A Fast Algorithm For Rate Optimized Motion Estimation

John C.-H. Ju, Yen-Kuang Chen, and S.Y. Kung
Department of Electrical Engineering, Princeton Univeristy

Abstract

Motion estimation is known to be the main bottleneck in real-time encoding applications, and the search for an effective motion estimation algorithm has been a challenging problem for years. This paper describes a new block-matching algorithm which is much faster than the full search algorithm and even produces better rate-distortion curves than the full search algorithms. We observe that piecewise continuous motion field reduces the bit rate for differentially encoded motion vectors. Our motion estimation algorithm exploits the spatial correlations of motion vectors effectively in the sense of producing better rate-distortion curves. Furthermore, we incorporate such correlations in a multiresolution framework to reduce the computational complexity. Simulation shows that this method is quite successful because of the homogeneous and reliable estimation of the displacement vectors.

1 Introduction

In most video compression algorithms, there is always a tradeoff between picture quality and compression ratio (and computational cost). Generally speaking, the lower the compression ratio, the better the picture quality. Some researchers have attempted to develop new (better) algorithms which can (1) *achieve higher picture quality with same amount of bits*, or (2) *achieve the same picture quality with less bits*.

It was believed that the less the sum of absolute difference (SAD), the less the number of bits for residue, and, then, the less the total bit rate. Hence, minimal SAD criterion is widely used in BMAs. Namely, the motion vector for this block is the displacement vector which carries the minimal SAD.

$$\textit{motion vector} = \arg\min_{\vec{v}}\{SAD(\vec{v})\} \tag{1}$$

Among several search algorithms to accomplish block matching, the full search methods, where the SADs of all possible displaced candidates within the search area in the previous frame are compared, give the best solution in the viewpoint of estimation error. However, it is observed that the full search BMAs

1. are **computationally** too costly for a practical real-time application [8, 9].
2. usually do not produce the **true motion field**, physical motion, in the video shots [4].
3. usually cannot produce the optimal **bit rate** in many coding standardse, as elaborated in Section 2.

2 Rate Optimized Motion Estimation

The total number of bits to encode an interframe includes the number of bits of coding motion vectors. In some coding standards, such as H.261, H.263, MPEG-1, MPEG-2, MPEG-4, which encode the motion vectors differentially via variable-length coding [7], the number of bits depends on the behavior of motion vectors. Therefore, **it is not always true that the less the SAD, the less the bit rate.** Those conventional

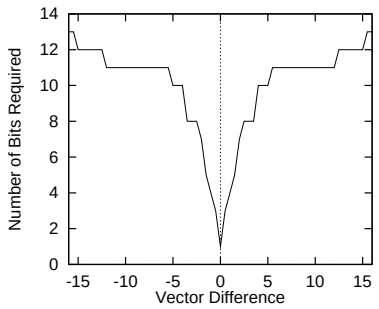

Figure 1: Variable length coding in motion vector difference used in H.263.

block-matching algorithms (BMAs), which treat the motion estimation problem as an optimization problem on SAD only, could suffer from the high price on the differential coding of motion vectors [2].

Figure 1 shows the bit requirement for different vector difference in H.263 standard. The smaller the difference, the less the bits required. A rate-optimized motion estimation algorithm should take account of the total number of bits:

$$\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\} = \arg\min_{\{\vec{v}_i\}} \{bits(residue_1(\vec{v}_1), Q_1) + bits(\vec{v}_1) + bits(residue_2(\vec{v}_2), Q_2) + bits(\Delta\vec{v}_2) + \cdots + bits(residue_n(\vec{v}_n), Q_n) + bits(\Delta\vec{v}_n)\} \quad (2)$$

where $\vec{v}_i$ is the motion vector of block i, $\Delta\vec{v}_i = \vec{v}_i - \vec{v_{i-1}}$[1], $bits(\Delta\vec{v}_i)$ is the number of bits to encode the $\Delta\vec{v}$, $residue_i(\vec{v}_i)$ is the residue of block i, and $bits(residue_i(\vec{v}_i), Q_i)$ is the number of bits required for this residue.

