# REGION-BASED CODING OF MOTION FIELDS FOR LOW-BITRATE VIDEO COMPRESSION

*Huipin Zhang and Frank Bossen*

Media Laboratory, DoCoMo USA Labs
San Jose, CA 95110, USA

## ABSTRACT

Low-bitrate video compression requires a compact encoding of motion fields. We encode motion fields by segmenting them into regions of homogeneous motion. Each region is formed by a collection of blocks. The segmentation is represented by the connectedness relationship between adjacent blocks. To reduce the amount of connectedness information, blocks may be of variable size, as determined by a quadtree structure. We use context-based binary arithmetic coding to further reduce the amount of bits required to code the segmentation. A segmentation algorithm based on rate-distortion optimization is presented. Simulations show that such a motion-field coding scheme achieves coding efficiency comparable to the state-of-the-art H.264 codec with modest improvements.

## 1. INTRODUCTION

Efficient coding of motion fields is critical for low-bitrate video applications, as the proportion of bits allocated to motion data can become significant. Compression of motion data is achieved through the exploitation of the spatial redundancy in a motion field. Traditionally this is accomplished by differential coding of motion vectors using robust predictors, such as the median predictor used in H.264 [1]. One drawback of this traditional approach is that, given a single motion boundary, the motion difference between the two sides of the boundary may have to be coded multiple times, resulting in inefficiencies. For example, consider a head-and-shoulder sequence with a static background and a moving foreground. The motion vector associated with the foreground needs to be transmited multiple times: once at the top of the head, and once at the right shoulder.

To avoid transmitting such motion vectors multiple times, a region-based approach is considered. We define regions as a collection of contiguous blocks of variable size. First a segmentation or region map of the motion field is coded, followed by one motion vector per region.

The segmentation of the motion field is an important component of the coding scheme. It can be achieved in several ways. In [2], a split-and-merge algorithm is developed in a graph-theoretic context to minimize a coding distortion. In [3], an iterative method for generating a polynomial motion field is presented, where regions resulting from a quadtree segmentation are merged. We here take a similar approach. The motion field is segmented in two phases: an initial segmentation phase and a region merging phase. In the first phase, we use a hierarchical block matching algorithm to generate an initial set of regions, each of which comprises a single block. In the second phase, adjacent blocks with similar motion are merged. In both phases we consider a rate-distortion cost comprising the prediction error, the cost of coding the segmentation and the cost of coding the motion vectors associated with the regions. After merging two regions, we reestimate a motion vector for the resulting region.

The remainder of the paper is organized as follows. We discuss coding of the segmentation and of motion vectors in Section 2. The segmentation algorithm is introduced in Section 3. Simulation results are presented in Section 4, and conclusions are drawn in Section 5.

## 2. REGION-BASED CODING OF MOTION FIELDS

We assume block-based motion compensation wherein all blocks have a same size $p \times p$. A region map defines a segmentation of the motion field, where a region index $r_{ij}$ is associated with each block at position $(i, j)$. A region is thus a collection of one or more blocks of size $p \times p$. We restrict regions to be collections of contiguous blocks. A motion vector $mv_k$ is associated with each region $k$. Thus, for each block at position $(i, j)$, a motion vector is defined by $mv_{r_{ij}}$. Coding the motion field involves coding the region map and coding the motion vectors associated with the regions.

### 2.1. Region map coding

Straightforward coding of the region map is not very efficient. Furthermore, permutations of the region numbering do not affect the motion field. We therefore consider a trans-

0-7803-8554-3/04/$20.00 ©2004 IEEE.

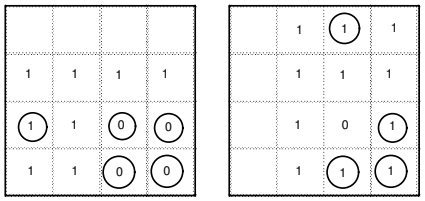


**Fig. 1**. Block connectedness of the segmentation shown in Figure 3(a): Top Connectedness Map (a), and Left Connectedness Map (b). Coded symbols are circled.

formation of the region map into connectedness maps. Two adjacent blocks are said to be *connected* if they belong to a same region (i.e., they share the same region index). The connectedness of blocks with their neighbors can be represented by two binary maps: a Top Connectedness Map (TCM) $[m_{ij}^T]$ and a Left Connectedness Map (LCM) $[m_{ij}^L]$, where $m_{ij}^T$ and $m_{ij}^L$ represent the connectedness of block at position $(i, j)$ with its top and left neighbors, respectively. The TCM and the LCM are sufficient for describing the region map, modulo permutations of the region numbering. A decoder can reconstruct the region map by using a filling algorithm. Figure 1(a) and (b) illustrates the two binary maps corresponding to the segmentation shown in Figure 3(a), where a 1 indicates that a block and its neighbor belong to a same region, and a 0 the opposite.

