MOTION VECTOR OPTIMIZATION OF CONTROL GRID INTERPOLATION AND OVERLAPPED BLOCK MOTION COMPENSATION USING ITERATIVE DYNAMIC PROGRAMMING*

Michael C. Chen and Alan N. Willson, Jr.
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, CA 90095-1600, USA
e-mail: willson@icsl.ucla.edu

ABSTRACT

Interdependence between motion vectors (MVs), introduced by control grid interpolation (CGI) and overlapped block motion compensation (OBMC) algorithms, is the key to improving temporal prediction performance of conventional block-matching motion compensation schemes. Unfortunately, this dependency makes the problem of finding optimal MVs *intractable*. While standard schemes that successively optimize each MV are susceptible to severe local minimum problems, we propose a dynamic programming (DP) paradigm, where each horizontal or vertical slice of MVs is jointly determined during an iterative optimization process. To retain reasonably low complexity, our algorithm effectively identifies an initial search region and then chooses a proper search scheme for each MV. In addition, a computationally-efficient multiscale search strategy is employed. The performance of the proposed method is compared with that of the standard optimization techniques, and our experimental results show that the proposed scheme always gives a better rate-distortion performance. Especially for CGI, the PSNR improvements and the percentage of bit-rate savings provided by our algorithm, in some cases, are in excess of 1.0 dB and 20%, respectively.

1 Introduction

For many video coding applications, the need to use the knowledge of motion in removing temporal redundancy has been evident by the fact that various industrial standards incorporate a motion estimation/compensation technique into their systems in order to achieve a high level of compression. Standard block-matching algorithms (BMAs), based on a translational movement model, are usually insufficient to cope with the various types of motion encountered. It is well known that blocking artifacts are produced. To overcome this deficiency, several motion compensation schemes have been developed. Sullivan and Baker [1] presented the CGI method, in which a geometrical transformation is embedded in the algorithm in order to create a smooth motion field. The OBMC approach, first proposed by Watanabe and Singhal [2] and later improved by Katto *et al.* [3] and Orchard and Sullivan [4] independently, applies a smoothing effect to the boundaries of all blocks by using an overlapped window.

Despite the high compensation performance of CGI and OBMC, finding MVs that optimize rate-distortion performance[†] is not a trivial task. Unlike their counterparts in BMAs, MVs are mutually dependent within the frameworks of CGI and OBMC, which results in distortion dependency between neighboring blocks. For a further complication, a differential MV coding scheme, which intends to reduce bit-rate for transmission, introduces another level of inter-block dependency. Consequently, the optimization of MVs cannot be solved by independently minimizing the rate and distortion of each block. Moreover, the non-causal character of CGI and OBMC methods implies that using sequential decision procedures leads to a suboptimal solution. Because exhaustive search is typically not feasible, in practice the optimization problem is decoupled such that efficient iterative polynomial-time procedures can be developed.

The primary approach to solve such a problem involves a two-stage procedure in which an initial estimate for MVs is first attained and then a greedy optimization approach is used to improve temporal prediction performance. Since a reliable initial solution for MVs is not always easy to obtain, the success of this approach hinges on the development of an effective optimization scheme. Whereas the optimization is usually tackled from a single MV level by all current algorithms, we have devised an iterative DP algorithm that views each row or column of MVs as an entity. Thereby, our approach substantially reduces the chances of being trapped in local minima during a search process.

Despite various patch and window selections for CGI and OBMC [4], [6]-[7], respectively, we have restricted our discussion to the cases where square-shaped patches and windows are used. However, the same principle can also be extended to other variations without much complication.

2 Problem Formulation

Under CGI and OBMC, an image region is partitioned into a group of rectangular blocks given by $\mathcal{A} = (A_{0,0}, ..., A_{N-1,N-1})$. For each block in $\mathcal{A}$, the vertices are chosen as control points where each point is assigned with a MV in the set $\vec{\mathcal{V}}_n = (\vec{v}^n_{0,0}, ..., \vec{v}^n_{N,N})$. Then, within a predefined search area Ω, MVs $\vec{\mathcal{V}}$ are selected to minimize the total rate and distortion. While BMA uses a MV to displace all pels within each block, CGI determines the displacement of each pel by interpolating the MV values of its associated control points based on a spatial transformation. On the other hand, OBMC

*This work was supported by the Office of Naval Research under Grant N00014-95-1-0231.

