CONTENT ADAPTIVE MOTION ESTIMATION FOR MOBILE VIDEO ENCODERS

Arshad Ahmed and S K Nandy

Supercomputer Education and Research Centre
Indian Institute of Science
Bangalore 560 012 INDIA
[arshad@cadl, nandy@serc].iisc.ernet.in

Paul Sathya

Central R & D
STMicroelectronics
Noida 201 301 INDIA
paul.sathya@st.com

ABSTRACT

Power consumption has emerged as an important constraint in the design of mobile video encoders. As motion estimation accounts for the majority of the total computations involved in video encoding, the algorithm and architecture used affect the quality and power levels of the final solution. In this paper, we present a block matching motion estimation algorithm whose computations are *content complexity* adaptive. The basic framework used is the multi-resolution mean pyramid technique. The algorithm is made macroblock adaptive by dynamically varying the number of candidate motion vectors passed to lower levels, depending on the frequency characteristics of the macroblock being matched and the complexity in the sequence for such characteristics. We use the concept of a *deviation pyramid* in order to estimate the macroblock frequency characteristics. Simulation results show that for typical videophony sequences, the algorithm reduces computational complexity by a factor ranging from 15.5 to 74.0, while maintaining PSNR values close to that obtained by using the full-search block matching algorithm. Simple operations are used in the algorithm to ensure applicability of the proposed algorithm for hardware implementation.

1. INTRODUCTION

The presence of multimedia capabilities on mobile terminals opens up a spectrum of applications such as video-conferencing, video telephony, security monitoring, information broadcast and other such services. But, making wireless video as pervasive as wireless speech communication, presents a number of technological challenges. These challenges relate to the development of efficient video compression techniques and error resilient transmission techniques. The MPEG-4 simple profile [1] which is intended for wireless video applications is representative of the current level of technology in low-bit rate, error resilient video coding. From the viewpoint of system design, all the proposed techniques have to be implemented in the highly power constrained, battery operated environment. Hence, to prolong battery life, we need to make the proposed techniques data aware such that system and algorithm parameters are reconfigured depending on the data being processed[2].

The source coding model of MPEG-4 (which is based on the H.263 standard[3]) employs block based motion compensation for exploiting temporal redundancy and discrete cosine transform for exploiting spatial redundancy. The motion estimation process is computationally intensive and accounts for nearly 70% [4] of the total encoding computations. Hence, there has been a tremendous interest in developing efficient block-matching algorithms.

This work was supported by STMicroelectronics

Algorithms based on the unimodal error surface assumption such as the n -step search and logarithmic search achieve a large magnitude of computational reduction. But, apart from being difficult to implement due to their irregular structure, the drop in PSNR due to local minima problems is unacceptable for H.263 coding where a 1 dB drop generally leads to a perceptible difference in visual quality [5]. Currently proposed motion estimation algorithms which dynamically alter computations based on content [6],[7] operate within the full-search block matching (FSBM) framework. Hence their computational complexity is still high compared to the n -step search. Moreover, the above algorithms operate at the frame level, *i.e.* the search range and hence the computational complexity is the same for all macroblocks of the frame irrespective of the activity in the macroblock.

Our proposed algorithm scales computations at the macro-block level. We use the computationally efficient, multi-resolution mean pyramid [8] as the framework for our algorithm. We scale computations by adaptively deciding the number of candidate motion vectors (CMVs) to be passed from each level of the pyramid. This decision is based on the frequency characteristics of the macroblock being matched. In addition, the number of CMVs are also made dependent on the *content complexity*. Since content complexity is sequence specific, we optimise the motion estimator for every sequence by performing online learning on the first few frames in a sequence. This results in near optimal computations while maintaining PSNR at a high value.

The remaining paper is organised as follows. Section 2 describes the multi-resolution mean pyramid technique and the content based motion estimation algorithm. Simulations and results are presented in Section 3. Section 4 concludes the paper.

2. CONTENT COMPLEXITY BASED MOTION ESTIMATION

The basic framework for our algorithm is the multi-resolution mean pyramid[8] which is described below.