The motion estimation problem is formulated as a shortest path (least bit count) finding problem (which considers the number of bits for texture as well as that for motion vectors), and then used dynamic programming or the Viterbi algorithm to find optimal motion vectors [2].

Different quantization Q_1, Q_2, $\cdots$, Q_n produces different bit rates or distortion of pictures. And, the optimal motion vectors could be different. A Lagrangian-type cost function $J = D + \lambda R$ is further exploited in motion estimation [2] in order to reach near optimal motion vector search in the rate-distortion sense.

Since this scheme is computational too complex in the real implementation, several modified methods that consider rate-distortion tradeoffs in a low complexity framework have been proposed [1, 5, 6].

Our Previous Work

In [3], we propose a rate-optimized motion estimation based on a "true" motion tracker. We observe that piecewise continuous motion field reduces the bit rate for differentially encoded motion vectors. Hence, a neighborhood relaxation method is proposed as the following:

$$motion\ of\ B_i = \arg\min_{\vec{v}} \{SAD(B_i, \vec{v}) + \sum_{B_j \in \mathbf{N}(B_i)} (\mu_{i,j} \times SAD(B_j, \vec{v} + \vec{\delta})\} \quad (3)$$

where $\mathbf{N}(B_i)$ means the neighboring blocks of B_i, a **small** $\vec{\delta}$ is incorporated to allow local variations of motion vectors among neighboring blocks due to the non-translational motions, and $\mu_{i,j}$ is the weighting factor for different neighboring blocks. If a motion vector can induce the SADs of the center block and its neighbors to drop, then it is selected to be the motion vector for the encoder.

It is an *ad hoc* approach which performs motion estimation based on rate optimization without actually count the number of bits for encoding motion vectors. Our motion estimation algorithm exploits the spatial correlations of motion vectors effectively in the sense of producing better rate-distortion curves.

[1] In [7], $\Delta\vec{v}_i \equiv \vec{v}_i - prediction\ of\ \vec{v}_i$. In this paper, we assume $prediction\ of\ \vec{v}_i = \vec{v_{i-1}}$ for simplicity.

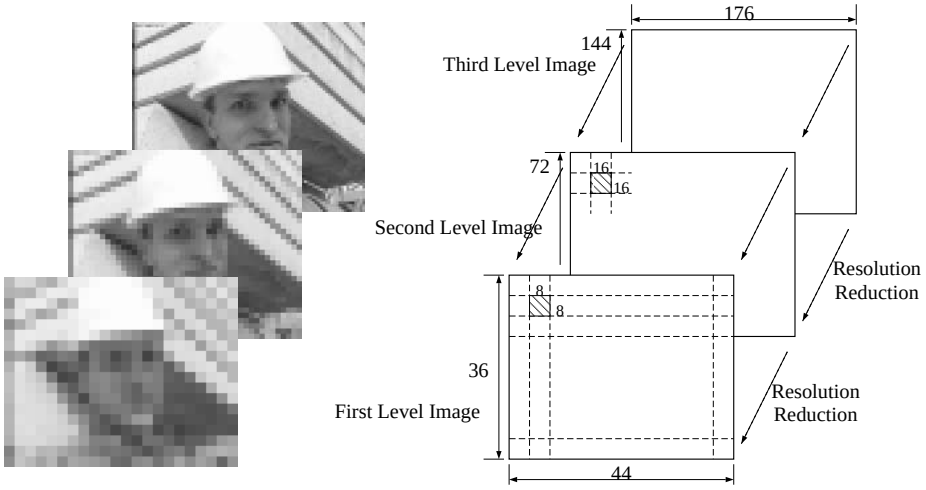

Figure 2: The third level images are the images of original resolution. The second level images are the images of a quarter resolution of the third level. (A pixel in second level is the average of four pixels in the corresponding position.) The first level images are a quarter of the second level.

