

ECE/CS 572

Mid-Report

Ogg Video Coding

Joe Crop
Alex Erwig
Vivek Selvaraj

May 6th, 2010

1. Introduction

The follow report will cover the open-source video codec known as Ogg Theora. Ogg Theora, like most video codecs is lossy. It is based on the VP3.1 source code donated by On2 Technologies. The key difference between the two codecs is that VP3 can actually be losslessly transcoded if desirable, whereas Ogg Theora cannot. The report will go on to describe the Ogg Theora codec and its practical implementation, political ramifications of the open-source versus commercial video coding, and simulation/implementation objectives of this report.



1.1 Format Specifications

Generally speaking, Theora currently supports progressive video data of arbitrary dimensions (up to 1048560 x 1048560) at a constant frame rate in one of several Y-Cb-Cr color spaces. Three different chroma sub-sampling formats are supported: 4:2:0, 4:2:2, and 4:4:4. The Theora format does not support interlaced material, variable frame rates, bit-depths larger than 8 bits per component, nor alternate color spaces such as RGB or arbitrary multi-channel spaces [1].

Theora is a block-based lossy transform codec that utilizes an 8 x 8 Type-II Discrete Cosine Transform and block-based motion compensation. This places it in the same class of codecs as MPEG-1, -2, -4, and H.263. However, Theora only supports I-frames and P-frames. B-frames, such as those found in MPEG codecs cannot be used. sequence, decodes them, and synthesizes a facsimile of the original video frames. Theora is a free-form variable bit rate (VBR) codec, this also implies that the decoder is entirely in charge of error detection/correction in the bit-stream.

2. The Ogg Thera Codec and Encoding/Decoding Methods

As mentioned above, the Theora codec uses 8 x 8 blocks per frame. 4 x 4 sets of blocks are organized in groups called super-blocks, and 2x2 sets of blocks are interpreted as macro-blocks. Frames have a width and height that are subsequently multiples of 16. However, inside frames there can be regions called picture regions that can be any size that contain the actual video data. Picture regions can be a maximum of 255 pixels from any frame border. One interesting aspect of Theora is that it uses a right-handed coordinate system. In other words, the origin of the frame is in bottom-left corner instead of the top-left that most codecs use. For each 8 x 8 block, the luma and chroma components are sampled. The chroma portions can be sub-sampled however, the luma portions can never be sub-sampled.