[†] In [5], we have shown that it is essential to take both rate and distortion into account in order to achieve near-optimal performance for a motion-compensated transform coding system.

calculates the pixel intensity using a linear sum of the estimated pixel values from the neighboring MVs.

Let $I_n(\vec{s})$, and $\tilde{I}_n(\vec{s})$ denote the pixel intensity at spatial index $\vec{s} = (x, y)$ of the n-th frame to be coded, and the decoded frame, respectively. Let $\vec{\mathcal{V}}_n$ represent a set of MVs of frame n and let $\vec{v}_n^{p,q}(\vec{s})$[‡] be the p, q-th MV associated with $\vec{s}$ at frame n. $\mathcal{D}(\mathcal{A})$ represents the total distortion of the region $\mathcal{A}$ and $\mathcal{R}(\vec{\mathcal{V}}_n)$ denotes the total rate of $\mathcal{V}_n$. $\mathcal{H}(\cdot)$ gives the number of bits using an entropy encoding. $w_{p,q}(\vec{s})$ and $a_{p,q}(\vec{s})$ are the interpolation coefficients for the p, q-th associated MV at $\vec{s}$ under CGI and OBMC, respectively. Assuming we adopt a 1-D differential MV coding method, then, for a given Lagrange multiplier value λ, the problem of finding rate-distortion optimal MVs for a given image region $\mathcal{A}$ can be formulated as

$$\min_{\vec{\mathcal{V}}_n}\left\{\lambda\mathcal{R}(\vec{\mathcal{V}}_n) + \sum_{\vec{s}\in\mathcal{A}}\left(I_n(\vec{s}) - \sum_{p,q} w_{p,q}(\vec{s})\tilde{I}_{n-1}(\vec{s} - \vec{v}_n^{p,q}(\vec{s}))\right)^2\right\} \tag{1}$$

$$\min_{\vec{\mathcal{V}}_n}\left\{\lambda\mathcal{R}(\vec{\mathcal{V}}_n) + \sum_{\vec{s}\in\mathcal{A}}\left(I_n(\vec{s}) - \tilde{I}_{n-1}\left(\vec{s} - \sum_{p,q} a_{p,q}(\vec{s})\vec{v}_n^{p,q}(\vec{s})\right)\right)^2\right\} \tag{2}$$

where $\mathcal{R}(\vec{\mathcal{V}}_n) = \sum_{i=0}^{N}\left(\mathcal{H}(v_{i,0}) + \sum_{j=1}^{N}\mathcal{H}(v_{i,j-1}^n - v_{i,j}^n)\right)$.

Due to two-dimensional dependencies between MVs, the solution to (1) and (2) cannot be obtained in polynomial time. One way to simplify the problem is to degenerate the interdependence so that the search's degree of freedom is within control. In the extreme case, all dependencies of a MV can be degenerated by holding the other MVs constant. By doing so, the problem (1) and (2) can be minimized by successively finding the best MV for each control point by minimizing the cost function. Being simple and straightforward, this approach is widely employed by all CGI and OBMC optimization algorithms. However, a major drawback to this approach is that the optimization performance depends heavily on an accurate initial MV estimate due to its rather simplified structural constraints. When the initial estimate MVs fail to approximate the optimal ones, the performance of the method can vary significantly with the sequences.

