Video Coding Fundamentals

James Ridgway

28th July 2013

1 Introduction

This document discusses some of the fundamental video coding principles that are relevant to the H.264 standard. This document starts with an analysis of compression as used in multimedia formats before addressing the specifics of video files.

2 Glossary

B-Frame	A bi-directional *frame* whose representation is predicted by its two neighbouring frames.
Frame	A raw (decoded) portion of data that contains either audio samples or an image.
I-Frame	Shorthand for an Intra Frame.
Intra Frame	A reference frame of a video that is independent from any other frames.
P-Frame	A predicted frame whose representation is predicted from preceding frames.
Spatial domain	The spatial domain refers to the normal space of a multidimensional signal. In the context of an image, this is the 2D pixel space.
Spatiotemporal domain	Spatial domain signals existing in the temporal domain.
Temporal cohesion	The correlation between signals observed at different moments in time.

3 Compression

Multimedia formats, whether they be image, audio or video, can use compression techniques to reduce the size of data that is needed to represent the original media. Most compression techniques rely on producing an "alternative representation" in a domain different to that of the original media [Muk11].

3.1 The Compressed Domain

Images are originally represented in the *spatial domain*, which as a function is represented as $I : \mathbb{Z}^2 \to \mathbb{N}^3$ whereby each coordinate of a 2D coordinate space maps to a three-dimensional RGB colour vector. Video, being sequences of images is represented in the *spatiotemporal domain* such that $V : \mathbb{N} \times \mathbb{Z}^2 \to \mathbb{N}^3$ [Muk11, Fri10].

Let's consider the implications of storing the output of these functions in an uncompressed format. If we were to consider a single 1920 x 1080 image just over 2 million pixels values will need to be stored. This is a substantial amount of information to store for a single image. In the context of a standard video file[1], a single second of video at 1920 x 1080 resolution would require between 49 and 63 million pixel values. Images and videos contain vast quantities of data and storing this data in a raw, uncompressed format is impractical for most situations, as a result, image and video formats typically use an alternative representation. Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) are two of the more popular methods for alternatively representing images and video, both of which belong to the compressed domain [Muk11, Fri10].

3.1.1 Discrete Cosine Transform

JPEG images use a DCT to convert an image from the spatial domain into the frequency domain. The spatial domain represents data based on intensity of pixels. DCT separates parts of an image based on frequency. Image signal energy is generally stored in low-frequency regions, therefore high-frequency information can be discarded or manipulated without causing significant degradation of image quality [WJN10].

The 2-dimensional DCT, $F(n, m)$, of an $N \times M$ pixel image is defined as follows:

$$F(n,m) = \frac{2}{\sqrt{NM}} C(n)C(m) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) \cos\frac{(2x+1)n\pi}{2N} \cos\frac{(2y+1)m\pi}{2M} \tag{1}$$

where

$$C(n) = C(m) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } m = 0 \\ 1 & \text{if } m \neq 0 \end{cases} \tag{2}$$

and $f(x, y)$ is the intensity of the pixel at the x^{th} row and y^{th} column.

3.2 Compression Considerations

Before data is represented in a compressed format important consideration should be given to the trade-off between computational cost and storage requirements.

The accuracy of the representation is another important consideration. For images and video, an approximation of the original source if often sufficient, which means that lossy compression schemes can be used, however, if the given application requires an exact representation then a lossless representation must be used [Muk11].

Image and video compression methods should consider the following attributes [Muk11]:

[1] Most videos range between 24 and 30 frames per second.

- **Reconstructibility** From an encoded (compressed) representation it should be possible to decode/reconstruct the original data. In the case of lossy compression the reconstruction can be an approximation.

- **Low Redundancy** The compressed representation should be as concise as possible. Images in the spatial domain can have a high redundancy rate due to high spatial correlation of neighbouring pixels. Correlation can also exist in the spatial domain if we consider the context of video, neighbouring frames will have high temporal correlation because the change in an image frame is very gradual from one consecutive frame to another.