3 Subblock Multiresolution Motion Estimation

For lower computational complexity, our new block-matching algorithm is based on successive refinement of motion vector candidates on images of different resolutions. Say, three different resolutions, as shown in Figure 2. A coarser resolution image is obtained by computing the mean of 2×2 pixels from finer levels to represent a pixel in the next coarser level. The image size is reduced by half along both horizontal and vertical directions. A motion estimation is first performed on the coarsest resolution and then the motion vectors of finer resolutions are refined based on the motion information obtained at coarser resolutions.

An area in finest resolution represents an area 16 times smaller in coarsest resolution. Therefore, the search area used at the coarsest resolution is also 16 times smaller. Thus, the computational complexity is dramatically reduced. The motion vector obtained from the coarsest resolution is also 4 times coarser in scale. As a result, local refinement in the finer resolution is required for higher accuracy.

Step 1 *The algorithm starts with a search on the images of most coarse resolution.*

The first level images are divided into subblocks of 8×8 pixels, as shown in Figure 2. Each of the subblocks can search in the $\pm 4 \times \pm 4$ possible candidate displaced positions. (The SADs are denoted as $SAD^8_{i,j}(\vec{v})$.) The SADs of macroblocks (of 16×16 pixels) then can be computed as

$$SAD^{16}_{i,j}(\vec{v}) \quad = \quad \sum_{\Delta i=0}^{1} \sum_{\Delta j=0}^{1} \{SAD^8_{2i+\Delta i, 2j+\Delta j}(\vec{v})\}$$

without too much computational overhead. In conventional multiresolution BMAs, only one of

$$\arg\min_{\vec{v}} \{SAD^8(\vec{v})\} \qquad \text{or} \qquad \arg\min_{\vec{v}} \{SAD^{16}(\vec{v})\}$$

is used as the motion candidate but not both. We observe that the motion vector for the macroblock is better at capturing the global common motion when the macroblock is inside a moving object. On the other hand, the motion vector for the subblock is better at capturing its own true motion when the macroblock covers two or more moving objects. Hence, we select the motion vectors which carry minimal SADs either for subblock or for macroblocks as the candidates.

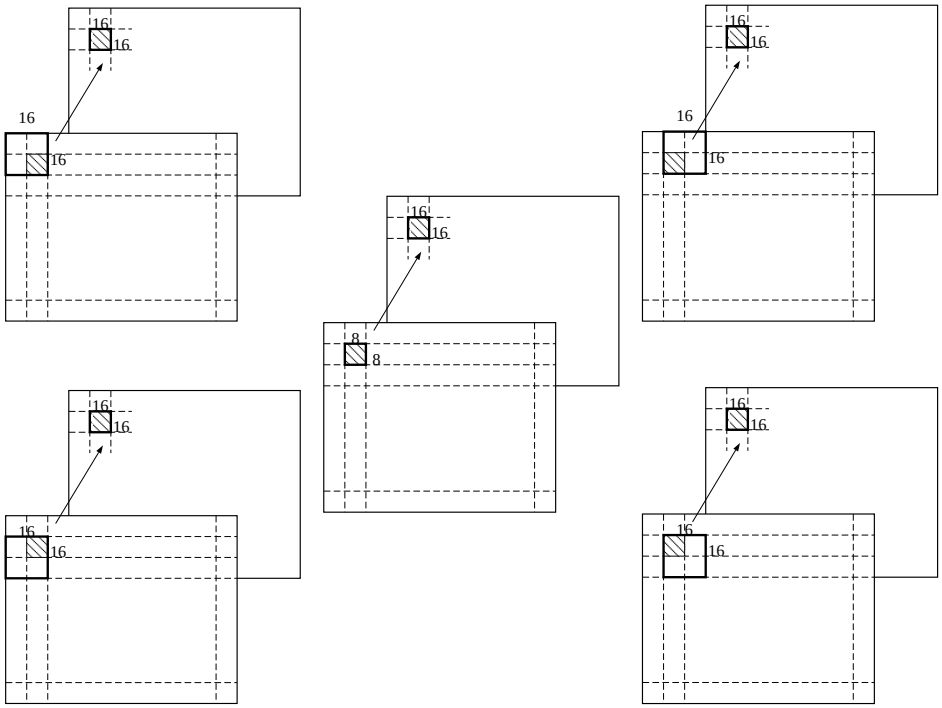

Figure 3: Each macroblock in the second level images is covered by four macroblocks in the first level and one subblock in the first level. Therefore, a macroblocks in this level will inherit the five motion vector candidates (one from the subblock and four from the macroblocks) from the first level as the base motion vectors.

