Understanding H.264 Video

Document ID: IC-COD-REP020-1.3-Approved

Dated: 11/06/2008

Status: Approved

Distribution: External: All

                Internal: All

Firmware: 9000 3.5.0

This document provides a basic introduction, to all levels of reader, to H.264 Video, officially known as ISO/IEC 14496-10, in the context of the IndigoVision 9000 series. The document will introduce some of the basic concepts of H.264 video and show how they relate to the IndigoVision 9000 products. Some of the H.264 coding techniques will also be briefly discussed.

Signatory Page

Codec Manager....…………………………………………...…Date: .....…......
Dr Mike Smart

Chief Technical Officer....…………………………….………..…Date: .....…......
Dr Barry Keepence

Contents

1   Introduction

IndigoVision 9000 series of products use H.264 video compression. This document introduces some of the key concepts of H.264 video in relation to the IndigoVision 9000 series. The document also explores some of the fundamental parts of H.264 video coding.

For a more in-depth analysis of video coding and H.264 there are several good introductory references [1][2]. Another good source of reference material is the MPEG Industry Forum [3].

This document refers to the IndigoVision 3.5.0 9000 firmware.

1.1 References

[1] "Video coding: an introduction to standard codecs", M. Ghanbari, IEE, 1999.

[2] "H.264 and MPEG-4 Video Compression. Video Coding for Next-generation Multimedia", I. Richardson, John Wiley, 2003.

[3] http://www.m4if.org/

[4] ISO/IEC 14496-10 Information technology – Coding of audio-visual objects – Part 10: Advanced video coding, First Edition, 1st December 2003.

[5] "Understanding MPEG-4 Video ", IC-COD-REP012, 11th June 2008.

[6] "Video Resolution and TVL", IC-COD-REP019, 16th November 2006.

[7] "Understanding ACF", IC-COD-REP011, 3rd June 2008.

[8] "Understanding Rate Control", IC-COD-REP007, 12th June 2008.

[9] "Understanding Analytics", IC-COD-REP017, 11th June 2008.

[10] "Understanding Frame Rate", IC-COD-REP008, 16th June 2008.

1.2 Terminology

Term	Definition
ACF	Activity Controlled Frame rate
CBR	Capped Bit Rate
FPGA	Field Programmable Gate Array
ISO	International Organization for Standardization
ITU	International Telecommunication Union
MPEG	Moving Picture Experts Group

Table 1: Terminology

2   Introducing H.264

H.264 (ISO/IEC 14496-10) is the latest official video compression standard, which follows on from the highly successful MPEG-2 and MPEG-4 (ISO/IEC 14496-2) video standards and offers advancements in both video quality and compression.

H.264 is also referred to as MPEG-4 Part 10, MPEG-4 Advanced Video Coding (AVC), and in the earlier stages of it's development as H.26L.

2.1 What is H.264 video?

H.264 is a video codec (**co**mpressor and **dec**ompressor) standard. A video codec is designed to compress and uncompress digital video in order to reduce the amount of bandwidth required to transmit and store the video. This is needed as the raw data rate of uncompressed CCIR601 active digital video[1] is in excess of 158Mbps – over 300 times the capacity of a 512kbps ADSL connection and only just over one hour recording on a 80GB hard disk.

Simply scaling the video, to SIF[2] resolution, and compressing with standard utilities such as WinZip or gzip could achieve 10:1 compression. However, at least 300:1 compression is needed to stream live video over an ADSL connection and to achieve 300 hours recording to a 80GB hard disk. This level of compression can be achieved with H.264.

2.2 H.264 as a standard

It is important, before looking at H.264 in some more detail, to understand the difference between making a comparison between a standard and an implementation of a standard. The two are very different. Thus when people say, "H.264 provides better video quality than MPEG-2" this is a little misleading.

H.264 is a video compression standard. The H.264 standard defines the syntax of a compliant bitstream, to which a compliant decoder must conform exactly, implementing all the necessary tools defined by the standard in order to decode the bitstream.

