception. Values lower than 100 ms typically result in too many skipped frames and values greater than 300 ms result in annoying perceptual artifacts, inability for users to interact naturally.

For non-interactive applications, like broadcast/streaming/storage, the larger end-to-end delay is tolerable. Therefore, the Max Delay value should be set to a larger value. Then the rate control needs to achieve the target bit-rate over a larger time interval.

3.7 *profileIdc*

The H.264 video codec standard has been designed to cater to a very broad application range of applications that covers all forms of digital compressed video from low bit-rate Internet streaming applications to HDTV broadcast and Digital Cinema applications with nearly lossless coding. This is achieved by using the different profiles of the H264 standard.

3.7.1 **Baseline Profile**

This is targeted primarily for low-cost applications. Constrained baseline profile (CBP) is used widely in video conferencing and mobile applications.

The features of the baseline profile are:

- I and P Slices
- Multiple Reference Frames
- In-Loop Deblocking Filter
- CAVLC Entropy Coding

NOTE: DM365 does not support multiple reference frames. ASO/FMO and RS are not supported by CBP.

3.7.2 **Main Profile**

The following features were originally intended as the mainstream consumer profiles for broadcast and storage applications.

- I and P Slices
- Multiple Reference Frames

- In-Loop Deblocking Filter
- CABAC Entropy Coding
- Interlaced and Progressive
- B Slices

3.7.3 High Profile

High profile supports the most sophisticated tools provided by the H.264 standard. Therefore, it is recommended to use high profile as the default setting for the DM365 encoder, except for video conferencing applications where the baseline profile should be used.

The following features are supported as part of the high profile encoding of H264:

- I and P Slices
- In-Loop Deblocking Filter
- CAVLC/CABAC Entropy Coding
- 4:2:0 Chroma Format
- Selection Between 8x8 or 4x4 Transform
- Progressive and Interlaced Encoding
- Support for Using scaling Matrices

3.8 *airRate*

Adaptive intra-refresh is as an error resiliency tool. On a network or channel slices/frame, loss is inevitable; in video conferencing, these losses cannot be handled by re-transmission, whereas, in streaming applications re-transmissions can handle these losses. This data loss can be mitigated with the use of adaptive intra refresh (AIR). In DM365 encoder random AIR pattern is implemented, i.e., the MBs encoded as intra refresh [IR] are randomly distributed across several frames. This ensures that the peak bit-rate consumption is distributed over several frames rather than one frame consuming too many bits adversely impacting the end-to-end delay.

The AIR value specifies the number of MBs that are coded as intra MBs per frame. Specifying a value of 30 for AIR, implies that for 720p30, that has 3600 MBs/frame, all macroblocks are refreshed once every 120 frames.

3.9 *intraFrameInterval*

The intra frame interval decides the GOP structure, which decides the distance in pictures between two intra pictures. For progressive encoding, specifying a value of N implies N-1 frames are encoded between two successive I-frames. For interlaced encoding, specifying a value of N implies 2N-2 fields are encoded between two successive I pictures.

Intra frames are used as an error resiliency tool. However, in applications like video conferencing it is not recommended to use periodic I-frames, because I-frames take significantly more number of bits compared to P-frames, which introduces additional difficulties for low-delay applications. Instead, it is recommended to use adaptive intra refresh as an error resiliency tool to avoid a single frame taking too many bits.

Keeping the intra frame interval value set to 0 ensures that the encoder only encodes the first frame as an IDR frame.

For the video surveillance and storage applications, it is recommended that periodic intra frames are used for the random access.

It is recommended that the intra frame interval be set 1 sec or less for broadcast and streaming applications since it enables fast forward and random access.

3.10 mvSADoutFlag

This flag enables dumping of the MV and SAD value of the encoded stream to the application. Motion vector and SAD values are used to detect the motion in the encoded video frame. This information can be used as input to algorithms that can detect the motion and raise an alarm, or insert intra frames or change the encode quality settings based on the application's need. This feature is primarily used in video surveillance applications to monitor the motion. Motion vectors can also be used to improve the video quality of dual channel video encoding.

3.11 Recommended Parameter Settings for Different Applications

3.11.1 Video Conferencing Application

A video conference (also known as a video teleconference) is an interactive telecommunication application, which allows two or more locations/users to interact via two-way video and audio transmissions simultaneously. It has been called as visual collaboration and it is a type of groupware. It differs from video phone in that it is designed not just for communication between individuals, but also to serve the needs for conferencing.

The recommended values of the encoder parameters for the video conferencing are specified in [Table 1](#).

