
Understanding MPEG-4 Video

Document ID: IC-COD-REP012-1.1-Approved
Dated: 22/03/2004
Status: Approved
Distribution: External: All
Internal: All

This document provides a basic introduction, to all level of readers, to MPEG-4 Video, officially known as ISO 14496-2, in the context of the IndigoVision VB8000 MPEG-4 series. The document will introduce some of the basic concepts of MPEG-4 video and show how they relate to the IndigoVision VB8000 products. Some of the MPEG-4 coding techniques will also be briefly discussed.



IndigoVision

Confidential Proprietary Information and Copyright Notice.

"Copyright © 2004 IndigoVision Limited. All rights reserved. This document is confidential. Please do not disclose this document, in whole or in part, to any third party."

Signatory Page

Chief Technical Officer.....Date:

Codec Manager.....Date:

Revision History

Version	Date	Author	Comments
1.0	26/05/2004	Mike Smart	First draft
1.1	13/07/2004	Mike Smart	Updated after review

Document Location

`$(core\codecs\docs\reports`

Contents

1	BACKGROUND	6
1.1	References	6
1.2	Terminology	6
2	INTRODUCING MPEG-4	7
2.1	What is MPEG-4?	7
2.2	What is MPEG-4 video?	7
2.3	MPEG-4 as a standard	7
2.4	Profiles and Levels	8
2.5	Relationship to H.263	8
3	MPEG-4 CODING	9
3.1	Minding your P and I's	9
3.2	Encoding an MPEG-4 I-VOP	9
3.3	Encoding an MPEG-4 P-VOP	10
3.4	Motion estimation	11
3.5	DCT	12
3.6	Quantisation	12
3.7	Entropy encoding	12
3.8	Rate control	13
3.9	Other tools	13
3.10	Decoding an MPEG-4 bitstream	13
3.11	Common MPEG-4 artefacts	13
4	INDIGOVISION MPEG-4	14
4.1	IV8102 MPEG-4 codec	14
4.2	Configuration	14

Tables

Table 1: Terminology 6

1 Background

IndigoVision VB8000 series of transmitters and receivers support both MPEG-4 video and audio. This document introduces some of the key concepts of MPEG-4 video in relation to the IndigoVision VB8000 series. The document also explores some of the fundamental parts of an MPEG-4 video encoder and decoder.

For more in-depth analysis of video coding and MPEG-4 there are several good introductory references [1] [2] [3]. Another good source of reference material is the MPEG Industry Forum [4].

1.1 References

- [1] "Video coding: an introduction to standard codecs", M. Ghanbari, IEE, 1999.
- [2] "Image and Video Compression Standards, Algorithms and Architectures", V. Bhaskaran, and K. Konstantinides, Kluwer Academic Publishers, 1997.
- [3] "Video Codec Design, Developing Image and Video Compression Systems", I. Richardson, John Wiley, 2002.
- [4] <http://www.m4if.org/>
- [5] ISO/IEC 14496-2 Information technology – Coding of audio-visual objects – Part 2: Visual, Second Edition, 1st December 2001.
- [6] "Internet Streaming Media Alliance Implementation Specification", ISMA v1.0, 28 August 2001.
- [7] "Understanding ACF", IC-COD-REP011-1.1, 19th May 2004.

1.2 Terminology

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

Table 1: Terminology

2 Introducing MPEG-4

This section introduces MPEG-4 and more specifically MPEG-4 Video.

2.1 What is MPEG-4?

MPEG-4 covers a wide spread of technology: video is simply one part. There are three main parts to the MPEG-4 ISO 14496 standard:

- ISO 14496-1: Systems
- ISO 14496-2: Visual
- ISO 14496-3: Audio

However, there are also a large number of other parts to the standard covering a host of topics including conformance testing, reference models, file formats and new extensions to the video specification, such as the infamous part 10: Advanced Video Coding. This document covers only ISO 14496-2: Visual, and more specifically MPEG-4 coding of rectangular video¹.