3 Proposed Iterative Dynamic Programming Approach

Initial Estimate for MVs

Consider an image partition where the block size is $l \times l$, almost all current optimization methods initialize MVs using standard block-matching algorithms with a block size of $l \times l$, as shown in Fig. 1. One major problem with this approach is that it does not take all pixels that a MV is associated with into account. Another drawback is its high sensitivity to noise while the rate effect is ignored. Therefore, the estimated motion field tends to deviate from the optimal one which, in most cases, is rather smooth. To remedy this problem, we propose an objective function in which a $2l \times 2l$ block size and a weighting function $w_{p,q}(\vec{s})$ are incorporated in the distortion terms. In addition, the rate term in (1) is retained in our objective function in order to regulate the initial estimated MVs. The objective function is provided by

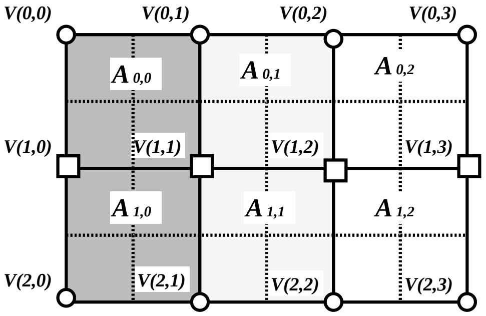


Fig. 1. Structure of a 3×4 CGI and OBMC. The image partition indicated by dotted lines is used to find initial MVs for conventional optimization methods. Circular grid points are held constant, whereas the refinement of square grid points is performed during the horizontal optimization process.

$$\hat{\lambda}\mathcal{R}(\vec{\mathcal{V}}_n) + \sum_{i=0}^{N}\sum_{j=0}^{N}\Phi(i,j) \tag{3}$$

$$\Phi(i,j) = \sum_{p,q=0}^{1}\sum_{\vec{s}\in A_{i-p,j-q}} w_{p,q}(\vec{s})^2\left(I_n(\vec{s}) - \tilde{I}_{n-1}\left(\vec{s} - \vec{v}_{i,j}^n\right)\right)^2.$$

It is clear that the minimization of our proposed objective function cannot be performed by optimizing each MV individually because of a 1-D inter-MV dependence relation in (3). However, it is shown in our previous work [5] that an efficient polynomial-time DP algorithm can be developed to find the optimal MVs without requiring any iteration.[§]

Whereas a simple yet effective approximation of (1) can be found for OBMC, finding a similar formulation for CGI is not as straightforward since an interpolation process is performed on MVs instead of estimated pixel values and since an accurate model of $\tilde{I}_{n-1}(\vec{s})$ is not attainable. One possible way to allow an efficient objective function as (3) is to express the CGI compensated pixel value as a linear combination of displaced pixel values, that is:

$$\tilde{I}_{n-1}\left(\vec{s} - \sum_{p,q} a_{p,q}(\vec{s})\vec{v}_n^{p,q}(\vec{s})\right) = \tilde{I}_{n-1}\left(\sum_{p,q} a_{p,q}(\vec{s})(\vec{s} - \vec{v}_n^{p,q}(\vec{s}))\right)$$
$$\approx \sum_{p,q}\hat{a}_{p,q}(\vec{s})\tilde{I}_{n-1}(\vec{s} - \vec{v}_n^{p,q}(\vec{s})). \tag{4}$$

[‡] To facilitate our explanation, we define that $\vec{v}_n^{p,q}(\vec{s})$ is equivalent to $\vec{v}_{i+p,j+q}^n$ when $\vec{s} \in A_{i,j}$.

[§] Nakaya and Harashima [7] introduced a similar concept called *shape preserving energy*. Unfortunately, their formulation entails a 2-D dependency which demands several iterations for an optimization method to converge. In addition, minimizing their cost function does not always lead to a better solution because in (1) and (2) MVs are more correlated in the horizontal dimension due to the character of the differential MV encoding.

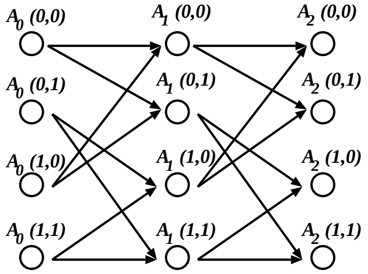


Fig. 2. Trellis diagram for optimizing the first row of MVs while keeping other MVs constant. Here each MV is assumed to have only two search candidates.

