



<b>Title</b>	Survey and Pareto Analysis Method for Coding Efficiency Assessment of Low Complexity H.264 Algorithms
<b>Authors(s)</b>	Ivanov, Yuri, Bleakley, Chris J.
<b>Publication date</b>	2006-09-01
<b>Publication information</b>	Ivanov, Yuri, and Chris J. Bleakley. "Survey and Pareto Analysis Method for Coding Efficiency Assessment of Low Complexity H.264 Algorithms," 2006.
<b>Conference details</b>	Irish Machine Vision and Image Processing Conference (IMVIP), Dublin, Ireland, 30 August - 1 September, 2006
<b>Item record/more information</b>	<a href="http://hdl.handle.net/10197/7125">http://hdl.handle.net/10197/7125</a>

Downloaded 2024-05-22 11:42:48

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd\_oa)



© Some rights reserved. For more information

# Survey and Pareto Analysis Method for Coding Efficiency Assessment of Low Complexity H.264 Algorithms

**Y. V. Ivanov, C.J. Bleakley**  
School of Computer Science and Informatics  
University College Dublin  
Belfield, Dublin 4  
{yury.ivanov, chris.bleakley}@ucd.ie

## Abstract

A large number of algorithms have been proposed by researchers to reduce H.264 computational complexity. Currently there is no method for reliably comparing the effectiveness of these algorithms. This paper proposes a method that allows direct comparison of the results obtained for various previously published low complexity H.264 encoding algorithms. The method is based on a new coding efficiency metric for unified bit rate and quality assessment. Pareto analysis is used to derive an optimal reference efficiency complexity curve using standard H.264 encoding tools and parameters. The paper demonstrates application of the method to the assessment of recently published low-complexity algorithms. The method shows that some published low complexity algorithms can be outperformed by simply adjusting the standard video encoder parameters.

**Keywords:** Video Compression, H.264, Complexity Scaling, Pareto Analysis.

## 1 Introduction

The H.264 standard [1] was developed by the Joint Video Team (JVT) for video compression in applications where bandwidth or storage capacity is limited. Experimental results show that H.264 provides better coding efficiency than MPEG-4 and H.263 at lower bit rates [2, 3] at the cost of significantly increased computational complexity.

A large number of algorithms have been proposed by researchers to reduce MPEG-4 and H.264 complexity [10-15]. Most of these algorithms are focused on new methods for the most computationally complex components of the video encoder, i.e. Motion Estimation (ME) [8], Discrete Cosine Transform (DCT) coding and Mode Decision (MD).

Unfortunately, there is no method for comparing the effectiveness of the various encoding algorithms. Most papers quote the percentage reduction in computational complexity relative to an arbitrary configuration of the JM reference encoder [5]. Variations in bit rate and perceptual quality are calculated in percentages relative to the reference encoder. Since different encoder configurations are used, the results described in different papers are not directly comparable and can be misleading. The impact of reduced complexity encoding schemes on both visual quality and bit rate is often not clearly stated.

This paper proposes a single metric for accessing the effectiveness of video encoding. This metric allows direct comparison of the results obtained for various previously published low complexity H.264 encoding schemes.

It is well known that the computational complexity of the H.264 encoder can be scaled by simply adjusting the encoding parameter configuration, for example the search range and number of modes to be searched. However, to the author's knowledge, there have been no publications which provide an analysis of the optimal encoding parameters for a required complexity point. This paper presents the results of a Pareto analysis which identifies the optimum operating points for the H.264 video encoder obtained by scaling complexity via modification of the parameter configuration. The analysis is important for two reasons. Firstly, it allows system designers to select the best encoder operating point for a given processor. Secondly, it allows researchers to assess the performance of novel low complexity encoding algorithms relative to that obtained by simply modifying the parameter settings of the encoder. We believe that the methodology applied is generally applicable to other complexity scaling problems.

The effectiveness of a number of published low complexity H.264 video encoding schemes is assessed by projecting their findings onto the Pareto curve. It was found that a number of papers use a sub-optimal reference encoder. While some ambiguity may arise from utilization of different software and hardware platforms, the results suggest that some published methods do not perform as well as simply scaling the video encoding parameters in an optimal fashion.

The paper is structured as follows. Section 2 gives a definition of complexity and introduces a coding efficiency metric for bit rate and quality assessment. Section 3 describes the method itself consisting of complexity analysis for the H.264 reference encoder and development of a Pareto-optimized H.264 complexity curve. Section 4 demonstrates application of our metric to recently published low complexity encoding algorithms. Finally, conclusions are given in Section 5.

## 2 Theory

There are two major components of computational complexity – time complexity and storage complexity. Time complexity is the number of computational operations required to execute a specific implementation of an algorithm. Storage complexity is the amount of memory required to perform the algorithm. In software implementation these two quantities determine the computational complexity of the algorithm on a specific hardware platform. This paper focuses on execution time, as derived from a reference software implementation of the H.264 encoder.

