A Fast Rate-Optimized Motion Estimation Algorithm for Low-Bit-Rate Video Coding

John C.-H. Ju, Yen-Kuang Chen, and S. Y. Kung

Abstract—**Motion estimation is known to be the main bottleneck in real-time encoding applications, and the search for an effective motion estimation algorithm (in terms of computational complexity and compression efficiency) has been a challenging problem for years. This paper describes a new block-matching algorithm that is much faster than the full search algorithm and occasionally even produces better rate-distortion curves than the full search algorithms. We observe that a 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 successful because of the homogeneous and reliable estimation of the displacement vectors. In nine out of our ten benchmark simulations, the performance of the full search algorithm and that of our subblock multiresolution method is about the same. In one out of our ten benchmark simulations, our method has improvement.**

Index Terms— **Motion estimation algorithm, multiresolution refinement, neighborhood relaxation.**

I. Introduction

VIDEO image compression plays an important role in many multimedia applications, such as video conferencing, videophone, video games, etc. The key to achieving compression is to remove temporal and spatial redundancies in video sequences. Block-matching motion estimation algorithms (BMA's) have been widely exploited in various international video compression standards to remove temporal redundancy.

In most video compression algorithms, there is a tradeoff between picture quality and compression ratio (and computational cost). Generally speaking, the lower the compression ratio is, the better the picture quality. Some researchers have attempted to develop new (better) algorithms that can do the following.

- *Achieve higher picture quality with the same amount of bits,* i.e., minimize the total distortion of intraframes (D_I) and interframes (D_i)

$$D_{\text{total}} = D_I + D_i$$

under a bit-rate constraint

$$\begin{aligned} B_{\text{total}} &= \sum B_I + \sum B_i \\ &= \sum B_I + \sum (B_{i\_res} + B_{i\_mv}) \\ &\le B_{\text{constraint}} \end{aligned}$$

where B_I stands for the number of bits for intraframes, B_i stands for the number of bits for interframe, B_{i_res} stands for the number of bits for interframe residues, and B_{i_mv} stands for the number of bits for interframe motion vectors.

- *Achieve the same picture quality with fewer bits*

$$B_{\text{total}} \quad \sum B_I + \sum (B_{i\_res} + B_{i\_mv})$$

under the same

$$D_{\text{total}} = D_I + D_i$$

In high-quality video compression (e.g., video broadcasting), quantization scales are usually low. Therefore, the number of bits for residue B_{i_res} dominates the total bit rate B_{total}. It was believed that the less the displaced frame difference (referred to as DFD or mean residue) is, the fewer the bits for residue B_{i_res} and, thus, the less the total bit rate B_{total}. Hence, a minimal DFD criterion is widely used in BMA's. Namely, the motion vector for this block is the displacement vector that carries the minimal DFD

$$\text{motion vector} = \arg\min_{\vec{v}} \{\text{DFD}(\vec{v})\} \tag{1}$$

Among several search algorithms to accomplish block matching [4], [14], the full search methods, where the DFD's 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 BMA's:

1) are *computationally* too costly for a practical real-time application;
2) usually do not produce the *natural motion field,* physical motion, which could produce better subjective picture quality;
3) cannot produce in general the *optimal bit rate* for very low-bit-rate video coding standards that differentially encode the motion vectors within a slice and use a significant portion of bits for motion vector encoding.

Manuscript received May 28, 1997; revised March 24, 1999. This paper was recommended by Editor-in-Chief W. Li.

J. C.-H. Ju was with Princeton University, Princeton, NJ 08544 USA. He is now with C-Cube Microsystems, Milpitas, CA 95035 USA.

Y.-K. Chen was with Princeton University, Princeton, NJ 08544 USA. He is now with Microprocessor Research Labs, Intel Corp., Santa Clara, CA 95052 USA.

S. Y. Kung is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA.

Publisher Item Identifier S 1051-8215(99)08174-4.

A Fast Block-Matching Motion Estimation Algorithms

Reducing the number of search positions can reduce the computation. Assume that the DFD increases monotonically as the search moves away from the position of the global minimum DFD. The *three-step search*, the *two-dimensional logarithmic search*, and the *conjugated direction search* block-matching algorithms restrict the number of search locations at large-scale motion vectors at first, and refine the predicted motion vector later [17], [23], [26].

Assume that the motion vector obtained from a larger block size provides a good initial estimate for motion vectors associated with smaller blocks that are contained by the larger block. The hierarchical methods, which use the same image size but different block sizes at each level, also restrict the number of search locations at large-scale motion vectors at first, and refine the predicted motion vector later [3], [13].

Instead of limiting the number of search locations, a 4 : 1 subsampling of pixels is used in DFD calculation to reduce the complexity of motion vector estimation by a factor of four [21]. Furthermore, two other techniques were also proposed to enhance the performance. One is 2 : 1 block subsampling and the other is 4 : 1 subblock motion-field estimation. Combined with 4 : 1 subsampling of pixels, the former reduces the number of operations by a factor of eight while the latter has a reduction factor of 16.

Reducing the number of search positions and the number of pixels in DFD calculation can also reduce computation. The multiresolution motion estimation algorithm relies on the idea of predicting an approximate large-scale motion vector in a coarse-resolution video and refining the predicted motion vector in a multiresolution fashion to obtain the motion vector in the finer resolution. These algorithms use different image resolutions with a smaller image size at a coarser level (i.e., of a pyramid form). A block at the coarser level covers a larger region than a block with the same number of pixels at the finer level, so that a smaller search area can be used at coarser levels. In addition, multiresolution BMA's also reduce the number of pixels in DFD calculation. These algorithms can be further divided into two groups: constant block size and variable block size.

1) In [20] and [28], the same block size is used at each level. If the image size is reduced by half as the level becomes more coarse, one block at a coarser level covers four corresponding blocks at the next finer level. Next, the motion vector of the coarser-level block is either directly used as the initial estimate for the four corresponding finer-level blocks [20] or interpolated to obtain four motion vectors of the finer level [28].
2) In [30], different block sizes are employed at each level to maintain a one-to-one correspondence between blocks in different levels. Next, the motion vector of each block is directly used as an initial estimate for the corresponding block at the finer level.