An H.264 encoder, conversely, can implement a subset of the syntax defined by the standard, providing it produces a compliant bitstream. Various implementations and algorithms within the encoder are also not defined by the standard, and are created by the designer of the encoder. As such different vendors H.264 encoders will produce streams of differing quality, for the same bitrate. So returning to the statement in the first paragraph, it is more appropriate to say, *"H.264 provides a richer syntax and toolset than MPEG-2 and as such allows the possibility of implementing a superior video encoder that can generate higher quality video for the same bitrate, or conversely, can generate the same quality video at a much lower bitrate".*

This can be demonstrated using the reference software encoder (JM11) freely available from the International Standards Organization (ISO) as an example

---

[1] 720x480 pixel 4:2:2 video at 30fps
[2] 352x240 pixel 4:2:0 video at 30fps

implementation of an H.264 encoder. The reference encoder allows a user to select which tools to use in order to encode a particular video sequence. Table 2 shows the result of encoding an identical video sequence using the H.264 reference encoder with different tools. Each output bitstream from each test is a fully compliant H.264 bitstream and each bitstream is of equivalent video quality.

Tools	Bitrate (kbps)	Execution time[3] (relative)
I-frame only encoding	2279	1
I and P-frames but with no motion estimation (0 search range)	1055	1.5
I and P-frames with a +/-16 search using a simplified search algorithm	453	14
I and P-frames using a full search algorithm with differing block size motion compensation	421	56

Table 2: Effect of tools on bitrate and complexity

Table 2 clearly shows that the more tools and algorithms that are used the greater the compression achieved for the same quality of video. However it is also clear that the addition of tools comes at the expense of increased complexity – in this case measured by the execution time of the encoding process. It is this increase in complexity that often causes some tools or algorithms to be omitted from the design of an H.264 encoder.

2.3 Relationship to MPEG-4 Part 2

MPEG-4 (ISO/IEC 14496) is a collection of standards defining the coding of audio-visual objects. The collection is divided into a number of *parts* describing video compression and audio compression standards, as well as system level parts, describing features such as the MPEG-4 file format. The video compression standard found in many products today, such as in the IndigoVision 8000 product series, is the traditional DCT-based MPEG-4 Part 2 (ISO/IEC 14496-2) standard.

The H.264 video compression standard has been incorporated into MPEG-4 as MPEG-4 Part 10 (ISO/IEC 14496-10). This means MPEG-4 now has two video compression standards available. However, these two video compression standards are non-interoperable, with each standard using different methods to compress and represent the data i.e. an MPEG-4 Part 10 (H.264) decoder cannot decode an MPEG-4 Part 2 bitstream, and vice versa.

---

[3] Total execution time of required coding tools only.

3   H.264 Coding

This section explores H.264 compression in a little more detail. However, this is still only a basic introduction to aid users of 9000 H.264 products. For a more in-depth discussion of H.264 and video coding see the references [1][2].

Inside a 9000 transmitter frames of video are captured from the camera and sent to the internal H.264 encoder to be compressed. Each frame of video is then compressed in one of two ways: as an I-frame or as a P-frame.

An I-frame is a video frame that has been encoded without reference to any other frame of video. A video stream or recording will always start with an I-frame and will typically contain regular I-frames throughout the stream. These regular I-frames, also called *intra frames, key frames* or *access points*, are crucial for the random access of recorded H.264 files, such as with rewind and seek operations during playback. The regularity of these I-frames is known as the I-frame interval. The disadvantage of I-frames is that they compress less than P-frames.

P-frames are *motion-compensated* frames: that is to say the encoder makes use of the difference between the current frame being processed and a previous frame of video, ensuring that information that does not change, e.g. a static background, is not repeatedly transmitted. Unlike purely difference-based codecs, such as delta-MJPEG, H.264 not only looks for differences but searches for motion that has occurred in the video. This means that motion-compensated codecs will typically outperform simple difference-based codecs when there is motion. The process of searching for motion is known as *motion estimation*.

3.1 Encoding an H.264 I-frame

This section explores the process of encoding a frame as an intra-frame.