- **Factorisation into substructures** Decomposing an object into its component parts can be useful for identifying those components whose contribution is relatively insignificant. Insignificant parts can be removed or reduced further, thus reducing the size of the compressed representation.

In §3.1.1 we saw how a DCT can be used to represent an image by the form of an *alternative representation*. Whilst a single video frame can be represented independently using any image representation technique, these methods do not harness the *temporal cohesion* [Muk11]. Consecutive frames can have high temporal redundancy because portions of the frame remain unchanged, and by examining the correlation between consecutive frames a more concise representation can be used [Muk11, Ric08, Le 91].

A better representation of videos frames with a reduced temporal redundancy is to use a Group of Pictures (GOP) technique. In a GOP, one of these frames will be a reference frame which is called an *Intra Frame* or *I-Frame*. Other frames are then predicted by this, and these predictions are represented as changes (deltas) from the preceding frame – these are known as *P-Frames*. Some predicted frames are predicted from its two neighbours, these are bi-directionally predicted frames and are called *B-Frames* [Muk11, Ric08].

Prediction errors and motion vectors are used to explicitly represent temporal cohesion, however, there are some instances where this is not the case. Some frames are represented as side information or parity bits that can be used decode the frame – these frames are only approximations. Side information and parity bit encoding is often used for low complexity compilers or in distributed environments so that each frame is encoded individually – this method of encoding is called *distributed video coding* [Muk11, GARR05]

4 Quantisation

Quantisation is an important part of lossy compression, and accounts for its "lossy"-ness. Compressed data is a smaller alternative representation of the original data. This process of converting from one representation to another can often involve an intermediate representation. Quantisation is the process of scaling down the range of symbols that are used in the intermediate representation. For instance, with a DCT a matrix of coefficients is produced whose values may range between -223 and 150, but after quantisation, these values may only range between 10 and 130. The reduced range caused by the quantisation process subsequently requires fewer bits to code the representation than the original range. Quantisation parameters for multimedia formats are chosen based on how individual components affect the average human perception [Muk11].

5 Coding Concepts [2]

An encoder (compressor) and decoder (decompressor) forming a complementary pair is known as a *codec* (enCOder/DECoder). The encoder is used to store or transmit video by converting the original raw video format to an alternative (compressed) representation. The decoder converts the compressed form back to the original video.

The H.264 codec consists of four main components: block-based motion compensation, transform, quantisation and entropy coding.

A codec uses a model (an efficient alternative representation) to reconstruct an approximation of the original video files. A codec should attempt to maximise on two conflicting goals: high fidelity (high quality) video with high compression levels. Decoding an alternative video representation that uses few bits often results in the decoder outputting a poor, low quality approximation.

A video encoder consists of three core components: a *temporal model*, a *spatial model* and an *entropy encoder*. The temporal model reads in a sequence of video frames and attempts to reduce the temporal redundancy by identifying similarities between the neighbouring video frames – this analysis usually involves computing a prediction of the current video frame. With H.264 the prediction can be computed from multiple previous or future frames. The prediction is improved by means of compensation for differences between the frames – this is known as motion compensation prediction. The temporal model outputs a residual frame and a set of motion vectors. The residual frame is computed by subtracting the prediction from the current frame, motion vectors are used to describe how the motion was compensated.

The residual frame from the temporal model is then fed into the input of the spatial model. The spatial model, like the temporal model is concerned with removing redundancy in its domain, in this case, the spatial model analyses neighbouring samples of the residual frame (produced by the temporal model) to reduce the spatial redundancy. Spatial reduction in H.264 is achieved by applying a transform followed by a quantisation process. The transform step produces a set of transform coefficients which are then quantised, removing insignificant values, and returning the quantised transform coefficients as the output of the spatial model.

The entropy encoder is the final component of the video encoder that produces an encoded output from the results of the spatial and temporal model. The entropy encoder processes the motion vectors from the temporal model and the coefficients from the spatial model to produce a compressed bit stream consisting of motion vectors, residual transform coefficients and header information.