Instead of reducing the number of search locations, another multiresolution method trades the number of search locations for better estimation quality [10]. It uses different image resolutions with the same image size of a pyramid form. Since the same image size is used at each level, the numbers of possible motion candidates are the same at each level. The block size is not the same at each level and is reduced by half as the level becomes coarser. A block at the coarser level covers the same region as that at the finer level. Then, in the coarsest level, a set of motion candidates is selected from the maximum motion candidate set using a full search with fewer pixels in DFD calculation. In each of the finer levels, the motion candidate set is further screened. At the last level, only a single motion vector is selected.

Assume that the motion field is piecewise continuous in the spatial domain and in the temporal domain. The initial search area could be reduced by exploiting corrections of motion vectors between spatial and temporal adjacent blocks [12], [29].

Based on spatio-temporal correlation integrating with a multiresolution scheme, a fast motion estimation, which is about two orders of magnitude faster than full search motion estimation algorithms, is introduced [4].

B Natural-Motion Estimation Algorithms

Pixels inside a same video object move consistently. Therefore, the natural motion field is piecewise continuous in the spatial domain [2]. Neighborhood constraints on the search of motion vectors are introduced to get piecewise continuous motion fields [22].

Generally, the motion vector obtained from a larger block is more noise resistant than that from a smaller block. The *hierarchical* [3] or *multigrid (quad-tree-structured)* [13], [18], [25] motion estimation algorithms rely on the idea of predicting an approximate large-scale motion vector using larger blocks and refining the predicted motion vector using smaller blocks.

Natural motion field is also piecewise continuous in the temporal domain. Assume that the motion of each block is much smaller than the block; a block-recursive motion estimation algorithm is introduced that exploits corrections of motion vectors between temporal adjacent blocks [12] in addition to spatial adjacent blocks.

In [9], we propose a new tracking method for motion-based scene segmentation.

1) At the outset, we disqualify some of the reference blocks that are considered to be unreliable to track.
2) We adopt a multicandidate prescreening to provide some robustness in selecting motion candidates.
3) Assuming that the natural motion field is piecewise continuous, we determine the motion of a feature block by consulting all its neighboring blocks' directions. This allows a chance that a singular and erroneous motion vector may be corrected by its surrounding motion vectors (similar to median filtering).

C Rate-Optimized Motion Estimation Algorithms

In some coding standards, e.g., H.261, H.263, MPEG-1, and MPEG-2, which encode the motion vectors differentially within a slice [16], the total number of bits B_{total} also includes the number of bits of coding motion vectors B_{i_mv}. In very low-bit-rate coding conditions, a significant portion of bits are

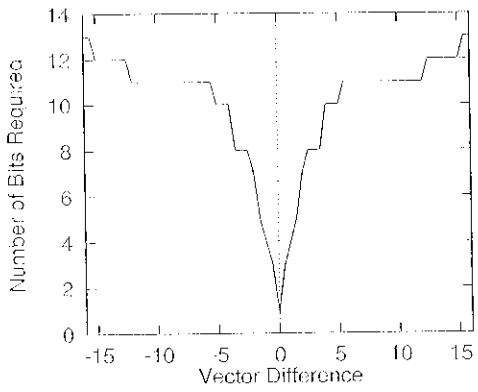

Fig. 1. Variable length coding in motion vector difference used in H.263.

B_{i_mv}. Therefore, *it is not always true that the less the DFD is, the less the bit rate*. Those conventional BMA's that treat the motion estimation problem as an optimization problem on DFD only could suffer from the high price on the differential coding of motion vectors [7].

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

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

where $\vec{v}_i$ is the motion vector of block i, $\Delta\vec{v}_i = \vec{v}_i - \vec{v}_{i-1}$,[1] $\text{bits}(\Delta\vec{v}_i)$ is the number of bits to encode the $\Delta\vec{v}$, $\text{DFD}_i(\vec{v}_i)$ represents the DFD of block i, and $\text{bits}(\text{DFD}_i(\vec{v}_i), Q_i)$ is the number of bits required for this frame difference.

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 dynamic programming or the Viterbi algorithm is used to find optimal motion vectors [7].

Different quantization $Q_1, Q_2, \cdots, Q_n$ produces different bit rates or distortion of pictures. Moreover, the optimal motion vectors could be different. A Lagrangian-type cost function $J = D + \lambda R$ is further exploited in motion estimation [7] in order to reach near optimal motion vector search in the rate-distortion sense.

Since this scheme is computationally too complex in the real implementation, several modified methods that consider rate-distortion tradeoffs in a low complexity framework have been proposed [6], [11], [15]. For example, although the full search on the minimal DFD criterion is used in [27], two special cases in determining the motion vectors are implemented:

1) $\text{DFD}_i(\vec{v}_{i-1})$ is reduced by 100 because of the savings obtained in encoding zero motion vector difference ($\Delta\vec{v}_i = \vec{v}_i - \vec{v}_{i-1} = 0$).
2) If two or more motion vectors have the same DFD, the tie is broken in favor of the shortest motion vector (by spiral searches).

[1] In [16], $\Delta\vec{v}_i = \vec{v}_i -$ prediction of $\vec{v}_i$. In this paper, we assume prediction of $\vec{v}_i = \vec{v}_{i-1}$ for simplicity.

D. Overview of Our Work

In this paper, we present an *ad hoc* approach that performs motion estimation based on rate optimization without actually counting 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 [8], as shown in Section II. Furthermore, we incorporate such correlations in a multiresolution framework to reduce the computational complexity, as shown in Section III. Experimental results show the promising potential of our approach.

II. Natural-Motion Estimation for Rate Optimization

A piecewise continuous motion field is attractive in reducing the bit rate for differentially encoded motion vectors. Hence, a "natural" motion tracker based on neighborhood relaxation offers an effective approach of rate-optimized motion estimation.

Equation (2) can be written as

$$\text{motion of } B_i \approx \arg\min_{\vec{v}} \{\text{bits}(\text{DFD}_i(\vec{v}), Q_i) + \text{bits}(\Delta\vec{v})\}$$

where B_i stands for the block i.

