VARIABLE SIZE FINITE-STATE MOTION VECTOR QUANTIZATION

Ali Bilgin
Dept. of Elec. and Comp. Eng.
University of Arizona,
Tucson, AZ 85721
E-mail: bilgin@ece.arizona.edu

Michael W. Marcellin
Dept. of Elec. and Comp. Eng.
University of Arizona,
Tucson, AZ 85721
E-mail: mwm@ece.arizona.edu

Hüseyin Abut
Dept. of Elec. and Comp. Eng.
San Diego State University,
San Diego, CA 92182
E-mail: abut@anadolu.sdsu.edu

ABSTRACT

In this paper, we present an algorithm which combines Vector Quantization (VQ) techniques with variable size Block Matching Algorithms (BMA) for motion compensated image sequence coding. First, an algorithm which decomposes the image frame into blocks of different sizes with similar motion parameters is proposed. Since the algorithm is based on splitting the block that would result in the highest decrease in distortion, it does not require a merging step. Next, VQ techniques are utilized to expedite the search for the block motion vector without significant sacrifice in temporal entropy reduction performance. We compare the performance and the computational requirements of the proposed algorithm with other BMAs.

1. INTRODUCTION

Motion compensation plays a very critical role in image sequence coding. In order to achieve significant data compression, the temporal correlation that exists in image sequences needs to be exploited. Algorithms based on block matching (BMA) are by far the most widely used motion estimation techniques. In a BMA, the image frame is usually partitioned into fixed size blocks. If a maximum motion displacement of W pels is allowed in both horizontal and vertical directions, there are $(2W+1)^2$ motion vector candidates in total, assuming full-pel accuracy. An exhaustive search BMA computes the distortion corresponding to each of these candidates using an accepted block distortion measure (BDM) and selects the best match as the motion vector for the block. Since exhaustive searches are computationally expensive, several fast search algorithms, including Modified Logarithmic Search (MLS) [1], Conjugate Directions Search (CDS) [2] and Three-Step Search (TSS) [3], have been proposed. All of these algorithms attempt to systematically reduce the number of search points over the search space.

Although the BMA is the most widely used motion estimation technique in image sequence coding, it has some drawbacks. One of these drawbacks is the assumption that the motion within each block is uniform. Since this assumption does not always hold, especially for large blocks, a large error in prediction is made by trying to represent the nonuniform motion of the block with a single motion vector. To overcome this problem, variable size block-matching algorithms have been introduced and studied in [4, 5]. The underlying premise of these techniques is to split the blocks with non-uniform motion to get unidirectional motion within every block.

Recently, new techniques that reinterpret the BMA as a Vector Quantization (VQ) problem have been introduced [6, 7]. It was shown in [7] that VQ based techniques can yield performance comparable to other fast search techniques (in terms of temporal redundancy reduction), while requiring only a fraction of the computations and side information. In this paper, we propose an extension to the algorithm of [7] which combines VQ techniques with variable size BMAs.

2. VARIABLE SIZE BMA

Since the assumption that the motion within every block is uniform does not always hold, BMAs suffer from poor motion compensation along moving edges. A possible approach to maintain the validity of this assumption is to use smaller block sizes at the expense of increased side information for motion vectors. In other words, there is a trade-off between efficient motion estimation and the amount of side information to be transmitted to the decoder. Large size blocks require less side information, however they are not as efficient, in terms of motion compensation, as smaller size blocks.

A possible solution to this problem is to use variable block sizes. Such algorithms have been introduced and discussed in [4, 5]. One such algorithm proposed in [4] is essentially a split/merge segmentation scheme based on regular decomposition of an image frame into blocks of varying sizes with more or less uniform motion parameters. First, the image frame is split into a number of blocks. For each block, the best motion vector in the search range is computed based on the Sum-Square-Error (SSE) measure. If this error is above a predetermined threshold, the block is split again. Splitting can be into two or four subblocks. The algorithm continues in this fashion until the SSE is under the threshold or the preset minimum block size is reached. It is important to note that at every stage, a new threshold is computed by obtaining the product of normalized SSE and the area of the new subblock. The final decomposition

---

THIS WORK WAS SUPPORTED IN PART BY THE NATIONAL SCIENCE FOUNDATION UNDER GRANT NO. NCR-9258374

0-7803-3192-3/96 $5.00©1996 IEEE

structure can be represented using a binary or quad tree depending on the type of splitting adopted. Next, a merging operation is performed.

