ON2 TECHNOLOGIES, INC.

WHITE PAPER

Advantages of TrueMotion VP6 Technology

February 17, 2004

Document version: 1.2

On2 Technologies, Inc.
21 Corporate Drive, Suite 103
Clifton Park, NY 12065
www.on2.com

Contents

INTRODUCTION

This white paper explains why TrueMotion VP6 is the best video compression technology available today, particularly when compared with current MPEG-4 and H.264 profiles. This paper covers technical advances in VP6, improvements over previous versions of the TrueMotion algorithm, the performance of the VP6 decoder, and real-time VP6 encoding/decoding.

For more information about how we conducted our comparison tests with Windows Media and Real 10, see the “Test Methodology” section of this paper.

THE CASE FOR VP6

The following are just a few reasons why VP6 surpasses all other video compression technology:

- Achieves a 10-15% improvement over VP6 beta.
- Supports multi-pass encoding.
- Highly configurable (constant or variable) datarate control.
- Direct access to the reconstruction buffer.
- Multiple platform support (Intel, Equator, TI, PowerPC).
- Configurable sharpness control that allows you to adjust the subjective quality of the output.
- Compresses high-definition (HD) material with no restrictions on the encoder. VP6 can play back 1920x1080 HD material on a 2.5 GHz PC and 1280x720 material on a 1.5 GHz PC.
- Supports real-time encoding at full D1 resolution.
- Up to 40% image quality improvement over VP5.
- Up to 50% faster playback than VP5.
- Produces output with Peak Signal-to-Noise Ratios (PSNR) that are consistently better than Windows Media 9 and H.264 on a wide range of test material at data rates ranging from dial-up (28.8 Kbps) to DVD and HD (6 to 8 Mbps).
- Optimized to produce the best quality video available on high-resolution material (640x480 and higher).
- Carries no burdensome “patent pooling” restrictions or complicated external licensing fees.
- Designed for inexpensive Digital Signal Processors (DSPs). VP6 is ideal for embedded chipsets in non-PC devices and set-top boxes. Unlike some standards-based codecs (JVT,

MPEG-4 v10), VP6 runs on general-purpose DSPs without requiring expensive add-on subprocessors.

- A purely software-based solution that can be upgraded easily.
- Introduces predefined “profiles”: *Simple* for fast playback on inexpensive processors, *General* for full D1 on set-top boxes, and *Advanced* for ensuring the best quality possible at extremely low data rates.
- Offers a “fast compress” mode that can encode in real time on a 1 gigahertz Pentium 4 processor with very little loss in quality.
- Achieves any requested data rate by choosing automatically to adjust quantization levels, adjust encoded frame dimensions, or drop frames altogether.
- Guaranteed playback of a file transmitted at a constant bit rate.
- Supports both native interlaced and progressive scan output (also features block prediction modes that enable a mix of interlaced and progressive scan material).
- Decode complexity in VP6 is considerably lower than H.264. This has allowed VP6 to be successfully ported to inexpensive digital signal processing (DSP) solutions running at full D1 resolution.
- High-quality post-processing that removes block and ringing artifacts.
- Sophisticated context modeling in the entropy encoder.
- Better prediction of low-order frequency coefficients to improve output quality.
- Improved quantization strategy that preserves more detail in the output.

IMPROVEMENTS SINCE VP5 AND VP6.1

Speed Improvements

- VP6 adds support for varying entropy encoding strategies based on complexity and/or overall frame size, including VLC coding and modified binary coding.
- A re-engineered token set improves “localization” by reducing memory footprint of entropy tables and memory bus traffic.

Quality Enhancing Features

- Supports the Heightened Sharpness Profile (VP6.2), which provides the ability to produce much sharper output files than VP6.1.
- Improvements overall frame quality consistency.
- Improvements in two-pass streaming.
- Improvements to motion compensation, included extended (long range) motion vectors and new modes, including quarter pel motion.

- New filtering strategy to improve overall sharpness.
- Improvements to motion compensation in UV planes to reduce potential color-artifacts.
- Improvements to interlace handling.
- Support for dynamically updated scan ordering.
- Provides better support for recovery frames (a frame to be received after a frame dropped in transport).
- Improvements in quality of very low bitrate (dialup rate) compressions.