The two binary maps contain redundancies that can be eliminated. To exploit such redundancy, contexts for arithmetic coding are defined for symbols in the TCM and the LCM. We assume that the TCM bit is coded prior to the LCM bit for each block. Figure 2 (a) shows the relationship between a TCM symbol $c$ and its conditioning symbols $a$ and $b$, where $c = m_{ij}^T$, $a = m_{i,j-1}^T$, and $b = m_{i-1,j}^L$. Figure 2 (b) shows the the relationship between an LCM symbol $d$ and its conditioning symbols $a$, $b$, and $c$, where $d = m_{ij}^L$. We defined the context $k_{ij}^T$ for coding $m_{ij}^T$ as $k_{ij}^T = 2a + b$, and the context $k_{ij}^L$ for coding $m_{ij}^L$ as $k_{ij}^L = 4a + 2b + c$. The design of contexts $k^T$ and $k^L$ is based on the observation that the probability distribution of block connectedness is highly dependent on the connectedness in adjacent blocks. For example, when both $a$ and $b$ are equal to 1, meaning that the blocks above, to the left, and diagonally adjacent belong to a same region, it is very likely that the current block belongs to that region as well.

Furthermore, both the TCM and the LCM contain symbols that can be derived from already coded symbols. These symbols are not coded as they can be derived from the context value. When $k_{ij}^L = 7$ (i.e., $a = b = c = 1$), the symbol $m_{ij}^L$ is not coded, as it may only take a value 1 for consistency. Similarly, when $a+b+c$ is equal to 2 the symbol $m_{ij}^L$ is not coded either, as it may only take a value 0 for consistency. Nevertheless, the number of coded TCM and LCM

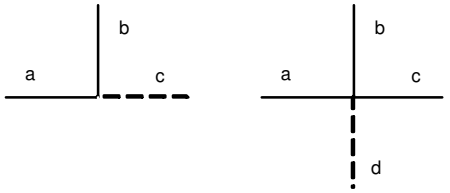


**Fig. 2**. Context design for context-based arithmetic coding of block connectedness: TCM symbols (left) and LCM symbols (right). Dashed lines represent connectedness symbols to be coded.

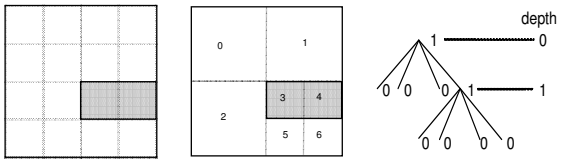


**Fig. 3**. Motion-field representation using the quadtree structure: (a) original motion field with two regions. (b) quadtree partition of regions. (c) quadtree representation and coding.

symbols remains large. To reduce this number, a quadtree structure of blocks is introduced as further described below.

Blocks belonging to a same region may be recursively combined into larger blocks of sizes $2p \times 2p$, $4p \times 4p$, etc. We use a quadtree structure to represent the block hierarchy. The block connectedness coding is thus preceded by the coding of a quadtree structure. The coding of the quadtree representation is straightforward: one binary symbol is used at each level to indicate whether to split a block into four smaller blocks. Figure 3 shows a simple example of a two-region segmentation and its associated quadtree.

Given the constraints on the TCM and LCM symbols and the quadtree structure, the number of symbols that are coded can become quite small. For example, only the 9 symbols circled in Figure 1(c,d) are coded.

### 2.2. Motion vector coding

Motion vectors are coded either with or without prediction from a previously coded neighboring region. The decision to use prediction is determined in a rate-distortion sense. For each region, a binary flag is thus coded to signal the decision to a decoder.

## 3. SEGMENTATION

We construct a segmented motion field in two phases: an initial segmentation phase and an iterative region-merging phase. In both phases, decisions are optimized in terms of a rate-distortion cost representing a trade-off between the distortion (i.e., the motion-compensated prediction error) and the cost in bits of coding the region map and the motion

