

A Low-Complexity Rate-Distortion Model for Motion Estimation in H.263

Feng Chen¹, John D. Villasenor¹ and Dong Seek Park²

¹Electrical Engineering Department
University of California, Los Angeles
405 Hilgard Avenue, Los Angeles, CA 90095, U.S.A.
E-mail: fchen@icsl.ucla.edu, villa@icsl.ucla.edu

²Samsung Electronics Company
Kiheung, Korea
E-mail: dspark@icsl.ucla.edu

ABSTRACT

The ITU-T low-bit-rate video coding standard H.263 supports efficient transmission of digital video over narrow-band telecommunication channels. The standard provides an excellent framework for low bit rate video coding, and leaves substantial design flexibility for tasks such as motion estimation to the implementer. We propose here an alternative to the sum of absolute difference (SAD) metric that is used as the criterion for identifying motion vectors in many video coding implementations. The SAD usually fails to identify the rate-distortion optimal motion vector for a block because it does not take into account of the number of bits used to code the motion vector. One solution which has been proposed in the past is to use the Lagrangian cost $J=D+\lambda R$ as the cost function in motion estimation. This allows near optimal motion vector search in the rate-distortion sense but is too complex to be used in a practical video coding system. In this paper we present an alternative motion estimation algorithm that considers rate-distortion trade-offs in a low complexity framework, resulting in both higher coding efficiency and faster encoding speed.

1. INTRODUCTION

The ITU-T low-bit-rate video coding standard H.263 provides an efficient framework for encoding digital video at bit rates of 64 kbps or less [1]. Like most standards, H.263 specifies a framework to ensure compatibility between users, and leaves room for innovations by individual implementers. One area which plays a crucial role in determining the overall coding efficiency, and which is not specified by the standard, is the means for identifying the best motion vectors.

Many practical video coders use the sum of absolute difference (SAD) as a criterion for motion estimation due to its low complexity. However, there are two major drawbacks to using SAD. First, although its complexity is relatively low compared to metrics such as mean square error due to the absence of multiplications, SAD still requires $2N^2$ additions (subtractions) for the matching of each $N \times N$

block for each candidate motion vector. As a result, motion estimation consumes a major portion of the encoding time for each frame. Second, SAD alone usually fails to identify the best motion vector for a block in the rate-distortion sense. This is because the SAD gives a fairly good indication of both the distortion of the reconstructed block and the number of bits needed to code the prediction residual, but contains no information on the number of bits needed to code the motion vector. To solve both problems, we propose here a low-complexity motion estimation algorithm consisting of the following three steps: 1) a segmentation process which identifies the background regions of a frame over which no motion estimation is used; 2) for the regions with motion, a fast motion estimation process which identifies a small set of candidate motion vectors using a reduced-resolution SAD criterion; and 3) a detailed motion estimation process for the candidate motion vectors from step 2 using a low-complexity rate-distortion model which combines SAD and the number of bits used for encoding the motion vectors as the decision criterion. The combination of the above three steps provides a motion estimation algorithm with superior coding performance at a much faster encoding speed relative to unmodified SAD.

The rest of the paper is organized as follows: Section 2 describes the fast motion estimation process which consists the step 1 and 2 mentioned above. Section 3 describes the low-complexity rate-distortion model mentioned in step 3 above. Section 4 contains the simulation results and section 5 concludes the paper with a summary of the algorithm and possible future work.

2. FAST MOTION ESTIMATION

To improve the speed of motion estimation, a two-step process is performed during motion estimation for each block. First, segmentation is performed to identify the regions of the frame that have not changed since the last frame. A simple frame differencing and thresholding technique is used to achieve this. This exploits the high fraction of the frame area that is "background" for many

frames in typical low bit rate video conferencing applications. We have found that this preprocessing step before motion estimation provides 30-60% computational savings for talking-head sequences. For the blocks that are identified as belonging to foreground regions, a fast motion estimation procedure is used to identify a specified number of candidate motion vectors. For the implementation described here, we use a reduced-resolution (subsam-pled) SAD as the criterion to pick the set of candidate motion vectors. If the subsampling factor is two in each dimension, we will achieve another factor of four speed improvement over the standard SAD approach. In consid-ering the rate-distortion optimization described below, the null vector (0,0) and the predicted motion vector (from DPCM coding of motion vectors of adjacent blocks) of the block are always included in the set of candidate motion vectors. We note that the fast motion estimation described here uses a fast-computed criterion and can be easily cou-pled with hierarchical motion estimation techniques such as the "log search" algorithm to give even faster perfor-mance.

