FAST VIDEO MOTION ESTIMATION ALGORITHM FOR MOBILE DEVICES

Ho-Jae Lee, Panos Nasiopoulos, Victor C.M. Leung

Department of Electrical and Computer Engineering
University of British Columbia, Canada
hol@ece.ubc.ca, panos@ece.ubc.ca, vleung@ece.ubc.ca

ABSTRACT

The power consumption of video enabled mobile devices is an ongoing challenge. The computational burden caused by the motion estimation process involved in video compression is the main cause for this consumption. We developed a new algorithm which drastically reduces the computation costs of the motion estimation process over the existing techniques. Our algorithm uses only 0.5% of the computational complexity of the full search method, making it the fastest presently available motion compensation method.

1. INTRODUCTION

In the past few years, significant changes in video compression and wireless data communications have stirred an evolution in portable multimedia applications. However, the presence of video in mobile devices has become one of the most difficult challenges for real time high quality applications.

Mobile communication standards such as 3GPP (3rd Generation Partnership Project), are specifically designed to support data communication between mobile and wired-networks or mobile networks only. Despite the existence of standards that support high speed data flow and real time video rates, the computational burden caused by the video compression process remains a technological challenge. Conventional video encoders spend up to 60% of the computational time on motion estimation [1]. Power consumption of video enabled mobile devices can be reduced by lowering the power consumption of the motion compensation process. In recent studies, Spatial and Temporal Correlation of motion vectors is exploited in order to lower the computational costs over the Full Search (FS) technique which scans the entire searching area [2]. Other motion estimation algorithms include the Prediction Model Search using Four-Step Search (PM4SS) [3], the Adaptive Predicted Direction Search algorithm (APDSA) [4], and the Hybrid Search model [5]. However, none of them has resulted in speeds that are practical for low power, real-time realization of mobile devices. Hence, there is an immense interest in developing fast and efficient motion compensation techniques which will enable this new market.

We have developed a novel new method which drastically improves the speed of the motion estimation process over the existing techniques. Our proposed method makes use of spatial-temporal correlation among neighboring macroblocks, macroblock subsets, and a special search scanning pattern to reduce the search time for finding motion vectors.

In section 2, we give an overview of existing fast motion estimation algorithms. In section 3, we present our new method and in section 4 we evaluate its performance. The conclusions are presented in section 5.

2. OVERVIEW OF CURRENT MOTION ESTIMATION ALGORITHMS

The objective of the motion compensation process is to determine the amount of motion on a block by block basis which minimizes the difference between consecutive frames. The Full Search (FS) algorithm, which is one of the most well known methods, is based on an exhaustive testing of all the candidate blocks within the search window, giving the minimum block distortion position that corresponds to the best matching block. This method, however, involves a large number of computations.

In order to reduce complexity and processing time, several other search algorithms have been proposed, including the New Three Step Search (NTS) [6] and the Novel Four Step (NFS) [7]. For a search area of ± 7 pixels, these two fast motion estimation algorithms reduce the motion estimation speed to only 8% of that need for FS.

One of the most common fast motion compensation algorithms is the Diamond Search (DS) approach. This method uses a diamond search pattern. Among the five surround checking points, the one that gives the minimum distortion is the motion vector of the best matching block. When the maximum displacement is $\pm$ 7 pixels both horizontally and vertically, it improves the computational speed by a factor of 15 when compared to FS [8].

Several advanced versions of DS have been proposed. One of them, the Rood based Motion Estimation Algorithm, improves the searching speed over that of the original DS algorithm by up to 40% while maintaining similar distortion error [9].

The Hybrid Search uses a compact plus-shaped and an X-shaped search and a diamond search to reduce the motion estimation computations [5]. The Adaptive Motion algorithm dynamically alters computations, depending on the complexity of the contents of each macroblock [10]. A multi-resolution mean pyramid is used to estimate the macroblock characteristics, resulting in substantial reduction

0-7803-9332-5/05/$20.00 ©2005 IEEE

in search range and block size. The Spatial-Temporal Motion Estimation Algorithm selects an adaptive search window for each macroblock, using one temporal and three spatial macroblocks [11].

3. PROPOSED NEW MOTION ESTIMATION ALGORITHM

We introduce a new motion estimation algorithm. In our first attempt we use only spatial correlation. The performance of our algorithm improves significantly when we combine spatial with temporal correlation information. The two steps are described in the following subsections.