2.1. Multi-resolution mean pyramid

A diagrammatic view of the process of generating the mean pyramid is shown in Fig.1. In equation form:

$$x_L^k(p, q) = \left[\frac{1}{4} \sum_{u=-1}^0 \sum_{v=-1}^0 x_{L-1}^k(2p+u, 2q+v) \right] \quad (1)$$
$$1 \leq L \leq 2, 1 \leq p \leq \frac{N_H}{2^L}, 1 \leq q \leq \frac{N_V}{2^L}$$

where L denotes the pyramid level, k denotes the frame number, p and q are pixel positions and N_H and N_V denote the horizontal and

Figure 1: Mean pyramid generation and multi-resolution motion estimation

vertical frame size respectively.

Mean Absolute Difference (MAD) of pixel values is used as a measure to determine the motion vectors (MV) at each level L and is given by

$$MAD_{m,n}^{L,k}(i,j) = \frac{1}{\tilde{I}\tilde{J}} \sum_{i=\tilde{I}}^{i+\tilde{I}-1} \sum_{j=\tilde{J}}^{j+\tilde{J}-1} |x_L^k(i,j) - x_L^{k-1}(i+m,j+n)|$$

$$\tilde{I} = I/2^L, \tilde{J} = J/2^L, -s_L \leq m, n \leq s_L \quad (2)$$

$$MV_L^k(i,j) = \arg \min_{-s_L \leq m, n \leq s_L} MAD_{m,n}^{L,k}(i,j) \quad (3)$$

where m, n denote the search coordinates for the macroblock at the position (i, j) . s_L is the level dependent search range and I, J denote the macroblock height and width respectively. We see from Eq.(2) that as we move from a lower level to a higher level, the computations involved in determining the MAD for one position decreases by a factor of 4. In addition, for the same search area the search range reduces by a factor of two in each direction. These savings result at the expense of obtaining decimated motion vectors rather than the exact motion vectors. Usually, FSBM is performed at the highest level of the mean pyramid in order to detect random motion and obtain a low-cost estimate of the motion associated with the macroblock. This estimate is progressively refined at lower levels by searching within a small area around the motion vector obtained from the higher level. This process is shown in Fig.1 where 2 CMVs are propagated from every level to the next lower level. In equation form,

$$MV_L^k(i,j) = 2MV_{L+1}^k(i,j) + \delta MV_L^k(i,j) \quad (4)$$

where $\delta MV_L^k(i,j)$ denotes the refinement of the motion vector at level L and is dependent on the search range s_L .

2.2. Deviation pyramid

It has been verified [8] that passing a large number of CMVs to lower levels improves the quality of search results. But each CMV at a particular level translates into a fixed number of computations. Hence, we have a complexity v/s PSNR tradeoff that needs to be explored. This tradeoff can be made efficient if we decipher under what conditions the best match at a particular level is incorrect. One reason due to which search results deteriorate is that the mean pyramid generated image at a particular level is not a *good* representation of the actual image. The mean operation used in pyramid generation is essentially an approximated form of a two-dimensional low pass filter. Hence, the high frequency components in the reference macroblock are lost at higher level representations. It should be noted that the same loss occurs for the correct search position and hence the match obtained is usually correct. But in the presence of competing incorrect solutions, the best match will be an incorrect match. As a result, the process of determining the number of CMVs should be based on:

1. Presence of high frequency components in the reference macroblock.
2. Presence of competing solutions in the search area.

The estimation of the frequency characteristics should be done in a reliable and computationally efficient manner. We use the *deviation pyramid* in order to perform this estimation. The presence of competing solutions is sequence specific and hence needs to be learnt by the motion estimator for every sequence.

The deviation pyramid for the reference macroblock is defined as

$$D_L^k(p,q) = \left[\frac{1}{4} \sum_{r=-1}^0 \sum_{s=-1}^0 |x_{L-1}^k(2p+r, 2q+s) - x_L^k(p,q)| \right] \quad (5)$$