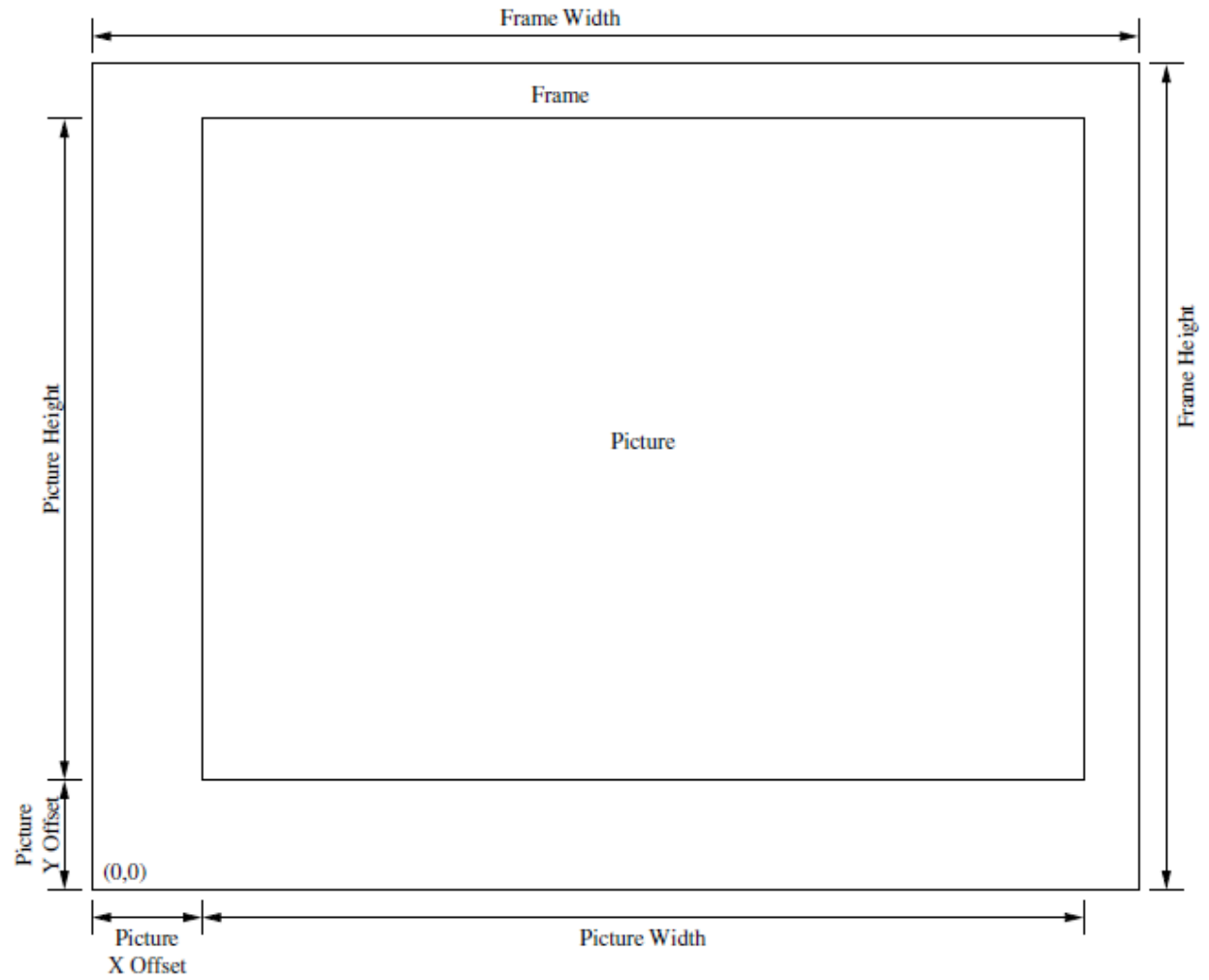


Figure: Theora frame and picture regions [1]

