16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

A COMPLEXITY SCALABLE H.264/AVC ENCODER FOR MOBILE TERMINALS

Amit Kumar[12]*, Daniele Alfonso*[2]*, Luca Pezzoni*[2] *and Gabriella Olmo*[1]

[1]Facolta' di Ingegneria dell'Informazione, Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129, Torino, Italy
email: amit-agr.kumar@st.com, gabriella.olmo@polito.it

[2]Advanced System Technology Labs., STMicroelectronics
Centro Direzionale Colleoni, 20013, Agrate Brianza, Italy
email: daniele.alfonso@st.com, luca.pezzoni@st.com

ABSTRACT

Video encoding is one of the most appealing features of modern mobile terminals, but also one of the most demanding for what concerns power consumption. We propose an H.264/AVC encoder able to adaptively self-adjust the accuracy of the motion-compensated prediction on the basis of the available power resources to achieve a progressive reduction of the computational requirements with graceful degradation of the rate-distortion performance.

1. INTRODUCTION

Nowadays we are witnessing an increasing demand of multimedia capabilities in modern mobile terminals, for which the primary requirement for next generation products is real-time H.264/AVC encoding and decoding [1][2] for a wide range of image formats, from QCIF/CIF for conversational applications to HDTV for personal video recording. As a consequence, the limited power resources of the mobile terminals are particularly stressed by the high computational requirements of video processing, and the product designers must find a smart way to efficiently exploit the available power without compromising the battery life.

Power-aware multimedia [3] is a relatively new design concept that considers not only the usual low-power goal, but also the possibility to dynamically adapt the computational requirements of multimedia processing on the basis of available power resources, e.g. by reducing the number of calculation performed during real-time video encoding depending on the current battery status.

In general, the performance of a video encoder can be defined by three factors –the bit-rate, the distortion and the complexity– which must be considered and minimized together, in order to be able to scale the complexity of the computation by allowing a graceful degradation of the compression performance, avoiding sudden quality drops that might be annoying to the end user.

It is already known that Rate-Distortion Optimization (RDO) is by itself a fairly complex minimization problem, which can be elegantly solved by a Lagrangian formulation [4]. This approach can provide excellent compression performance, but it also requires a consistent amount of computation that limits its practical applicability in low-power devices [5]. The Lagrangian formulation becomes even more computationally expensive if a third variable –the complexity– is introduced in the formulation of the problem [6].

For that reason, we have followed a more practical approach, by developing a complexity scalability method for real-time H.264/AVC encoding based on statistical considerations, which allows achieving incremental complexity reduction with graceful degradation of the rate-distortion performance, introducing negligible additional computation in the video coding system.

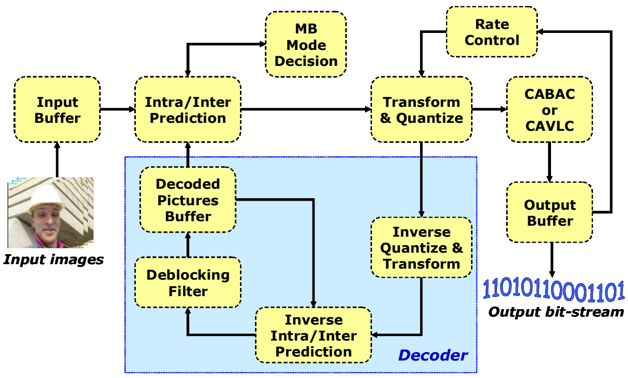

Figure 1: H.264/AVC encoder simplified block scheme.

2. H.264/AVC ENCODER COMPLEXITY

Along with the fact that H.264/AVC provides same or better visual quality at almost half of the bit-rate with respect to its ancestors in the ISO/IEC MPEG and ITU-T H.26x families, it is also very complex from computational point of view.

In the H.264/AVC standard, video sequences are split into pictures, either frames or fields, which are further split into macroblocks (MB). H.264/AVC allows macroblock partitions of 16x16, 16x8, 8x16 and 8x8 pixels, and 8x8 partitions can be further sub-partitioned to 8x4, 4x8 and 4x4 pixels blocks.