Complexity scaling of a video encoder, such as H.264, is a trade-off between complexity, bit rate and visual quality. When reducing complexity  $C$  it is expected that there will be some increase in bit rate  $R$  and in distortion  $D$ :

$$R = f_r(C), \quad D = f_d(C) \quad (1)$$

In general, the functions  $f_r$  and  $f_d$  have different characteristics. The computational complexity of the H.264 encoder for different encoding tool combinations has been partially evaluated in [2, 3]. A brief summary is given in Table 1.

**Table 1. Summary of Tool-By-Tool Encoder Complexity Analysis**

<i>Encoding tool</i>	<i>Impact on the PSNR and bit rate</i>	<i>Impact on the complexity</i>
1/4 <sup>th</sup> Sub-pixel accuracy	At low rates PSNR +0.07 dB, +4.7% bit rate. At high rates PSNR +0.04 dB, -12 % bit rate.	Memory access frequency (MAF) +15%.
Variable Block Sizes (VBS)	PSNR +0.07dB to +0.02dB, bit rate -5% to -17%.	Complexity increases linearly: +2.5% for each additional mode. Most of the bit rate reduction (75%–85%) is achieved using 4 modes.
Hadamard Transform	+0.02 to +0.12 dB PSNR, +2.4% bit rate.	MAF +20%.
CABAC	Up to 16% bit reduction.	MAF +25 to 30%.
Search Range Size	Negligible impact on PSNR and bit-rate.	MAF is higher up to 60 times.

It can be clearly seen from these results that the complexity of the video encoder can vary widely with different settings for the tools. Obviously not every combination is optimal in the rate-distortion sense. Unfortunately, the results do not provide information regarding the best combination of tools required to set the H.264 encoder complexity  $C$  to a specific operating point  $C_i$ . These measurements only provide a general picture of how H.264 complexity can be scaled.

Furthermore, since different low complexity algorithms impact bit rate and distortion in different ways, it is often difficult to determine which algorithm is in fact more efficient at a given operating points. Since changing complexity effects both bit rate and distortion, the need arises to unify both quantities into a single metric. At present, the rate-distortion model is widely used in video coding for making optimal decisions where both bit rate and distortion are important. The model is based on the following Lagrange formula [7]:

$$\min \{J\}, \quad \text{where } J = D + \lambda \cdot R \quad (2)$$

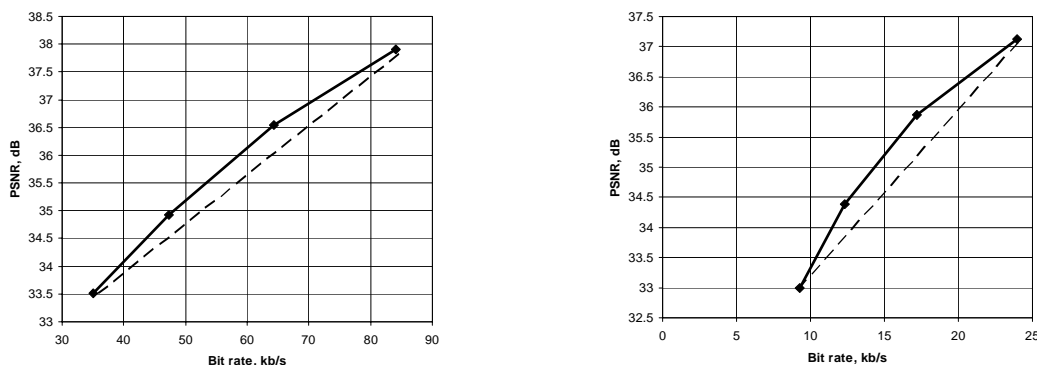
where  $D$  is a distortion measure (usually Sum of Absolute Differences) and  $R$  represents bit rate. During video encoding, the Lagrange rate-distortion function  $J$  is minimized for a particular value of the Lagrange multiplier,  $\lambda$ .

Based on this, we introduce a coding efficiency metric,  $W$ , which is dependent on visual quality loss,  $\Delta D$ , and bit rate change,  $\Delta R$ , relative to a reference full complexity encoder:

$$W = \Delta R + \mu \Delta D \quad (3)$$

where  $\Delta R$  is a percentage,  $\Delta D$  is PSNR in dB and  $\mu$  is a constant relating bit rate loss and distortion increase. Thus, for any given computation complexity,  $C_i$ , the most efficient encoder can be identified as that providing minimum  $W$ .

The constant  $\mu$  can be interpreted as the percentage increase in bit rate equivalent to 1 dB loss in PSNR. Previous work reported in [16] determined that a 10% decrease in bit rate is roughly equivalent to a loss of 0.5 dB in PSNR. In our work,  $\mu$  was determined experimentally. Bit rate and PSNR were measured for four video sequences at QP settings of 26, 28, 30 and 32. CIF and QCIF sequences with high and low bit rates were selected for the analysis. The results were plotted, as shown in Figure 1, and the gradient determined by fitting a linear model. The gradient was found to vary between 3.62 and 29.6 with a mean of 12.9. Hence,  $\mu$  was set to 13 for calculation of  $W$ . To check the robustness of the method, extreme values for  $\mu$  were also tested in the Pareto analysis and were determined to have little impact on the findings.