Table 1. Recommended Parameters for Video Conferencing

| Parameter Name | Recommended Settings | Remarks |
|--------------------|---|---|
| profileIdc | 66 (Base line profile) | 100 (high profile) without B frames is recommended for closed loop video conferencing with low delay. |
| initQ | -1 | |
| rcQMax | 44 | If the target is very low bit-rate, you can increase this value to avoid too many frame skips. |
| rcQMin | 8 | |
| perceptualRC | 0 | Disabled |
| EncQuality | STD_QUALITY | |
| rcAlgo | 0 | CBR |
| maxDelay | 300 to 500 ms | |
| airRate | QCIF – 1 CIF – 4 D1 – 12 720p - 30 | |
| intraFrameInterval | 0 | Only first frame as intra frame |
| SliceSize | 8000 to 11200 bits | Used for error concealment and to meet the network packetization limits. To use this feature, EncQuality should be set to HIGH_QUALITY. |
| targetBitRate | QCIF – 64 kbps @ 15fps / 128 kbps @ 30 fps CIF – 256 to 384 kbps @ 15 fps / 384 to 512 kbps @ 30 fps D1 – 1000 to 1500 kbps @ 30 fps 720p – 2000 to 4000 kbps @ 30 fps | |

3.11.2 Security/Surveillance Application

Video surveillance applications, compared to other categories of video applications, are very broad in terms of requirements. It requires very high compression to achieve high bandwidth, low complexity to meet cost constraints, a good amount of flexibility for random access, enable use of events trigger and stream scalability to serve several clients on network.

In video surveillance there are typically two streams: display stream (real time display of the video sequence) and storage stream (higher fidelity video stream for archiving and/or analysis purpose).

Typically video surveillance applications use:

- For the display stream: low bit-rates and low delays.
- For the storage stream: higher bit-rates and larger delays.

The recommended values of bit-rate and frame rates are listed in [Table 2](#) for different standard resolutions.

The recommended values of the encoder parameters for security/surveillance are specified in [Table 2](#).

Table 2. Recommended Parameters for Video Surveillance

| Parameter Name | Recommended Settings | Remarks |
|----------------------------|--|--|
| profileIdc | 100 | |
| initQ | -1 | |
| rcQMax | 44 | If the target is very low bit-rate, you can increase this value to avoid too many frame skips |
| rcQMin | 8 | |
| entropyMode | 1 | CABAC |
| transform8x8FlagIntraFrame | 1 | Enable |
| transform8x8FlagInterFrame | 0 | Disable for 720P, enable for single channel < 720P |
| unrestrictedMV | 1 | Enable |
| seqScalingFlag | 1 | AUTO |
| perceptualRC | 1 | |
| EncQuality | STD_QUALITY | |
| rcAlgo | 0 | CBR |
| maxDelay | Display: 300 to 500 ms Storage: 1000 to 7000 ms | |
| airRate | 0 | |
| intraFrameInterval | Display: 5 second Storage: 1 second | Periodic intra frame is recommended for the random access |
| mvSADoutFlag | 1 | Enable MV and SAD value of the encoded stream. To detect the motion this information is used in security applications. For more details, see the device-specific user's guide. |
| targetBitRate | For display: QCIF – 64 kbps @ 15 fps/128 kbps @ 30 fps CIF – 256 to 384 kbps @ 15 fps/384 to 512 kbps @ 30 fps D1 – 1000 to 1500 kbps @ 30 fps 720p – 2000 to 4000 kbps @ 30 fps For storage: QCIF – 128 kbps @ 15 fps/256 kbps @ 30 fps CIF – 384 to 512 kbps @ 15 fps/512 to 1000 kbps @ 30 fps D1 – 2000 to 4000 kbps @ 30 fps 720p – 4000 to 8000 kbps @ 30 fps | |
| lfDisableIdc | 0 | Loop filter enabled |
| meMultiPart | 0 | Single motion vector per MB |

3.11.3 Broadcasting/Streaming Application

Broadcasting is the distribution of audio and/or video signals that transmit programs to an audience. The audience may be the general public or a relatively large sub-audience.

There are two major types of broadcast: live and pre-recorded content.

- Live content

Live content is transmission as it occurs. For example, broadcast output of a video camera over the internet or to a television. The encoder encodes the content in real time without writing the content to a clip, e.g., sports and live events.

- Prerecorded content

Pre-recorded content consists of video or audio that is recorded and written to a digitized clip. You can then edit the clip before converting it to a streaming format and broadcast it across a network. To the viewer, the presentation looks just like a live broadcast, e.g., movies and serials.

Similarly *streaming* is the delivery of media, such as movies and live presentations, over a network in real time to a single client. A streaming server transmits the media to a client device, which plays the media as it is delivered. With streaming, no files are downloaded to the viewer's local storage. You can stream media at various rates, from modem to broadband bit-rates.

With the streaming server, you can deliver the following and similar content:

- Live events in real time
- Video on demand
- Playlists of pre-recorded content

The main difference between streaming and broadcast is in the target audience. In streaming, the recipient is a single client; while in broadcast, the recipient is a large audience.

Recommended values of the encoder parameters for the broadcasting and streaming applications are specified in [Table 3](#).