Because it is difficult to mathematically express the bit costs for different DFD's and $\Delta\vec{v}$, the above is first simplified into the following approximation:

$$\text{motion of } B_i \approx \arg\min_{\vec{v}} \left\{\frac{\alpha_1}{Q_i}\text{DFD}_i(\vec{v}) + \alpha_2||\Delta\vec{v}||\right\} \approx \arg\min_{\vec{v}} \{\text{DFD}_i(\vec{v}) + \beta||\Delta\vec{v}||\} \quad (3)$$

because $\text{bits}(\text{DFD}_i(\vec{v}), Q_i)$ and $\text{bits}(\Delta\vec{v})$ grow when $\text{DFD}_i(\vec{v})$ and $||\Delta\vec{v}||$ grow, respectively. $\beta = \alpha_2 Q_i/\alpha_2$.

Here we would like to use an idea commonly adopted in relaxation methods. First, we assume that the rate-optimized motion vector of the block's neighbor produces close-to-minimal DFD. Say that B_j is its neighbor and $\vec{v}_j^*$ is the known optimal motion vector, i.e.,

$$\vec{v}_j^* = \arg\min_{\vec{v}} \{\text{DFD}_j(\vec{v})\} \quad (4)$$

Assume that B_j is a neighbor of B_i, $\vec{v}_j^*$ is the optimal motion vector, and $\text{DFD}_j(\vec{v})$ increases as $\vec{v}$ deviates from $\vec{v}_j^*$ according to

$$\text{DFD}_j(\vec{v}) \approx \text{DFD}_j(\vec{v}_j^*) + \gamma||\vec{v} - \vec{v}_j^*|| \quad (5)$$

or

$$||\Delta\vec{v}|| = ||\vec{v} - \vec{v}_j^*|| \approx \gamma^{-1}\left(\text{DFD}_j(\vec{v}) - \text{DFD}_j\left(\vec{v}_j^*\right)\right) \quad (6)$$

Substituting (6) into (3)

$$\text{motion of } B_i \approx \arg\min_{\vec{v}} \left\{\text{DFD}_i(\vec{v}) + \mu \sum_{B_j \in \mathcal{N}(B_i)} \left(\text{DFD}_j(\vec{v}) - \text{DFD}_j(\vec{v}_j^*)\right)\right\} \quad (7)$$

where $\mathcal{N}(B_i)$ means the neighboring blocks of B_i. $\mu = \beta/\gamma = \alpha_2 Q_i/\alpha_1\gamma$.

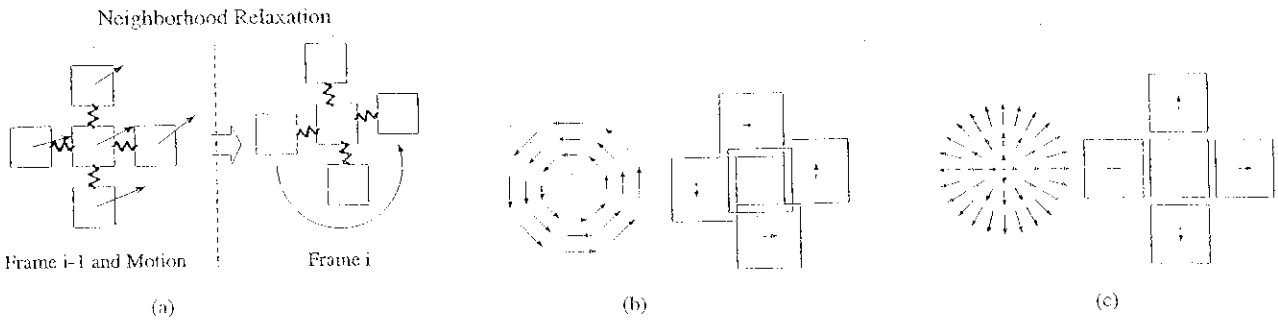

Fig. 2. (a) Neighborhood relaxation will consider the global trend in object motion as well as provide some flexibility to accommodate nontranslational motion. In (a), local variations $\vec{\delta}$ among neighboring blocks are included in order to accommodate other (i.e., nontranslational) affine motions such as (b) rotation, and (c) zooming/approaching.

Here we can use an idea commonly adopted in relaxation method, i.e., we can let v_j^t (and $\mathrm{DFD}_j(\vec{v}_j^t)$) remain constant during the block i updating of the neighborhood relaxation. Therefore, they can be dropped from (7), resulting in

$$\text{motion of } B_i \approx \arg\min_{\vec{v}} \left\{ \mathrm{DFD}_i(\vec{v}) + \mu \sum_{B_j \in N(B_i)} \mathrm{DFD}_j(\vec{v}) \right\} \tag{8}$$

If a motion vector can induce the DFD's of the center block and its neighbors to drop, then it is selected to be the motion vector for the encoder. That is, when two motion vectors produced similar DFD's, the one that is much closer to the neighbors' motion will be selected. The motion field produced by this method will be smoother than that of (1).

The above approach will be inadequate for nontranslational motion, such as object rotation, zooming, and approaching [24]. For example, in Fig. 2(b), assume that an object is rotating counterclockwise. Because (8) assumes that the neighboring blocks will move in the same translational motion, it may not adequately model the rotational motion. Since the neighboring blocks may not have uniform motion vectors, a further relaxation on the neighboring motion vectors is introduced [9].

$$\text{motion of } B_i = \arg\min_{\vec{v}} \left\{ \mathrm{DFD}(B_i, \vec{v}) + \sum_{B_j \in N(B_i)} \mu_{i,j} \times \mathrm{DFD}(B_j, \vec{v} + \vec{\delta}) \right\} \tag{9}$$

where a *small* $\vec{\delta}$ is incorporated to allow local variations of motion vectors among neighboring blocks due to the nontranslational motions and $\mu_{i,j}$ is the weighting factor for different neighboring blocks.[2] As illustrated in Fig. 2, this in principle can track more flexible motions, such as rotation, zooming, shearing, etc.

We incorporated the above algorithm into the baseline H.263 video codec provided by Telenor R&D [27].

[2]The shorter the distance between B_i and B_j is, the larger the $\mu_{i,j}$. The larger the Q_i is, the larger the $\mu_{i,j}$. In practice, we use the four nearest neighbors with $\mu_{i,j} \in [0.05, 0.40]$.