3. LOW-COMPLEXITY R(D) MODEL

Once the set of candidate motion vectors (the null vec-tor, the predicted motion vector and the set of motion vec-tors that produce the smallest value of subsampled SADs) are obtained, a detailed motion estimation algorithm is used to select the best motion vector for the block from the set. The optimal criterion in the rate-distortion sense for motion estimation is the total Lagrangian cost $J=D+\lambda R=D+\lambda(R_{mv}+R_{res})$, where D is the final distortion of the reconstructed frame and R , the total number of bits used, is composed of R_{mv} , the number of bits used to code the motion vectors and R_{res} , the number of bits to encode the prediction residual blocks [2,3]. The scalar λ is the Lagrange multiplier which corresponds to the negative slope of the rate distortion curve when J is minimized. By varying the value of λ , one traces out the convex hull of the rate distortion curve. The value of λ that corresponds to a rate $R = R_{budget}$ gives the optimal (R,D) for overall distortion minimization given the bit budget constraint R_{budget} , and the resulting bit rate for each block of the video frame constitutes the optimal bit allocation among these blocks. If all of the blocks are independently coded, then we can achieve the rate-distortion optimal motion vector for the current block by finding the motion vector that results minimal value of J given λ [4]. This process, however, is computationally expensive because it involves finding the final distortion of the block after DCT and quantization as well as the number of bits needed to encode both the motion vector and the quantization indi-ces for each candidate motion vector. In coding standards

such as H.263, the motion vectors are coded differentially using a predicted value of the motion vectors from the pre-vious coded blocks. This introduces inter-block depend-ency and transforms the optimization process into a joint optimization problem whose solution requires the method of Viterbi decoding. Although intelligent pruning algo-rithms can be developed to reduce the complexity of such optimization methods [7], the complexity of the final algo-rithm still remains formidable for practical implementa-tions.

We have incorporated a low-complexity motion esti-mation model which approximates the rate distortion opti-mal motion vector search mentioned above yet introduces almost no added complexity compared to simple applica-tion of the SAD criterion. The model introduced here fits within the framework proposed in [6]. Figure 1 illustrates how this algorithm works: before motion estimation, a most-preferred motion vector is established. In the case when there is no inter-block coding for motion vectors, the most-preferred motion vector is the motion vector that requires the least number of bits to encode (provided the entropy coding table is given), or equivalently, the motion vector that has the largest probability of occurrence. In most video coders this is usually the null vector (0,0). In the case when the motion vectors are coded differentially, the difference motion vector between the current and pre-dicted motion vectors is encoded. Therefore the motion vector that will give the most probable difference motion vector will be the most preferred motion vector. If again the null vector (0,0) is the most probable, the most preferred motion vector will simply be the predicted motion vector for the current block. Once the most-preferred motion vector for the block is established, the difference motion vector between a candidate motion vector and the most preferred motion vector is computed and sent to an entropy coding table to determine the number of bits needed to encode this difference motion vector. The number of bits is then sent to a bias model which produces a bias. The cost function of this candidate motion vector is simply the SAD offset by this bias. While there is substan-tial flexibility in choosing the exact form of this model, we have found that a simple linear model usually works fairly well. Figure 2 shows two scatter plots that fit distortion and R_{res} using a set of six motion vectors that produces the smallest SAD for each training block of a frame. The plots show that in both cases, a linear fit works fairly well.

4. SIMULATION RESULTS

We incorporated the above algorithm into the baseline H.263 video codec provided by Telenor [5] and performed

simulation on a number of QCIF video sequences. Figure 2 shows the results of encoding 170 frames of the "car-phone" sequence at about 22-25 kbps at a frame rate of 8 frames/sec. The upper plot shows the comparison of PSNR of the reconstructed video between the baseline (Telenor) codec and the algorithm described here. Our goal was to compare bandwidth at fixed PSNR, and this plot confirms that the PSNR for both algorithms was essentially identical. The lower plot in Figure 2 shows the difference in bits/frame between the two algorithms. The average bit rate for the improved coder is about 22.7 kbps and the average bit rate for the baseline coder is about 24.9 kbps. The overall bandwidth saving using the improved algorithm is about 10%. Furthermore, the encoding speed of the resulting coder is about 5-7 times faster than the original implementation of the TMN H.263 coder of version 1.4a. Similar performance improvement can be observed on other talking head sequences.

5. CONCLUSION

An efficient H.263-compatible motion estimation technique is proposed which enables both high encoding speed and improved coding efficiency. The algorithm consists of three steps: 1) removal of the background region of the frame; 2) use of a subsampled SAD criterion to identify a small set of candidate motion vectors for a block; and 3) detailed motion estimation using the combined information from SAD and bits used for the motion vector. Steps 1 and 2 provide significantly faster implementation of the motion estimation process while step 3 achieves better coding efficiency. Since SAD serves as a reasonable model of the distortion D and the number of bits R_{res} associated with the prediction residual block, but contains no information on the number of bits R_{mv} needed to code the motion vector, a bias value computed using R_{mv} is subtracted from SAD to form a cost function that contains all three pieces of information in the expression for the Lagrangian cost J . When compared with an implementation based only on SAD, the algorithm improves both the coding efficiency and the computational burden.

There are a number of possible ways to improve the performance of the current coder. In terms of encoding speed, one can easily combine a hierarchical motion estimation technique such as a logarithmic search with the subsampled SAD criterion proposed here. This would achieve another order of magnitude reduction in motion estimation complexity without much of degradation in terms of prediction accuracy. One could also adapt the value of the Lagrangian multiplier λ during the encoding process to achieve an more accurate description of the rate-distortion trade off and further improve the compression performance.