It is known that most part of the computational complexity of the encoding process is due to Motion Estimation (ME), which is needed for motion-compensated prediction.

Hence, as a first step, we have considered to reduce the number of MB coding modes tested during Inter encoding process.

In general, arbitrary removal of smaller blocks sizes will provide reduction in complexity, because of the reduced number of motion vector tested, but it might give unpredictable behaviour in terms of increased bit-rate and quality loss for different video sequences and different quantization steps. In fact, MB mode selection during encoding process depends on video contents (texture and motion) and on several coding parameters like quantization step, temporal decomposition structure, and so on.

In order to select the optimal macroblock mode in rate-distortion (R-D) sense, Lagrangian minimization has been successfully applied to mode selection problem in [7]. However, computation for R-D costs for all modes becomes huge because a large number of candidate modes are provided in H.264/AVC. Hence, we have considered the usage statistics of macroblock modes in order to remove the least used, hence having the least impact on bit-rate and visual quality as compared to removing any arbitrary coding mode, under the assumption that if a mode is least used, it is a direct consequence of its low importance in terms of its impact on compression efficiency.

For all the simulations we have used a proprietary H.264/AVC encoder including a fast motion estimation method called “SlimH264”, which is another part we have considered for complexity reduction.

2.1 SlimH264 motion estimation

SlimH264 supports all the MB and sub-MB partition modes of the H.264/AVC standard for any number of reference frames. It is a hierarchical-recursive block matching algorithm, working in two distinct motion search phases, called Coarse Search and Fine Search.

SLimH264 is hierarchical because Coarse Search results are used to initialize the Fine Search and it is recursive since it uses the motion vectors available from the previous blocks as candidate vectors to find the best temporal prediction of the current block, thus exploiting the temporal as well as spatial correlation between consecutive pictures.

The SlimH264 motion estimation algorithm achieves performance very close to the Full-Search block matching approach, using about 3% of the computation for a typical [-64,+64] pixels search range [8].

2.1.1 Coarse Search

In this step the algorithm finds a motion vector for each 16x16 macroblock by searching the most similar predictor in the picture which precedes in display order the current picture under estimation, evaluating the following candidates:

- Three motion vectors belonging to three spatially adjacent macroblocks (with respect to the current one) are being tested, called spatial predictors
- Three motion vectors belonging to three temporally adjacent macro block (with respect to the current one) are being tested, called temporal predictors.
- A fixed number of small motion vectors added to the best of above mentioned spatial/temporal vectors, called updates.

2.1.2 Fine Search

During this phase, the coarse motion vectors are refined to get the best motion vector for each block type, reference frame and direction, using the same predictors/updates strategy, where the temporal predictors are Coarse Search results already computed.

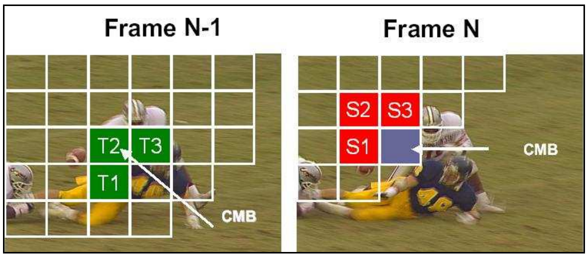

Figure 2: Spatial (S1, S2, S3) and temporal (T1, T2, T3) motion vector candidates in Coarse Search step. CMB is the current macroblock under estimation.

3. USAGE STATISTICS BASED MB AND MV REMOVAL

During encoding process, quantization acts like a low pass filter, so that higher quantization parameter (QP) leads to mostly use bigger MB partitions, because less spatial details have to be encoded, whereas lower QP leads to mostly use smaller MB partitions. These factors need to be considered before removing any MB mode during the encoding process, otherwise the penalty will be either in terms of increased bit-rate or poor visual quality or both.

We propose a method considering the usage statistics of MB coding modes before removing it. To obtain further complexity reduction, it also removes some of the motion vectors from testing on the basis of their usage statistics and furthermore it reduces some of the steps of the motion estimation algorithm.