User Interface Improvements

- Added a datarate control field.
- Added a keyframe frequency field.
- Added a sharpness slider.
- Added a control for automatically removing second-pass files after encoding.
- Added a VP6 Decompressor Windows control panel for configuring post processing settings.

VP6 Profiles

VP6 introduces the concept of profiles to the VP lexicon. Four profiles are currently defined for VP6.

- **Simple Profile -** designed specifically for fast playback on cheap processors. It holds the following features.
  - The ability to switch to a faster entropy encoding strategy to ensure smooth playback.
  - The ability to decode different parts of the bitstream on different sub-processors (for instance the vlx and the core), to ensure better overall system utilization.
- **General** - full D1 playback on set top boxes.
  - Disables complex adaptive prediction filter.
  - Disables adaptive sub-pixel filtering and complex filters.
  - Introduces Bifurcated bitstream.
- **Advanced Profile (VP6.1) -** designed to ensure the best quality possible at extremely low datarates (dial-up to less than 200 Kbps).
  - Best quality entropy encoding strategy.
  - Adaptive sub-pixel motion estimation.
- **Heightened Sharpness Profile (VP6.2)** - allows you to adjust the amount of subjective sharpness in the output. In our testing, this profile achieved visibly superior sharpness at much lower data rates than VP6.1.

VP6 DECODING

Hi-Definition (HD)

The VP6 Simple Profile encoding plays back HD resolutions on a PC and ¾ HDTV on Equator's 405 MHz BSP.

- **1920x1080** - a 3.5 megabit, progressive 24 frame per second clip plays back on a 2.5 GHz Intel Pentium 4 PC with a good video card (fast transfer rate and enough video memory). No restrictions on the encoder are required.
- **1280x720** - a 2 megabit, progressive clip at 24 frames per second plays back on a 1.5 GHz Intel Pentium 4 PC with a good video card (fast transfer rate and enough video memory). No restrictions on the encoder are required.
- **1280x720** - a 2 megabit progressive clip at 24 frames per second, plays back on Equator's 400 MHz MAP BSP.

Decoding Speed Compared with MPEG-4 and MPEG-2

Initial testing indicates that best-quality VP6 images have roughly the same decode complexity as the fastest MPEG-4 profiles (without B-frame prediction) at roughly the same data rate. We estimate roughly 1.5 times the complexity of MPEG-2.

Decoding Speed Compared with H.264

VP6 Advanced Profile has substantially less decode complexity than H.264 profiles. Our initial testing indicates that best-quality H.264 is roughly 4-8 times more complex than VP6 Advanced Profile. "Best-quality H.264" in this case is defined as material encoded with a profile in which the B-frame and CABAC entropy encoding options are *enabled*.

In addition, VP6 Simple Profile has significantly less decoding complexity than the *fastest* JVT/H.264 profiles. When compared using these profiles, H.264 is roughly twice as complex as VP6. "Fastest JVT H.264" in this case is defined as material encoded with a profile in which the B-frame and CABAC entropy encoding options are *disabled*.

VP6 IS READY FOR REAL-TIME

Encoding and decoding content on the same processor is often a requirement for real-time video applications. For example, you may have a video conferencing application that requires the processor to encode an outgoing stream while also decoding an incoming stream in real time. In such applications, the VP6 real-time encoder can be configured to let the application decide how much processor time to spend encoding.

Using the real-time encoder results in only a slight decrease in the quality of the compressed stream when compared with off-line VP6 encoding. This is especially noticeable in cases where the source material contains high motion, complex textures, and so on. Additional lossless entropy encoding tradeoffs may be required that will impede quality slightly.