Using the approximation (4), we arrive at a similar objective function for CGI, given by

$$\min_{\vec{V}_n}\left\{\hat{\lambda}\mathcal{R}(\vec{V}_n)+\sum_{i=0}^{N}\sum_{j=0}^{N}\Psi(i,j)\right\} \qquad (5)$$

$$\Psi(i,j)=\sum_{p,q=0}^{1}\sum_{\vec{s}\in A_{i-p,j-q}}\hat{a}_{p,q}(\vec{s})^2\left(I_n(\vec{s})-\tilde{I}_{n-1}\left(\vec{s}-\vec{v}^{\,n}_{i,j}\right)\right)^2.$$

Finding $\{\hat{a}_{p,q}(\vec{s})\}$ that will lead to the best compensated performance requires an extensive training with images and is beyond the scope of this paper. However, when the $\{\vec{v}^{\,p,q}_n\}$ are relatively close to each other and the variations of $\{\tilde{I}_{n-1}(\vec{s})\}$ are small, it appears that $\{a_{p,q}(\vec{s})\}$ allows a rather close approximation. Since this condition is usually met by most natural video sequences, we have found empirically that directly adopting $\{a_{p,q}(\vec{s})\}$ gives a satisfactory performance.

MV Refinement

As can be perceived by examining (1) and (2) carefully, the fundamental problem that prevents an efficient algorithm from being developed is the co-existence of horizontal and vertical dependencies between MVs. Consequently, no valid sequential ordering can be found for the MV search process. Based on this observation, our algorithm, in contrast to the conventional approach, only degenerates either the horizontal or vertical dependency by restricting the refinement process to one column or row of MVs at a time. Thus, the planar MV optimization of (1) and (2) is converted to a series of sequential row or column MV optimizations.

Here, we formulate the optimization problem of a MV slice as a problem of finding the shortest path through a spanning tree. In particular, the spanning tree is reduced to a more compact trellis because of the one-dimensional dependency attribute. Thus, the problem can be solved optimally using DP techniques based on the Viterbi algorithm [8]. To illustrate this problem formulation, Fig. 1 shows a 3 × 4 CGI structure, where the circular grid points are held constant and the rectangular grid points are under refinement during the optimization process. The objective is to find $\vec{V}(1)=\{\vec{v}(1,i)|i\in(0,1,2,3)\}$ such that

$$\min_{\vec{V}}\{\mathcal{D}(\mathcal{A})+\lambda\mathcal{R}(\mathcal{A})\}=\min_{\vec{V}}\sum_{i=0}^{2}\{\mathcal{D}(\mathcal{A}_i)+\lambda\mathcal{R}(\mathcal{A}_i)\} \qquad (6)$$

TABLE I. Comparisons of computational complexity for various schemes.

Method	No. Position/Iter.	Iteration
Exhaustive	N^m	1
Conven.	$m\times k$	≥ 1
Conven. (Multiscale)	$9m\times\log\sqrt{k}$	≥ 1
Iterative DP	$m\times k^2$	≥ 1
Multiscale DP	$81m\times\log\sqrt{k}$	≥ 1