2.2 What is MPEG-4 video?

MPEG-4 Video is a video codec (**co**mpressor and **de**compressor) 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 video² is in excess of 158Mbps – over 300 times the capacity a 512kbps ADSL connection and only just over one hour recording on a 80GB hard disk.

Simply scaling the video, to SIF³ 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 MPEG-4.

MPEG-4 is a lossy codec. This means that the compression and decompression does not reproduce exactly the same as the original video but achieves the high compression ratios required at the expense of some quality. Typically, the greater the compression required the greater the loss in video quality.

The MPEG-4 Visual [4] standard was first published in 1999 by the ISO. It was built on top of the success of the MPEG-1 and MPEG-2 standards, and was originally targeted at very low bitrate coding. Over the years the standard was expanded to incorporate a wide range of applications from mobile phones to broadcasting.

2.3 MPEG-4 as a standard

It is important, before looking at MPEG-4 video in some more detail, to understand the difference between making a comparison between a standard and an

¹ 14496-2 also covers the coding of non-rectangular video objects.

² 720x480 pixel 4:2:2 video at 30fps

³ 352x240 pixel 4:2:0 video at 30fps

implementation of a standard. The two are very different. Thus when people say, "MPEG-4 provides better video quality than MPEG-2" this is a little misleading.

MPEG-4 is a standard specified by the ISO. The MPEG-4 standard defines the syntax of an MPEG-4 compliant bitstream, to which the decoder must conform exactly, implementing all the necessary tools defined by the standard in order to decode the bitstream.

An MPEG-4 encoder, conversely, can implement any 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 codec. As such different vendors MPEG-4 encoders will produce streams of differing quality. Further, returning to the statement in the first paragraph, it is more appropriate to say, "*MPEG-4 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*".

2.4 Profiles and Levels

The MPEG-4 Visual specification provides a vast array of tools, which can be used for coding video for a spectrum of applications. Because a decoder that implemented every tool would be extremely expensive to design and implement a number of subsets, or *profiles*, have been defined as part of the MPEG-4 standard.

The most basic profile is called *MPEG-4 Simple Profile* and supports the decoding of simple rectangular video. This profile to date remains one of the most widely supported profiles by MPEG-4 vendors. An extension to Simple Profile is known as *Advanced Simple Profile* has also become popular [6].

Within each profile the standard defines a number of *levels*. Each level dictates a level of complexity on the MPEG-4 bitstream, such as bitrate and video resolution. This controls the complexity of the decoder. For example an MPEG-4 Simple Profile Level 3 compliant decoder must be able to be to decode a SIF (or equivalent in size) resolution MPEG-4 bitstream up to 256kbps.

2.5 Relationship to H.263

H.263 was developed by the ITU standards organisation and shares many similarities with MPEG-4. In fact MPEG-4 was originally based around the baseline H.263 specification. Indeed in the MPEG-4 Simple Profile the tool known as *MPEG-4 short header* is equivalent to baseline H.263.

Over time the two specifications have diverged. The ITU has published amendments to the H.263 specification in the form of H.263+ and H.263++, and ISO has extended MPEG-4 short header to Simple Profile and above.

Today, a compliant MPEG-4 Simple Profile decoder will be able to decode a baseline H.263 stream, due to the fact it must support the short header tool. However, this is about the extent of any interoperability between the two standards. As to which is a better codec it depends on which of the extended tools of MPEG-4 or H.263+/++ are implemented and also how well they are implemented.

3 MPEG-4 Coding

This section explores in a little more detail MPEG-4 Simple Profile encoding and decoding. However, this is still only a basic introduction to aid users of VB8000 MPEG-4 transmitters and receivers. For in-depth discussions of MPEG-4 and video coding see the references [1] [2] [3].