Every frame of video to be encoded as an I-frame is subdivided into a series of 16 by 16 pel-sized non-overlapping blocks called *macroblocks.* Each macroblock is encoded by the H.264 encoder using two main processing units: Transform/ Quantization and Entropy coding, as shown in Figure 1. This produces the H.264 I-frame part of the bitstream.

However, before looking at these processing units in more detail it is important to note in Figure 1 that each macroblock is also decoded or *reconstructed*, within the encoder using the inverse transform/quantization and deblocking stages. This reconstruction process is required in order to encode subsequent frames as P-frames.
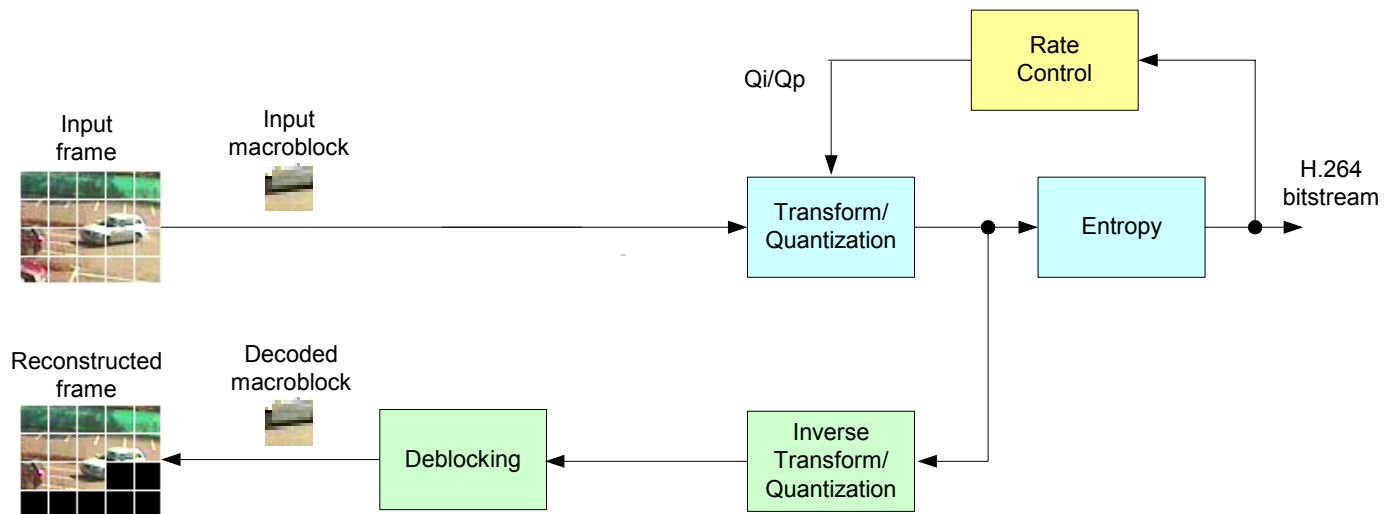
Figure 1: Simplistic view of encoding an H.264 I-frame

3.2 Encoding an H.264 P-frame

The previous section provided a very basic explanation of how an H.264 I-frame is encoded. This section examines the process of encoding a frame as a P-frame and how compression can be greatly improved by the use of *motion compensation*.

Figure 2 shows the encoding of an H.264 P-frame. As described previously motion compensation makes use of similarities that exist between the current input frame and a previously encoded frame. This previously encoded frame is called the *reference frame* and is in fact a previously reconstructed frame[4].

Motion estimation is the process of examining the reference frame in the locale of the input macroblock for a set of pixels that closely match the input macroblock. In the example shown in Figure 2 the motion estimation unit has found a relatively close match 8 pels to the left of the input macroblock in the reference frame. The displacement between the input macroblock and the point where the best match was found is known as the *motion vector*.

