

# Measuring Video Quality With the TMS320DM6446 DVSDK

---

Cesar Iovescu

## ABSTRACT

The transmission of video sequences is enabled by compressing the video content with different video encoding algorithms at the source and decoding the stream at the destination. The compression process discards part of the original information and decreases the video quality of the decoded stream.

Measuring the video quality is measuring how *close* the decoded stream is to the original stream.

**Objective Video Quality** — The most common metric to measure the video quality objectively is the peak signal to noise ratio (PSNR). This application report shows how to use the DaVinci EVM digital video evaluation module (DVEVM) and SDK digital video software development kit (DVSDK) to measure the quality of video sequences using the PSNR metric.

**Subjective Video Quality** — People have different perceptions of the same video sequence; therefore, there is a subjective perspective to the video quality. Subjective video quality involves groups of people viewing the sequences and grading the quality.

This application report contains project code that can be downloaded from <http://www.ti.com/lit/zip/SPRAAK1>.

---

## Contents

1	Video Quality Measuring Methodology .....	2
2	The Digital Video Test Bench (DVTB) .....	3
3	Computing the PSNR .....	4
4	Example of Measuring the Video Quality With the DVSDK .....	5
5	Other Tools for Measuring Video Quality .....	7
6	Conclusion .....	7
7	References .....	7

## List of Figures

1	Generation of the Reconstructed Video Sequences .....	2
2	PSNR Computation .....	2
3	PSNR Utility Input Parameters .....	5
4	Sequence #1, H.264 – MPEG4 PSNR Measurements .....	6
5	Sequence #2, H.264 – MPEG4 PSNR Measurements .....	6
6	Sequence #3, H.264 – MPEG4 PSNR Measurements .....	7

## List of Tables

1	PSNR Measurements – MPEG4 SP, D1, 30 fps .....	5
2	PSNR Measurements – H.264 BP, D1, 30 fps .....	6

## 1 Video Quality Measuring Methodology

The video quality measuring method proposed in this report presents two steps:

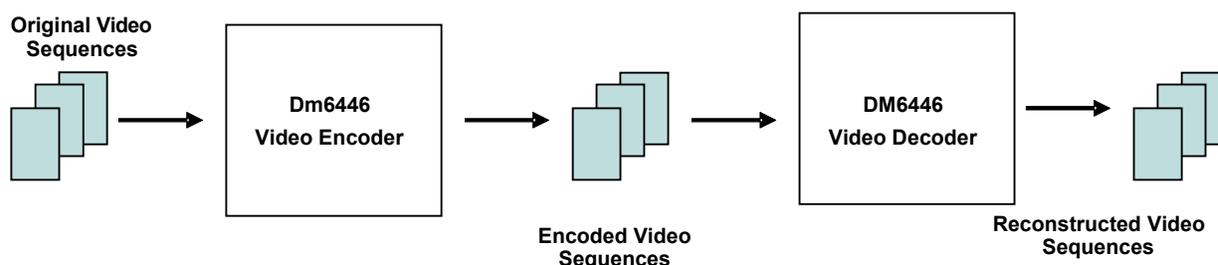
- Generate the reconstructed (decoded) video sequences.
- Compute the PSNR between the original and the reconstructed video sequences.

Before using the digital video test bench (DVTB), the DaVinci DVEVM and DVSDK environment must be set up.

For more information about setting up the DaVinci environment, refer to the DVEVM Hardware Setup and the DVEVM Software Setup chapters in the *DVEVM Getting Started Guide* ([SPRUE66](#)).