In this work we have used a modified version of the algorithm of [4] to eliminate the merging process. In our algorithm we perform the splitting in an iterative fashion. At each iteration, only the block that provides the largest decrease in overall distortion is split. Splitting halts when no block provides significant improvement, or when a certain maximum number of blocks are obtained. The binary tree decomposition algorithm is described in more detail as follows:

1. Let $s(n_1, n_2, k)$ be the intensity of the pixel at location (n_1, n_2) of the kth frame. Decompose this frame into M $N \times N$ non-overlapping blocks B_i, $i = 1, \cdots, M$. Let $size(B_i)$ be the number of pixels in B_i, T_B be the total number of blocks allowed per frame, and T_D be a fixed distortion threshold.

2. Find the motion vector $\mathbf{d}_i = (d_i^x, d_i^y)$ for every B_i that achieves the minimum block distortion

$$D_i = \sum_{(n_1,n_2)\in B_i} D(s(n_1, n_2, k), s(n_1 + d_i^x, n_2 + d_i^y, k + 1)).$$

3. For every B_i; if $D_i > T_D \times size(B_i)$, temporarily split B_i into a left child B_i^L and a right child B_i^R, and compute $\Delta D_i = D_i - (D_i^L + D_i^R)$. If $D_i \le T_D \times size(B_i)$, set $\Delta D_i = 0$.

4. Find B_k such that $\Delta D_k > \Delta D_i$, for $\forall i, i \neq k$. Split B_k into B_k^L and B_k^R. Compute $D_k^L, D_k^R, \Delta D_k^L, \Delta D_k^R$. Delete B_k from the block list and add B_k^L and B_k^R to the list. Set $M = M + 1$.

5. If $M = T_B$ or $D_i < T_D \times size(B_i), \forall i$, stop. Otherwise go to 4.

For variable size BMAs, the decomposition tree needs to be transmitted to the decoder together with the motion vector information. If a binary tree is used for decomposition then $2L - 1$ bits are required to represent a tree with L leaf nodes, i.e. L blocks. Therefore, the total side information needed will be $L \times b + 2L - 1$, where b is the number of bits required for each motion vector. An example is illustrated in Figure 1.

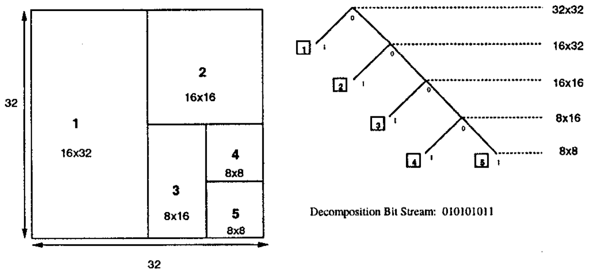

Figure 1: Binary Tree Decomposition.

A major problem with variable size block-matching algorithms is the number of computations required. Although the exact number depends on the particular decomposition, they require roughly two to four times as many computations as fixed size block matching.

3. FINITE STATE MOTION VECTOR QUANTIZATION

The motion fields of typical image sequences are smooth and exhibit a high degree of spatial-temporal correlation. Since a BMA attempts to represent these smooth motion fields with a discrete number of motion vectors, there is usually a high degree of correlation among them [8]. Table 1 presents the entropy content of the motion vectors of the Miss America sequence. In this table, Y, A, and B denote the motion vector of the corresponding block in the reference frame, the previous motion vector, and the upper motion vector, respectively, as illustrated in Figure 2. The motion vectors were computed using exhaustive search BMA with 16 × 16 blocks, based on Mean Absolute Difference (MAD) as the BDM and a maximum motion displacement of 6 pels was allowed. This corresponds to an exhaustive search of 169 candidate vectors, or equivalently, a maximum possible zero-order entropy of approximately 7.4 bits per motion vector.

Table 1: Inter/intra frame motion vector entropies of Miss America sequence (Block size 16 × 16, maximum displacement=6).

$H(X)$	$H(X\|Y)$	$H(X\|A)$	$H(X\|B)$
5.0703	4.0011	4.3009	4.1680

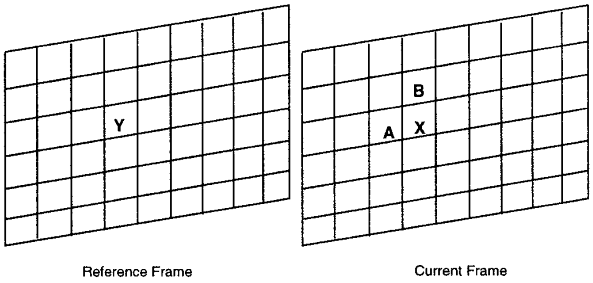

Figure 2: Blocks for Table 1.

It is clear from the table that the motion vectors are correlated in both spatial and temporal directions. Although the motion vectors computed using exhaustive search BMA can be further compressed during transmission by any of several well known entropy coding techniques, the amount of computation required to compute them remains extremely high.