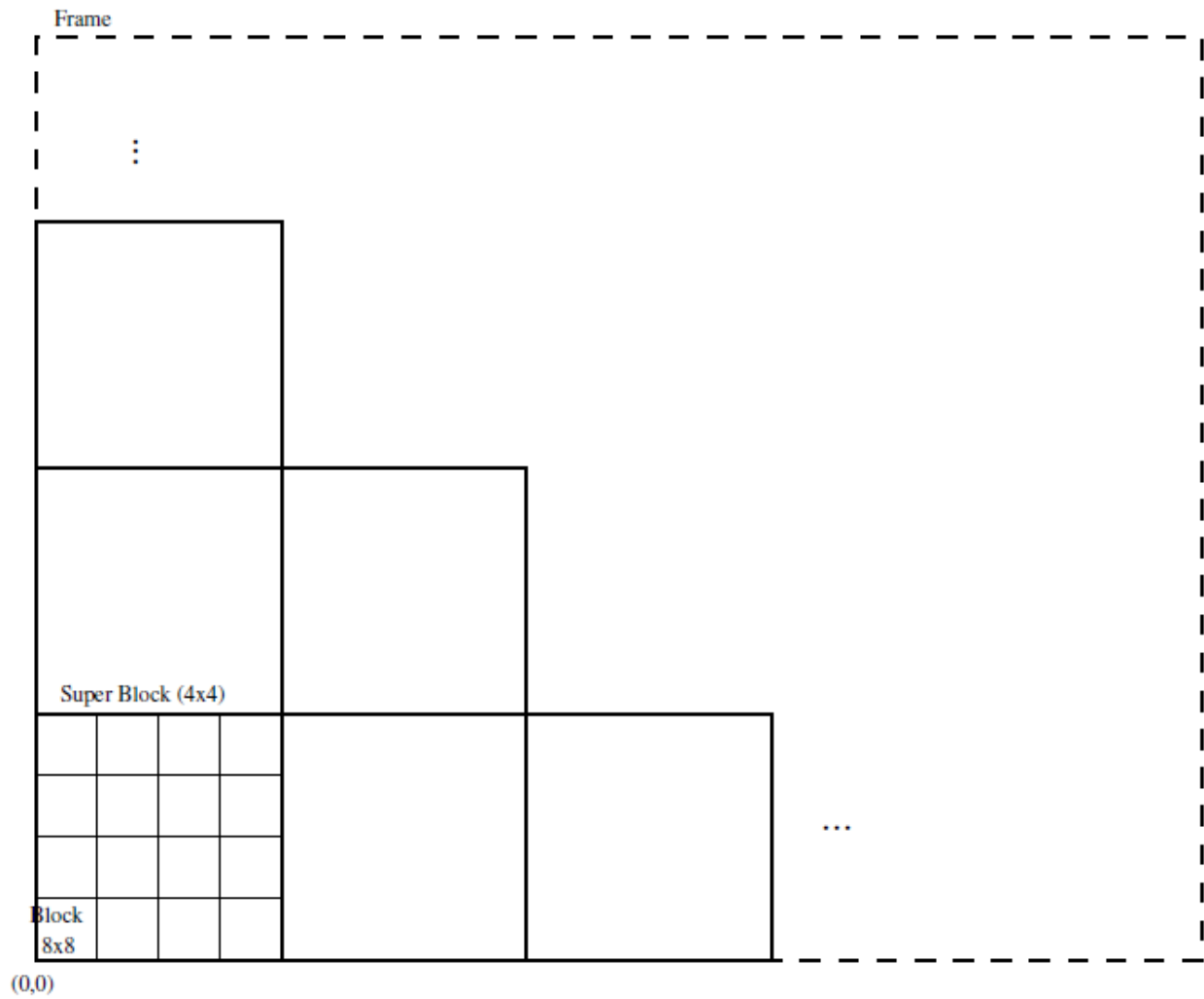


Figure: Theora blocks and super-blocks diagram [1]