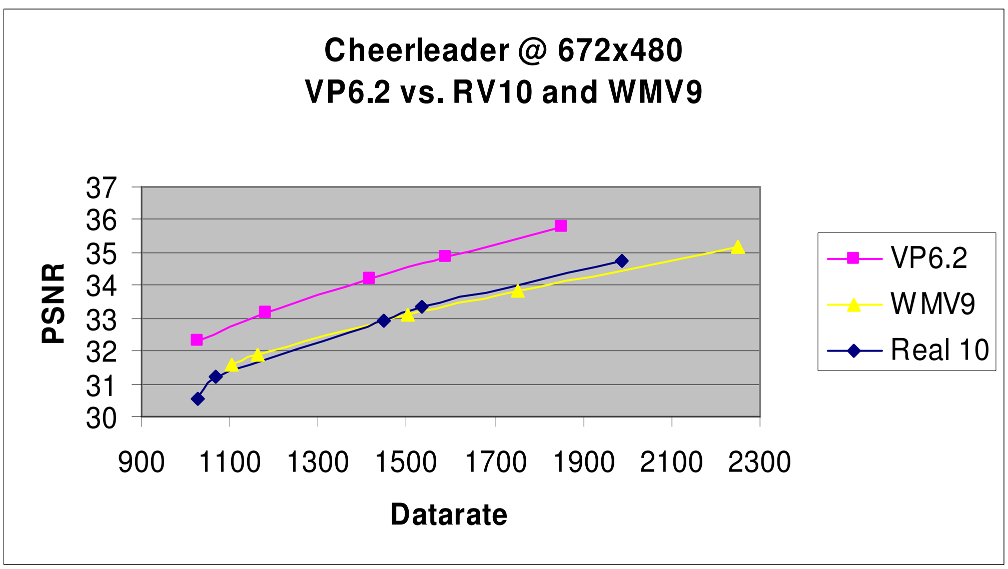
Research by On2's engineers has found that conventional symmetric encoding/decoding is possible with VP6 (although it is not yet implemented). For more information, contact your On2 Customer Support representative.

Proprietary Comparative Quality: VP6.2 vs. RealVideo 10 and Windows Media

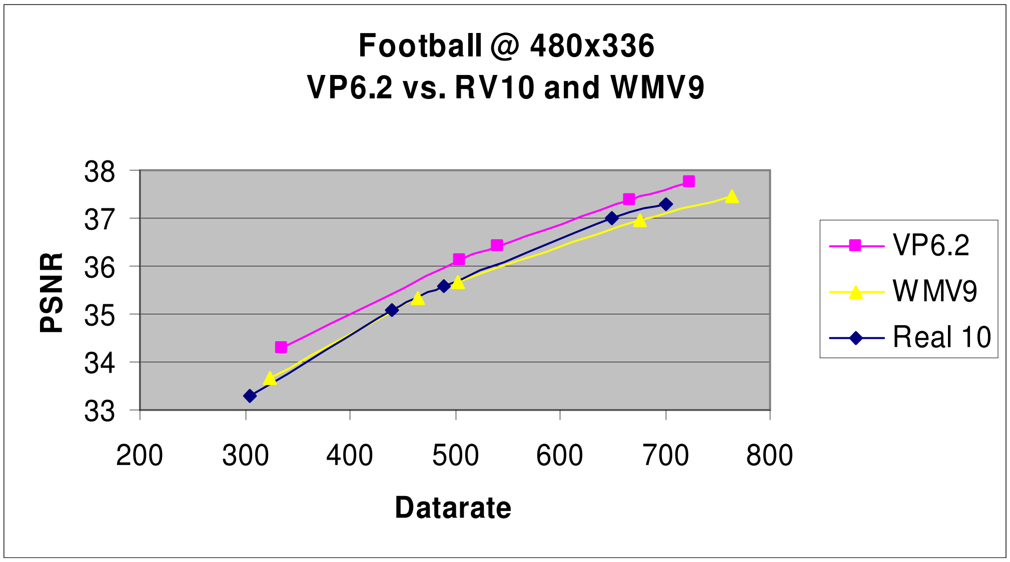
The clips used for comparison between the codecs on this page are industry standard MPEG-2 test clips. These are the clips that professionals use when making decisions. They are short, well known and extremely difficult. A compressor can't play datarate tricks with these clips (for example, throwing 5 times average datarate on hard sections).

The following graphs show PSNR as a function of datarate. PSNR is a measure of image quality. Higher PSNR means better quality images.