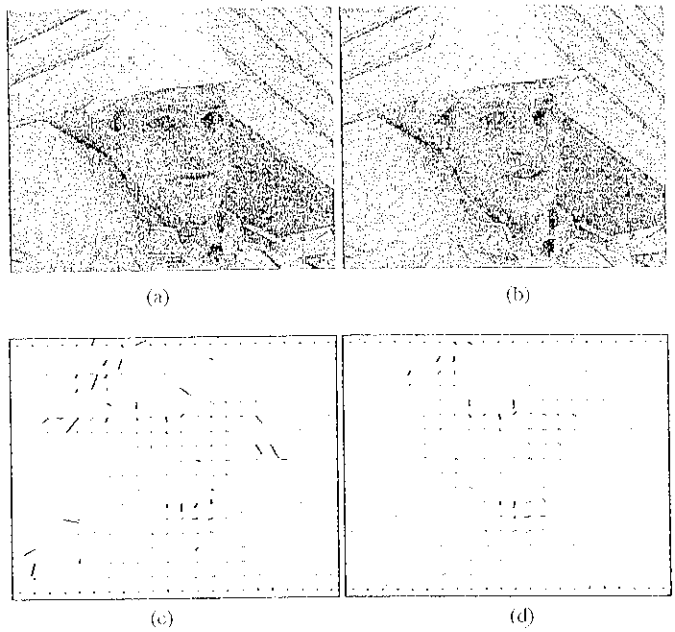

Fig. 3. (a) and (b) show the one-hundred-fifth and one-hundred-eighth frames of the *Foreman* sequence, respectively; (c) shows the motion vectors found by the original approach, which is based on the minimal residue criterion; and (d) shows the motion vectors found by our neighborhood relaxation method. The motion field is smoother, and, as a result, the bits for coding motion vectors is fewer.

The motion vectors found by the original minimal-residue-based approach and our neighborhood relaxation method are shown in Fig. 3. The motion field of our method is smoother, and, as a result, the bits for coding motion vectors are fewer. Using a fixed quantization parameter, our method can achieve 13.9% bit-rate reductions (25.4% bit-rate reductions in coding motion vectors) as well as higher (+0.02 dB) signal-to-noise ratio (SNR) in coding the one-hundred eighth frame of the *Foreman* sequence.

If a smaller DFD is the result of closely tracking the noise effect (which is commonly the case with a full search method), then a small residue does not necessarily result in good SNR. A lower SNR could occur simply because the residue tends to have predominantly higher frequency components and the DCT-based quantization tends to lose higher frequency components [6], [19]. Our natural-motion tracker is deliberately made to be immune to noise. Hence, it can give even higher SNR [8].

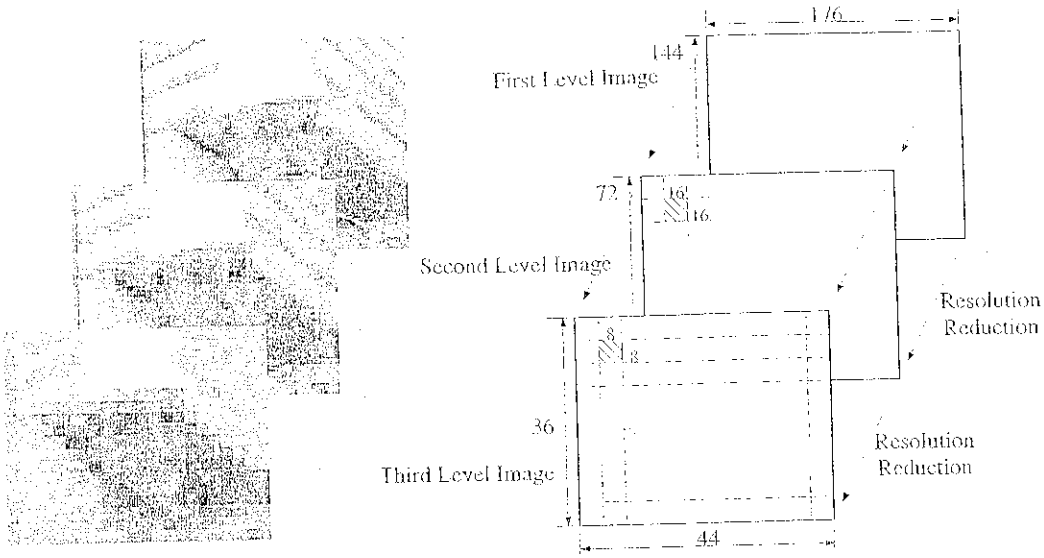

Fig. 4. The first-level images are those of original resolution. The second-level images are those of a quarter resolution of the first level. (A pixel in the second level is the sum of four pixels in the corresponding position.) The third-level images are a quarter of those of the second level.

III. Subblock Multiresolution Motion Estimation

In order to reduce the computational complexity, our new block-matching algorithm is based on successive refinement of motion vector candidates on images of different resolutions. For example, three different resolutions, as shown in Fig. 4, are considered. A coarser resolution image is obtained by computing the sum of 2×2 pixels from finer levels to represent a pixel in the next coarser level, as shown here

$$I^{(k+1)}(x, y, t) = \sum_{\Delta x=0}^{1} \sum_{\Delta y=0}^{1} I^{(k)}(2x + \Delta x, 2y + \Delta y, t)$$

where (x, y) is for the pixel location, t is the frame index, and k is the level of the picture. When $k = 1$, the $I^k(\ )$ is the original picture without resolution reduction.

The image size is reduced by half along both horizontal and vertical directions. The 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 the finest resolution is equivalent to an area 16 times smaller in the 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 four 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 third-level images are divided into subblocks of 8×8 pixels, as shown in Fig. 4. Each of the subblocks can search in the $\pm 4 \times \pm 4$ possible candidate displaced positions. The DFD's are denoted as

$$\mathrm{DFD}_8^{(k)}(i, j, \vec{v}) = \sum_{x=0}^{7} \sum_{y=0}^{7} \left| I^{(k)}(8i + x, 8j + y, t) - I^{(k)}(8i + x + v_x, 8j + y + v_y, t - 1) \right|$$

where $k = 3$ and $\vec{v} = [v_x, v_y]^T$.