Step 2 *The motion vector candidates are refined on the images of the finer resolution.*

As shown in Figure 3, a macroblock in this level will inherit the five motion vector candidates (one from subblock and four from macroblocks) from the first level as the base motion vectors. Then, the subblocks will search in the $\pm 1 \times \pm 1$ window around these five motion vectors. The motion vectors which carry minimal SADs are selected (either for the subblocks of 8×8 pixels or the macroblocks of 16×16 pixels) again as the motion candidates for the third level.

Step 3 *In the final step of this method, only macroblocks of the finest resolution require motion estimation.*

A macroblock in this level, again, will inherit five motion vectors from the second level as the base motion vectors and, then, searches in the $\pm 1 \times \pm 1$ window around these five motion vectors. The motion vector which carries minimal SAD is selected.

4 Simulation Results

We incorporated the above algorithm into the baseline H.263 video codec provided by Telenor R&D [10].

Figure 4 shows the motion vectors found by the full search approach, multiresolution method without neighborhood relaxation, and our subblock multiresolution motion estimation. The motion field of our method is smoother than that of the full search. As a result, the number of bits for coding motion vectors is lower. Using a fixed quantization parameter, our method can achieve **10.7%** bit-rate reductions (21.3% bit-rate reductions in coding motion vectors) as well as **higher** (+0.01 dB) signal-to-noise ratio (SNR) in coding the 108th frame of the "foreman" sequence.

Note that Figure 4 also shows the motion vectors found by the multiresolution method **without** neighborhood relaxation. It also produces smoother motion field than the original full search method. Thus, it lowers