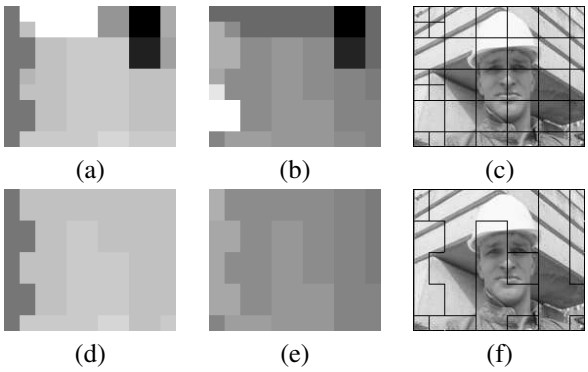

**Fig. 4**. An illustration of a segmented motion field before and after merging regions. Before: horizontal motion field (a), vertical motion field (b) and segmentation (c); after: horizontal motion field (d), vertical motion field (e) and segmentation (f).

vectors associated with each region. In the first phase we use a hierarchical motion estimation algorithm [4] to generate an initial motion field. The result is a partition of the frame into blocks of variable size that is constrained by a quadtree structure. Each block has size $2^k p \times 2^k p$, where $k \geq 0$, and can be seen as a region consisting of $2^{2k}$ blocks of size $p \times p$. An initial motion vector is associated with each region. In the second phase, we iteratively merge regions, similarly to [3]. Starting from the initial set of regions and their motion vectors, each iteration merges the two neighboring regions that lead to the largest reduction of the rate-distortion cost among all pairs of regions. The process stops when the rate-distortion cost cannot be further decreased by merging any two regions. Figure 4 shows an example of a segmented motion field before and after region merging.

The merging process involves two successive steps: region merging and motion vector updating. The decision of merging two regions is based on an estimate of a joint motion vector for the two regions under consideration. We estimate the joint motion vector by considering motion vectors in a neighborhood of the motion vectors of each region, and by selecting the one that minimizes the rate-distortion cost.

## 4. NUMERICAL EXPERIMENTS

We conducted numerical experiments to determine the efficiency of the motion coding technique with respect to H.264. We characterize the coding efficiency using a rate-distortion curve that describes the relationship between the prediction error and the motion bitrate. The prediction error is measured by the Mean Absolute Difference (MAD) between the input frame and the motion-compensated frame, and is averaged over all frames in a sequence. The reference frame used for motion compensation is always the original frame, such as to evaluate motion coding independently of texture coding.

We set the smallest block size in the quadtree structure to 4 by 4, and the largest to 32 by 32 for QCIF sequences, and 64 by 64 for CIF sequences. Motion vectors are represented with quarter-pixel resolution.

To generate reference results with H.264, we used version JM75a [5] of the reference software where we disable all intra block modes such as to generate a temporal prediction for all blocks in a frame. We also modified the rate-distortion selection of macroblock modes as follows. We removed the influence of texture coding by setting the prediction error to be zero in the texture coder. Furthermore we replaced the distortion criterion from the Sum of Squared Difference (SSD) to the Sum of Absolute Difference (SAD). The Lagrange multiplier is adjusted accordingly by using $\lambda_{\text{MOTION}}$. We also modified the software to use original frames as the reference frames for motion estimation and compensation. The following parameters were set in the configuration file: arithmetic coding (CABAC) was enabled; the motion range was set to 16 for QCIF sequences and to 32 for CIF sequences; the Hadamard transform was disabled (since we are optimizing for the SAD).

Figure 5 shows the rate-distortion curves for four test sequences: *Stefan* (QCIF, 30 frames per second (fps), 50 frames), *Foreman* (QCIF, 30fps, 300 frames), *Mobile* (CIF, 30fps, 300 frames), and *Rugby* (CIF, 25fps, 220 frames). The horizontal axis represents the average number of motion bits per frame and the vertical axis represents the MAD. Multiple samples on the curves are obtained by varying the Lagrange multipliers. In H.264 this is accomplished by varying the quantization parameter *qp*.

The results show that the region-based approach outperforms H.264 on low-motion sequences (*Stefan* and *Mobile*) where 20 to 25 percent less bits are needed to obtain a same distortion. In high-motion sequences, we observe an improvement over H.264, but only at lower bit rates. As the bit rate increase H.264 becomes more efficient. We expect that the observed reduction of the motion bitrate can be translated into a reduction of a few percent of the overall bitrate when taking texture coding into account, although this has not been verified.

From a complexity standpoint, the region-based approach is more computationally intensive than H.264. The average encoding time per frame is about twice that of H.264. The increase in complexity is mainly due to the elaborate region-merging process.

## 5. CONCLUSION

A region-based motion-field coding scheme was introduced. It consists of coding a region map and the motion vectors associated with each region. The region map contains three elements: a quadtree structure and two connectedness maps.

The region map and the motion vectors are derived using a rate-distortion optimized segmentation algorithm. Starting from an initial segmentation, regions are iteratively merged and their associated motion vectors refined. Numerical experiments show that the coding technique achieves coding efficiency comparable to that of the state-of-the-art H.264 codec, with modest improvements. Further improvements may be obtained by using more elaborate segmentation algorithms. For example, region merging operations may be interleaved with splitting operations.