Although the quantisation stages cause a loss of information, this process is roughly reversible, and as such, the decoder mechanism works in reverse to that of the encoder. The output produced by the decoder mechanism will only ever (in the case of H.264) be an approximation to the original input because of the quantisation stages.

5.1 Temporal Model

The residual frame produced by the temporal model is produced by subtracting the predicted frame from the actual video frame. The resize of the residual frame is dependent on the accuracy of the prediction process – the smaller the residual frame, the fewer bits

[2] Unless explicitly stated otherwise information in this section is based on [Ric08] and [Muk11].

needed to code it. Prediction accuracy can be improved by calculating and propagating compensation for motion from the reference frame(s) through to the current frame.

Motion compensation can significantly improve prediction calculations because two successive video frames are usually highly correlated. Most of the information captured in successive residual frames relates to the movement of objects in the scene, therefore a better prediction can be produced by compensating for this change in motion between frames. Changes between video frames can be caused by the motion of objects, changes in the scenery and adjustments in light levels. With the exception of lighting and changes to the scenery, these changes directly correspond to the movement of pixels between frames. The movement of individual pixels between successive frames can be estimated. The displacement movement of pixels in a video frame is known as an *optical flow* [AWSZ05].

In theory, it is possible to use the optical flow to predict the majority of the pixels in the current frame, provided the optical flow is accurate, by displacing each pixel as described by the optical flow.

Unfortunately, this is a very computationally intensive process, as each pixel will have to be transformed, and each frame decoded, on a pixel-by-pixel basis using the optical flow vectors. Whilst workable in theory, this would result in a large amount of residual data, which is at odds with the desirability of a compact residual frame.

5.2 Macroblocks Motion Estimation

A macroblock is typically a 16×16 pixel block of the current frame, in the wider context of block-based motion estimation a block is any $N \times M$ sample of a frame. Macroblocks are used by a variety of codecs including MPEG-1, MPEG-2, H.261, H.263 and H.264.

Macroblock motion estimation starts by dividing each frame into macroblocks. In turn, each macroblock is taken, and the reference frame is searched for a matching macroblock. Macroblocks from the current frame are paired with macroblocks in the current frame by choosing a candidate block that minimises the difference between the macroblock in the current frame and itself – this process provides a residual block. Finally, the residual block is encoded and stored, alongside the difference between the current macroblock and the candidate macroblock, called a motion vector.

Using a 16×16 size macroblock can cause some problems with certain motions and object outlines. If a macroblock and its corresponding candidate macroblock have a large difference (residual energy), then the number of bits required to code this macroblock increases and inflates the bit-rate. This issue can be addressed rather simply by decomposing a macroblock into smaller 8×8, or even 4×4 macroblock size. Using smaller macroblocks results in a larger number of blocks, which can be disadvantageous, therefore a better solution is to use an adaptive block size approach as used by H.264.

References

[AWSZ05] I. Ahmad, X. Wei, Y. Sun, and Y. Zhang. Video transcoding: An overview of various techniques and research issues. *IEEE Transactions on Multimedia*, 7(5):793–804, October 2005.

[Fri10] J. Fridrich. *Steganography in Digital Media: Principles, Algorithms and Applications*. Cambridge University Press, 2010.

[GARR05] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero. Distributed video coding. *Proceedings of the IEEE*, 93(1):71–83, January 2005.

[Le 91] D. Le Gall. Mpeg: a video compression standard for multimedia applications. *Commun. ACM*, 34(4):46–58, April 1991.

[Muk11] J. Mukhopadhyay. *Image and Video Processing in the Compressed Domain.* CRC Press, 2011.

[Ric08] I. E. G. Richardson. *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia.* John Wiley & Songs, 2008.

[WJN10] E. Walia, P. Jain, and N. Navdeep. An Analysis of LSB & DCT based Steganography. *Global Journal of Computer Science and Technology*, 10(1):4–8, 2010.