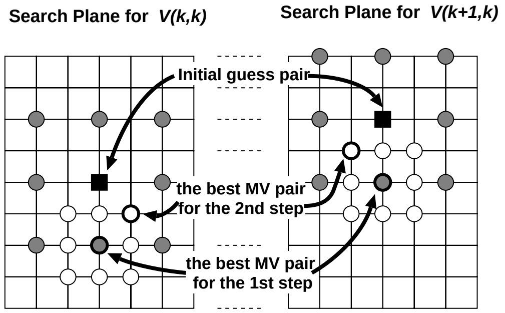
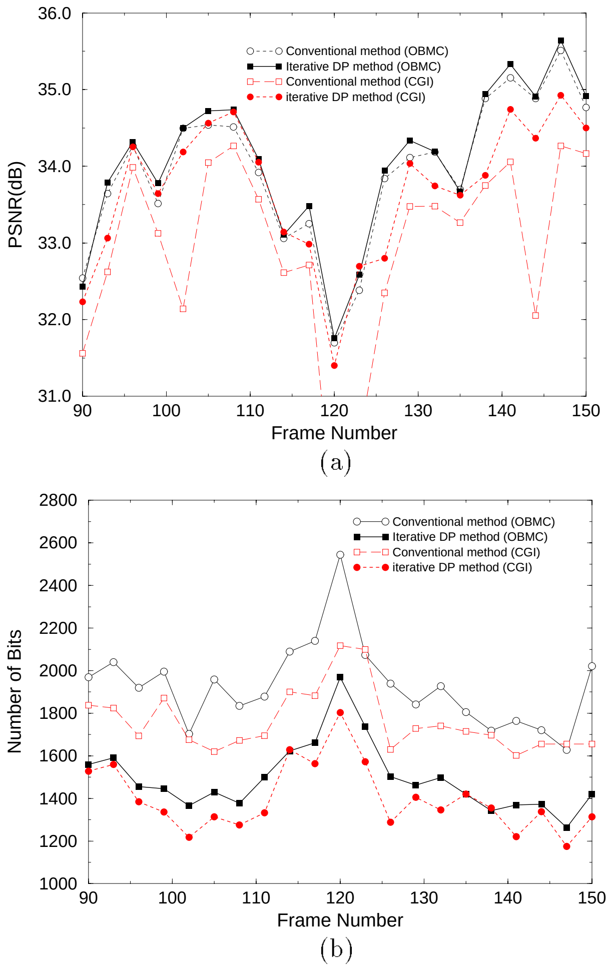

Search Plane for V(k,k)
Search Plane for V(k+1,k)
Initial guess pair
the best MV pair for the 2nd step
the best MV pair for the 1st step

Fig. 3. An example of a two-step multiscale optimization strategy using the proposed DP approach.

where $\mathcal{A}=\{A_{i,j}|i\in(0,1),j\in(0,1,2)\}$ and $\mathcal{A}_i=\{A_{0,i}\cup A_{1,i}\}$.

$A_{0,i}$ and $A_{1,i}$ are grouped as $\mathcal{A}_i$ since they are affected by the same pair of MVs, i.e., $(\vec{v}(1,i),\vec{v}(1,i+1))$. Thus, the number of combinations of $(\vec{v}(1,i),\vec{v}(1,i+1))$ determines the number of states for $\mathcal{A}_i$. To be more exact, there are k^2 states of $\mathcal{A}_i$ to choose from if each MV has k search candidates. The trellis diagram for $k=2$ is shown in Fig. 2. It is noteworthy that the states between adjacent stages are not fully connected; an arc is added only if $\mathcal{A}_i$ and $\mathcal{A}_{i+1}$ share the same $\vec{v}(1,i+1)$.

Table I compares computational complexity, in terms of the number of search positions per iteration, for various algorithms, where N and k are the search window sizes for the initial MV estimate and the MV refinement stages, respectively, and m is the number of MVs in an image. As can be seen, the $O(k^2)$ iterative DP algorithm is more complex than the $O(k)$ conventional method. Since distortion evaluation is very costly for CGI and OBMC, the increase of complexity can severely hamper the implementation of DP algorithms for a large k. Therefore, we adopt a multiscale search strategy, which is similar to that of our predictive logarithmic search [9], in the MV refinement stage. This substantially reduces the computational complexity to $O(\log k)$, which converts to a factor of 208 and 14 in computational savings for a 15×15 and 7×7 refinement window, respectively. To illustrate our multiscale scheme, a graphical example with a 7×7 refinement window is shown in Fig. 3, where the square-shaped points represent the initial estimated MVs and the gray and white circular-shaped points are the search positions in the first and second search step, respectively.



(a)

(b)

Fig. 4. Comparisons of the optimization performance, in terms of (a) PSNR and (b) total number of bits, of the proposed and the conventional optimization schemes on the carphone sequence.