The deviation pyramid represents the error introduced by the process of averaging. In order to obtain a single quantity representative of the frequency content of the macroblock we sum up the deviation pyramid generated for the reference macroblock at each level. A reference macroblock with low frequencies sums up to a small value, whereas the presence of high frequencies results in a large aggregate. The Average Deviation Estimate (ADE) of the macroblock at position (i, j) is given by

$$ADE_L^k(i,j) = \frac{1}{\tilde{I}\tilde{J}} \sum_{r=0}^{\tilde{I}-1} \sum_{s=0}^{\tilde{J}-1} D_L^k(i+r, j+s) \quad (6)$$

where $\tilde{I} = I/2^L$ and $\tilde{J} = J/2^L$. The presence of incorrect competing solutions at higher search levels may result in the correct CMV having a MAD value greater than the minimum MAD value of that level. Though the rank of the correct solution among the CMVs varies from macroblock to macroblock, it is possible to estimate the band of values above the minimum solution within which the correct solution may be found. This quantity is termed the *distortion band* and is given by,

$$band^L(i,j) = MAD_{corr}^L(i,j) - MAD_{min}^L(i,j) \quad (7)$$

The distortion band corresponds to the presence of competing solutions at that ADE for the given sequence. We would like to predict the required distortion band based on the ADE of the macroblock. Using this predicted value $band_{pred}$, we can compute a threshold MAD value and all motion vectors whose MAD falls below this level can be passed as CMVs, i.e.

$$MAD_{cutoff}^L(i,j) = MAD_{min}^L(i,j) + band_{pred}^L(i,j) \quad (8)$$

$$\text{where } band_{pred}^L(i,j) = \mathcal{F}\{ADE_L^k(i,j)\} \quad (9)$$

where $\mathcal{F}$ is a function which relates the ADE to the required distortion band. In order to determine the generic form of $\mathcal{F}$ we plot in Fig.2 the ADE of the macroblock against the distortion band, for the first 5 frames of three QCIF format sequences - *Carphone*, *Foreman* and *Claire*. The macroblock size is 16×16 , the search range is ± 16 and the refinement range at levels 1 and 0 is ± 1 . The number of CMVs is fixed at 9 for all levels. Based on these plots we can make the following observations:

1. The distortion band is sequence specific and level specific. Hence $\mathcal{F}$ should also be sequence and level specific.
2. Across different levels and different sequences, we find that the distortion band at lower ADEs is very small which validates our point that the effect of competing incorrect solutions can be significant for reference macroblocks with high frequency components.
3. From the foreman sequence, it is clear that the relation between ADE and required distortion band can be non-linear.

Figure 2: Variation of distortion band with ADE

The dual requirements on the function $\mathcal{F}$ are a good fit to ensure low prediction error and low complexity to aid a hardware realization. A linear fit uses the least number of parameters but as observed, it could lead to large errors in prediction and hence is not a viable solution. In order to accurately capture the variation of the distortion band with ADE at a low cost, we use the technique of uniform quantization on the ADE axis. In this framework, determining the parameter set pertains to calculating the maximum value of the distortion band within two decision thresholds along the ADE axis. Hence from an implementation viewpoint, the process of determining the *content complexity* is reduced to estimating and assigning a distortion band value to each of the quantised average deviation estimates (QADE) at the levels 2 and 1. Hence Eq.(9) can be rewritten as,

$$band_{pred}^L(i, j) = band_{est}^L\{QADE_L^k(i, j)\} \quad (10)$$

Learning the values of $band_{est}$, which corresponds to training the motion estimator for a particular sequence is done over the first few frames of the sequence. Even though the technique of non-uniform quantization gives comparatively lower quantization errors, the complexity involved in iteratively determining the decision thresholds precludes its hardware implementation.

The schematic diagram of the proposed motion estimator is shown in Fig.3. The functioning of each module is described below. During the training of the motion estimator, the distortion band estimator is operational and the number of CMVs passed between adjacent levels is kept at the maximum value. The distortion band estimator performs the following two operations:

1. It matches the decimated value of the final motion vector with the CMVs passed between adjacent levels and hence determines the MAD of the correct solution at each level.
2. Based on the MAD of the correct solution and the best solution at every level, the distortion band is calculated as in Eq.(7)

From a hardware realization viewpoint the matching operations can be implemented through simple XOR gates. A single subtraction is required to determine the distortion band at each representation level. The distortion band predictor is look-up table based. The table stores the maximum value of the distortion band corresponding to each of the QADEs at levels 1 and 2 obtained during training.