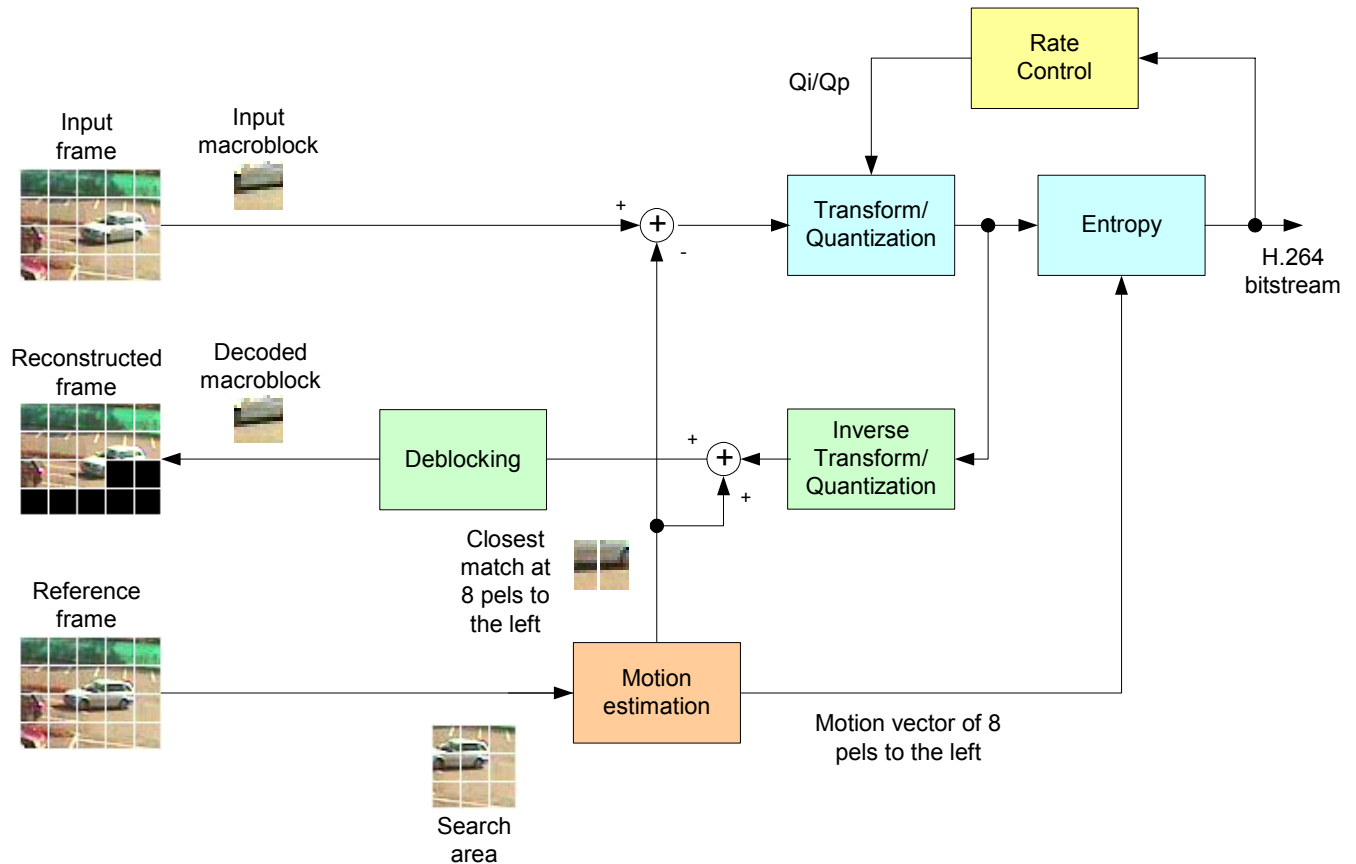[4] The example shows a car reversing from a space in a parking lot.

Figure 2: Simplistic view of encoding an H.264 P-frame

Once a good match has been found the difference between the input macroblock and the closest match found by the motion estimation unit is computed. It is this difference, or *error*, macroblock that is then encoded by the transform/quantization and entropy stages. Combined with the motion vector information the H.264 P-frame part of the bitstream is generated.