Inside a VB8000 transmitter frames of video are captured from the camera and sent to the internal MPEG-4 encoder to be compressed. Each frame is then compressed in one of two ways, as explained in the next section.

3.1 Minding your P and I's

There are two ways to encode a video frame in an MPEG-4 Simple Profile codec: as an I-frame or as a P-frame⁴. 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 MPEG-4 files, such as with rewind and seek operations during playback and the regularity of these I-frames is known as the I-frame interval. However, the disadvantage of I-frames is that they tend to be much larger 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 encoded 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, MPEG-4 not only looks for differences but searches for, and makes use of, 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*.

3.2 Encoding an MPEG-4 I-VOP

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

Every frame of video to be encoded as an I-VOP is subdivided into a series of 16 by 16 pel-sized non-overlapping blocks called *macroblocks*. Each macroblock is encoded by the MPEG-4 encoder using three main processing units: DCT, Quantisation and Entropy Encoder, as shown in blue in Figure 1. This produces the MPEG-4 I-VOP part of the bitstream.

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 path indicated in green. This reconstruction process is required in order to encode subsequent frames as P-frames.

⁴ Technically speaking in MPEG-4 these are referred to as I-VOPs and P-VOPs, where a VOP refers to a Video Object Plane. For this document we will use VOP and frame to mean the same.