Determining the QADE at levels 1 and 2 involves Eq.(5-6) followed by uniform quantisation. Eq.(5-6) applied at level L have the same cost as determining the MAD at a single search position at level $L - 1$. Hence the operations involved in making the multi-resolution technique adaptive have a low computational, power, and area cost. It should be noted that in the training mode, the motion estimator is operational albeit with high power consumption as maximum number of CMVs are passed between adjacent levels for every

Figure 3: Schematic diagram of the proposed motion estimator

macroblock. During normal operation the distortion band estimator is turned off and the algorithm attempts to maintain the quality of motion vectors similar to that obtained during training, with computations optimised for the sequence.

3. SIMULATIONS

As the proposed motion estimator is intended for mobile video encoders, we test its working with 7 standard sequences in the QCIF (176×144) format. The sequences used are: *Claire*, *Carphone*, *Foreman*, *Grandma*, *Mother* (Mother and Daughter), *MissA* (Miss America), *Salesman*. Simulations have been done on 300 frames of each sequence, except for the *MissA* sequence where 150 frames have been used. In addition, we have tested the algorithm on 210 frames of the sequence *Football* having frame size 352×240.

The macroblock size is fixed at 16×16 and the search range is ± 16 along both axes. In our simulations, the maximum number of CMVs passed is fixed at 9. We reuse search data[9] at all levels of the search pyramid. The search range at levels 0 and 1 is fixed at ± 1 along both axes. As a result, the search area due to 2 CMVs at levels 1 and 0 can overlap a maximum of 3 out of the 9 positions. It is simple to detect this occurrence and reduce the search area accordingly. In order to estimate the speedup factor, we have assumed that the addition operation involved in pyramid generation, contributes to half the cost of the basic operation involved in MAD calculation, which involves computing an absolute difference and performing an addition. In calculating the speedup factors we have also taken into consideration the reduction in the average computations per macroblock for the FSBM and n-step algorithm due to the macroblocks along the frame edges.

Simulation results are given in Tables 1-3. In Table 1 the figures in parenthesis denote the PSNR drop compared to FSBM. As seen in Table 1, the PSNR of n-step search drops for the sequences *Carphone*, *Foreman* and *Football* whereas the proposed algorithm maintains PSNR close to that obtained by FSBM. In Table 2 we see that the proposed algorithm scales computations depending on the complexity of the sequence. The close PSNR match between FSBM and the proposed algorithm and the range of computational scalings validates the utility of sequence specific training. Fig. 4 shows the variation of average CMVs per frame for levels 1 and 0 for two sequences.

In order to verify the effectiveness of the proposed algorithm, we have compared it against a fixed version of multiresolution motion estimation. The results are shown in Fig. 5. The numbers in parentheses denote the fixed number of CMVs passed to the levels 1 and 0. We see that for a given fixed CMV value the drop in PSNR is sequence dependent. Hence using a prefixed low value for the CMVs leads to a perceptible drop in PSNR and using a prefixed high value of CMVs leads to unnecessary computations. The adaptive algorithm scales computations and hence the speedup factor while ensuring that the PSNR drop compared to FSBM is maintained at a

Figure 4: Variation of average CMVs per frame

Figure 5: Comparison of adaptive and fixed multiresolution motion estimation low value.

It was observed that for sequences with plain backgrounds, a large number of search positions give approximately the same MAD and this leads to an increase in the CMVs. A solution to this problem of plain backgrounds was proposed in [10] which uses a static/non-static classifier operating at the macroblock level in the FSBM framework. The drop in PSNR is dependent on the accuracy of the classifier which makes a static/non-static decision based on the similarity between the higher order bits of the reference macroblock pixels and the exactly overlapping position of the search area. Two solutions in our framework for this phenomenon are as follows:

Solution A

When the QADE at both levels 1 and 2 is zero and the best MAD at level 2 is less than 0.4, we limit the CMVs passed to level 1 to two. When the QADE at level 1 is zero and the best MAD at level 1 is less than 0.6, we pass one CMV to level 0.

Solution B

This solution is similar to Solution A except that under the same conditions at level 1, we interpolate the best motion vector at level 1 to obtain the final MV position. This completely eliminates computations at level 0.