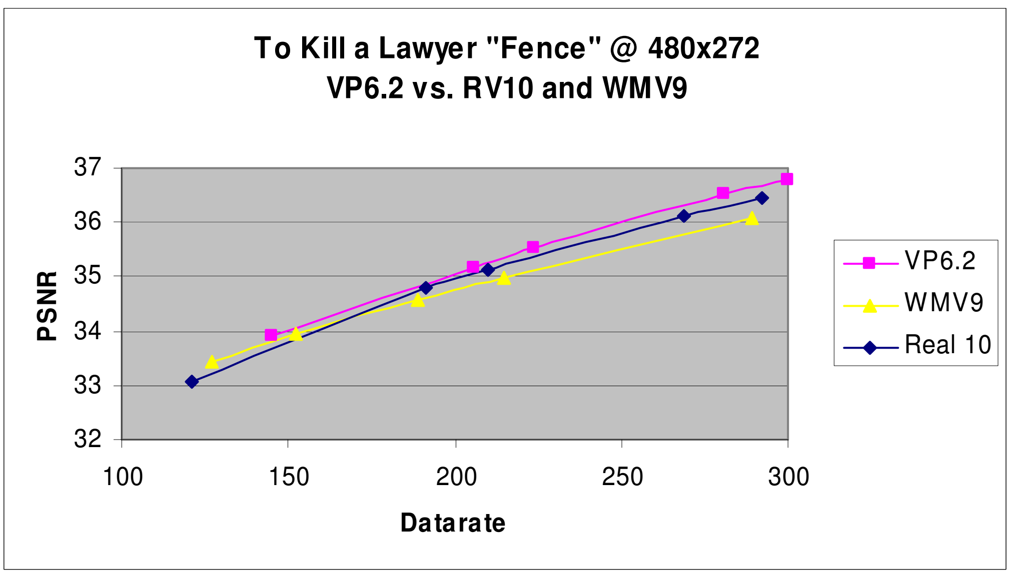
The first clip we used was a 300-frame NTSC MPEG-2 test clip called *cheerleader*. We edited the source by dropping the second of the two fields and resampling to 352x240. We also subsampled the YUY2 source to YV12 (the internal format for each of the codecs we tested).



The second clip used was a 300-frame NTSC MPEG-2 test clip called *football*. We edited the source by blending the two fields together to provide a deinterlaced source, then we subsampled the results to YV12 to ensure that we weren't testing internal color conversion code.



The third clip was a section from the full-length feature film *To Kill a Lawyer*. We named this clip *fence*.



Sample Frames

VP6 @ 571 Kbps

Windows Media 9 @ 571 Kbps

We attempted to run the MPEG-2 interlaced carousel clip through Windows Media but ran into problems. We were unable to get Windows Media 9 to provide output that had not been deinterlaced upon decompression for our PSNR graphs. As a result, we are unable to provide a PSNR graph for interlaced material comparing VP6 to Windows Media 9. Visually the VP6 was far superior to Windows Media 9 at the same datarate even with the deinterlacing enabled.

Quality Comparisons with VP6 and H.264

The test results in this section illustrate how VP6 (VP6.1) outperforms H.264 in video quality (PSNR).