Once again however the reverse path decodes the encoded macroblock. The decoded error macroblock is then added to the closest match found by the motion estimation. The deblocking unit then filters the result in order to form the reconstructed frame.

3.3 Motion estimation

The motion estimation unit is worth further mention because it is one of the most computationally expensive parts and critical to the performance of the H.264 encoder.

As stated in the previous section the motion estimation examines the reference frame for similarities to the input macroblock. The result of this search is generally one of three: an exact match has been found, a close match has been found or no match has been found. The previous section demonstrated what happens when a close match is found.

In the case where an exact match is found only the motion vector needs to be transmitted, and no error macroblock is coded. In the case where no match is found the input macroblock has to be encoded as an intra macroblock, as in Figure 1. Of course, the latter case is not very efficient.

The area in which the motion estimation search is completed is known as the *search area* and the size of this search area is determined by the *search range*. Clearly, the greater the search range the greater the chances of finding a good match. The method of performing the search is known as the *search algorithm*. Finally, it is possible to search quite finely around the closest match to find an even better match using a process called ¼-*pel motion estimation*.

Motion estimation is a complex procedure and often encoders, especially real-time software or DSP-based encoders, will use reduced search areas, use a restrictive search algorithm or not perform ¼-pel motion estimation in order to achieve real-time performance. However, this can often result in poor quality video and significantly reduced compression.

3.4 Transform and quantization

One of the main differences between H.264 and previous codecs, such as MPEG-4 and MPEG-2, is that H.264 no longer uses the 8x8 pel DCT. H.264 uses a much simpler and reversible transform that splits each input or error macroblock into a series of 4x4 pel blocks and then simply converts the blocks into a state more conducive to compression. However, it should be noted that this process achieves no actual compression in itself.

The second stage of the process is known as quantization and where the majority of the compression is achieved. This is also the stage where the majority of information can be lost and artefacts introduced.

The quantization process is controlled by a parameter known as Qp, where Qp can take a value between 0 and 51 inclusive. If Qp is set to 0 then the quantization unit performs little processing on the transformed data, meaning that little data is lost, quality remains high but the compression achieved is low.

As Qp increases in value the quantization unit starts removing information. However, the encoder is designed to remove only the most insignificant details first and often this lost information is imperceptible to the human eye. Quality remains good but the compression achieved starts to increase.

As Qp increases further towards the maximum value of 51, more and more information is discarded, and quality has to be sacrificed. However, compression will increase significantly as Qp increases.

3.5 Entropy encoding

The final stage in the forward path is the entropy encoder unit, also known as the variable-length encoder unit. This is a lossless process based on the statistical examination of the bitstream. Patterns that occur regularly are simply converted to a small number of bits, whereas patterns that occur irregularly are converted into a longer number of bits.

3.6 Deblocking

The deblocking filter was introduced in the H.264 standard as part of the reconstruction loop of the encoding process in order to reduce the "blocky" artefacts often noted with video encoders, such as MPEG-4, especially at the lower bitrates.

Figure 3: Effect of deblocking in H.264

Figure 3 shows the effect of deblocking on the quality of the compressed video, providing for much improved subjective quality.

3.7 Rate control

The rate control unit controls the bitrate of the bitstream generated by the H.264 encoder. It performs this task by analysing the rate at which the entropy encoder is producing data and comparing this figure with the requested target bitrate. If the entropy encoder is producing too much data the rate control unit simply raises the Qp of the quantization unit. If too little data is being produced the Qp is lowered. Remember, the larger the Qp the better the compression but the lower the quality.

Many algorithms exist for controlling Qp for optimal performance. Some of these algorithms use the option of dropping frames of video as well as adjusting Qp. These algorithms trade-off the quality of each frame with the jerkiness in the video caused by frame dropping. Further, the bitrate profiles generated by these algorithms will differ and the choice of algorithm is dependent on the network and target application.

Rate control is covered in detail in the white paper *Understanding Rate Control* [8].