These solutions are based on the reasoning that if the current best matches are likely to give a high PSNR and the deviation values are low, then the final best solution is unlikely to be significantly different from the best solution tracked at the current level. From Table 3 we see that the speedup factor improves significantly for

Sequence	FSBM	n-step	Adaptive pyramid
Carphone	32.10	31.83 (0.27)	32.08 (0.02)
Foreman	31.87	31.36 (0.51)	31.85 (0.02)
Claire	42.95	42.94 (0.01)	42.94 (0.01)
MissA	41.34	41.27 (0.07)	41.28(0.06)
Mother	40.62	40.58 (0.04)	40.57 (0.05)
Grandma	42.57	42.57 (0.00)	42.57 (0.00)
Salesman	40.03	39.98 (0.05)	39.99 (0.04)
Football	26.00	23.98 (2.02)	25.94 (0.06)

Table 1: Average PSNR comparisons

Sequence	Average CMVs		Speed up w.r.t	
	Level 1	Level 0	FSBM	n-step
Carphone	2.09	3.64	25.33	1/1.23
Foreman	5.28	4.85	15.50	1/2.01
Claire	1.82	1.48	43.54	1.40
MissA	3.62	4.88	19.92	1/1.56
Mother	1.65	1.92	37.74	1.21
Grandma	2.43	2.18	32.79	1.05
Salesman	1.13	1.16	50.00	1.60
Football	5.37	4.76	20.10	1/1.60

Table 2: Computational complexity comparisons using average CMVs and speedup factors

Sequence	Solution A			Solution B		
	Speedup w.r.t FSBM	n-step	PSNR drop	Speedup w.r.t FSBM	n-step	PSNR drop
Claire	49.98	1.60	0.03	74.04	2.38	0.05
MissA	40.41	1.30	0.08	49.21	1.58	0.10
Mother	50.92	1.63	0.05	60.00	1.93	0.06
Grandma	51.72	1.66	0.02	62.31	2.00	0.03

Table 3: Speedup and PSNR drop with background elimination all sequences with plain backgrounds, with a small drop in PSNR compared to FSBM.

4. CONCLUSIONS

We proposed a content adaptive motion estimation algorithm that scales computations at the macroblock level. Simulations have verified that PSNR is maintained close to that obtained using FSBM while computations are scaled depending on the content complexity. Modifications have been proposed to deal with plain backgrounds and have been verified. Simple operations have been used throughout the algorithm in order to maintain applicability for hardware implementation.

5. REFERENCES

- [1] MPEG-4 Video Group, "Overview of the MPEG-4 Standard," ISO/IEC JTC1/SC29/WG11 N2323, Dublin, Ireland, July 1998.
- [2] M. Goel and N. R. Shanbhag, "Dynamic Algorithm Transforms: A Systematic Approach to Low-power Reconfigurable Signal Processing," *IEEE Trans. on VLSI Systems*, Vol.7, No.4, pp.463-476, Dec. 1999.
- [3] ITU-T Recommendation H.263, "Video coding for low bit rate communication".
- [4] J. Kneip *et al.*, "The MPEG-4 Video Coding Standard- A VLSI point of view," *Proc. of IEEE Workshop on Signal Processing Systems*, Cambridge, MA, pp.43-52, 1998.
- [5] N. Farber, B. Girod and J. Villasenor, "Extensions of ITU-T Recommendation H.324 for Error-Resilient Video Transmission", *IEEE Comm. Magazine*, pp.120-128, June 1998.
- [6] S. Park and W. Burleson, "Reconfiguration for power savings in real-time motion estimation," *IEEE ICASSP'98*, Vol.5, pp.3037-3040, 1998.
- [7] J. Minocha and N. R. Shanbhag, "A Low-power Data Adaptive Motion Estimation Algorithm," *Proc. of IEEE Workshop on Multimedia Signal Processing*, Copenhagen, Sept. 1999.
- [8] K. M. Nam *et al.*, "A fast hierarchical motion vector estimation algorithm using mean pyramid," *IEEE Trans on Circ. and Sys. for Video Tech.*, Vol.5, No.4, pp.415-428, Aug. 1995.
- [9] Y. H. Yeh and C. Y. Lee, "Cost-effective VLSI Architectures and Buffer Size Optimization for FSBM Algorithms," *IEEE Trans. on VLSI Systems*, Vol.7, No.3, pp.345-358, Sept. 1999.
- [10] V. Moshnyaga and K. Nakasima, "Reduction of background computations in adaptive block-matching motion estimation," *IEEE ISCAS 2000*, Vol.1, pp.667-670, May 2000.