3.1 MB modes removal

As removal of an arbitrary MB mode during encoding process might give unexpected results in terms of bit rate and/or visual quality, an approach based on usage statistics has been followed for removal of MB modes tested during encoding process.

As a first step, we profiled the H.264/AVC encoder, finding that complexity contribution of each MB mode tested is around 10-12% of total encoding time.

Following the above observation we developed an algorithm that collects the statistics of the MB modes usage for the first Group Of Pictures (GOP) of the sequence being coded, and then removes the least used MB modes starting from the second GOP to have a significant complexity reduction along with minimum impact on Rate-Distortion performance.

Figures 3-4 show that the removal of least used MB mode(s) (first one least used and so on) provides a reduction in complexity around ~10% of total encoding time for each mode removed with little effect on bit-rate and quality.

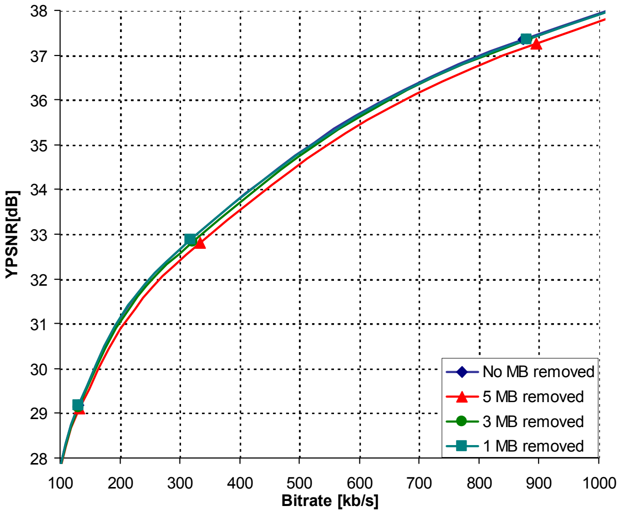

Fig.3. Rate Distortion diagram for "City" sequence.

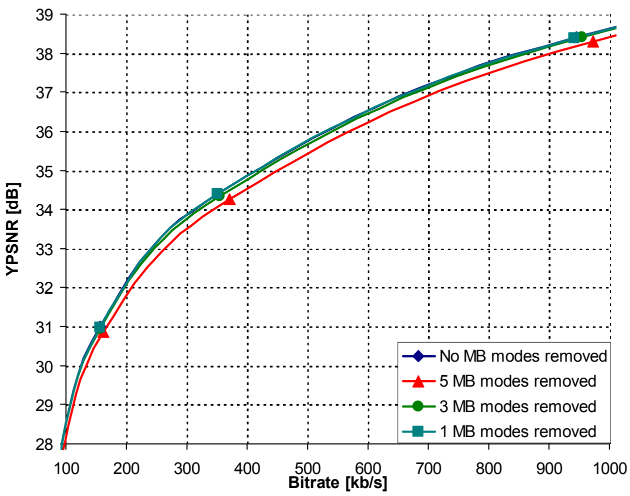

Fig.4. Rate Distortion diagram for "Foreman" sequence.

3.2 Motion vectors removal

During the motion estimation process SlimH264 tests a fixed number of motion vectors for each sub-partition of each MB mode under consideration. From this methodology, the following procedure has been applied to reduce the number of motion vectors tested during encoding process.

Fine Search is performed for 8x8 MB mode and the final MV is then stored and used as predictor for corresponding 8x4, 4x8 and 4x4 sub-partitions, together with the testing of the predicted motion vector defined by the H.264/AVC standard for differential MV encoding in the bit-stream. In this way, we test only 1 predictor instead of 6 for each sub-MB mode during Fine Search.

Using final MV of 8x8 MB for sub-partitions 8x4, 4x8 and 4x4, we are able to save around 70% of MV predictors testing during Fine Search.