3.1 Motion Estimation Based on Spatial Correlation over DS

During the first step, the difference between the current and previous frame is calculated by direct subtraction. The resulting difference frames are subdivided into 16x16 macroblocks and the sum of the absolute difference (SAD) for each macroblock is calculated. These SAD values are sorted in descending order and the average and standard deviation of each macroblock are derived.

Current methods search for the best matching macroblock by scanning the search window from left to right. In our case, the scanning process follows the sorted SAD values. Smaller SAD values indicate that the corresponding motion vector will also be smaller. Therefore, macroblocks that correspond to larger SADs have higher priority and their motion vectors are computed first.

Although our objective is to reduce the amount of calculations involved in the motion estimation process, we also need to ensure that we do not compromise the image quality. In order to achieve both objectives, we divide the macroblocks of a frame into three categories as shown in table 1. A different method is used for calculating the motion vectors for each category. The block search process uses only a subset of the total 256 pixels in the macroblock. Figure 1 shows eight different ways of extracting the 16 pixels from one 8x8 macroblock (shaded blocks). This process can be expanded to 16x16 macroblocks. For category 1, performance evaluations have shown that a combination of any two subsets provides enough information for accurate calculation of motion vectors.

In order to maintain the image quality, the search algorithm we employ for macroblocks whose corresponding SDA values are higher than the average does not take into consideration any spatial or spatial/temporal correlation between motion vectors.

The Diamond Search pattern is used for estimating the motion vectors in this category as well as the other two.

If the SAD value is between the sum of the average and standard deviation and the average value (i.e., category 2), this is an indication that there is little or no disparity between two consecutive frames. For this category, only one 16 pixel subset is needed for the calculation of the motion vector. This approach reduces the estimation complexity by up to 75%.

Finally, if the SAD value is smaller than the average (category 3), we take the spatial correlation into consideration. Using values of already known motion vectors (shaded areas in figure 2) around the target macroblock, we calculate the motion vector as the average of the surrounding motion vectors. This vector becomes the initial search point for the macroblock. This is different from conventional motion estimation algorithms which use the macroblock center as the initial point for the search.

For the second and third category, if the motion vectors of several successive macroblocks are (0,0), we can assume that the rest of the macroblocks are stationary or quasi-stationary and their motion vectors are also (0,0).

3.2 Motion Estimation Based on Spatial and Temporal Correlation over DS

In this step, we combine temporal and spatial correlation to improve the speed of the motion estimation process. As in the approach described above, the difference between the current and previous frame is calculated by direct subtraction and the sum of the absolute difference (SAD) for each 16x16 macroblock is calculated. However, in addition to that, we also take advantage of the temporal correlation between two consecutive frames. The motion vectors of the previous frame are used to determine the predicted motion vector of the corresponding macroblocks in the current frame. Then the difference between macroblocks from the current frame and the corresponding search area is calculated, taking into consideration the temporal correlation from the previous frame. The new difference frame is subdivided into 16x16 macroblocks and the sum of the absolute difference (SAD) for each macroblock is calculated. We compare the two SAD values that we obtained for each macroblock and since the smaller one will yield a better approximation for the motion vector, we generate a new SAD array using the smallest value.

Table 1. List of methods used for each category.

	Sorted SAD values	**Used Method for calculating motion vectors**
Category 1	Down to $m + \sigma$	Use any combination of two subsets shown in Fig. 1.
Category 2	Between $m + \sigma$ and m	Use one of the subsets of Fig. 1. If motion vectors of several successive MBs are (0,0), the rest of the MVs are also set to (0,0).
Category 3	Less than m	Use one of the subsets of Fig. 1 and spatial correlation. If MVs of several successive MBs are (0,0), the rest of the motion vectors are also set to (0,0).