Without too much computational overhead, the DFD's of nonoverlapping macroblocks (of 16×16 pixels) then can be computed as

$$\mathrm{DFD}_{16}^{(k)}(i, j, \vec{v}) = \sum_{\Delta i=0}^{1} \sum_{\Delta j=0}^{1} \mathrm{DFD}_8^{(k)}(2i + \Delta i, 2j + \Delta j, \vec{v})$$

In conventional multiresolution BMA's, only one of either

$$2\left(\arg\min_{\vec{v}} \left\{\mathrm{DFD}_8^{(k)}(i, j, \vec{v})\right\}\right)$$

or

$$2\left(\arg\min_{\vec{v}} \left\{\mathrm{DFD}_{16}^{(k)}(i, j, \vec{v})\right\}\right)$$

is used as the motion candidate for the finer level, but not both. We observe that the motion vector for the macroblock (of 16×16 pixels) 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 (of 8×8 pixels) is better at capturing its own natural motion when the macroblock covers two or more moving objects. Hence, we select the motion candidates from vectors that carry minimal DFD's either for subblock or for macroblocks as the candidates, as shown here

$$\begin{aligned}\mathcal{V}^{(k-1)}(i, j) = \Bigg\{ & 2\left(\arg\min_{\vec{v} \in \mathcal{V}^{(k)}([i/2], [j/2])} \left\{\mathrm{DFD}_8^{(k)}(i, j, \vec{v})\right\}\right) \\ & 2\left(\arg\min_{\vec{v} \in \mathcal{V}^{(k)}(i^-, j^-)} \left\{\mathrm{DFD}_{16}^{(k)}(i^-, j^-, \vec{v})\right\}\right) \\ & 2\left(\arg\min_{\vec{v} \in \mathcal{V}^{(k)}(i^-, j^+)} \left\{\mathrm{DFD}_{16}^{(k)}(i^-, j^+, \vec{v})\right\}\right) \\ & 2\left(\arg\min_{\vec{v} \in \mathcal{V}^{(k)}(i^+, j^-)} \left\{\mathrm{DFD}_{16}^{(k)}(i^+, j^-, \vec{v})\right\}\right) \\ & 2\left(\arg\min_{\vec{v} \in \mathcal{V}^{(k)}(i^+, j^+)} \left\{\mathrm{DFD}_{16}^{(k)}(i^+, j^+, \vec{v})\right\}\right)\Bigg\}\end{aligned} \tag{10}$$

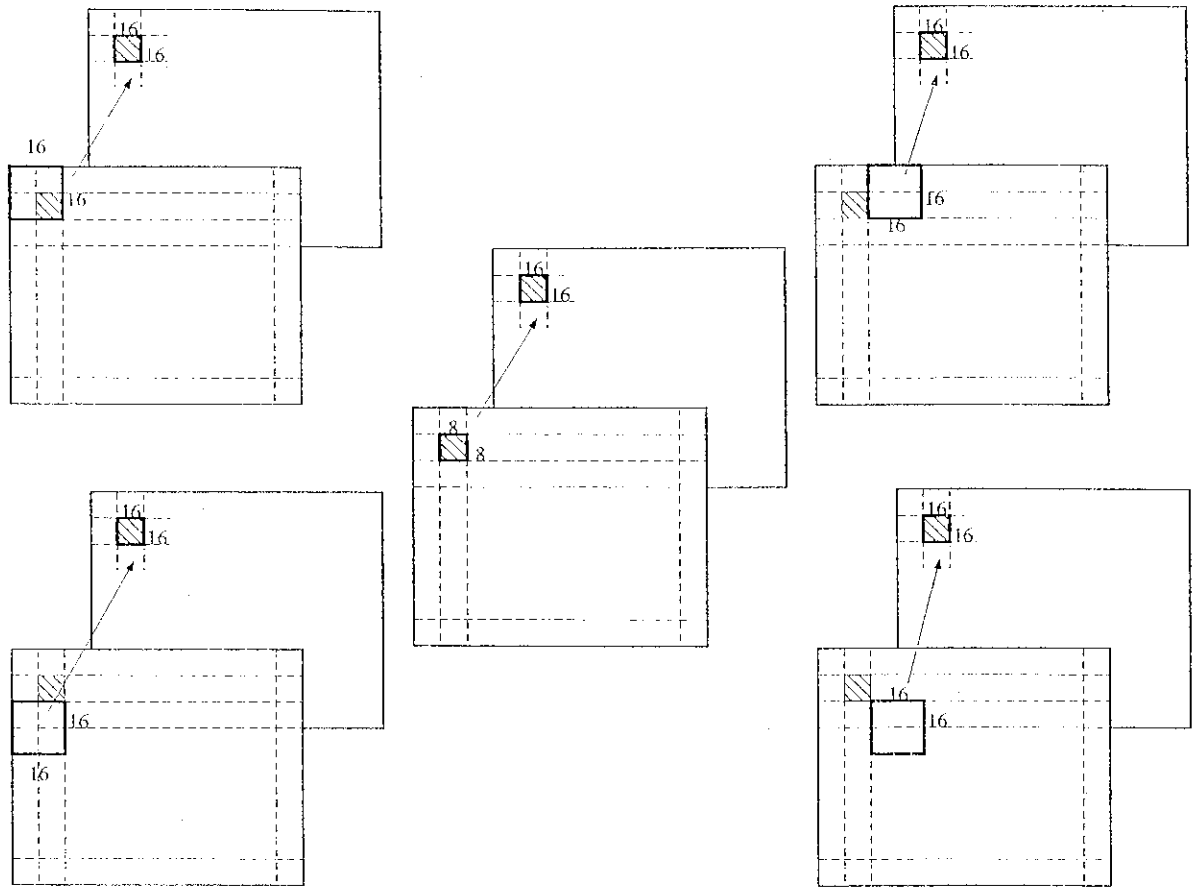

Fig. 5. Each macroblock (of 16×16 pixels) in the second-level image is covered by four macroblocks in the third-level image and one subblock (of 8×8 pixels) in the third-level image. Therefore, a macroblock on this level will inherit the five motion vector candidates (one from the subblock and four from the macroblocks) from the third level as the base motion vectors.

where

$$i^- = [(i-1)/2],\ i^+ = [(i+1)/2]$$
$$j^- = [(j-1)/2],\ j^+ = [(j+1)/2]$$

and the initial motion vector candidate set $\mathcal{V}^{(3)} = \{\pm 4 \times \pm 4\}$.

