



WHITE PAPER

On2's TrueMotion VP7 Video Codec

July 11, 2008

Document Version: 1.0

On2 Technologies, Inc.

21 Corporate Drive

Suite 103

Clifton Park, NY 12065

www.on2.com

May 22, 2008

On2's TrueMotion VP7 Codec

On2 was founded in 1992 as a video compression company, and its TrueMotion line of codecs have now seen seven development generations. The codecs have gained wide adoption in major applications like Adobe's Flash Player, Skype, Viewpoint Media Player, AIM Triton, The Sims 2 from Electronic Arts, and the Move Networks media player. As a result, today On2 codec technology claims well over a billion installs. TrueMotion video formats play an important role in today's video ecosystem.

This article is an introduction to On2's most recent TrueMotion codec: VP7.

A video compressor's goal is simple: take raw video data in, and output a much smaller amount of compressed data. A decompressor later converts the compressed data back to raw video that closely resembles the original raw video.

At a high level, On2 VP7 accomplishes this goal in much the same way as other video codecs. It uses motion compensation to exploit temporal redundancy, frequency-based block transforms to exploit spatial redundancy, a loop filter to deal with block transform artifacts, and entropy encoding to exploit statistical correlation. Despite these high-level similarities, several patent-pending innovations set On2 VP7 apart.

Golden Frames

One of the first things that jumps out at new users of On2's TrueMotion codecs is the *golden frame*. Like other compressors, TrueMotion retains a recently decompressed frame to use as a predictor for the current frame. Some codecs retain the last several frames; others use a frame from the future that is decompressed out of order, and then used as a predictor (as in P frames shipping before B frames). TrueMotion codecs and VP7 in particular instead retain a frame's worth of decompressed data from the arbitrarily distant past. The codec can update any part of that frame at any future point in time. We call this secondary reference frame a golden frame, and have found many uses for it.

	References Golden Frame Buffer	Updates Golden Frame Buffer	References Last Frame Buffer	Updates Last Frame Buffer
Key (K)				
Normal (N)				
Golden (G)				
Droppable (D)				
Recovery (R)				

Figure 1: On2 VP7's Frame Types

Note: These frame types are just guidelines. Through the VP7 SDK, users control exactly when frames use or update the golden or last frame reference buffers.

Foregrounds Matter Most

The first use we found was segmentation of foreground and background video. For example, in most video conferencing applications there is a person talking in front of a static background. The speaker occludes the background but, as the speaker shifts in his seat, he reveals new parts of the background. By updating golden frames with only non-moving high quality blocks, TrueMotion codecs maintain high-quality background images despite fast-moving foregrounds.

Packet-Loss Recovery

We also use golden frames in situations where packets are lost. In typical video conferencing systems, when the receiver notices packet loss it signals the sender. When the sender receives this signal, it recovers by encoding a frame that has no dependencies on prior frames (in other words, a keyframe). Since this frame must be encoded from scratch, it's usually a very big frame, which causes choppy video playback. Alternately, it's a poor-quality frame that looks blurrier than surrounding frames, and causes a distracting visual pulse.

On2 VP7 video conferencing systems have a more palatable option: the sender can send a frame that references only the golden frame. We call this a recovery frame. Since we aren't encoding a frame from scratch, the result is better quality and a smaller frame than is otherwise possible.