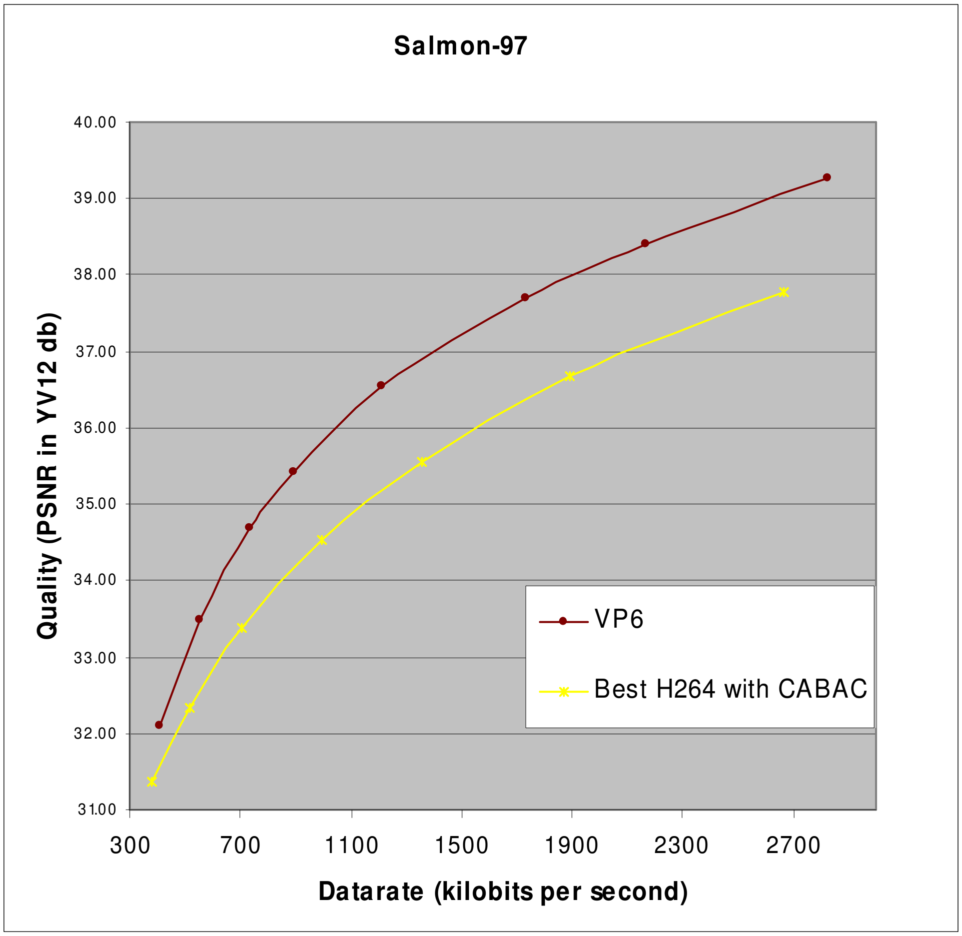
The graphs in this section show the results of these runs for from 300 kilobits/sec to three megabits/sec. We selected a range of quantizers giving results in the appropriate data rate range. In most cases the points on the H.264 curves are two quantizer steps apart. We then selected fixed quantizer values from VP6 to give roughly the same number of points across the same data rate range.

We used the following parameters in our profiles for comparing H.264 versus VP6.

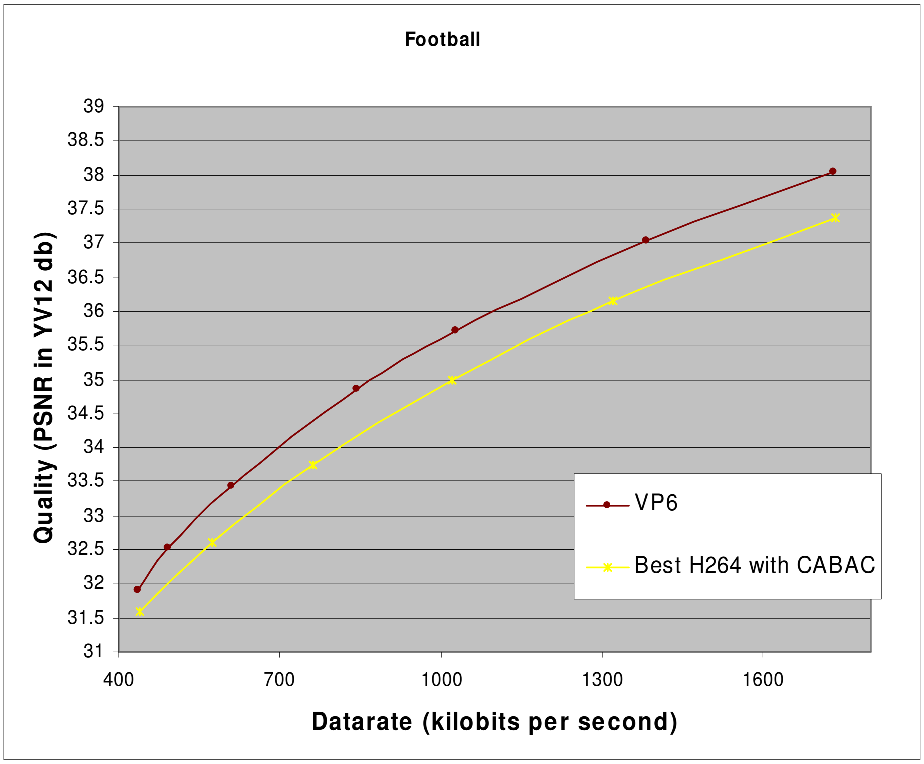
Profile	CABAC	RdOpt	Hadamard	Motion	Search Radius	B Frames	Reference Frames
Best	on	on	On	1/4 pixel	39	2	1