4 Simulation Results

To test the performance of our proposed DP algorithms, we have performed simulations on the carphone and Susie image sequence with a frame rate of 8.3 and 10 frame/sec, respectively. An 8×8 block size, a search range of [-16, 15] were employed. In addition, we adopted a bilinear transform and a 16×16 bilinear window for CGI and OBMC, respectively. In the first three iteration cycles, a 15×15 refinement window was used. Then it was reduced to a size of 7×7. Since conventional closed-loop motion-compensated transform coders use a non-perfectly reconstructed previous-frame image for motion estimation/compensation, to emulate this situation, each frame of image, in our simulations, is transform coded with a PSNR quality greater 37 dB before it is used for temporal prediction.

We applied the iterative DP and its multiscale version to the first 40 frames of the Susie image sequence under a CGI motion compensation framework. According to our experimental results, our multiscale approach only experiences, on average, less than 0.1 dB PSNR degradation and less than a 5% of bit-rate increase while it requires significantly less computation.

Fig. 4 compares the rate-distortion performance (with λ=200) of the iterative DP and conventional single-MV optimization schemes using a multiscale strategy for CGI and OBMC. It is not a surprise to find that the PSNR improvements provided by our proposed algorithm are rather small for OBMC as compared to those for CGI. This outcome can be explained by the fact that MVs from block-matching algorithms give a much closer approximation to the PSNR-optimal MVs for OBMC than those for CGI. As a result, while PSNR performance of a conventional optimization scheme is competitive with that of our method, it suffers from more than 0.8 dB performance loss for CGI. As far as bit-rate performance is concerned, using our iterative DP method leads to a 20% and 22% bit-rate saving for CGI and OBMC, respectively. This indicates that our proposed objective function is more robust against various noise sources, e.g., illumination changes, imperfect motion models and quantization effects. Therefore, the initial MVs tend to give a much better rate-distortion performance. Furthermore, the solutions obtained by jointly optimizing a row or a column of MVs are less likely to deviate much from the optimal ones even when the initial estimate MVs are not reliable.

5 Conclusion

We have presented an iterative DP algorithm for CGI and OBMC. We have shown that the initial MVs estimated by standard block-matching tend to be noisy and do not give a good rate-distortion performance. We proposed using a better objective function for estimating initial MVs and showed how this function can be efficiently optimized without any iteration. While an accurate initial solution is not always attainable for CGI, our iterative DP approach demonstrated a considerable rate-distortion performance gain over the conventional optimization scheme.

References

[1] G. J. Sullivan and R. L. Baker, "Motion compensation for video compression using control grid interpolation," in *Proc. IEEE ICASSP'91,* May 1991, pp. 2713-2716.

[2] H. Watanabe and S. Singhal, "Windowed motion compensation," in *Proc. SPIE Conf. Visual Commun. Image Proc.,* vol. 1605, Nov. 1991, pp. 582-589.

[3] J. Katto, J. Ohki, and M. Ohta, "A wavelet codec with overlapped motion compensation for very low bit-rate environment," *IEEE Trans. Circuits Syst. for Video Technology,* vol. 4, pp. 328-338, June 1994.

[4] M. T. Orchard and G. J. Sullivan, "Overlapped block motion compensation: an estimation-theoretic approach," *IEEE Trans. Image Processing,* vol. 3, pp. 693-699, Sept. 1994.

[5] M. C. Chen and A. N. Willson, Jr., "Rate-distortion optimal motion estimation algorithm for video coding," to appear in *Proc. IEEE ICASSP'96,* May 1996.

[6] Y. Wang and O. Lee, "Active mesh—a feature seeking and tracking image sequence representation scheme," *IEEE Trans. Image Processing,* vol. 3, pp. 610-624, Sept. 1994.

[7] Y. Nakaya and H. Harashima, "Motion compensation based on spatial transforms," *IEEE Trans. Circuits Syst. for Video Technology,* vol. 4, pp. 339-356, June 1994.

[8] D. P. Bertsekas, *Dynamic Programming—Deterministic and Stochastic Models.* Englewood Cliffs, NJ: Prentice Hall, 1987.

[9] M. C. Chen and A. N. Willson, Jr., "A high accuracy predictive logarithmic motion estimation algorithm for video coding," in *Proc. IEEE ISCAS'95,* vol. 1, May 1995, pp. 617-620.