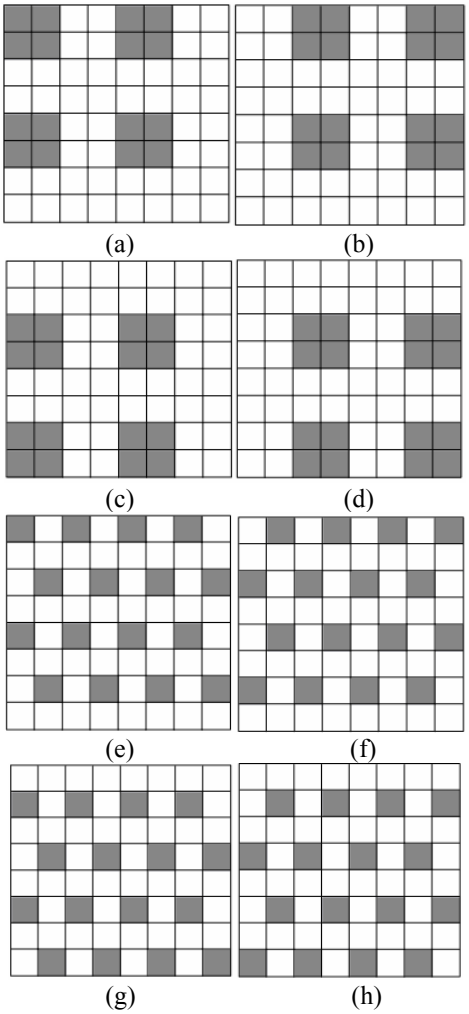

Figure 1. Eight types of pixel subsets for one macroblock

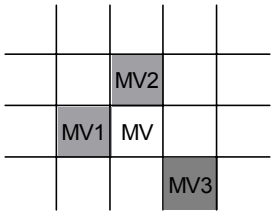

Figure 2. Target vector calculated as the average of already known motion vectors (shaded areas) around it.

The three different categories are defined the same way as in the spatial correlation case. In addition, the process for finding the motion vectors is similar to the one described in the previous subsection, except of the final step. In other words, for the first two categories, the motion vector is calculated in a similar manner.

However, for category 3, if several successive motion vectors are (0,0), we can assume that the rest of macroblocks are stationary or quasi-stationary and their motion vectors are also set to (0,0). For the same category, if several successive motion vectors are the same as the corresponding vectors from the previous frame, the rest of the macroblocks vectors are set to have the values from the previous frame as well.

4. PERFORMANCE EVALUATION

In this section, we evaluate our proposed method and we compare it with existing motion estimation algorithms. In this simulation, we used the Telenor H.263 encoder 1.7 with variable bitrate, quantization parameter of 10, 0 frame skip, and 30 fps. The macroblock size is fixed to 16x16 pixels and the maximum value of displacement for motion vectors is ± 7 pixels. Our technique uses the new subset of pixels over the Diamond Search pattern. Variable bit rates were used to ensure constant picture quality. Six video sequences, "Akiyo", "Claire", "Grandma", "Miss_am", "News" and Salesman", were used for our evaluations.

We compare our technique with the existing motion compensation methods in terms of SNR, average bit rates and computational complexity. Table 2 shows the SNR values and table 3 shows the average bit rates for all the tested methods and all six video streams. We observe that the picture quality is kept the same for all these tests.

Table 4 shows the total number of processed points used by every method to determine the motion vectors. The last two columns show the results obtained by our method when only spatial correlation is used and when spatial and temporal correlations are combined. It is clear that the performance of our method improves significantly when spatial and temporal correlations are used. When compared with all the other methods, we observe that our approach drastically improves the motion estimation search speed. In particular, our algorithm has only 0.5% of the computational complexity of the full search method and is almost 19 times faster than NFS, the best among the conventional methods.

5. CONCLUSIONS

We presented a new motion estimation technique, which makes use of spatial and temporal correlation among neighboring macroblocks, macroblock subsets and a special search scanning pattern to reduce the computational time for finding the motion vectors.

Performance evaluations showed that, for the same picture quality, our algorithm uses only 0.5% of the computational complexity of the full search method and is almost 19 times faster than NFS, the best presently available method. We observe that our method drastically reduces the computation costs of the motion estimation process over the existing techniques and may be the answer to low power consumption problem facing mobile devices.

6. REFERENCES

[1] Sun-Hyoung Han, Sung-Woo Kwon, Tae-Young Lee, Moon-Key Lee; "Low power motion estimation algorithm based on temporal correlation and its architecture", *Signal Processing and its Applications, Sixth International Symposium,. 2001*, Volume: 2, pp 647 -650, Aug. 2001

[2] Sun-Hyoung Han; Sung-Woo Kwon; Tae-Young Lee; Moon-Key Lee, "Low power motion estimation algorithm based on temporal correlation and its architecture", *Sixth International Symposium on Signal Processing and its Applications 2001*, Volume: 2, pp 647 -650, 13-16 Aug. 2001

[3] Jie-Bin Xu, Lai-Man Po, Chok-Kwan Cheung, "A new prediction model search algorithm for fast block motion estimation", *International Conference on Image Processing,* Volume: 3, pp 610 -613 vol.3, 26-29 Oct. 1997