3.8 Other tools

A more in-depth discussion of H.264 coding and H.264 tools and terminology, such as INTRA prediction, B-frames, CAVLC/CABAC, motion vector partitioning, multiple reference frames, unrestricted motion vectors, and data partitioning are all beyond the scope of this document. Please see references [2][4].

3.9 Decoding an H.264 bitstream

The decoding process of an H.264 bitstream is intentionally identical to the reverse path shown in Figure 2. The exception is that the bitstream is first passed through an entropy decoder before the data is passed to the inverse transform and quantization unit. Embedded motion vectors are passed to a motion compensation unit, which reads the closest match data from the decoder's version of the reference frame. Of course the encoder and the decoders reference frames are identical because the encoder has in effect a mirror of the decode process.

4   IndigoVision H.264

The previous sections have discussed H.264 in general. This section examines IndigoVision's H.264 codec and video configuration options and attempts to correlate these options with the information presented in the previous sections.

4.1 IV9105 H.264 codec

The IndigoVision 9000 products use IndigoVision's own custom designed FPGA-based H.264 hardware codec: the IV9105. The IV9105 was designed and developed by IndigoVision, as a natural development of the previously successful MPEG-4 IV8105. This new IV9105 high performance codec offers some distinct advantages

- Custom H.264 codec tailored to IndigoVision. This means the H.264 encoder is designed for video typically used by IndigoVision customers. For example, extra compression can be achieved when there is low activity in the video – a situation common in many surveillance applications.

- High performance, low cost coding of 4SIF 30fps compliant H.264.

- Reduced bitrate requirements for similar quality video over MPEG-4. Typical savings of 20-25% can be achieved, with further savings under still conditions. Actual savings dependent on video quality and content. A typical scene is shown in Figure 4.

Figure 4: Typical scene to be compressed

- Use of a hardware encoder means the time to encode every frame of video is constant, regardless of bitrate and motion. This means high-quality video can be maintained during fast-moving activity without frames being dropped.

- Efficient compression due to highly computationally complex operations such as motion estimation being performed completely in hardware, without any need for software intervention.

- Advanced video pre-filtering to help reduce noise and improve compression.

- Low processing host overhead means processing power available for value-add features such as high quality audio and analytics.

- Based on IndigoVision's extensible MainStream architecture, coupled with the programmable nature of hardware FPGAs means that developments in codec performance and functionality can be passed on to customers through firmware upgrades, in the same manner as software updates.

- An example of the extensible structure was demonstrated in the 3.3 9000 firmware release where dual-resolution encoding was introduced.

An example of the savings that can be achieved on a scene, such as the one shown in Figure 4, is demonstrated below in Figure 5. In this example the same video sequence has been encoded using four different encoders: 8000 MPEG-4, 9000 H.264, an MPEG-4 encoder with no motion estimation, and an MJPEG encoder. All were encoded at 25fps (with the exception of MJPEG at 5fps) to the same subjective video quality.
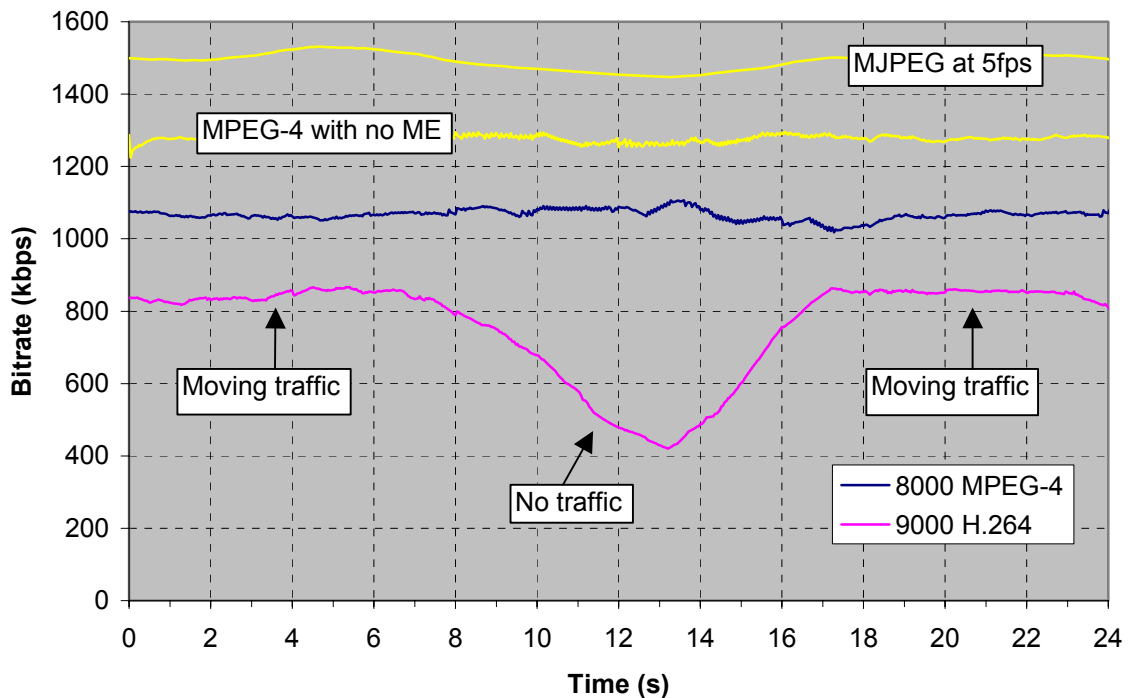

