Software H.264 Encoder Engine for Online Video Delivery Service Cost Reduction

Masaki Kitahara[†], Naoki Ono, and Atsushi Shimizu

Abstract

This article describes the software H.264 encoder engine that we have developed in NTT Cyber Space Laboratories. It implements technologies for fast encoding, high compression performance, and reduced encoder operation cost, which are all important features for reducing the cost of online video delivery services. In tests, it outperformed conventional encoders. It is also suitable for use in creating content for online video delivery services.

1. Introduction

Video delivery services such as IPTV (Internet protocol television) and video-on-demand have recently gained in popularity. Video generally involves much more data than audio, so the increasing popularity of high-resolution video has led to demands for video encoding. For example, the amount of data used for uncompressed HDTV (high-definition television) video is close to 1 Gbit/s.

Video content delivery over networks requires video encoding technology that can encode video with a good compression ratio, which is defined as the compressed size divided by the original size, so a smaller ratio equates to better performance. To meet this demand, the use of H.264, which is known to achieve twice the compression performance of MPEG-2, has spread widely over the years. H.264 can achieve such high performance because it enables the use of various coding tools, which makes it adaptable to various kinds of video characteristics. However, this means that an H.264 encoder must adaptively select coding tools from a large number of candidates according to the characteristics of the input video in order to achieve a good compression ratio. In other words, there is a trade-off between compression performance and encoding time: H.264 encoders tend to need long encoding times. To reduce the service cost of online video delivery services, it is necessary to encode video with high compression performance so that the bitrate of the compressed video is low in order to reduce storage costs. Therefore, to reduce the encoder operation cost in such services, fast encoding is needed. The basic functionalities and system requirements are listed in **Table 1**.

NTT Cyber Space Laboratories has a long history of developing codec LSIs (large-scale integrated circuits) and equipment. We utilized know-how acquired from those past developments and developed technologies for fast encoding, high compression performance, and encoder operation cost reduction, which are all important for reducing the cost of online video

|--|

Encoding format	MPEG-4 part 10 AVC/ITU-T H.264				
Profiles	Main Profile, High Profile, High 4:2:2 Profile				
Image size	CIF (352 x 288) to 4K x 2K (3840 x 2160)				
System requirements	CPU	SSE4.1 or above			
	OS	MS-Windows XP SP3			
	Library	Microsoft C/C++ Runtime Library			

CIF: common intermediate format

[†] NTT Cyber Space Laboratories

Yokosuka-shi, 239-0847 Japan

OS: operating system



Fig. 1. Conventional multithreaded encoding.

delivery services. These technologies are described below and compared with other encoders.

2. Technology for fast encoding

We have implemented a multithreaded encoding method that utilizes multicore central processing units (CPUs), which are common in recent personal computers (PCs), with negligible degradation in compression performance.

Conventionally, multithreaded encoding has been achieved by dividing images in the input video into sub-images and encoding the sub-images independently thread by thread. Such a method is illustrated in Fig. 1. In H.264, images in the input video are encoded in units of macroblocks, which are blocks of pixels that consist of 16 pixels in the horizontal direction and 16 pixels in the vertical direction; when a macroblock is encoded, information about how adjacent already-encoded macroblocks were encoded is utilized in order to achieve high compression performance. Furthermore, encoding begins with the macroblock at the top left of the image (or the sub-image if conventional multithreaded encoding is used) and each macroblock in the top horizontal macroblock line is encoded from left to right. When the top right macroblock is encoded, macroblocks in the horizontal macroblock line below are encoded from left to right, and the rest of the horizontal macroblock line is processed in the same way. When the encoding method described in Fig. 1 is used, at the moment core 2 is encoding macroblock B, core 1 is encoding macroblock C, assuming that the encoding speed is the same for all the cores. This means that macroblock A, which is one of the macroblocks adjacent to macroblock B, has not yet been encoded by core 1, so its encoding information is not yet available for use in encoding macroblock B; this results in low compression performance. Since the number of such cases increases with the number of divided images (faster encoding), it is not possible to achieve a significant speedup in encoding in the conventional method without significant degradation in compression performance. Thus, even if the PC had many cores, it would not possible to utilize its full potential with the conventional method.

In the newly developed software H.264 encoder engine, we have implemented a new method created by applying know-how acquired during encoder LSI development. In this method, the encoding of macroblocks is pipelined, so that the number of threads can be increased with negligible loss in compression performance. Thus, it is possible to utilize the full potential of newer multicore CPUs designed for PCs. As a result, our new H.264 encoder engine can encode at twice the speed of our conventional one. Furthermore, even faster encoding will be possible in the future by using PCs with more cores than those on the market at present.



Fig. 2. Adaptation of GOPs to scene change.

3. Technology for high compression performance

We have implemented a method that adapts the structure of groups of pictures (GOPs) to scene changes to achieve higher compression performance. In H.264, frames in the input video are grouped together to construct GOPs. Furthermore, each frame can be one of three types: intra-coded (I), predictivecoded (P), or bidirectionally predictive-coded (B). I frames are encoded solely using information inside the frame to be encoded without reference to other frames, P frames are encoded using past I or P frames and are also used as a reference for further prediction, and B frames are encoded using both past and future frames. An example where each GOP is of equal length (6 frames), and there are 2 B frames between an I frame and a P frame or between two P frames is shown in Fig. 2(a). As shown here, it is common to use I frames for the first frame in a GOP to enable random access.