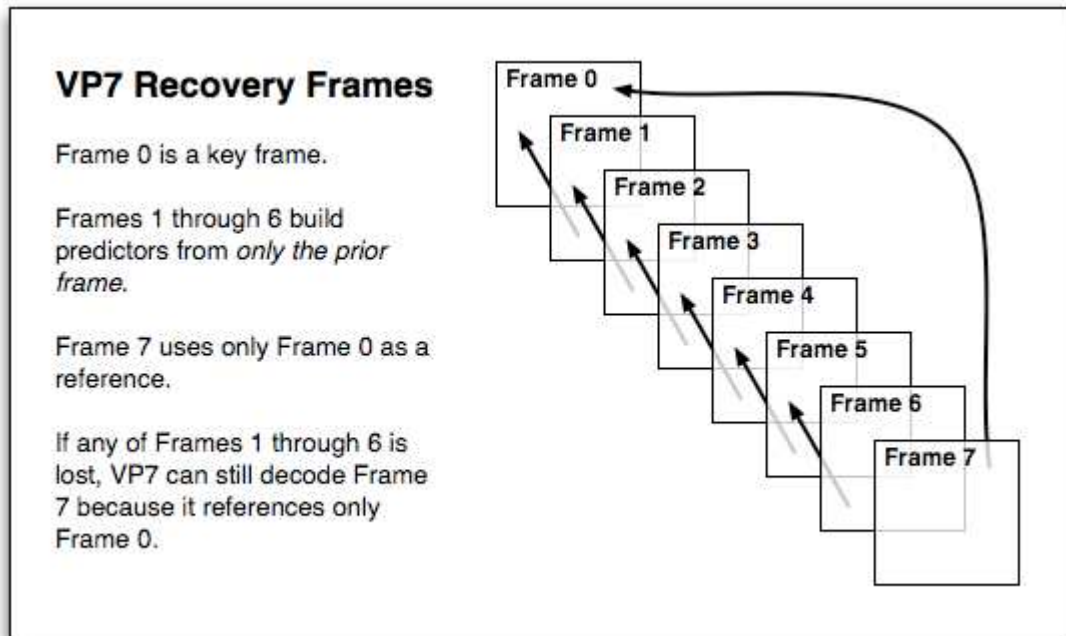


Figure 2: Using Recovery Frames to Adapt to Packet Loss

Video Conferencing

Multi-party video conferencing also exploits the golden frame. In multi-party conferencing systems, users have disparate connection bandwidth. The typical solution is to throttle bandwidth to a presumed least common denominator: all users receive data at a rate that the slowest connection can receive.

On2 VP7-based systems have an alternative. Through use of the golden frame, normal frames, and droppable frames, VP7 achieves four levels of limited temporal scalability. This means we can produce a single bitstream that degrades as needed for each party (see Figure 3). High-def parties pay no penalty for the slower connections in the conference. Best of all, there are no additional CPU costs to this approach.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
2 fps	K																	
5 fps							R						R					
15 fps			N		N				N		N				N		N	
30 fps		D		D		D		D		D		D		D		D		D

Figure 3: Sample Temporal Scalability Pattern and On2 VP7

Each stream requires all of the lower frame rates to decode. So a user with the lowest bandwidth may receive a 5 fps stream while the user with the highest bandwidth receives the full 30 fps.

Golden Quality

On2 VP7 also uses the golden frame to improve quality. In very slow moving pans or zooms, a periodic high-quality golden frame preserves image quality by restoring detail lost to repeated application of a loop filter or sub-pixel motion. The results can be quite stark (Figure 4). In other situations the golden frame serves as a good predictor for when something on screen returns to a position held earlier.

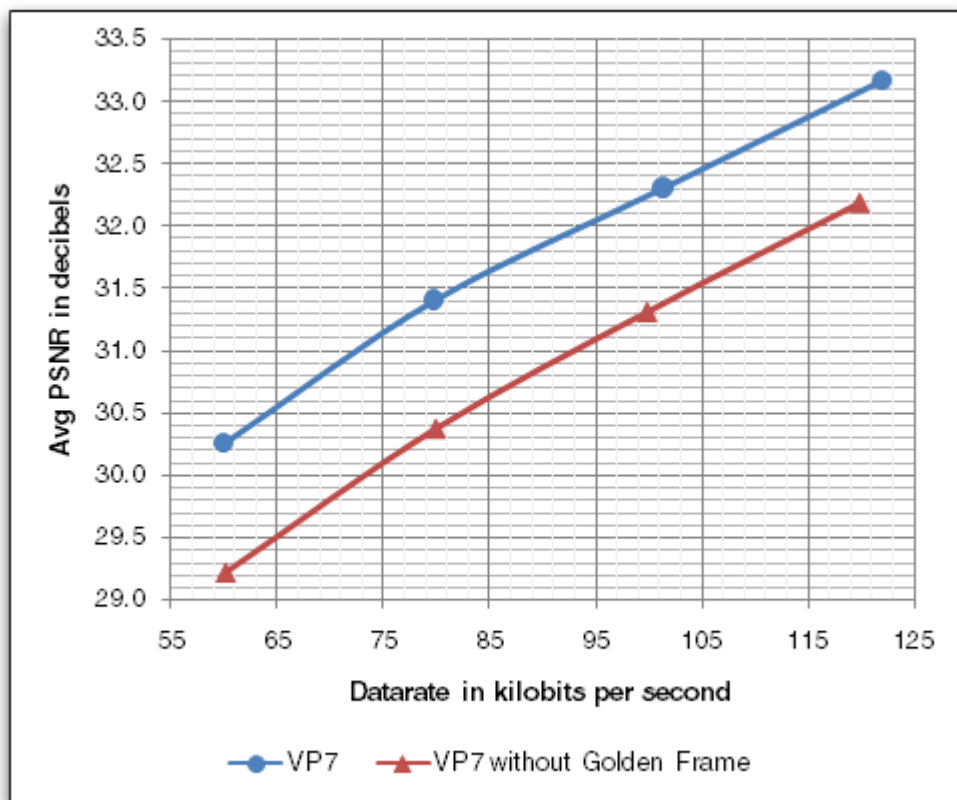


Figure 4: PSNR on Mobile and Calendar with and without Golden Frames

Realtime Quality

On2 VP7 is successful in video conferencing applications because of its outstanding video quality at low CPU usage.