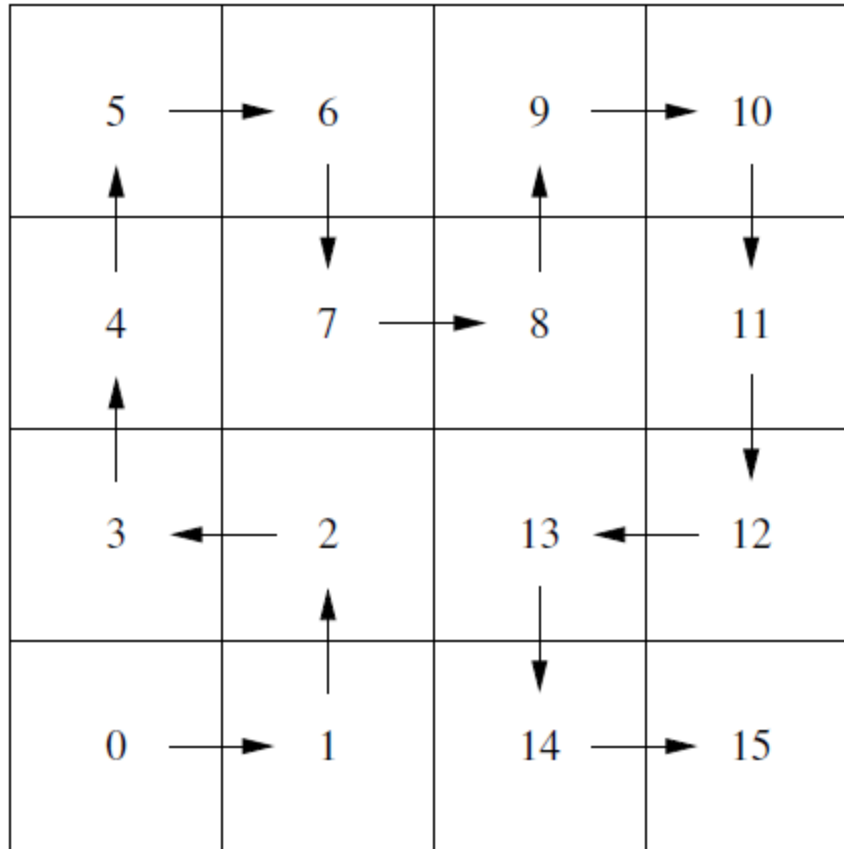


Figure: block ordering within super-block (Hilbert curve) [1]

2.1 Motion Prediction and Coding Modes

A block is predicted using one of two reference frames, selected according to the coding mode. A reference frame is the fully decoded version of a previous frame in the stream. The first available reference frame is the previous intra frame, called the golden frame. The second available reference frame is the previous frame, whether it was an intra frame or an inter frame. If the previous frame was an intra frame, then both reference frames are the same [1].

2.2 DCT Coefficient Encoding

There have been many improvements that Ogg Theora has made over the conventional MPEG coding scheme. The most notable improvement has been on Discrete Cosine Transform coefficient encoding. When coding DCT coefficients, a more advanced token set is used over MPEG. The primary change is the inclusion of tokens that represent runs of multiple zeros and ones as they are of frequent occurrence. The table below shows the standard DCT coefficient

Token Value	Extra Bits	Number of Coefficients	Description
7	3	1...8	Short zero run.
8	6	1...64	Zero run.
9	0	1	1.
10	0	1	-1.
11	0	1	2.
12	0	1	-2.
13	1	1	± 3 .
14	1	1	± 4 .
15	1	1	± 5 .
16	1	1	± 6 .
17	2	1	$\pm 7 \dots 8$.
18	3	1	$\pm 9 \dots 12$.
19	4	1	$\pm 13 \dots 20$.
20	5	1	$\pm 21 \dots 36$.
21	6	1	$\pm 37 \dots 68$.
22	10	1	$\pm 69 \dots 580$.
23	1	2	One zero followed by ± 1 .
24	1	3	Two zeros followed by ± 1 .
25	1	4	Three zeros followed by ± 1 .
26	1	5	Four zeros followed by ± 1 .
27	1	6	Five zeros followed by ± 1 .
28	3	7...10	6...9 zeros followed by ± 1 .
29	4	11...18	10...17 zeros followed by ± 1 .
30	2	2	One zero followed by $\pm 2 \dots 3$.
31	3	3...4	2...3 zeros followed by $\pm 2 \dots 3$.

Figure: DCT Coefficient Table [1]

2.3 Theora Headers

Before decoding can begin, a decoder MUST be initialized using the bit-stream headers corresponding to the stream to be decoded. Theora uses three header packets; all are required, in order, by this specification. This leads to a minor disadvantage over other, more popular codecs because the header size is often much larger.

The three headers are the: identification header, comment header, and setup header. The

identification header identifies the stream as Theora, provides a version number, and defines the characteristics of the video stream such as frame size. The comment header includes user text comments and a vendor string for the application/library that produced the stream. Finally, The setup header includes extensive codec setup information, including the complete set of quantization matrices and Huffman decode tables needed to decode the DCT coefficients [1]. Headers are described in more detail in chapter 6.