Step 2) The Motion Vector Candidates Are Refined on the Images of the Finer Resolution. As shown in Fig. 5, a macroblock on this level will inherit the five motion vector candidates (one from subblock and four from macroblocks) from the third 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 that carry minimal DFD's are selected (for either the subblocks of 8×8 pixels or the macroblocks of 16×16 pixels) again as the motion candidates for the first level.

Step 3) In the Final Step of This Method, Only Macroblocks of the Finest Resolution Require Motion Estimation. A macroblock on this level, again, will inherit five motion vectors from the second level as the base motion vectors, and then search in the $\pm 1 \times \pm 1$ window around these five motion vectors. The motion vector that carries minimal DFD is selected.

Fig. 6 shows the motion vectors found by the multiresolution method without neighborhood relaxation and our subblock

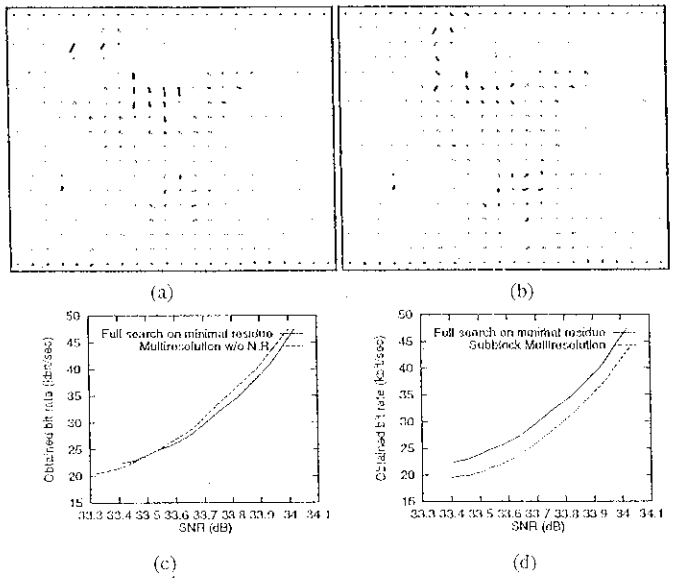

Fig. 6. The simulation result based on the one-hundred-fifth and one-hundred-eighth frames of the *Foreman* sequence as shown in Fig. 3. (a) shows the motion vectors found by the multiresolution approach *without* neighborhood relaxation and (b) shows the motion vectors found by our subblock multiresolution search method. The motion field is smoother, and, as a result, the bits for coding motion vectors is fewer. (c) and (d) show the rate-distortion curves by 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.

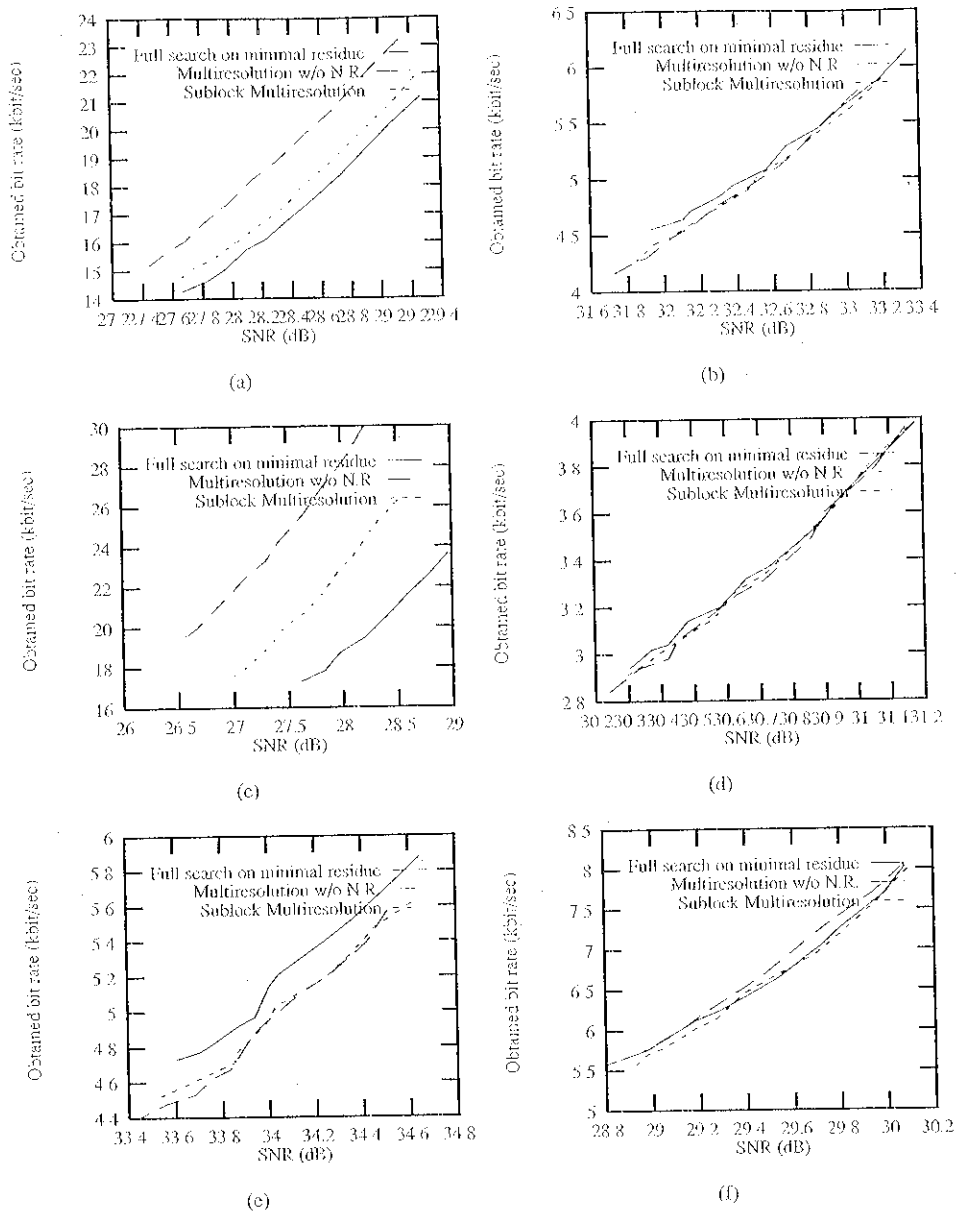

Fig. 7. The rate-distortion curves for all the H.263 sequences: (a) *Carphone* (average over the 380 frames), (b) *Claire* (490 frames), (c) *Foreman* (175 frames), (d) *Grandma* (860 frames), (e) *Miss-am* (150 frames), and (f) *Mthr-dotr* (300 frames).

multiresolution motion estimation (based on the one-hundred-fifth and one-hundred-eighth frames of the *Foreman* sequence as shown in Fig. 3). 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 12.2% bit-rate reductions (25.7% bit-rate reductions in coding motion vectors) as well as a higher (+0.02 dB) SNR in coding the one-hundred-eighth frame of the *Foreman* sequence.