### 1.1 Generating the Reconstructed Video Sequences

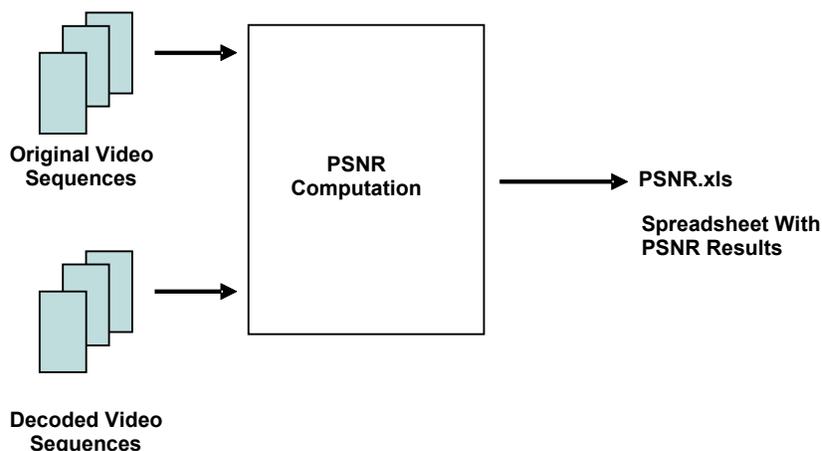
The methodology for generating the reconstructed video sequence is presented in [Figure 1](#). The DVTB is used to generate the encoded sequences. The DVTB utility is described in more detail in the following sections of this application report. Because it is not possible to access the reconstructed frames after the encoding process, it is necessary to decode the sequences and generate the reconstructed (decoded) sequences.



**Figure 1. Generation of the Reconstructed Video Sequences**

### 1.2 Computing the PSNR

The PSNR is computed by using a simple DOS utility that takes the original and reconstructed video sequences as input. The output generated is an excel spreadsheet that contains the PSNR values for the Y, Cb, Cr components for each frame.



**Figure 2. PSNR Computation**

## 2 The Digital Video Test Bench (DVTB)

The DVTB is a Linux® utility that was developed to execute end to end data flows using the DaVinci DVSDK . The DVTB uses the codec engine VISA APIs and Linux driver peripheral APIs to encode/decode audio and video streams.

The user can configure the codecs and/or peripherals before starting a data flow. This enables the user to exercise different use case scenarios and evaluate the system by configuring the codecs and/or peripherals as required.

Before using the DVTB, the DaVinci (DVEVM and DVSDK) environment must be set up.

### 2.1 Installation

For more information about setting up the DaVinci environment, refer to the DVEVM Hardware Setup and the DVEVM Software Setup chapters in the *DVEVM Getting Started Guide* ([SPRUE66](#)).

The DVTB is installed in a  $\$(DVEVM\_INSTALL\_ROOT)/dvtb$  folder during the installation of the `dvsdk_setuplinux_#_##_##_###.bin` file.

For more information about installing the DVTB, refer to the DVSDK Setup chapter in the *DVSDK Getting Started Guide* ([SPRUEG8](#)).

### 2.2 Using the DVTB to Generate Reconstructed Video Sequences

This section provides a short overview of using the DVTB. For more information, see the DVTB documentation located in:

$\$(DVEVM\_INSTALL\_ROOT)/dvtb/docs/DVTB-User-Guide.pdf$

The following steps are to be performed in the HyperTerminal comm session window.

1. Mount the hard disk drive
 

```
root@158.218.33.135:~# mount /dev/hda1 /mnt/disk
```
2. Copy a yuv video sequence, `video.yuv` to the hard disk drive. The video encoders from Texas Instruments accept 422 interleaved, little endian and 420 planar format.
3. Go to the folder where the `dvtb` utility is located.
 

```
root@158.218.33.135:~# cd /opt/dvevm
```
4. Load the kernel driver modules (ignore error messages)
 

```
root@158.218.33.135:/opt/dvevm# ./cfg-dvtb.sh
```
5. Create the following script that includes the `dvtb` commands that are used to configure the encoder. In this example the H.264\_E is used. Save the script as `script.cfg` for example.
 

```
setp engine name encode
setp videnc codec h264enc
setp videnc inputChromaFormat 1
setp videnc maxFrameRate 25000
setp videnc refFrameRate 25000
setp videnc targetFrameRate 25000
setp videnc intraFrameInterval 25
setp videnc maxHeight 480
setp videnc maxWidth 720
setp videnc inputHeight 480
setp videnc inputWidth 720
setp videnc numframes 252
setp videnc numframes 252
setp videnc targetBitRate 2000000
func videnc -s /mnt/disk/video.yuv -t /mnt/disk/video.h264
```
6. Run the script with the `dvtb`

```
root@158.218.33.135:/opt/dvevm# ./dvtb -s script.cfg
```

7. Create the following script that includes the dvb commands used to configure the decoder used.
 

```
setp engine name decode
setp viddec codec h264dec
setp viddec maxHeight 720
setp viddec maxWidth 480
func viddec -s /mnt/disk/video.h264 -t /mnt/disk/video_recon.yuv
```
8. The reconstructed sequence is available in the file /mnt/disk/video\_recon.yuv

### 3 Computing the PSNR

#### 3.1 Definition of the PSNR

The PSNR is a common measure of the quality of a reconstructed video sequence. After generating the reconstructed video sequence, the quality is analyzed using the PSNR measure.