In order to perform fair calculations between the two codecs, we disabled data rate control in VP6. We forced the codec to use fixed quantization.

Test Clip "Salmon" (non-interlaced)



Test Clip "Football" (Non-Interlaced)



The following still images from the football sample show the superiority quality of the best VP6 profile over the best H.264 profile. Note the significantly higher number of visible artifacts in the H.264 image.

VP6 Sample

H.264 Sample

The following are selections from the still images (the arm pad of the football player), magnified by 300%.

VP6 Sample

H.264 Sample

TEST METHODOLOGY

This section explains the methodology we used to create the VP6 and Windows Media 9 files that we used to create our quality tests and PSNR graphs. A FAQ is also included to answer common questions.

To Build The Test Clips For Windows Media

1. Downloaded the video compression manager from Microsoft.
2. Using VirtualDub (http://www.virtualdub.org/), we opened a raw clip and selected Windows Media 9 as our compressor. (We also ran the same file through Windows Media Encoder using settings similar to those used above to insure that we get the same results with their standard encoder. We did get the same results.)
3. We opened the configure screen and selected the following options:

   On the first tab, we did progressive frames. We did no resizing or cropping.

On the Compression tab, we set:

- Method = One-Pass CBR. (We found that for these short clips two-pass provided worse PSNR results.)
- Bit Rate = We changed this parameter for each point in the graph.
- Buffer Size = 5000 milliseconds
- Smoother Motion - Sharper Image = 0 (with it set at higher numbers we frequently got frame drops which resulted in frames with a very poor PSNR. With the number at 0 we got every frame encoded and a much better PSNR number)
- Decoder Complexity - Complex
- Performance - All the way to the right towards Better Quality
- Key Frame Interval - 99999. The clips used for testing are short we didn't want either codec to throw unnecessary keyframes.

4. Compressed the file.
5. Opened the file we just created in VirtualDub and selected **Compression > Uncompressed RGB**.
6. Saved the file as a raw RGB file.
7. Ran the **peaksnr.exe** program on the file comparing it to the raw file.
8. Recorded the PSNR we got and the file size and plotted a point in the graph.

To Build The Test Clips For VP6

1. Installed the VP6 Video for Windows codec.
2. Using VirtualDub, opened a raw clip and selected VP61 as our compressor.
3. Opened the configure screen and selected the following options.

   On the first tab:

   - Bit Rate = We changed this parameter for each point in the graph.
   - Mode = Two-Pass Best Quality
   - End Usage = Stream From Server (to match CBR from Windows Media clips)
   - Material = Progressive
   - Noise Reduction = 0
   - Auto Keyframe = 0
   - Max Frames Btw Keys = 99999 (the clips are short and we didn't want either codec to put in unnecessary key frames)

   On the second tab

   - Undershoot = 90% (default)

- Adjust Quantizer = Checked Min (4) and Max = (63) we adjusted this number to make sure vp6 could hit the lowest datarates
- Temporal Resampling = (unchecked) with this check vp6 drops frames and has the same problem as wm9 when smooth motion is set to something other than 0
- Spatial Resampling = (unchecked)
- Peak Bitrate = 100 %
- PreBuffer = 5 (similar to Windows Media 9)
- Optimal Buffer = 5
- Max Buffer = 5

4. Compressed the file.
5. Made sure the following two parameters were set as specified via Regedit in the registry:

   HKEY_LOCAL_MACHINE\Software\On2 Technologies\VFW Encoder/Decoder Settings\VP61\strPostProcessingLevel = 4

   HKEY_LOCAL_MACHINE\Software\On2 Technologies\VFW Encoder/Decoder Settings\VP61\strCPUFree = 0
6. Opened the file we just created in VirtualDub and selected **Compression > Uncompressed RGB**.
7. Saved the file as a raw RGB file.
8. Ran the **peaksnr.exe** program on the file comparing it to the raw file.
9. Recorded the PSNR we got and the file size and plotted a point in the graph.