2.4 The Decoding Process

After decoding the headers of the video stream, the video-data decoding process begins. Theora decoding is very similar to the MPEG standard. It is done in the following process:

- Decode coded block information (inter frames only).
- Decode macro block mode information (inter frames only).
- Decode motion vectors (inter frames only).
- Decode block-level q information.
- Decode DC coefficient for each coded block.
- Decode 1st AC coefficient for each coded block.
- Decode 2nd AC coefficient for each coded block.
- :
- Decode 63rd AC coefficient for each coded block.
- Perform DC coefficient prediction.
- Reconstruct coded blocks.
- Copy uncoded blocks.
- Perform loop filtering.

These steps don't necessarily have to be in the prescribed order. however, the order is obviously dependent upon order of the data that has streamed.

3. The Politics of Ogg

The current state of web video standards is something of a battleground. The major players in this battle are Apple, Microsoft, Google, Mozilla, and Opera. Apple has long supported h.264 as the future standard, and is a major force in championing it as the video standard for HTML5, the world wide web standard. This is significant, as every browser compliant with HTML5 will have native support for h.264. Apple's support is unsurprising since Apple is a member of the MPEG-LA, the patent holders of h.264. Microsoft has joined forces with Apple in this regard has recently announced its support for h.264 by announcing exclusive native support for it in IE9. [2]

Opposed to these major tech titans are a number of smaller companies, including browser creators Mozilla and Opera Software, and some content providers like Wikimedia. This group is

fighting to have HTML5 include a free, open source codec for video. Mozilla has made their intentions very clear. They will never have native support for h.264 in their Firefox because the software laws in many countries would require them to have an h.264 license, and not only is the cost for this license high, it would not transfer to other firefox users such as the numerous Linux distributors.[3] Apple and Microsoft, being licensors of the h.264 rather than licensees, have much different terms. Firefox and Opera both have native support for Theora and together account for 27% of the market.

The tech giant Google represents yet another major factor. Google's browser Chrome currently supports both h.264 and Theora natively. As the largest source of digital video on the web (Youtube), they are in a strong position. They have also been long supporters of open source code and open standards. However, they also have a potential dark horse in the race with their recent acquisition On2 technologies. On2 was the creator of the VP3 codec on which Theora is based. They have continually evolved their own technology with their latest iteration, VP8, which may also be better than h.264. There is a strong possibility that Google will open source this codec, creating a potential 3-way battle.[4]

4. Simulation Objectives

Given that h.264 is an established codec with a large body of code and optimizations, it is expected to be generally superior to Ogg Theora. The fundamental question is whether it is 'good enough' for web video. Even the just the availability of a competing standard could be enough to ensure the MPEG-LA consortium will not attempt to charge license fees for web-delivered non-commercial content. This is the question we intend to answer.

There are numerous comparisons between h.264 and Theora available on the internet already, many are old and no longer reflect the current state between the codecs, or are flawed in some way. The goal is to compare the standards in as fair and objective manner as possible using the latest encoders.

To accomplish this, we will encode uncompressed video with both codecs at different quality settings. Quality settings will be used rather than bitrate settings in order to isolate either codec's ability to maintain a given bitrate. Since we give the primary consideration to web video which can be buffered, bit rate is not directly relevant. The compressed videos will be objectively compared using the Structural SIMilarity index (SSIM), an image quality comparison method based on psycho-visual error sensitivity.[5]. The compression options will be adjusted until we have an optimally encoded Theora and h.264 video at the same image quality. Then the total file size will be compared. This comparison will also allow us to highlight the types of compression artifacts both codecs tend to produce. At minimum, we will need to encode both a slow moving scene and a fast moving scene. We also need to know the relative compression capabilities at near high quality and low quality.

The SSIM index calculates the overall image quality compared to a reference image based on visual perception factors. Mean squared error is a poor way to gauge image quality as the following picture shows. All of the converted images of the baby have the same error, but they differ substantially in image quality.