Table 3. Recommended Parameters for Broadcasting

| Parameter Name | Recommended Settings | Remarks |
|----------------------------|--|---|
| profileIdc | 100 | |
| initQ | -1 | |
| rcQMax | 44 | If the target is very low bit-rate, you can increase this value to avoid too many frame skips |
| rcQMin | 8 | |
| entropyMode | 1 | |
| transform8x8FlagIntraFrame | 1 | CABAC |
| transform8x8FlagInterFrame | 0 | Enable |
| unrestrictedMV | 1 | Disable for 720P, enable for single channel < 720P |
| seqScalingFlag | 1 | Enable |
| airRate | 0 | AUTO |
| intraFrameInterval | 12 to 15 fps | |
| targetBitRate | QCIF – 128 kbps @ 15 fps / 256 kbps @ 30 fps CIF – 384 to 512 kbps @ 15 fps / 512 to 1000 kbps @ 30 fps D1 – 2000 to 4000 kbps @ 30 fps 720p – 4000 to 8000 kbps @ 30 fps | |
| lfDisableIdc | 0 | Loop filter enabled |
| meMultiPart | 0 | Single motion vector per MB |
| maxDelay | 5000 ms to 7000 ms | |

3.11.4 Storage/Local Playback/DVR Application

Storage applications such as DVR records video in the digital format to a disk or local memory medium within the device such as set top boxes, portable media player, and software from PCs that enable video record and playback from the disk.

For high-end devices, e.g., DVR, the recommended values are specified in [Table 4](#).

Table 4. Recommended Parameters for High-End Devices Doing Local Storage

| Parameter Name | Recommended Settings | Remarks |
|----------------------------|---|---|
| profileIdc | 100 | |
| initQ | -1 | |
| rcQMax | 44 | If the target is very low bit-rate, you can increase this value to avoid too many frame skips |
| rcQMin | 8 | |
| entropyMode | 1 | |
| transform8x8FlagIntraFrame | 1 | CABAC |
| transform8x8FlagInterFrame | 0 | Enable |
| unrestrictedMV | 1 | Disable for 720P, enable for single channel < 720P |
| seqScalingFlag | 1 | Enable |
| airRate | 0 | AUTO |
| intraFrameInterval | 1 second | |
| targetBitRate | QCIF – 128 kbps @ 15 fps / 256 kbps @ 30 fps CIF – 512 kbps to 1000 kbps @ 30 fps D1 – 4000 to 8000 kbps @ 30 fps 720p – 4000 to 15000 kbps @ 30 fps | Most of the video recording / storage application use higher bit-rates and higher delays |
| lfDisableIdc | 0 | Loop filter enabled |
| meMultiPart | 0 | Single motion vector per MB |
| maxDelay | 5000 ms to 7000 ms | |

For low-end devices, e.g., mobile phone, the recommended values are specified in [Table 5](#).

Table 5. Recommended Parameters for Low-End Devices Doing Local Storage

| Parameter Name | Recommended Settings | Remarks |
|--------------------|--|---|
| profileIdc | 100 | |
| initQ | -1 | |
| rcQMax | 44 | If the target is very low bit-rate, you can increase this value to avoid too many frame skips |
| rcQMin | 8 | |
| perceptualRC | 1 | Enabled |
| EncQuality | HIGH_QUALITY | |
| rcAlgo | 1 | VBR |
| maxDelay | 5000 to 7000 ms | |
| airRate | 0 | |
| intraFrameInterval | 1 second | |
| targetBitRate | QCIF – 64 kbps @ 15 fps / 128 kbps @ 30 fps CIF – 384 kbps to 512 kbps @ 30 fps D1 – 1000 to 2000 kbps @ 30 fps 720p – 2000 to 4000 kbps @ 30 fps | |
| lfDisableIdc | 0 | Loop filter enabled |
| meMultiPart | 0 | Single motion vector per MB |

4 References

- H.264: Advanced Video Coding for Generic Audiovisual Services (03/2010) available at <http://www.itu.int/rec/T-REC-H.264-201003-1>

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

| Products | | Applications | |
|-----------------------------|--|----------------------------|--|
| Amplifiers | amplifier.ti.com | Audio | www.ti.com/audio |
| Data Converters | dataconverter.ti.com | Automotive | www.ti.com/automotive |
| DLP® Products | www.dlp.com | Communications and Telecom | www.ti.com/communications |
| DSP | dsp.ti.com | Computers and Peripherals | www.ti.com/computers |
| Clocks and Timers | www.ti.com/clocks | Consumer Electronics | www.ti.com/consumer-apps |
| Interface | interface.ti.com | Energy | www.ti.com/energy |
| Logic | logic.ti.com | Industrial | www.ti.com/industrial |
| Power Mgmt | power.ti.com | Medical | www.ti.com/medical |
| Microcontrollers | microcontroller.ti.com | Security | www.ti.com/security |
| RFID | www.ti-rfid.com | Space, Avionics & Defense | www.ti.com/space-avionics-defense |
| RF/IF and ZigBee® Solutions | www.ti.com/lprf | Video and Imaging | www.ti.com/video |
| | | Wireless | www.ti.com/wireless-apps |

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2010, Texas Instruments Incorporated