In its lowest complexity mode, the On2 VP7 encoder uses scarcely 30% more cycles than the decoder — yet maintains outstanding quality. When there are plenty of available cycles, realtime VP7 can produce quality

matching the best offline encodes. This happens automatically: VP7 times each frame it encodes, and automatically adjusts its complexity to ensure the best quality possible with available cycles.

TrueMotion VP7 accomplishes this through a set of adaptive algorithms and heuristics. The algorithms determine which motion vectors and modes are most likely to produce best results. They adapt to the material, and even adapt the adaptation frequency. Modes and motion vectors that work well are tried more frequently. Modes that aren't producing good results are either shut off entirely or suppressed until the error for all modes exceeds an adaptive threshold. The thresholds and the period of time between adjustments are both adapted. When VP7 is laboring to compress fast enough, the thresholds and periods between trials grow. When VP7 isn't having trouble, modes are tried more frequently.

In extreme cases where VP7 has more time than necessary, it reverts to its slowest and best possible modes, even to the point of allowing full search and rate distortion optimization. At the opposite extreme it's possible for VP7 to produce its results by doing only five or six motion searches on an entire frame, and checking only two or three different modes per macroblock.

Leave Real-Enough Alone

To ensure a realtime mode that's outstanding for static-camera video conferencing, On2 VP7 makes heavy use of the fact that sometimes prediction is near perfect. If the predictor VP7 has found (by means of motion or mode search) very nearly matches the raw frame it's encoding, VP7 avoids doing a lot of work. In that case, VP7 foregoes a forward transform, quantization, tokenization, dequantization, inverse transform, and reconstruction code. The only remaining operation needed is to stuff a set of tokens into the bitstream that represent all 0s, and to copy predictors directly into the frame buffer. If this happens often, the encoder can actually run faster than the decoder, which can't shortcut reading tokens from the bitstream.

This technique can have a profound effect on perceived video stream quality. It denoises the background by ignoring minor changes (fluctuations due to camera noise). It also allows the encoder to focus all available cycles on the small portion of the video screen containing significant changes: the part that's moving. As a result, the parts of the image that viewers care most about get the most cycles — namely, the speaker's face.

Decoder Complexity

To assure decode speed, the On2 VP7 bitstream has an innovative, low-complexity design.

Like other codecs, VP7 uses an in-loop deblocking filter (commonly called a loop filter) to address the issue of artifacts around block edges. This approach applies an adaptive one-dimensional blurring / low pass filter across block boundaries. VP7's filter is designed to work solely in character arithmetic, with any intermediate operation

that might otherwise outrange (exceed character boundaries 0-255) getting clamped. This ensures maximum width of SIMD instruction use and means that if a given processor has support for 64 bit SIMD, VP7's loop filter can be made to run nearly eight times faster than the same operation would in C.

On2 VP7's prediction filters are simpler than many other codecs. A predictor is always created from a single frame, and never by interpolation between forward and backward reference frames. If any subpixel motion is used, it's done with a single stage filter that's applied to that single frame.

Simplified Entropy Logic

On2 VP7 also employs a simpler entropy encoding style than other advanced codecs. Entropy decoding involves simple character-based arithmetic, which can be calculated with either a multiply or several lookups. While VP7 uses sophisticated and adaptive context modeling, it doesn't adapt as each bit is parsed from the bitstream.

Flexible Decode Implementation

On2 VP7's bitstream is partitioned in a way that preserves many options when building a fast decoder. All of the modes and motion vector information are stored in one substream, and all of the residual information is stored in another, separate substream. The job of creating an entire predictor frame can be separated and run on a separate core from the processor that's parsing and producing the residual for later reconstruction. Or, a VP7 decoder can be run more traditionally, by handling one macroblock at a time, pulling single mode and motion vectors from one substream, and then the residual from that macroblock from the other. Any compromise at all between the two extreme options can be made, to insure low data and instruction cache miss rates.

Conclusion

This is only an introduction to VP7's advantages. Thanks to the techniques described here — and others not discussed — VP7 has seen exceptional marketplace adoption. Golden frames, outstanding realtime quality, and low decoder complexity combine to make On2 VP7 a leading choice for the networked video we use today, and future applications not yet invented.