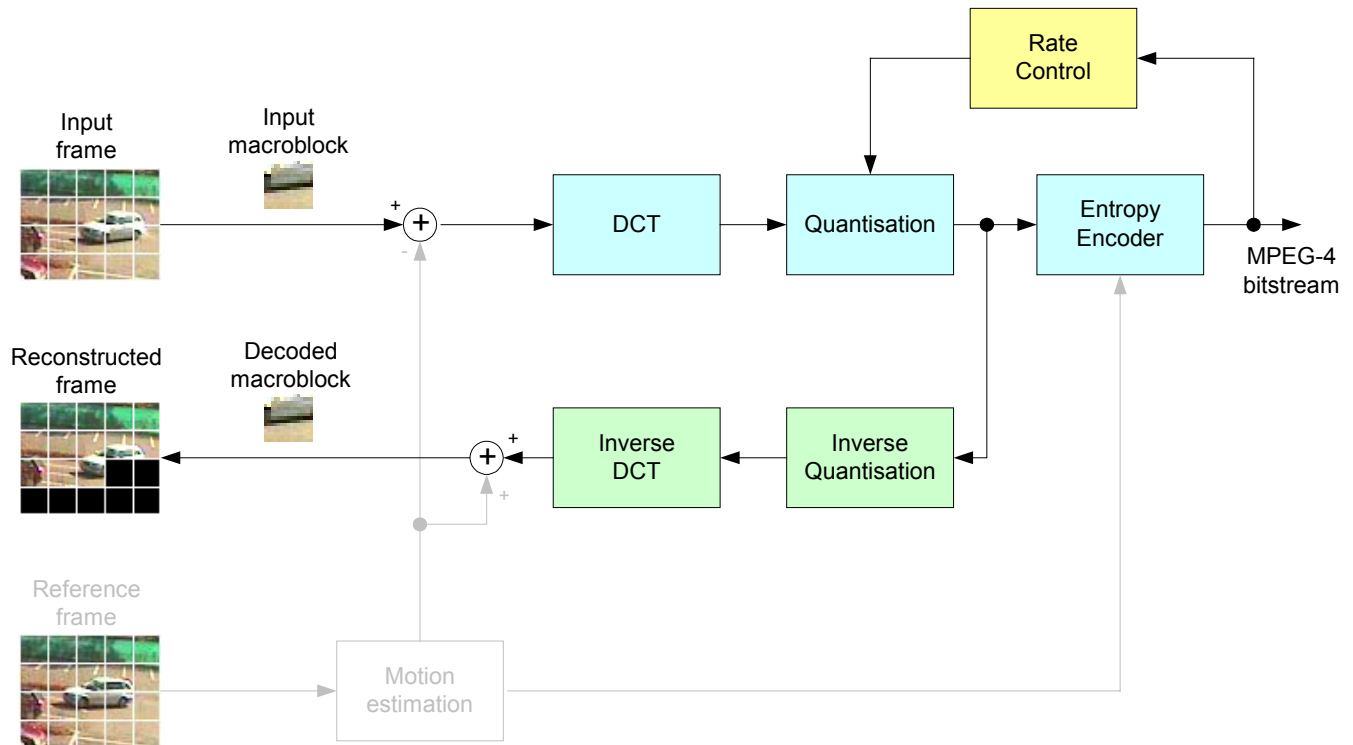


Figure 1: Simplistic view of encoding an MPEG-4 I-VOP

3.3 Encoding an MPEG-4 P-VOP

The previous section provided a basic explanation of how an 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 MPEG-4 P-VOP. As described in Section 3.1 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*⁵ and is in fact a previously reconstructed frame⁵.

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*.

⁵ The example shows a car reversing from a space in a parking lot.
IndigoVision Ltd