Also statistical approach has been used to reduce the number of motion vectors tested. As previously described, SlimH264 algorithm uses temporal, spatial and updates vectors as candidates for optimal motion vector determination. Statistical usage of these three classes of vectors can be collected for the first GOP and then the least used vectors can be removed starting from the subsequent GOP.

4. COMPLEXITY SCALABILITY

By combining the two statistical-based methods for computational complexity reduction discussed until now, three stages of complexity reduction have been chosen in order to be selectable depending upon the battery life or computational power limits. They can be called as High, Medium and Low Complexity stages. We defined the three stages by no hard and fast rule, but considering the following facts:

- There should be significant difference between any two stages, so that switching to a lower level of complexity should provide significant extra life to battery or lower the computational power by significant amount.
- In lowest complexity level there should be significant use of motion estimation during encoding process because complete elimination of motion-compensated prediction (i.e. Intra only coding) will lead to excessive coding performance drop.

Following sections will describe the techniques used to achieve the High, Medium and Low levels of complexity.

4.1 High complexity level (20% reduction)

Following the methodology of SlimH264, sub-partitions below 8x8 pixels can take advantage of computations already done for corresponding 8x8 MB mode. Coarse and Fine Searches are then performed in the usual way (as described in sections 2.1.1-2) for MB modes down to 8x8, and then final motion vector of MB 8x8 is used as the only predictor for corresponding 8x4, 4x8 and 4x4 blocks, along with the predicted motion vector as specified by the H.264/AVC standard. In this way, Fine Search for sub-MB modes can be almost completely avoided.

This level of complexity reduction only influences the motion estimation algorithm, without removing any MB mode testing during encoding process.

4.2 Medium complexity level (35% reduction)

Next level of complexity reduction has been chosen as 35%, which is a significant difference from High level complexity reduction which is 20%. To achieve this level some macroblocks and motion vectors during motion estimation process have been removed based on statistics usage. As we have seen, removal of any MB mode during encoding process provides a complexity reduction of ~10-12% with a negligible effect on bit rate and visual quality (section 3.1). For first GOP, usage statistics of each macro block is collected. Also usage statistics of three different motion vector predictors classes (spatial, temporal and update) are collected separately. Based on usage statistics, three least used MB modes, one least used spatial predictor and one least used subset of update vectors have been removed for testing from successive GOP's during encoding process, providing a total complexity reduction of ~35% over total encoding process.

4.3 Low complexity level (55% reduction)

This level exploits usage statistics based MB modes removal during encoding process. Usage statistics of MB modes have been collected for first GOP and based on them 5 least used MB modes have been removed from the encoding process of successive GOP's. This method provides a complexity reduction of ~55% over total encoding time. Following sections will describe the experimental results obtained for complexity scalable H.264/AVC encoder.

5. EXPERIMENTAL RESULTS

The proposed complexity scalability framework has been evaluated in terms of R-D performance for some CIF and SIF sequences with a variety of motion and texture content to validate the robustness of the approach. Sequences Foreman, Akiyo, Demoiselle, Sign-Irene, Silent, Table, News with 352x288 pixels and Stefan with 352x240 pixels with frame rate of 30 Hz have been chosen for evaluation. Complexity is estimated by measuring the encoder total execution time using GNU "time" command.

Following parameters are common to all the simulations:

- Linux system with kernel 2.6.20.16 running on Intel Pentium 4 CPU at 2.4 GHz.
- Encoder configured in Baseline Profile (i.e. no CABAC and no B pictures).
- GOP type is IPPP with Intra period 30.
- Hadamard transform is used.
- Search range is ±64 pixels.
- No Lagrangian RDO algorithm.
- Quantization step QP_I ={ 13, 20, 27, 34, 41 } for I pictures and $QP_P = QP_I$+1 for P pictures.

The performances of the proposed method are very good in terms of bit-rate and visual quality, as shown in figures 5, 6 and 7 for sequences Akiyo, Foreman and Stefan: from the Rate-Distortion curves it can be seen that there is moderate difference in terms of quality loss and extra bit rate.