The problem of limiting the search points over the motion vector space can be interpreted as a VQ problem. If the maximum displacement allowed in either direction is W, corresponding to $(2W + 1)^2$ motion vector candidates, a codebook with only N codewords, where $N \ll (2W + 1)^2$, can be designed using the Generalized Lloyd Algorithm (GLA). Then the number of computations and side information can be reduced proportionally by the ratio: $\log_2(N)/\log_2((2W + 1)^2)$. Algorithms using this approach are investigated in [6] and [7].

It is frequently observed that the motion fields of successive frames exhibit a high degree of short-term interframe correlation in addition to the intraframe correlation. In other words, since the motion field does not usually change rapidly, the motion vectors of corresponding blocks in successive frames are correlated. The statistics in Table 1 also support this assumption. Thus, it is reasonable to assume that the corresponding blocks of successive frames have similar motion vectors and they will be in neighboring Voronoi regions in the VQ framework. This property has been used to further improve the VQ codebook design in [7]. Using VQ design techniques, a VQ of codebook size N is designed with respect to the MSE distortion measure. For every codeword of this "supercodebook", $K-1$ nearest neighbor codewords are found to form an N-state finite machine. In other words, for every codeword, $K-1$ additional state codewords are selected from the supercodebook of size N and these codewords form state codebooks of size K. The motion estimation is performed by using the codewords in the state codebook specified by the motion vector of the corresponding block in the previous frame. The codeword yielding the minimum distortion according to our BDM is selected as the motion vector. It was shown in [7] that this Finite State Motion Vector Quantization (FSMVQ) technique required about half the number of BDM evaluations of conventional fast algorithms and generally performed as well in terms of temporal entropy reduction performance. FSMVQ also offered considerable savings on the motion vector data rate.

4. VARIABLE SIZE FINITE STATE MOTION VECTOR QUANTIZATION

As discussed earlier, the most critical trade-off in BMA is between the motion compensation and the amount of side information required. Variable size BMAs offer improved temporal entropy reduction performance, however they require more side information. They also have higher computational requirements. Since FSMVQ offers good temporal entropy reduction performance, together with low side information and fewer computations, it is suitable to be used as a variable size BMA.

If an image frame is decomposed into blocks of different sizes as described in Section 2, it is expected that the blocks of smaller size will have, in general, different motion characteristics than the blocks of larger size. For example, stationary blocks are not expected to be subdivided further since they tend to exhibit uniform motion within the block. Therefore, separate codebooks can be designed to benefit from this property. A block diagram for this architecture has been depicted in Figure 3. This would result in better entropy reduction performance with fewer computations.

Although FSMVQ of [7] is suitable to be implemented as a variable size BMA, some modifications are needed. Since the corresponding block in the reference frame is not necessarily the same size as the current block in variable size BMAs, a new approach for determining the current state needs to be developed. In this work, we have adopted the following approach which resulted in good performance: Initially, i.e. at the root node of the decomposition tree, a weighted sum of the motion vectors of the corresponding "macroblock" in the reference frame is computed and the codeword closest to this sum is chosen as the current state. Later, the codeword closest to the parent node's motion vector is used as the current state. Similar to the fixed size FSMVQ in [7], if the resultant distortion of the block is above a threshold, the search is performed over the entire VQ codebook. Although this increases the computations, it improves performance on instances where current state information is not highly correlated with the true motion vector of the block, like in object boundaries and scene changes.

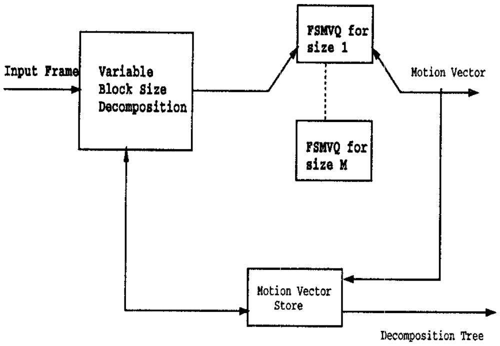

Figure 3: Variable Block Size Motion Estimation using FSVQ.

5. EXPERIMENTAL RESULTS

Experiments were conducted on the 8-bit luminance component of the Miss America sequence. 8 pels were cropped off from the end of every line to allow the decomposition of the frames into 32 x 32 non-overlapping blocks with no residue. First 90 frames of the sequence were used in training and the remaining 60 were used during the testing phase. The maximum displacements in both horizontal and vertical directions were chosen to be 6 pels and absolute difference was adopted as the BDM. A binary tree was used for decomposition and the initial splitting was done in the vertical direction. The codebooks needed in our FSMVQ algorithm were designed with a supercodebook size of 20 codewords and a state codebook size of 5 codewords for every block size. The minimum block size was selected to be 4 x 4 pels.