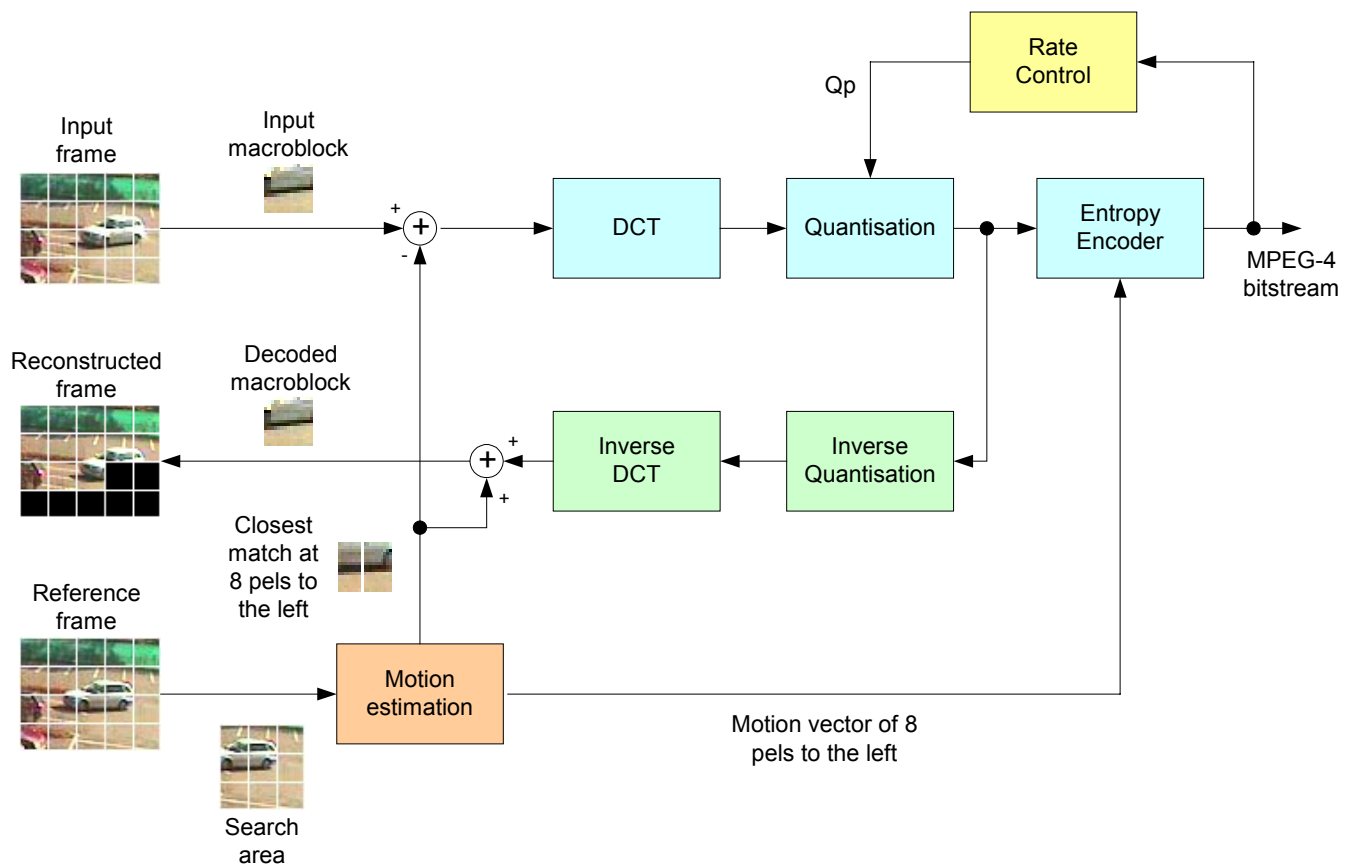


Figure 2: Simplistic view of encoding an MPEG-4 P-VOP

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 three forward path stages shown in blue. Combined with the motion vector information the MPEG-4 P-VOP part of the bitstream is generated.

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

3.4 Motion estimation

The motion estimation unit is worth further mention because it is one of the most computationally expensive parts and most critical to the performance of the MPEG-4 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 *1/2-pel motion estimation*.

Motion estimation is a complex procedure and often encoders, especially real-time software encoders, will use reduced search areas, use a restrictive search algorithm or not perform 1/2-pel motion estimation in order to achieve real-time performance. However, this can often result in poor quality video and significantly reduced compression.

3.5 DCT

The discrete cosine transform (DCT) is at the heart of most standards-based video codecs including H.261/3 to MPEG-1/2/4. The DCT splits each input or error macroblock into a series of 8x8 pel blocks and then simply converts the blocks into a state more conducive to compression. However, no actual compression is achieved at this stage.

3.6 Quantisation

This quantisation stage is where the majority of compression is achieved. This is also the stage where the majority of information can be lost and artefacts introduced.

The quantisation process is controlled by a parameter known as Q_p , where Q_p can take a value between 1 and 31 inclusive. If Q_p is set to 1 then the quantisation unit performs little processing on the DCT data, meaning that little data is lost, quality remains high but the compression achieved is low.

As Q_p increases in value the quantisation 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 Q_p increases further towards the maximum value of 31 more and more information is discarded, and quality has to be sacrificed. However, compression has increased significantly.

3.7 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.

3.8 Rate control

The rate control unit controls the bitrate of the bitstream generated by the MPEG-4 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 quantisation 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.

There are many different algorithms for controlling Qp for optimal performance, and some of these algorithms also use the option of dropping frames of video as well as adjusting Qp. These latter algorithms trade-off the quality (Qp) of each frame with the jerkiness in the video caused by frame dropping. Further, the bitrate profiles and characteristics of these algorithms will differ, and often the choice of algorithm is dependent on the network and target application.

3.9 Other tools

Other MPEG-4 tools and terminology, such as AC/DC prediction, B-VOPs, Method 1 quantisation, reversible VLC, four motion vectors, unrestricted motion vectors, data partitioning are all beyond the scope of this document. Please refer to [1] [2] [3][5].

3.10 Decoding an MPEG-4 bitstream

The decoding process of an MPEG-4 bitstream is intentionally identical to the reverse path shown in green in Figure 2. The exception is that the bitstream is first passed through an entropy decoder before the data is passed to the inverse quantisation 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.

3.11 Common MPEG-4 artefacts

The main sources of artefacts to be found in an MPEG-4 video sequence will be typically be related to the quantisation process discussed in Section 3.6. The most obvious artefact is blockiness. This happens when the Qp value is set high. This is typical when a low target bitrate is selected or when an I-VOP has been produced and the rate controller is attempting to compensate the large number of bits it has just generated.