A logarithmic scale is used to compute the PSNR because the signals have a wide dynamic range.

The PSNR formula is as follows:

$$PSNR = 10 \log_{10} \left( \frac{2^b - 1}{MSE} \right) dB$$

Where  $b$  denotes the number of bits used to represent the pixel. The mean square error (MSE) of two images of  $H \times W$  pixels is defined as:

$$MSE = \frac{1}{H \times W} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} (Orig(i,j) - Recon(i,j))^2$$

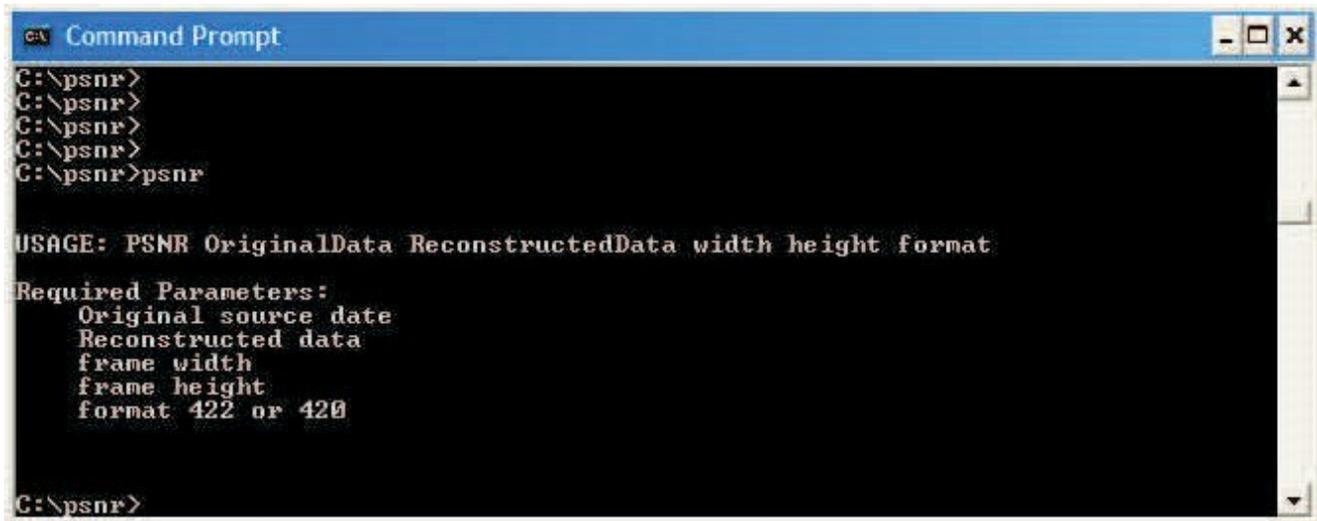
Orig( $i,j$ ) is the original pixel value and Recon( $i,j$ ) is the reconstructed pixel value. The higher the pixel value the better the quality of the reconstructed image.

An image has the YCbCr components. Y is the luma component and the Cb, Cr are the blue and red chroma components. The PSNR can be computed for each component, however the luma component Y is the dominant one and usually it is sufficient to compute the PSNR for the luma component Y.

### 3.2 Using the PSNR Utility

A PSNR utility is provided with this application report to compute the PSNR using the original and reconstructed video sequences. The following steps show how to use the utility

1. Open a DOS command prompt
2. Type *psnr* to display the required input parameters. The output is similar to that shown in [Figure 3](#).



```

C:\psnr>
C:\psnr>
C:\psnr>
C:\psnr>
C:\psnr>psnr

USAGE: PSNR OriginalData ReconstructedData width height format

Required Parameters:
  Original source data
  Reconstructed data
  frame width
  frame height
  format 422 or 420

C:\psnr>
  
```

**Figure 3. PSNR Utility Input Parameters**

The output produced by this utility is a spreadsheet that contains the PSNR values for the Y, Cb, Cr components for every image in the video sequence. The average PSNR also is provided.