Since P and B frames use past or future frames for encoding, when a scene change occurs in such frames, the compression performance degrades significantly because there is no correlation between the frame to be encoded and the frame that is used for encoding. The conventional method for avoiding this was to change the frame type of such a frame to an I frame, leaving the GOP structure basically unchanged. However, such a simple solution leads to an increase in the number of I frames. Since the compression performance of I frames is low, this is not an optimal solution for coping with scene changes.

An example of the method implemented in the new encoder engine is shown in **Fig. 2(b)**. When a scene change occurs, a new GOP is created from that frame. With this method, the number of I frames can be smaller than in the conventional method. Furthermore, since the GOP length can be limited, this method is applicable to the case where the GOP length needs to be made shorter than a particular value, such as in IPTV system specifications.

When this method is used together with the fast encoding technology described in section 2, the bitrate of compressed video can be 20% smaller compared with our conventional software H.264 encoder engine at the same encoding speed.

4. Technology for encoder operation cost reduction

In general, when the bitrate of a long section of compressed video exceeds the target bitrate, it is impossible to decode the video within the specified frame rate because this section of compressed video takes longer to deliver over a network that has bandwidth equal to the target bitrate. Such cases occur when the video contains a section of content that is complex and difficult to compress; an example is a scene with water splashing. In such cases, the person



Fig. 3. Example of an audio-visual encoder application.

	Competitor's encoder	Our conventional encoder engine	Our newly developed encoder engine	Remarks
Compressed bitrate*1	7 Mbit/s	7 Mbit/s	5.6 Mbit/s	-
Encoding time*2	x3.3	x5.2	x2.6	Encoder processing time at the same bitrate
Automatic retry of encoding	No	No	Yes	-
Encoding in 4:2:2 format video	Yes	No	Yes	-

Table 2. Comparison with other encoders.

*1 Bitrate able to provide the video quality of digital terrestrial television broadcasting in Japan *2 Encoder processing time relative to the actual video length when HDTV video was encoded

operating the encoder must encode the video again using different encoder parameters to decrease the bitrate of the complex parts. One problem with this is that the changed encoder parameters can influence other parts of the video, resulting in the video having degraded video quality. Another problem is that since encoding with different parameters can occur multiple times and the encoder operator must check the quality of the video each time, the encoder operation cost is higher.

In our new H.264 encoder engine, we have implemented a method that automatically detects anomalies during encoding and automatically encodes such parts with different encoder parameters multiple times. This method avoids quality degradation of the whole video and achieves a lower encoder operation cost.

5. Use of the H.264 encoder engine for application development

The H.264 encoder engine that we have developed

is solely for encoding images. To develop an application that can encode video content containing both images and sound, one needs other components such as an audio encoder engine and a multiplexer engine. An example of the architecture of such an audiovisual encoder application is shown in **Fig. 3**.

6. Comparison of performance and functionalities

The performance and functionalities of our new software H.264 encoder engine are compared with those of our conventional encoder engine and a competitor's encoder engine in **Table 2**. Our new encoder achieved a 20% reduction in the bitrate of compressed video at the same encoding speed. It supports the 4:2:2 format, which is important for professional use, and it also has functionalities that are not implemented in competitors' encoder engines such as automatic retry of encoding (described in section 5).

7. Conclusion

Our new software H.264 encoder engine achieves fast encoding, high compression performance, and a lower encoder operation cost, which are important features for reducing the cost of online video delivery services. It is therefore highly suitable for use in creating content for online video delivery services. Our future work includes developing encoder applications based on the software H.264 encoder engine and extending the technologies described here for use in other kinds of services.



Masaki Kitahara

Research Engineer, Visual Media Communications Project, NTT Cyber Space Laboratories. He received the B.E. and M.E. degrees in

industrial and management engineering from Waseda University, Tokyo, in 1999 and 2001, respectively. Since joining NTT in 2001, he has been engaged in the development of compression and rendering algorithms for 3D video, video compression algorithms, and software. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE).



Atsushi Shimizu

Senior Research Engineer, Visual Media Communications Project, NTT Cyber Space Laboratories.

He received the B.E. and M.E. degrees in electronic engineering from Nihon University, Tokyo, in 1990 and 1992, respectively. Since joining NTT in 1992, he has been engaged in video compression algorithm and software development. He is a member of IEICE, IIEEJ, and the Institute of Image Information and Television Engineers.



Naoki Ono

Senior Research Engineer, Visual Media Communications Project, NTT Cyber Space Laboratories.

He received the B.E. and M.E. degrees in information engineering from Niigata University in 1989 and 1991, respectively. Since joining NTT in 1991, he has been engaged in video compression algorithm and software development. He is a member of IEICE and the Institute of Image Electronics Engineers of Japan (IIEEJ).