Further artefacts like graininess can be down to poor implementations or attempted short cuts in an MPEG-4 encoder design. Alternatively, artefacts such as the 'halo' effect visible around a person's head, is simply a product of the compression process.

4 IndigoVision MPEG-4

The previous sections have discussed MPEG-4 in general. This section examines IndigoVision's MPEG-4 codec and video configuration options and attempts to correlate these options with the information presented in the previous sections. First the IndigoVision IV8102 MPEG-4 codec is introduced.

4.1 IV8102 MPEG-4 codec

The IndigoVision VB8000⁶ transmitters and receivers use IndigoVision's own custom MPEG-4 hardware codec: the IV8102; which was designed and built by IndigoVision. This codec offers 4SIF full frame rate MPEG-4 encoding and decoding, with some distinct advantages

- Custom hardware MPEG-4 chip tailored to IndigoVision.
- High performance, highly parallel, low cost, encoding and decoding of 4SIF 30fps compliant MPEG-4 video.
- Deterministic encode and decode time, regardless of bitrate and motion means high-quality video can be maintained during fast-moving activity without any 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 pre-filtering of video data to help reduce noise and improve compression.
- Low processing host overhead means processing power available for value-add features such as high quality audio, motion detection and ACF.
- Based on IndigoVision's extensible MainStream architecture.

4.2 Configuration

IndigoVision allow a small number of the MPEG-4 parameters to be configured via the VB8000 (MPEG-4) Video Configuration web page, available on all transmitters. Figure 3 shows an example page from a VB8000 transmitter.

There are five parameters that directly affect the MPEG-4 encoder in the VB8000 transmitter: Bit-Rate, Rate Control, Frame Rate, I-frame Interval and Resolution.

⁶ VP881 and VP882 models only.
IndigoVision Ltd

Video Configuration	
Device Type	Transmitter
Bit-Rate	<input type="text" value="1024"/> Kbps (Range : 32 to 4096)
Rate Control	<input type="text" value="CBR"/>
Transmit one frame for every	<input type="text" value="1"/> frame(s) captured (Range : 1 to 30)
I-Frame Interval	<input type="text" value="4000"/> ms (Range : 30 to 10000 ms)
Resolution	<input type="text" value="2SIF"/>
Brightness	<input type="text" value="128"/> (Range : 0 to 255)
Contrast	<input type="text" value="-70"/> (Range : -128 to 127)
Saturation	<input type="text" value="-1"/> (Range : -128 to 127)
Hue	<input type="text" value="0"/> (Range : -128 to 127)
<input type="button" value="Submit"/>	

Figure 3: IndigoVision VB8000 video configuration

- *Bit-Rate*: This is the requested target bitrate, in Kbps, that is sent to the MPEG-4 rate control unit, as described in Section 3.8. Typically, increasing the bitrate increases the quality of video.
- *Rate Control*: Defines the algorithm used by the rate control unit described in Section 3.8. There are currently two modes of operation: CBR and ACF. CBR is simple capped bitrate control where the rate controller attempts to maintain average output bitrate on or below the requested target bitrate. The ACF algorithm adjusts the frame rate of video dependent on the amount of activity detected in the scene. ACF is described in more detail in [7].
- *Transmit one frame for every*: This is the frame rate divisor and is used to control the frame rate of the MPEG-4 video. Selecting a value of one means that every frame captured from the camera is passed to the MPEG-4 encoder.
- *I-frame Interval*: Controls how far apart the I-frames are in the stream. This was discussed in Section 3.1.
- *Resolution*: Not strictly part of the MPEG-4 compression standard but controls the resolution of the data input to the MPEG-4 encoder.

No configuration of the MPEG-4 decoder is possible. Various filters, such as de-interlacing, are supported in VBDK and Control Center but these are not strictly part of the MPEG-4 decoder.