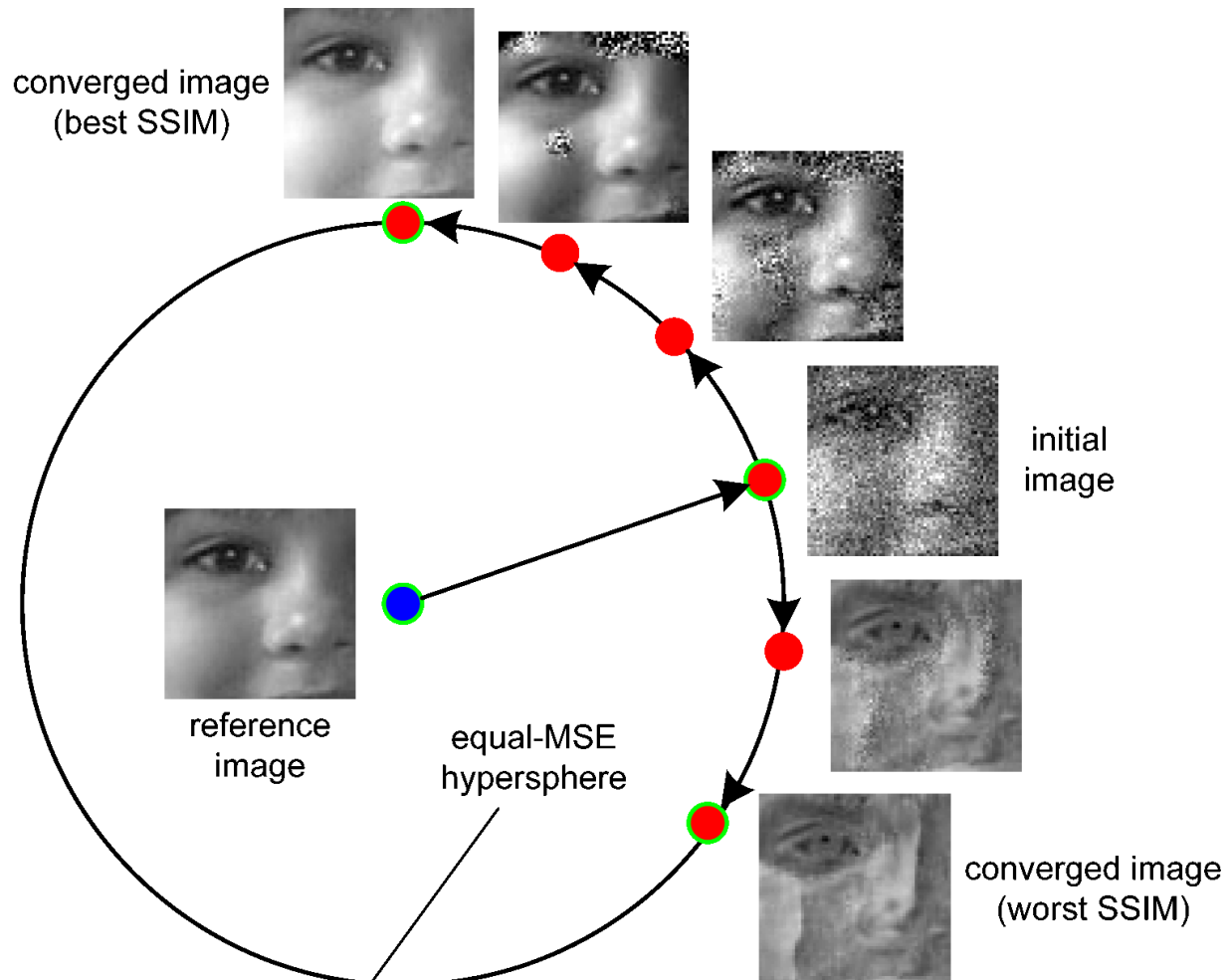


Figure: Examples of SSIM image qualities [6]