Note that Fig. 6 also shows the motion vectors found by the multiresolution method *without* neighborhood relaxation. It also produces a smoother motion field than the original full search method. Thus, it lowers the bit rate by 6.7% (it reduces the bits for motion vectors by 22.8%). Alas, it *degrades* the SNR by −0.06 dB.

Fig. 7 shows the rate-distortion curves for all H.263 test QCIF sequences[3] [1]. It is clear that when the quantization step is coarse, the cost in terms of residue coding is relatively smaller and the cost in terms of 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 Fig. 7(b). [Note that the reverse phenomenon can be observed in the upper right corner of Fig. 7(b).]

Fig. 7 also shows that our new algorithm is more robust than the previously proposed multisolution algorithm. The H.263 test sequence library can be categorized into the following classes:

[3]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.

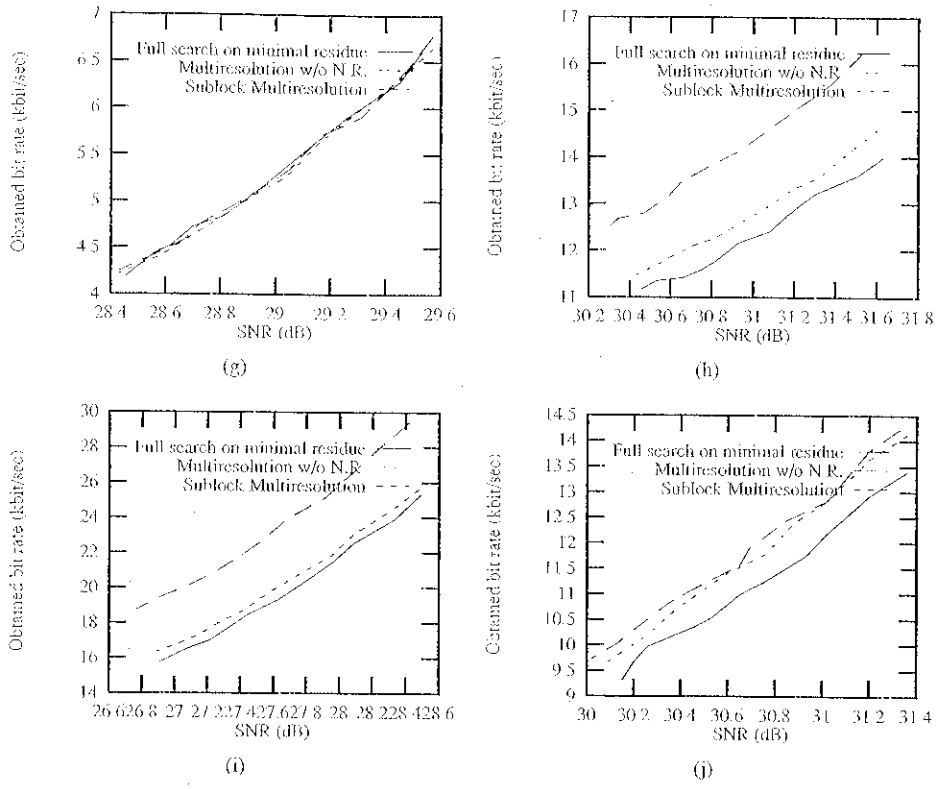

Fig. 7. *(Continued.)* The rate distortion curves for all the H.263 sequences: (g) *Salesman* (445 frames) and (h) *Suzie* (150 frames). Because there is a scene change in the sixtieth frame of the *Trevor* sequence, we divide the simulation of *Trevor* into two parts. (i) is the first part of *Trevor* (60 frames), and (j) is the second part (90 frames).

1) low spatial detail and medium amount of movement (e.g., *Miss-am, Mthr-dotr*);
2) medium spatial detail and medium amount of movement (e.g., *Carphone, Foreman, Suzie*, second part of *Trevor*);
3) high spatial detail and low amount of movement (e.g., *Claire, Grandma, Salesman*);
4) high spatial detail and large amount of local movement (e.g., first part of *Trevor*).

When there is high spatial detail and a low amount of movement, there are not many differences among three algorithms in the (b), (d), and (g) sequences. (a), (c), (h), and (i) show significant improvement (from 0.2 dB up to 0.7 dB) from the proposed algorithm to the conventional multiresolution search algorithm. In all the test sequences except (c), the differences between the proposed algorithm and full search are within the 0.2 dB range. In sequence (e), the proposed algorithm performs 0.2 dB better than the full search method.

IV. Conclusion

We first establish a basic framework based on spatial correlation and then integrate it with a multiresolution scheme.

The proposed algorithm is based on successive refinement of motion vector candidates. It starts with a search on the images of the most coarse resolution where approximate motion vectors are used as a set of motion vector candidates. In each successive searching process, the candidate vectors are refined on the images of the finer resolution to achieve the final motion vector, step by step. By repeating this process, at the images of the finest resolution, a single motion vector can be selected. Since the initial full search at the coarsest level and the motion candidates' refinement use a smaller search area, the computation complexity of the proposed algorithm is less than those of the full search algorithms.

Full search BMA's based on minimal residue usually fail to identify the optimal motion vectors for rate distortion because they do not count the number of bits used to code the motion vectors. In some coding standards, such as H.263 and MPEG-2, which encode the motion vectors differentially within a slice, it is not always true that the less the residue is, the less the bit rate. Simulation shows that our method is successful in bit-rate reduction because the homogeneous motion field reduces a significant amount of bits in coding motion vectors.