ACKNOWLEDGEMENT

This work was supported by Samsung Electronics Company.

REFERENCES

- [1] ITU-T Recommendation H.263, "Video coding for low bitrate communication", Draft May 2, 1995.
- [2] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. on Image Processing*, Vol. 3, No. 5, pp. 533-545, September 1994.
- [3] W.C. Chung, F. Kossentini, and M.J.T. Smith, "Rate-distortion constrained statistical motion estimation for video coding", *Proc. IEEE Int. Conf. Image Processing*, Vol. 3, pp 184-187, October 1995.
- [4] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. on Acoustic, Speech and Signal Processing*, Vol. 36, No. 9, pp. 1445-1453, September 1988.
- [5] Telenor Research, "TMN (H.263) encoder/decoder, version 1.4a, <ftp://bonde.nta.no/pub/tmn>," *TMN(H.263) codec*, May 1995.
- [6] D. T. Hoang, P. M. Long and J. S. Vitter, "Efficient cost measures for motion compensation at low bit rates," *Proc. of IEEE Data Compression Conference*, pp. 102-111, March 31 - April 3, 1996.
- [7] M. Chen and A. N. Willson, Jr., "Rate-distortion optimal motion estimation algorithm for video coding," *Proc. of IEEE International Conference on Acoustic, Speech, and Signal Processing*, Vol. 4, pp. 2096-2099, May 1996.

Figure 1. Low complexity motion estimation model

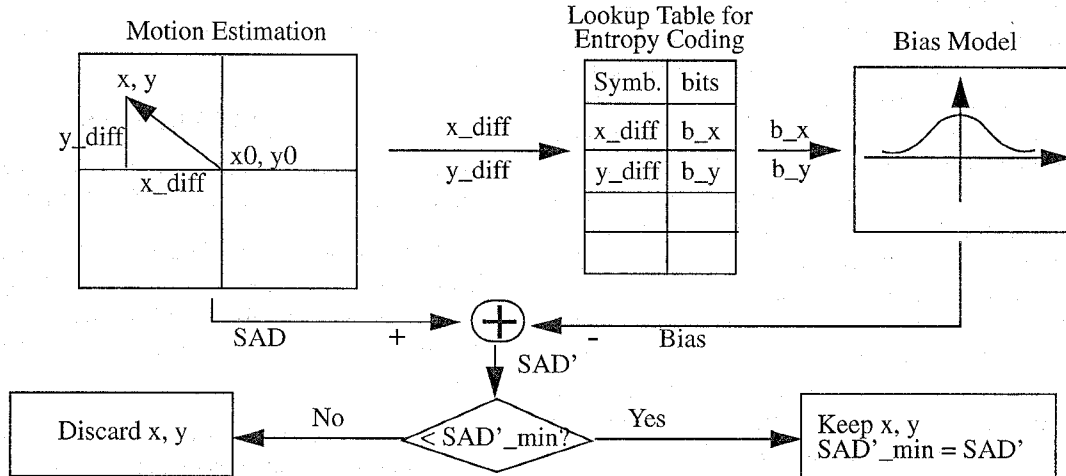


Figure 2. SAD vs. bits and distortion

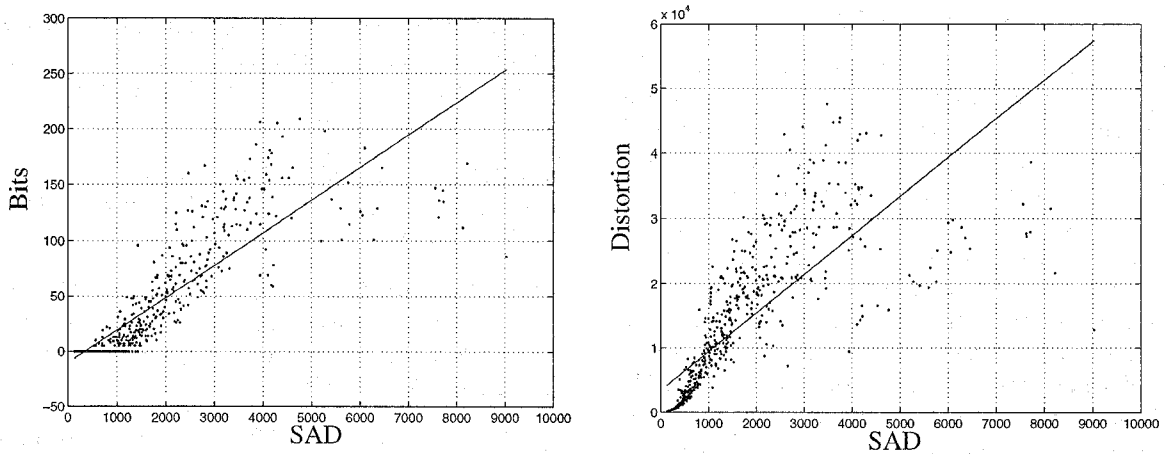


Figure 3. Coding "carphone" at 20 kbps (8 frames/sec): improved vs. baseline