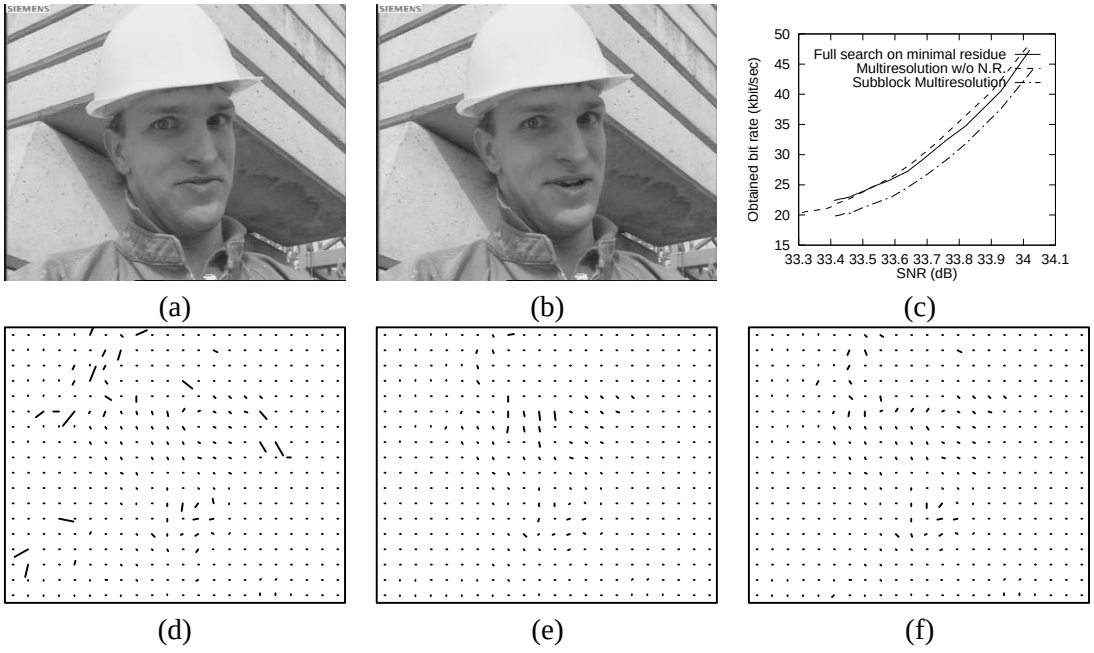

Figure 4: (a)(b) show the 105th frame and the 108th frame of the "foreman" sequence. (c) shows the rate-distortion curve for the original full search method, the multiresolution method without neighborhood relaxation and our method. It is clear that our method could give better quality and better bit-rate. (d) shows the motion vectors found by the full search approach on the minimal residue. (e) shows the motion vectors found by the multiresolution approach **without** neighborhood relaxation. (f) shows the motion vectors found by our subblock multiresolution search method. The motion field is smoother and, as a result, the number of bits for coding motion vectors is less.

the bit rate by 7.2% (it reduces the bits for motion vectors by 25.5%). But, it **degrades** the SNR by -0.06 dB.

Figure 5 shows the rate-distortion curves for all H.263 test QCIF sequences[2]. It is clear that when the quantization step is coarse, the cost on residue coding is relatively smaller and the cost on coding the motion vectors becomes dominant. In this case, our method results in better picture quality and bit rate, as illustrated in the lower-left corner of Figure 5(b). (Note that the reverse phenomenon can be observed in the upper-right corner of Figure 5(b).) If a video has high spatial detail and large amount of local movement (e.g., in "trevor" sequence, there are 6 people moving), then our method cannot work well, as shown in Figure 5(i).

Figure 5 also shows that our new algorithm is also more robust than the previously proposed multi-solution algorithm. In (c)(g)(h)(i), our method cannot perform as well as the full search method, but it performs much closer to that of full search than the conventional multiresolution does. In (a)(d)(e)(f), the performance of our method is better than that of the full search method and the conventional multiresolution.

References

[1] F. Chen, J. D. Villasenor, and D. S. Park, "A Low-Complexity Rate-Distortion Model for Motion Estimation in H.263," in *Proceedings of ICIP'96*, vol. II, pp. 517–520, Sept. 1996.

[2] M. C. Chen and A. N. Willson, Jr., "Rate-Distortion Optimal Motion Estimation Algorithm for Video Coding," in *Proceedings of ICASSP'96*, vol. IV, pp. 2098–2111, May 1996.

[2] A block of 16×16 pixels is used as a macroblock for motion vector estimation. Only forward prediction is implemented in the experiments. The maximum horizontal and vertical search displacement is ± 16.