## 4 Example of Measuring the Video Quality With the DVSDK

This section provides the practical application of the previous steps to measure the video quality for three different video sequences encoded with the H.264 BP and MPEG4 SP encoders.

Each sequence has specific characteristics (e.g., motion, texture) that influence the performance of the codecs. For example, sequence number 2 has a high motion content, and the performance of the codecs is the worst for this sequence. The sequences have D1 (720x480) resolution.

Measurements have been taken at different bit-rates with 30 fps frame rate. The measurements are summarized in [Table 1](#) and [Table 2](#).

**Table 1. PSNR Measurements – MPEG4 SP, D1, 30 fps**

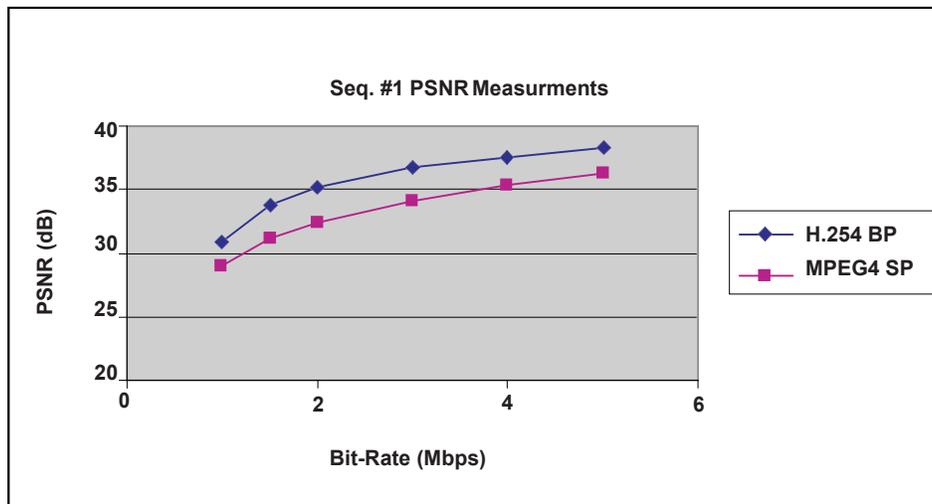
Bit-Rate (Mbps)	PSNR (dB) Sequence Number 1	PSNR (dB) Sequence Number 2	PSNR (dB) Sequence Number 3
1	29.05	22.53	27.06
1.5	31.13	24.7	29.28
2	32.44	25.85	30.64
3	34.15	27.46	32.47
4	35.33	28.66	33.68
5	36.21	29.7	34.61

**Table 2. PSNR Measurements – H.264 BP, D1, 30 fps**

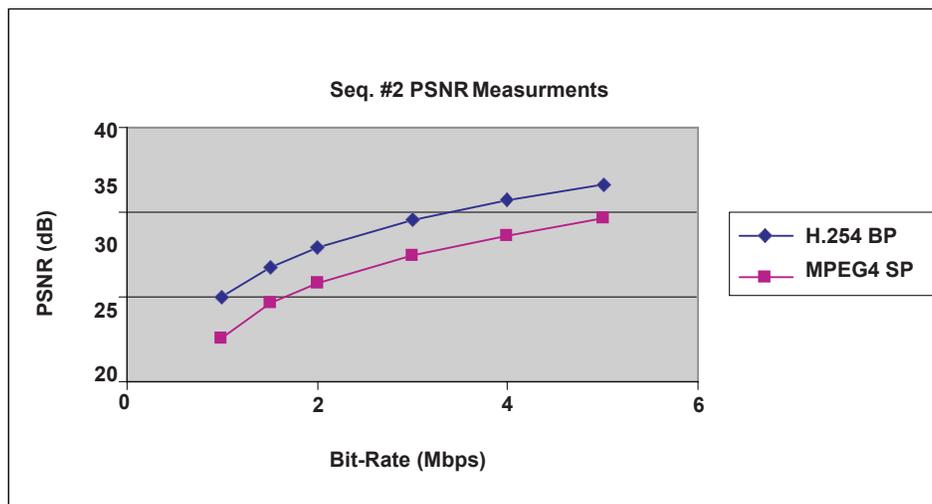
Bit-Rate (Mbps)	PSNR (dB)		PSNR (dB)
	Sequence Number 1	Sequence Number 2	Sequence Number 3
1	30.78	25.02	30.43
1.5	33.82	26.74	32.66
2	35.14	27.88	33.88
3	36.70	29.5	35.37
4	37.59	30.65	36.2
5	38.24	31.58	36.8