Figure 4 shows the performance of the variable size decomposition for different numbers of blocks. Here the rate includes both entropy of the motion compensated residue and the side information, i.e. the decomposition tree and the motion vectors.

Figure 5 shows the performance of different variable size BMAs using 400 blocks and Table 2 shows their average performances over the testing set. Here also, average rate includes both the entropy of the residual signal and the side information. Since these algorithms use variable size blocks, the following approach was used to compare their computational requirements. The number of pels in the block in each evaluation of the BDM is summed over each frame and this value is used to measure the amount of computations. It can be seen that FSMVQ performs comparable to the Modified Logarithmic Search. However it

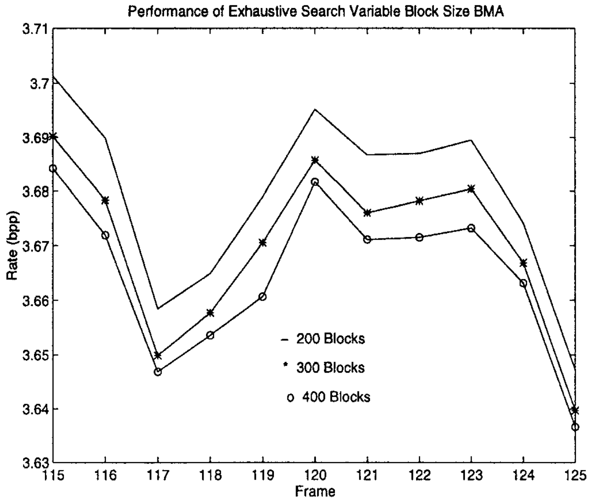

Figure 4: Variable Size Exhaustive Search BMA with Different Numbers of Blocks.

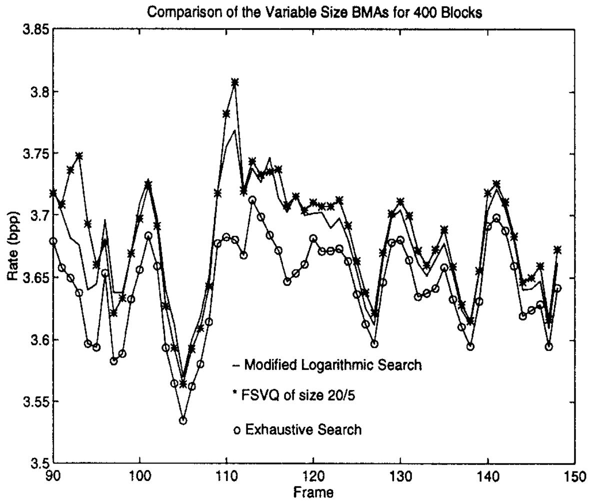

Figure 5: Comparison of Variable Size BMAs with 400 Blocks.

requires approximately %60 fewer computations.

Table 2: Comparison of Variable Size BMAs with 400 Blocks on Frames 90–149 of Miss America Sequence.

Search Algorithm	Average Rate (bpp)	Average Computations per Frame
Exhaustive Search	3.64208	71,001,853
Modified Logarithmic Search [1]	3.67487	6,188,947
FSMVQ	3.67547	2,453,900

6. REFERENCES

[1] S. Kappagantula and K. R. Rao, "Motion predictive interframe coding", *IEEE Transactions on Communications*, vol. 33, pp. 1011–1015, Sept. 1985.

[2] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation", *IEEE Transactions on Communications*, vol. 33, pp. 888–896, Aug. 1985.

[3] K. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing", in *National Telecommunications Conference*, Nov. 1981, pp. G5.3.1–G5.3.5.

[4] M. H. Chan, Y. B. Yu, and A. G. Constantinides, "Variable size block matching motion compensation with applications to video coding", *Proceedings of the IEE*, vol. 137, pp. 205–212, Aug. 1990.

[5] H. Jelveh and A. K. Nandi, "Improved variable size block matching motion compensation for video conferencing applications", in *Digital Signal Processing 91, Proceedings of the International Conference*, Sept. 1991, pp. 391–396.

[6] Y. Y. Lee and J. W. Woods, "Motion vector quantization for video coding", *IEEE Transactions on Image Processing*, vol. 4, pp. 378–382, Mar. 1995.

[7] A. Bilgin, H. Abut, and M. W. Marcellin, "A novel vector quantizer based architecture for block-matching motion estimation", in *Proc. of Twenty Nineth Asilomar Conference on Signals and Computers*, Oct. 1995, vol. 2.

[8] M. T. Orchard, "A comparison of techniques for estimating block motion in image sequence coding", in *SPIE, Visual Communications and Image Processing IV*, 1989, vol. 1199, pp. 248–258.