H.264 / MPEG-4 Part 10 White Paper

Prediction of Inter Macroblocks in P-slices

1. Introduction

The Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG are finalising a new standard for the coding (compression) of natural video images. The new standard [1] will be known as H.264 and also MPEG-4 Part 10, "Advanced Video Coding". This document describes the methods of predicting inter-coded macroblocks in P-slices in an H.264 CODEC.

Inter prediction creates a prediction model from one or more previously encoded video frames. The model is formed by shifting samples in the reference frame(s) (motion compensated prediction). The AVC CODEC uses block-based motion compensation, the same principle adopted by every major coding standard since H.261. Important differences from earlier standards include the support for a range of block sizes (down to 4x4) and fine sub-pixel motion vectors (1/4 pixel in the luma component).

2. Tree structured motion compensation

AVC supports motion compensation block sizes ranging from 16x16 to 4x4 luminance samples with many options between the two. The luminance component of each macroblock (16x16 samples) may be split up in 4 ways as shown in Figure 2-1: 16x16, 16x8, 8x16 or 8x8. Each of the sub-divided regions is a macroblock partition. If the 8x8 mode is chosen, each of the four 8x8 macroblock partitions within the macroblock may be split in a further 4 ways as shown in Figure 2-2: 8x8, 8x4, 4x8 or 4x4 (known as macroblock sub-partitions). These partitions and sub-partitions give rise to a large number of possible combinations within each macroblock. This method of partitioning macroblocks into motion compensated sub-blocks of varying size is known as **tree structured motion compensation**.

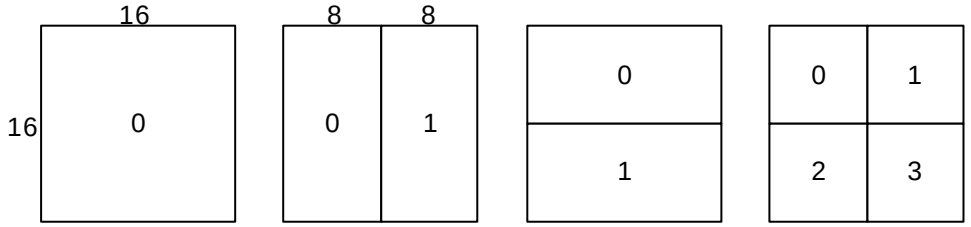

Figure 2-1 Macroblock partitions: 16x16, 8x16, 16x8, 8x8

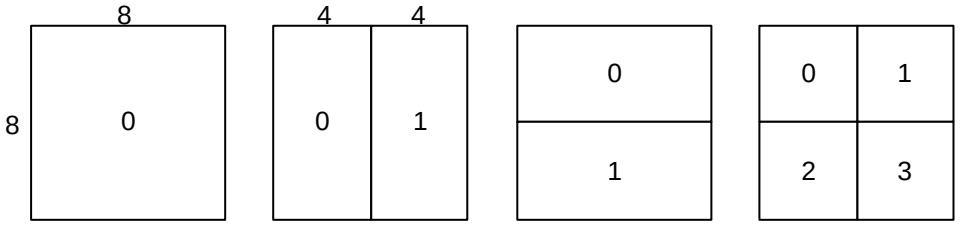

Figure 2-2 Macroblock sub-partitions: 8x8, 4x8, 8x4, 4x4

A separate motion vector is required for each partition or sub-partition. Each motion vector must be coded and transmitted; in addition, the choice of partition(s) must be encoded in the compressed bitstream. Choosing a large partition size (e.g. 16x16, 16x8, 8x16) means that a small number of bits are required to signal the choice of motion vector(s) and the type of partition; however, the motion compensated residual may contain a significant amount of energy in frame areas with high detail. Choosing a small partition size (e.g. 8x4, 4x4, etc.) may give a lower-energy residual after motion compensation but requires a larger number of bits to signal the motion vectors and choice of partition(s). The choice of partition size therefore has a significant impact on compression

performance. In general, a large partition size is appropriate for homogeneous areas of the frame and a small partition size may be beneficial for detailed areas.

The resolution of each chroma component in a macroblock (Cr and Cb) is half that of the luminance (luma) component. Each chroma block is partitioned in the same way as the luma component, except that the partition sizes have exactly half the horizontal and vertical resolution (an 8x16 partition in luma corresponds to a 4x8 partition in chroma; an 8x4 partition in luma corresponds to 4x2 in chroma; and so on). The horizontal and vertical components of each motion vector (one per partition) are halved when applied to the chroma blocks.

Example: Figure 2-3 shows a residual frame (without motion compensation). The AVC reference encoder selects the "best" partition size for each part of the frame, i.e. the partition size that minimizes the coded residual and motion vectors. The macroblock partitions chosen for each area are shown superimposed on the residual frame. In areas where there is little change between the frames (residual appears grey), a 16x16 partition is chosen; in areas of detailed motion (residual appears black or white), smaller partitions are more efficient.