Extra bit-rate and YPSNR loss, averaged over all the above mentioned CIF sequences are as follows:

- With 20% reduction in complexity, average extra bit rate is ~0.08% with an average YPSNR loss of ~0.08dB.
- With 35% complexity reduction, average extra bit rate is ~1% and average YPSNR loss is 0.06dB.
- With 55% complexity reduction, average extra bit rate is ~4% and average YPSNR loss is 0.12dB.

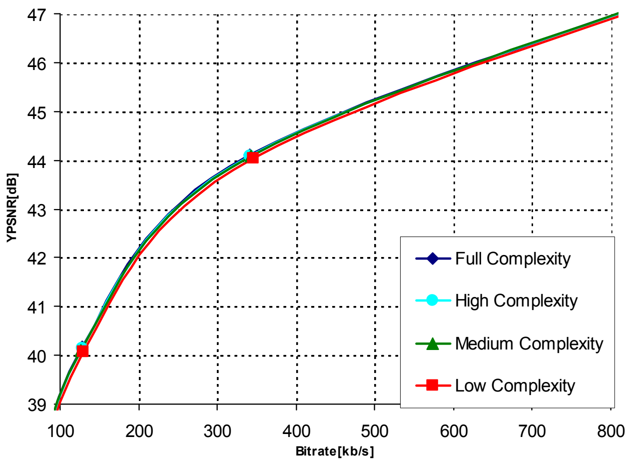

Fig.5. Rate-Distortion diagram for "Akiyo" sequence.

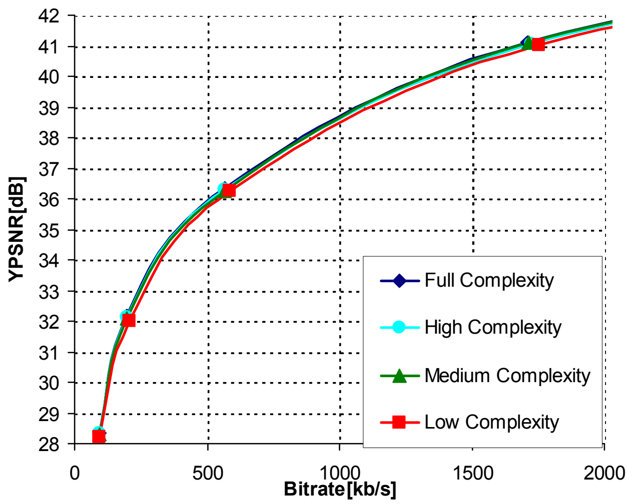

Fig.6. Rate-Distortion diagram for "Foreman" sequence.

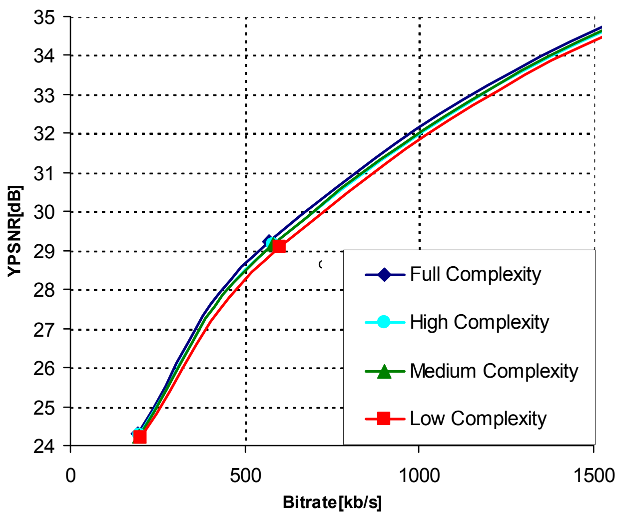

Fig.7. Rate-Distortion diagram for "Stefan" sequence.

Performance for the other sequences have been summarised in the Tables 1-3, showing average extra bit-rate and quality loss for the three different levels of complexity reduction. Extra bit rate and YPSNR loss for all the complexity levels mentioned are compared with the one with full complexity.