After performing these measurements it is interesting to compare the performance of the H.264 BP and MPEG4 SP encoders for these sequences. [Figure 4](#), [Figure 5](#) and [Figure 6](#) compare the PSNR performance for the encoders for the three sequences.

The difference between the two codecs is about 1.5 dB. These numbers have been measured for the codec releases included in the DVSDK v. 1.1.



**Figure 4. Sequence #1, H.264 – MPEG4 PSNR Measurements**



**Figure 5. Sequence #2, H.264 – MPEG4 PSNR Measurements**

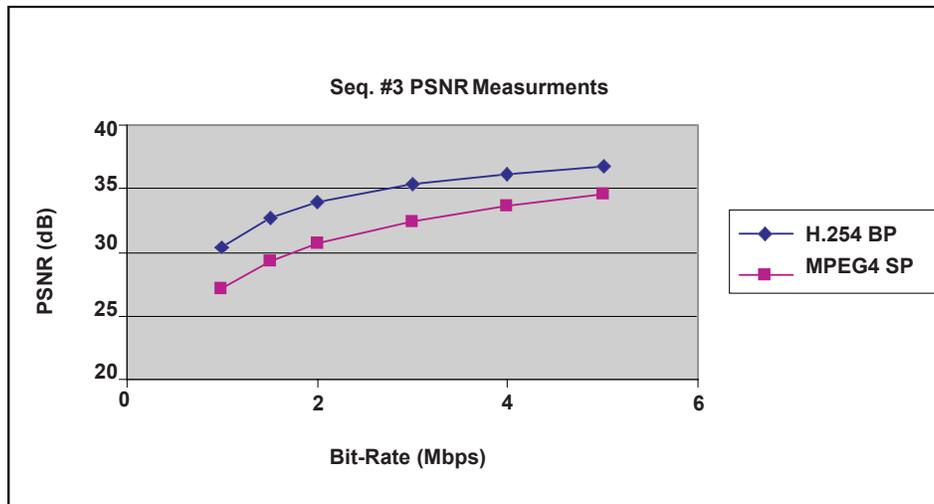


Figure 6. Sequence #3, H.264 – MPEG4 PSNR Measurements

## 5 Other Tools for Measuring Video Quality

This section provides a list of other tools that are useful for evaluating video quality.

### 5.1 YUV Viewer

A YUV viewer tool is indispensable for viewing raw YUV video sequences. Several are available on the web.

A good YUV viewer should have the following features:

- Wide selection for type input (YUV, RGB ....)
- Capability to step frame by frame
- Zooming capability

### 5.2 Byte-Extracting Utility

Sometimes it is necessary to study only a frame, or a subset of frames from a larger sequence.

To accomplish this, a byte-extracting utility is necessary. Such a utility can be found on the web or can be written.

## 6 Conclusion

This application report has shown how to use the DVSDK to measure the video quality of video sequences compressed with the encoders included in the DVSDK. The results provided in this application report apply only to codecs available in the DVSDK v. 1.1. Texas Instruments is working continually to improve the quality of its codecs.

## 7 References

- *DVEVM Getting Started Guide* ([SPRUE66](#))
- *DVSDK Getting Started Guide* ([SPRUEG8](#))

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale ([www.ti.com/legal/termsofsale.html](http://www.ti.com/legal/termsofsale.html)) or other applicable terms available either on [ti.com](http://ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2019, Texas Instruments Incorporated