5. Our theora_stat Tool

Our goal is to design and implement a program that acts much like the mpeg_stat program for OGG Theora encoded videos. The theora_stat tool would be implemented to parse through a theora encoded video and provide statistical information about the video. The tool would be invoked in command line with the video file as one of the parameters and other optional parameters. The tool would derive the following data –

- length of the video
- Frames per second

- Compression rate
- Bit rate
- GOP/frame type sequence
- Quantization scale
- Number of Macro blocks
- Number of Intra & Inter frames
- And other relevant data that can be extracted.

5.1 Theora Header Formats

There are four types of packets/headers in the Theora specification [1]. All Theora bit-streams begin with three header packets –

- the identification header (HEADERTYPE = 0x80)
- the comment header (HEADERTYPE = 0x81)
- the setup header (HEADERTYPE = 0x82)
- Then there would be any number of Frame headers (video data packets) depending on the length of the video. Here, the MSB of HEADERTYPE is unset.

5.2 Identification Header

The Identification header is 42 bytes in length. The following information can be extracted from the Identification header –

- The major version number.
- The minor version number.
- The version revision number.
- The width of the frame in macro blocks.
- The height of the frame in macro blocks.
- The total number of super blocks in a frame.
- The total number of blocks in a frame.
- The total number of macro blocks in a frame.
- The width of the picture region in pixels.
- The height of the picture region in pixels.
- The frame-rate numerator.
- The frame-rate denominator.
- The pixel aspect-ratio numerator.
- The pixel aspect-ratio denominator.
- The color space.
- The pixel format.
- The nominal bitrate of the stream, in bits per second.
- The quality hint.
- The amount to shift the key frame number by in the granule position

5.3 Comment Header

The comment header is a variable length UTF-8 encoded string that gives information about the video file like title, artist, version, date, location, copyright, license, organization, director, producer, composer, actor, tag and description.

5.4 Setup Header

The Setup header has information like base matrices, scale values used to build the de-quantization tables and the Huffman tables used to unpack the DCT tokens.

5.5 Frame Header

The following information can be gathered from the frame header –

- Frame type (FTYPE): INTRA/INTER
- The number of qi values (NQIS):
- QIS array. The qi values will be used for all DC coefficients in all blocks. The AC coefficients can be de-quantized using any qi value on the list, selected on a block-by-block basis.
- Run-length encoded sequence for the block coded flags and the block-level qi values.
- an array indicating the coded macro blocks
- motion vectors for inter-frames

6. References

- [1] Xiph.org Foundation "Theora Specification" August 5, 2009. [Online]. Available: <http://theora.org/doc/Theora.pdf> [Accessed: May, 5, 2010]
- [2] S. Shankland. "Microsoft takes H.264 stand in Web Video Debate," April 30, 2010. [Online]. Available: http://news.cnet.com/8301-30685_3-20003838-264.html [Accessed: May 6, 2010]
- [3] M. Shaver, "HTML5 Video and Codecs," January 23, 2010. [Online]. Available: <http://shaver.off.net/diary/2010/01/23/html5-video-and-codecs/> [Accessed: May 6, 2010]
- [4] R. Lawler, "Google to Open Source VP8 for HTML5 video," April 10, 2010. [Online]. Available: <http://newteevee.com/2010/04/12/google-to-open-source-vp8-for-html5-video/> [Accessed: May 6, 2010]
- [5] Z. Wang et al, "Image Quality Assessment: From Error Visibility to Structural Similarity" IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 13, NO. 4, APRIL 2004. [Online] Available: <http://www.cns.nyu.edu/pub/eero/wang03-reprint.pdf>.
- [6] Z. Wang et al, "Maximum SSIM" [Online]. Available: http://www.ece.uwaterloo.ca/~z70wang/research/ssim/maxmin_SSIM.gif [Accessed: 6 May, 2010]