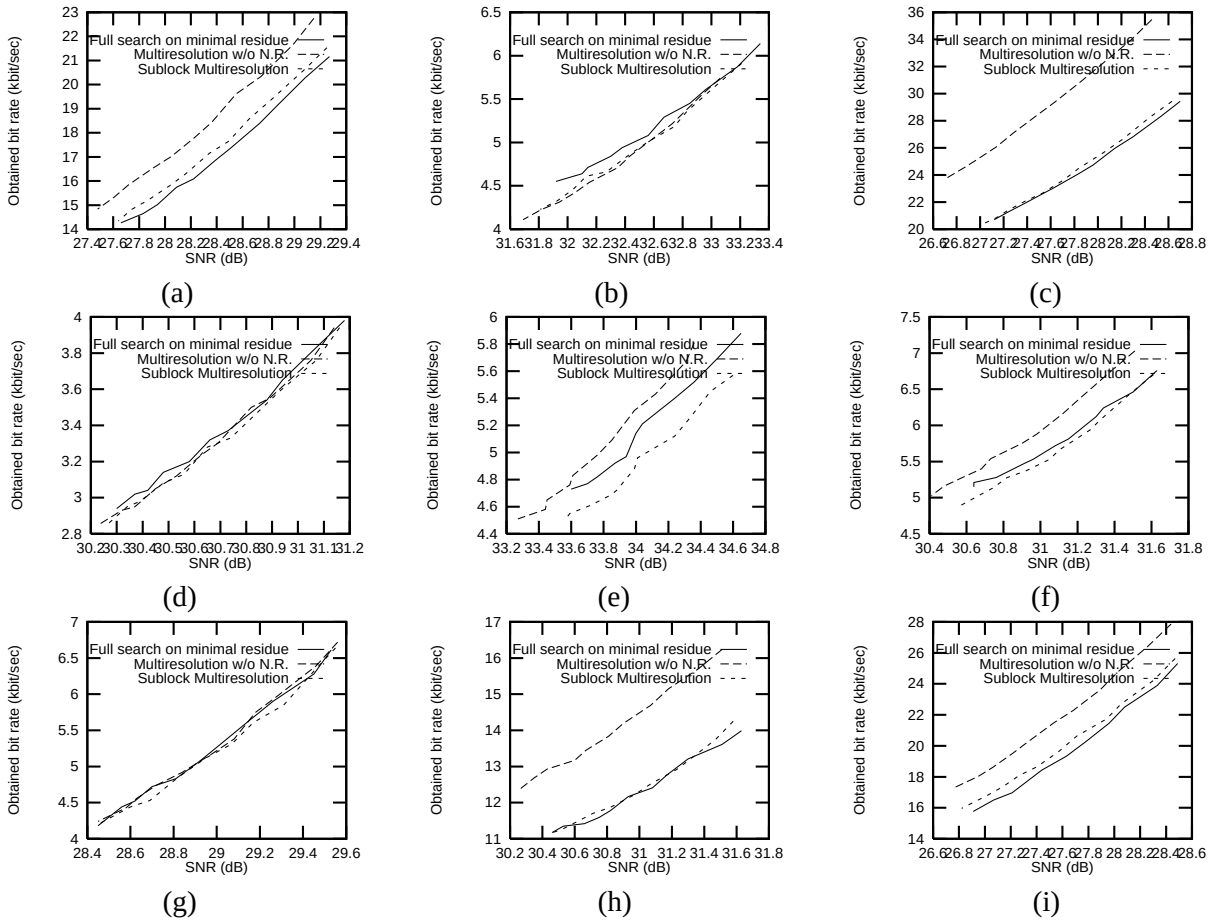

Figure 5: Rate-distortion curves for (a) carphone, (b) claire, (c) foreman, (d) grandma, (e) miss-am, (f) mthr-dotr, (g) salesman, (h) suzie, and (i) trevor sequences. In each sequence, the result is average over the first 102 frames.

[3] Y.-K. Chen and S. Y. Kung, "Rate Optimization by True Motion Estimation," in *Proceedings of IEEE Workshop on Multimedia Signal Processing*, (Princeton, NJ), pp. 187–194, June 1997.

[4] Y.-K. Chen, Y.-T. Lin, and S. Y. Kung, "A Feature Tracking Algorithm Using Neighborhood Relaxation with Multi-Candidate Pre-Screening," in *Proceedings of ICIP'96*, vol. III, pp. 513–516, Sept. 1996.

[5] W. Chung, F. Kossentini, and M. T. Smith, "Rate-distortion-constrained Statistical Motion Estimation for Video Coding," in *Proceedings of ICIP'95*, vol. 3, pp. 184–187, Oct. 1995.

[6] B. Girod, "Rate-constrained Motion Estimation," in *SPIE Proceedings of Visual Communication and Image Processing*, vol. 2308, pp. 1026–1034, Nov. 1994.

[7] ITU Telecommunication Standardization Sector, "ITU-T Recommendation H.263 Video Coding for Low Bitrate Communication." ftp://ftp.std.com/vendors/PictureTel/h324/, May 1996.

[8] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion Compensated Interframe Coding for Video Conference," in *Proc. of Nat. Telecommun. Conf.*, vol. 2, (New Orleans, LA), pp. G5.3.1–5.3.5, Nov/Dec 1981.

[9] B. Liu and A. Zaccarin, "New Fast Algorithms for the Estimation of Block Motion Vectors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, no. 2, pp. 148–157, Apr 1993.

[10] Telenor R&D, "H.263 Encoder Version 2.0." ftp://bonde.nta.no/pub/tmn/software/, June 1996.