[4] Jae-Yeal Nam, Jae-Soo Seo, Jin-Suk Kwak, Myoung-Ho Lee, Yeong Ho Ha, "The adaptive predicted direction search algorithm (APDSA) New fast-search algorithm for block matching motion estimation using temporal and spatial correlation of motion vector", *IEEE Transactions on Consumer Electronics*, Volume: 46 Issue: 4, pp 934 -942, Nov. 2000

[5] R. Li, B. Zeng and M.L. Liou, "A new Three-Step Search Algorithm for Block Motion Estimaton", IEEE Trans. On Circuits and Systems for Video Techn., vol. 4, no. 4, Aug. 1994, pp. 438-442

[6] L.M. Po and W.C. Ma, "A novel four-step search algorithm for fast block motion estimation", IEEE Trans. Cirs. Syst. And Video Technol., vol. 6, June 1996, pp.313-317.

[7] Shan Zhu; Kai-Kuang Ma, "A new diamond search algorithm for fast block matching motion estimation**",** Information, Communications and Signal Processing, 1997. ICICS., Proceedings of 1997 International Conference on , Volume: 1, pp 292 -296, 9-12 Sept. 1997

[8] Chun-Ho Cheung; Lai-Man Po, "A novel cross-diamond search algorithm for fast block motion estimation", *IEEE Transactions on Circuits and Systems for Video Technology,* Volume: 12 Issue: 12, pp 1168 -1177, Dec. 2002

[9] Su-Bong Hong; Hyoseo Lee; Geun-Young Chun; Hyunki Baik; Par, M., "FCBHS: a fast center-biased hybrid search algorithm for fast block motion estimation", *International Conference on Information Technology: Coding and Computing, 2002. Proceedings.* pp 254 -259, April 8-10, 2002

[10] Ahmed, A.; Nandy, S.K.; Sathya, P., "Content adaptive motion estimation for mobile video encoders", *ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems, 2001.*, Volume: 2, pp 237 -240 vol. 2, 6-9 May 2001

[11] Jong-Hyun Lim; Hae-Wook Choi, "Adaptive motion estimation algorithm using spatial and temporal correlation", *PACRIM. 2001 IEEE Pacific Rim CCSP, 2001.*, Volume: 2, pp 473 -476, 26-28 Aug. 2001.

Table 2. Average SNR of Y component of 100 frames

Seq.	FS (dB)	DS (dB)	NTSS (dB)	NFS (dB)	Proposed (spatial only) (dB)	Proposed (dB)
akiyo	34.34	34.28	34.26	34.28	34.18	34.19
claire	36.17	36.06	36.07	36.06	35.92	35.92
grandma	33.27	33.25	33.25	33.25	33.16	33.15
miss_am	37.14	36.90	36.76	36.87	36.69	36.43
news	32.46	32.43	32.43	32.43	32.41	32.37
salesman	31.71	31.71	31.71	31.71	31.68	31.62

Table 3. Average Bit Rate

Seq.	FS (kbps)	DS (kbps)	NTSS (kbps)	NFS (kbps)	Proposed (spatial only) - kbps	Proposed (kbps)
akiyo	27.48	27.78	28.50	28.05	28.80	28.94
claire	26.91	26.99	27.70	27.16	27.39	27.89
grandma	25.04	25.13	25.42	25.20	26.49	27.87
miss_am	27.56	30.54	34.08	31.37	32.04	39.45
news	62.91	63.88	65.03	64.55	68.70	68.80
salesman	39.98	40.02	40.83	40.12	40.97	42.62

Table 4. Total number of processed points

Seq.	FS	DS	NTSS	NFS	Proposed (spatial only)	Proposed
akiyo	1817200(100%)	113267(6.2%)	194612(10.7%)	109526(6.0%)	12530(0.7%)	10513(0.6%)
claire	1817200(100%)	113719(6.3%)	192929(10.6%)	109814(6.0%)	11600(0.6%)	10719(0.6%)
grandma	1817200(100%)	113712(6.3%)	194844(10.7%)	109712(6.0%)	11305(0.6%)	9828(0.5%)
miss_am	1817200(100%)	116481(6.4%)	197052(10.8%)	110565(6.1%)	12771(0.7%)	8534(0.5%)
news	1817200(100%)	114192(6.3%)	195368(10.8%)	109787(6.0%)	9528(0.5%)	8204(0.5%)
salesman	1817200(100%)	113763(6.3%)	195054(10.7%)	109748(6.0%)	9149(0.5%)	8145(0.4%)