To Build The Test Clips For RealVideo 10

1. Downloaded the Helix RealVideo10 Producer command line program.
2. Ran Producer with the following command:

```
producer -i <in-file>.avi -o <out-file>.rmvb -ad profile.rpad
```

3. The configuration file **profile.rpad** contained the following data:

```
<audience xmlns="http://ns.real.com/tools/audience.1.0">
<avgBitrate type="uint">300000</avgBitrate>
<maxBitrate type="uint">660000</maxBitrate>
<streams>
 <videoStream>
  <pluginName type="string">rn-videocodec-realvideo</pluginName>
   <codecName type="string">rv10</codecName>
   <encodingType type="string">vbrBitrate</encodingType>
   <encodingComplexity type="string"> high</encodingComplexit>
   <quality type="uint">77</quality>
   <maxStartupLatency type="double">4</maxStartupLatency>
   <maxFrameRate type="double">29.97</maxFrameRate>
   <maxKeyFrameInterval type="double">10</maxKeyFrameInterval>
   <enableLossProtection type="bool">false</enableLossProtection>
```

```
 </videoStream>
</streams>
</audience>
```

4. Saved the file as a raw RGB file.
5. Ran the **peaksnr.exe** program on the file comparing it to the raw file.
6. Recorded the PSNR we got and the file size and plotted a point in the graph.

FAQ About the Tests and VP6

1. Why did you use VCM instead of Windows Media Encoder?

   VCM allowed us to use AVI file format, so that we could run the PSNR calculations more easily. We were aware of a method using avisynth that allowed calculation of PSNR but thought that the method was too difficult to explain on the Web site. We decided that other people could run the tests in that method.

2. Aren't both codecs using YV12 for their internal format? If so, wouldn't it be better to calculate the PSNR in YV12 and remove the color conversion from the equation?

   Yes and Yes. The reason we didn't do this is two-fold. VirtualDub does not allow for compressions directly from YV12 AVI files, and we couldn't get Windows Media VCM to decode to a YV12 file format. So we had both codecs do the color conversion. In essence we are testing the PSNR of both codecs plus their color space conversion routines.

3. Why did you do that funky stuff in the registry for VP6?

   For a couple of reasons. First, if you use the default settings the results you get aren't completely repeatable. By default VP6 monitors the amount of time it takes to decode a frame and then picks a postprocessor based on how much time it has left. Since other things might be happening on the computer at the same time, this means you might not always get the same postprocessing level and thus the PSNR differs each time you run it. In addition setting the level to **4** effectively shuts off the VP6 film grain approximation filter, which degrades the PSNR by about 0.1 db.

4. Why didn't you compare yourselves to one of the MPEG-4 derivatives?

   Again, Microsoft is aggressively attacking our market. We recognize this fact and focus most of our attention accordingly.

   In a technical sense, it is our opinion that many of the MPEG-4 derivatives are clearly inferior to Windows Media 9 and as such comparisons to VP6 are unnecessary.

   In addition, some of our competitors in the MPEG-4 space have licenses that forbid the public from using a personal-use license to post comparisons on a Web site. It would be prohibitively expensive for us to purchase commercial licenses for these codecs.

5. Why give it away for free how will you make money?

   Its only free for personal use. Anyone making any money off the codec through commercial means is expected to purchase a commercial license. By making it free for personal use we hope to gain wider adoption, and to get more media attention.

DOCUMENT REVISION HISTORY

Document Version	Description	Name/Date
1.0	Created the document.	Jim Bankoski 5/9/2003
1.1	Edited, updated charts, expanded features list.	Jim Bankoski, John Luther 5/12/2003
1.2	Updated for VP6.2	John Luther 2/17/2004