## 6. REFERENCES


[1] ITU-T Recommendation H.264, *Advanced Video Coding for Generic Audiovisual Services*, May 2003.

[2] S. Liu and M. Hayes, "Segmentation-based coding of motion difference and motion field images for low bit-rate video compression," in *IEEE Proc. Int. Conf. Acoust., Speech, Signal Processing*, 1992, pp. 525–528.

[3] M. Karczewicz, J. Nieweglowski, and P. Haavisto, "Video coding using motion compensation with polynomial motion vector fields," *Signal Processing: Image Communication*, vol. 10, pp. 63–91, 1997.

[4] S.-J. Choi and J.W. Woods, "Motion-compensated 3D subband coding of video," *IEEE Trans. on Image Processing*, vol. 8, no. 2, pp. 155–167, Feburary 1999.

[5] ISO/IEC Joint Video Team (JVT), *H.264/AVC reference software JM 7.5*, http://bs.hhi.de/ suehring/tml/download/, January 2004.


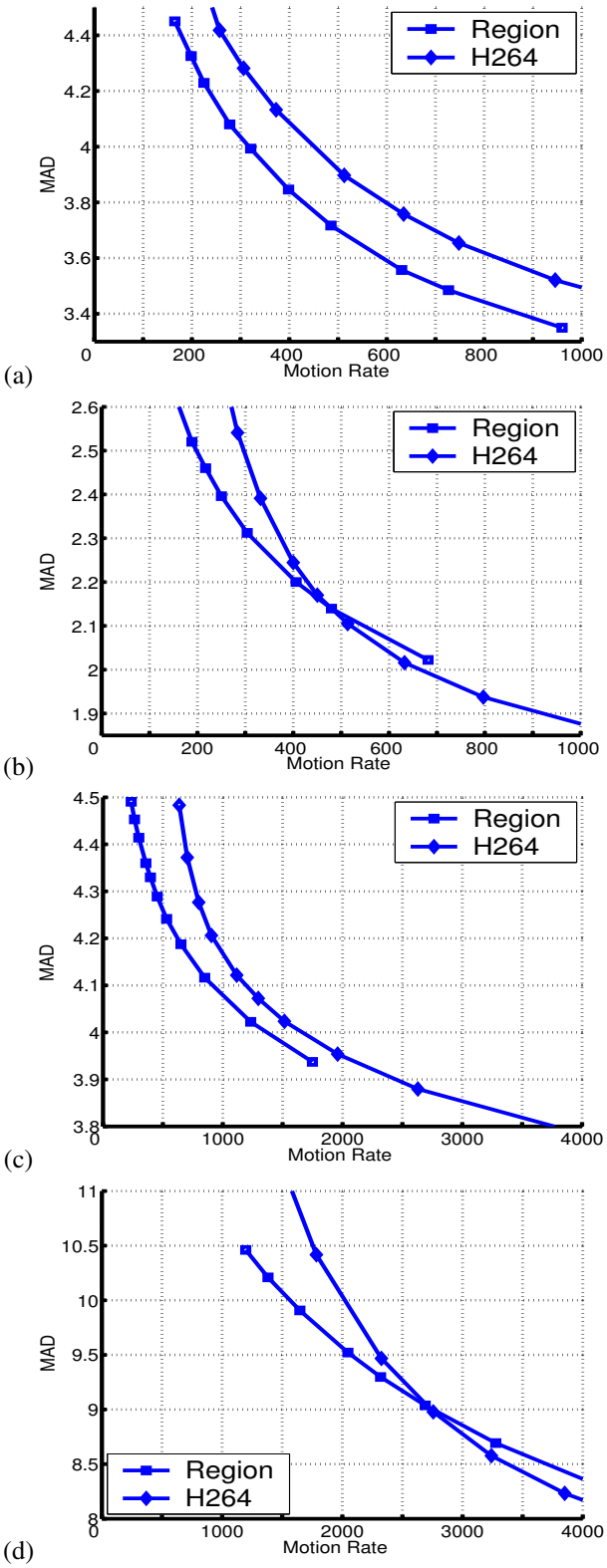


**Fig. 5**. Rate-distortion curves for *Stefan* (a), *Foreman* (b), *Mobile* (c), and *Rugby* (d) sequences; x-axis represents motion bits per frame and y-axis represents MAD. Curves marked with *squares* are obtained with the region based coding scheme and the curves marked with *diamonds* are obtained with H.264.