Conventional multiresolution algorithms use only information from coarser levels to refine the motion vector in finer levels and do not exploit spatial correlations of the motion vectors. Although its computational complexity is less than that of our method,[4] its motion vectors could come from tracking after a local minimum. Consequently, simulation shows that its coding efficiency is inferior to either the full search BMA's or our method.

The computational complexity required by this motion estimation is around 17 times less than that by the full search algorithms. The overall speedup of the whole video coding using this fast motion estimation is about 6.6 times.

[4] The multiresolution algorithm without neighborhood relaxation is about 30% less complex than our multiresolution algorithm with neighborhood relaxation.

REFERENCES

[1] "H.263 test sequence." [Online] Available FTP: ftp://bonde.nta.no/pub/tmn/qcif_source.

[2] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Systems and experiment performance of optical flow techniques," *Int. J. Comput. Vision*, vol. 13, no. 1, pp. 43–77, 1994.

[3] M. Bierling, "Displacement estimation by hierarchical block matching," in *Proc. SPIE Visual Commun. and Image Processing*, vol. 1001, pp. 942–951, 1988.

[4] J. Chalidabhongse and C.-C. J. Kuo, "Fast motion vector estimation using multiresolution-spatio-temporal correlations," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 7, pp. 477–488, June 1997.

[5] Y. Chan and S.-Y. Kung, "Multi-level pixel difference classification methods," in *Proc. ICIP'95*, Oct. 1995, vol. 3, pp. 252–255.

[6] F. Chen, J. D. Villasenor, and D. S. Park, "A low complexity rate distortion model for motion estimation in H.263," in *Proc. ICIP'96*, Sept. 1996, vol. II, pp. 517–520.

[7] M. C. Chen and A. N. Willson, Jr., "Rate-distortion optimal motion estimation algorithms for motion-compensated transform video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 147–158, Apr. 1998.

[8] Y. K. Chen and S. Y. Kung, "Rate optimization by true motion estimation," in *Proc. IEEE Workshop Multimedia Signal Processing*, June 1997, pp. 187–194.

[9] Y.-K. Chen, Y.-T. Lin, and S. Y. Kung, "A feature tracking algorithm using neighborhood relaxation with multi-candidate pre-screening," in *Proc. ICIP'96*, Sept. 1996, vol. III, pp. 513–516.

[10] K.-W. Chun and J.-B. Ra, "An improved block matching algorithm based on successive refinement of motion vector candidates," *Signal Process. Image Commun.*, no. 6, pp. 115–122, 1994.

[11] W. Chung, F. Kossentini, and M. J. Smith, "Rate-distortion-constrained statistical motion estimation for video coding," in *Proc. ICIP'95*, Oct. 1995, vol. 3, pp. 184–187.

[12] G. de Haan, P. W. A. C. Biezen, H. Huijgen, and O. A. Ojo, "True-motion estimation with 3-D recursive search block matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 368–379, Oct. 1993.

[13] F. Dufaux and M. Kunt, "Multigrid block matching motion estimation with an adaptive local mesh refinement," in *Proc. SPIE Visual Commun. and Image Processing*, 1992, vol. 1818, pp. 97–109.

[14] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," *Proc. IEEE*, vol. 83, pp. 858–876, June 1995.

[15] B. Girod, "Rate constrained motion estimation," in *Proc. SPIE Visual Commun. and Image Processing*, Nov. 1994, vol. 2308, pp. 1026–1034.

[16] ITU Telecommunication Standardization Sector (May 1996) ITU-T recommendation H.263: Video coding for low bitrate communication. [Online] Available FTP: ftp://ftp.std.com/vendors/PictureTel/h324/

[17] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conference," in *Proc. Nat. Telecommun. Conf.*, Nov./Dec. 1981, vol. 2, pp. G5.3.1–5.3.5.

[18] J.-B. Lee and S. D. Kim, "Moving target extraction and image coding based on motion information," *IEICE Trans. Fundamentals*, vol. E78-A, no. 1, pp. 127–130, Jan. 1995.

[19] X. Lee and Y.-Q. Zhang, "A fast hierarchical motion-compensation scheme for video coding using block feature matching," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 6, no. 6, pp. 627–635, Dec. 1996.

[20] J. Li, X. Lin, and Y. Wu, "Multiresolution tree architecture with its application in video sequence coding: A new result," in *Proc. SPIE Visual Commun. and Image Processing*, 1993, vol. 2094, pp. 730–741.

[21] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 148–157, Apr. 1993.

[22] M. T. Orchard, "Predictive motion-field segmentation for image sequence coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 54–70, Feb. 1993.

[23] A. Puri, H.-M. Hang, and D. L. Schilling, "An efficient block matching algorithm for motion-compensated coding," in *Proc. IEEE ICASSP'87*, 1987, pp. 25.4.1–25.4.4.

[24] J. M. Rehg and A. P. Witkin, "Visual tracking with deformation models," in *Proc. IEEE Int. Conf. Robotics and Automation*, Apr. 1991, vol. 1, pp. 844–850.

[25] V. Seferidis and M. Ghanbari, "Generalized block matching motion estimation using quad-tree structured spatial decomposition," in *Proc. Vis. Image Signal Process.*, vol. 141, no. 6, pp. 446–452, Dec. 1994.

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

[27] Telenor R&D (June 1996) H.263 encoder version 2.0. [Online] Available FTP: ftp://bonde.nta.no/pub/tmn/software/

[28] K. M. Uz, M. Vetterli, and D. LeGall, "Interpolative multiresolution coding of advanced television with compatible subchannels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, no. 1, pp. 86–99, 1991.

[29] K. Xie, L. Van Eycken, and A. Oosterlinck, "A new block based motion estimation algorithm," *Signal Process. Image Commun.*, vol. 4, pp. 507–517, Nov. 1992.

[30] S. Zafar, Y. Q. Zhang, and B. Jabbari, "Multiscale video representation using multiresolution motion compensation and wavelet decomposition," *IEEE J. Select. Areas Commun.*, vol. 11, pp. 24–35, Jan. 1993.