Figure 5: MPEG-4 and H.264 bitrates for equivalent subjective quality

4.2 H.264 transmitter configuration

As with the 8000 MPEG-4 series IndigoVision allows a small number of the H.264 encoder parameters to be configured via the 9000 *Encoder Configuration* web page, available on all transmitters. Figure 6 shows an example page from a 9000 transmitter.

There are a number of parameters that directly affect the H.264 encoder in the IndigoVision 9000 transmitter: Resolution, Bit Rate, Rate Control, Optimization, Frame Rate Divisor, and I-frame Interval.

Figure 6: IndigoVision 9000 video configuration

- **Resolution:** Controls the resolution of the data input to the H.264 encoder. Resolutions supported include SIF, 2SIF and 4SIF.

- **Bit Rate:** This is the requested target bitrate, in Kbps, that is sent to the H.264 rate control unit, as described in Section 3.6. Typically, increasing the bitrate increases the quality of video.

- **Rate Control:** This drop-down menu controls the algorithm used by the rate control unit described in Section 3.6. There are two modes of operation: CBR and ACF. CBR is a simple capped bitrate control where the rate controller attempts to maintain average output bitrate on or below the requested target bitrate. This rate control does not drop frames. ACF refers to Activity Controlled Frame Rate. For more details on ACF see *Understanding ACF* [7].

- **Optimization:** The rate control algorithms can work in two different ways. Optimizing for storage or optimizing for video quality. For a more in-depth description of rate control see *Understanding Rate Control* [8].

- **Frame Rate Divisor:** The frame rate divisor is used to control the frame rate of the H.264 video. Selecting a value of one means that every frame captured from the camera is passed to the H.264 encoder. Selecting a value of 2 and the video will be encoding at half of the full frame rate. See the white paper *Understanding Frame Rate* [10].

- **ACF Passive Frame Rate Divisor:** Used to control the frame rate of the video when using ACF rate control and ACF is in a passive mode. A value of 30 means for an NTSC camera the frame rate will drop to 1fps when no activity in the video is detected. For more details see *Understanding ACF* [7].

- **I-Frame Interval:** The I-frame interval controls how far apart in time the I-frames are encoded in the stream. This was discussed in Section 3.

- **Embed Analytics Data:** Select this option to add analytics configuration and real-time data into the H.264 bitstream. This embedded information can be displayed in Control Center. For more details on analytics see *Understanding Analytics* [9].

Some advanced encoder settings are possible by checking the "Advanced Encoder Parameters" box on the *Advanced Network Configuration* web page. These advanced topics are covered in detail in *Understanding Rate Control* [8].