Sequence	Extra Bit-rate	YPSNR loss [dB]	Complexity Reduction
Stefan	0,95%	0,052	18,94%
Foreman	0,34%	0,046	19,83%
Akiyo	-0,44%	0,02	21,10%
Demoiselle	-0,23%	0,046	20,17%
Sign-Irene	-0,88%	0,052	20,62%
Silent	-0,23%	0,022	19,67%
Table	0,94%	0,04	19,70%
News	0,00%	0,036	20,06%

Table 1. Performance with 20% complexity reduction.

Sequence	Extra Bit-rate	YPSNR loss [dB]	Complexity Reduction
Stefan	0,99%	0,032	36,00%
Foreman	0,43%	0,052	37,85%
Akiyo	-0,75%	0,036	39,38%
Demoiselle	0,84%	0,03	37,68%
Sign-Irene	-0,05%	0,046	40,02%
Silent	0,00%	0,022	38,17%
Table	0,85%	0,036	37,34%
News	-0,01%	0,048	38,76%

Table 2. Performance with 35% complexity reduction.

Sequence	Extra Bit-rate	YPSNR loss [dB]	Complexity Reduction
Stefan	3,70%	0,07	52,03%
Foreman	3,32%	0,094	53,09%
Akiyo	-0,23%	0,092	53,93%
Demoiselle	2,19%	0,072	54,22%
Sign-Irene	-0,19%	0,116	56,24%
Silent	0,73%	0,058	55,20%
Table	3,18%	0,076	53,12%
News	0,97%	0,106	53,56%

Table 3. Performance with 55% complexity reduction.

6. CONCLUSIONS

We propose a complexity scalable algorithm for a proprietary H.264/AVC encoder with fast motion estimation, tested with CIF and SIF sequences. Simulation results shows that three different levels of complexity - High, Medium and Low - can be achieved with a little loss in coding efficiency. The proposed method requires very low implementation complexity and it can be applied to real-time H.264/AVC video encoding in mobile terminals, e.g. for videoconferencing applications, in order to optimise the battery life.

ACKNOWLEDGEMENTS

This work has been supported in part by the European Union under the SEA project.

REFERENCES

[1] D. Marpe, T. Wiegand, G.J. Sullivan, "The H.264/ MPEG-4 Advanced Video Coding standard and its applications", *IEEE Communications Magazine*, vol. 44, no. 8, pp. 134-144, August 2006.

[2] ITU-T and ISO/IEC JTC 1, "Advanced video coding for generic audio-visual services", *ITU-T Rec.H.264 and ISO/IEC 14496-10 (MPEG-4 AVC)*, Version 4: July 2005.

[3] C.-J. Lian, S.-Y.Chien, C.-P. Lin, P.-C. Tseng and L.-G. Chen, "Power-aware multimedia: concepts and design perspectives", *IEEE Circuits and Systems magazine*, Vol.7, No.2, pp. 26–34, 2nd quarter 2007.

[4] G.J. Sullivan, T. Wiegand, "Rate-Distortion optimization for video compression", *IEEE Signal Processing Magazine*, pp.74-90, November 1998.

[5] T. Anselmo, D. Alfonso, "Fast Rate-Distortion Optimization in the H.264/AVC Standard", in *Proc. 2nd MobiMedia*, Alghero, Italy, September 18-20, 2006.

[6] Y. Wang, S.F. Chang, "Complexity adaptive H.264 encoding for light weight streams", in *Proc. ICASSP 2006*, Toulouse, France, May 14-19, 2006.

[7] T. Wiegand, M. Lightstone, T.G. Campbell and S.K. Mitra, "Rate-Distortion optimized mode selection for very low bit rate video coding and emerging H.263 standard", *IEEE Trans. Circuits Syst. Video Tech., vol 6, no. 2, pp. 182-190, Apr. 1996.*

[8] D. Alfonso, D. Bagni, L. Pezzoni, E. Piccinelli, "Fast motion estimation with detection of scene changes and interlaced/progressive content for H.264/AVC encoding", in *Proc. of Int. Conf. on Multimedia, Image Processing and Computer Vision (MICV)*, Madrid, Spain, March 30-April 1, 2005.