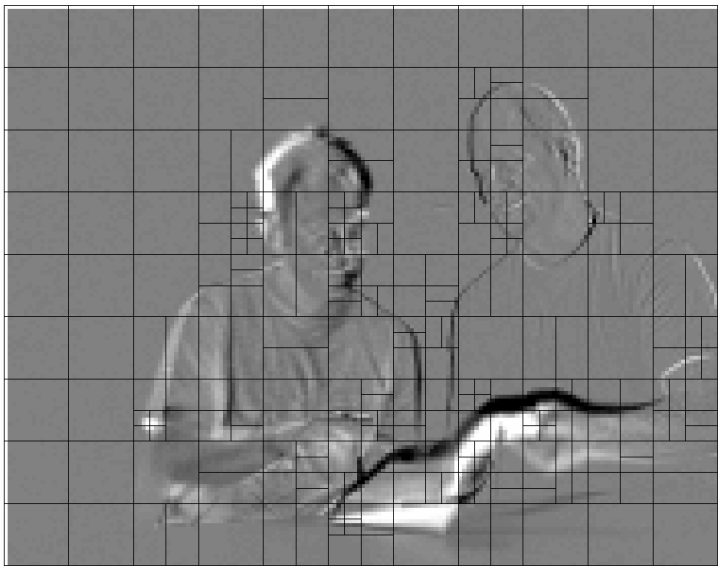

Figure 2-3 Residual (without MC) showing optimum choice of partitions

3. Sub-pixel motion vectors

Each partition in an inter-coded macroblock is predicted from an area of the same size in a reference picture. The offset between the two areas (the motion vector) has ¼-pixel resolution (for the luma component). The luma and chroma samples at sub-pixel positions do not exist in the reference picture and so it is necessary to create them using interpolation from nearby image samples. Figure 3-1 gives an example. A 4x4 sub-partition in the current frame (a) is to be predicted from a neighbouring region of the reference picture. If the horizontal and vertical components of the motion vector are integers (b), the relevant samples in the reference block actually exist (grey dots). If one or both vector components are fractional values (c), the prediction samples (grey dots) are generated by interpolation between adjacent samples in the reference frame (white dots).

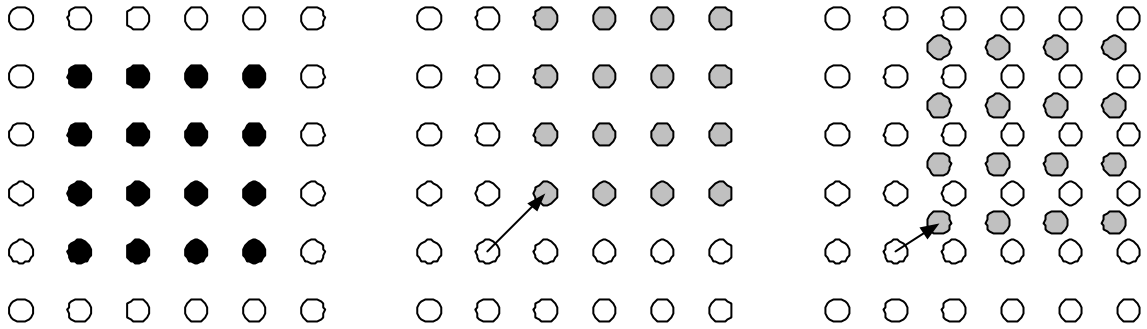

Figure 3-1 Example of integer and sub-pixel prediction

Sub-pixel motion compensation can provide significantly better compression performance than integer-pixel compensation, at the expense of increased complexity. Quarter-pixel accuracy outperforms half-pixel accuracy.

In the luma component, the sub-pixel samples at half-pixel positions are generated first and are interpolated from neighbouring integer-pixel samples using a 6-tap Finite Impulse Response filter. This means that each half-pixel sample is a weighted sum of 6 neighbouring integer samples. Once all the half-pixel samples are available, each quarter-pixel sample is produced using bilinear interpolation between neighbouring half- or integer-pixel samples.

If the video source sampling is 4:2:0, 1/8 pixel samples are required in the chroma components (corresponding to ¼-pixel samples in the luma). These samples are interpolated (linear interpolation) between integer-pixel chroma samples.

4. Motion vector prediction

Encoding a motion vector for each partition can take a significant number of bits, especially if small partition sizes are chosen. Motion vectors for neighbouring partitions are often highly correlated and so each motion vector is predicted from vectors of nearby, previously coded partitions. A predicted vector, MVp, is formed based on previously calculated motion vectors. MVD, the difference between the current vector and the predicted vector, is encoded and transmitted. The method of forming the prediction MVp depends on the motion compensation partition size and on the availability of nearby vectors. The "basic" predictor is the median of the motion vectors of the macroblock partitions or sub-partitions immediately above, diagonally above and to the right, and immediately left of the current partition or sub-partition. The predictor is modified if (a) 16x8 or 8x16 partitions are chosen and/or (b) if some of the neighbouring partitions are not available as predictors. If the current macroblock is skipped (not transmitted), a predicted vector is generated as if the MB was coded in 16x16 partition mode.

At the decoder, the predicted vector MVp is formed in the same way and added to the decoded vector difference MVD. In the case of a skipped macroblock, there is no decoded vector and so a motion-compensated macroblock is produced according to the magnitude of MVp.

5. References

1 ITU-T Rec. H.264 / ISO/IEC 11496-10, "Advanced Video Coding", Final Committee Draft, Document **JVT-G050**, March 2003