**Figure 1. Example of Rate-Distortion graphs for Carphone, QCIF (left) and Container, CIF (right). Dashed lines indicate a gradient of R-D function. Calculated  $\mu$  values are 11.3 and 3.62 respectively.**

Given the coding efficiency metric,  $W$ , it is now possible to compare the performance of various encoder configurations. This can be considered as an optimization problem. Given a

particular computational complexity requirement, what is the optimum encoder parameter configuration? Given that the encoder parameter configurations form a discrete set, we chose to solve the problem by Pareto analysis [9]. The efficiency of the encoder is assessed across a range of parameter configurations. These results are a projected on to a graph relating coder efficiency to computational complexity. The optimum encoder parameters can then be identified as those points leading to the points  $(C_i, W_i)$  which form the Convex Hull of the Individual Minima (CHIM). Parameter configurations corresponding to points inside the CHIM are sub-optimal.

### 3 Method of Coding Efficiency Assessment

The complexity distribution across all tools for the full complexity mode was profiled, i.e. full search, full VBS, search range 8, CABAC, full Hadamard, sub-pel accuracy on and de-blocking filter on. A wide range of different QCIF and CIF video sequences (298 frames each) with variable content were encoded at different QP settings using the JM 9.5 reference encoder, running on 3GHz Pentium IV with 1Gb RAM. The average results are shown in Table 2.

**Table 2. Complexity Distribution Between Different Encoding Tools in H.264**

<i>Encoding tool</i>	<i>% of complexity</i>
Motion Estimation (including Hadamard Transform)	66
Mode Decision and CABAC	22
Deblocking Filter	5
Transform Coding	4
Other (including I/O)	3

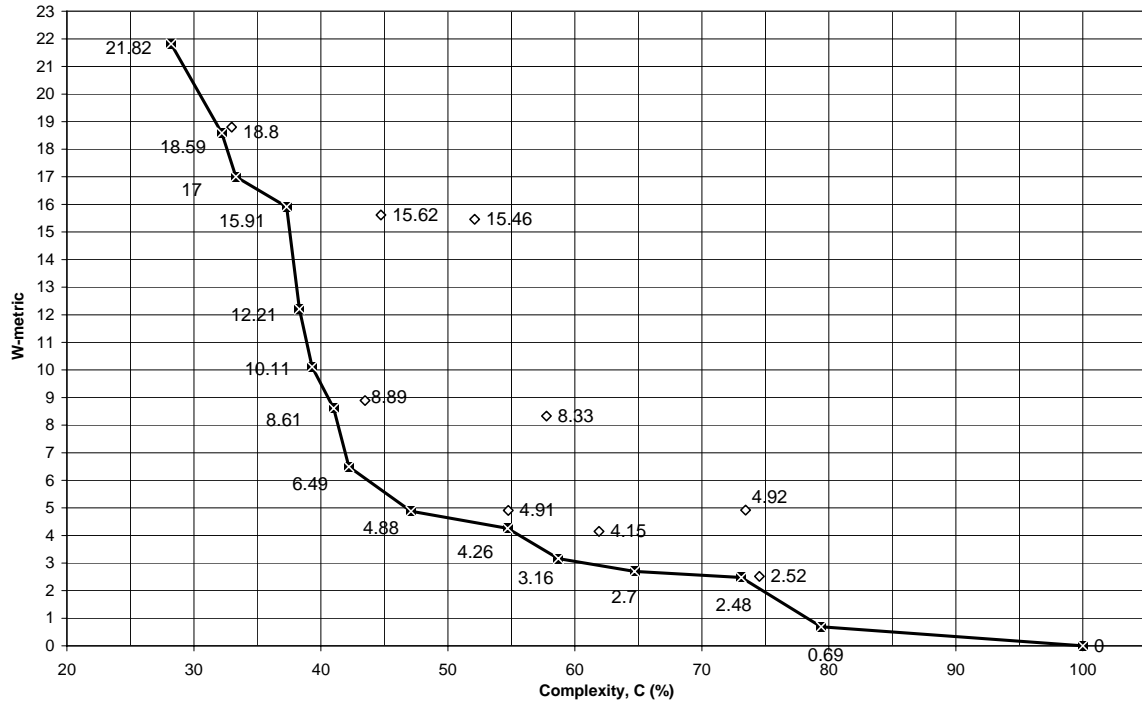
It can be observed that the computational complexity of Motion Estimation dominates and therefore provides most potential of complexity scaling (i.e. search range size, Hadamard on/off, varying of sub-pixel accuracy), which is consistent with [8]. However, since the variation of VBS modes has a high impact on MD complexity, which is 22% with CABAC, VBS can also be considered as an important tool for the purposes of complexity scaling. Switching off the deblocking filter reduces complexity further by 5%. Transform coding in H.264 has negligible impact on complexity (only 4%) since the H.264 standard adopts a separable integer transform with properties similar to DCT instead of the actual DCT [4].

In the next experiments in order to scale the complexity of the encoder from the full mode, search range size and number of VBS modes were varied. The limit of scaling is when the encoder operates in the lowest VBS mode with the search range size is equal to one. Additional simulations show that switching off sub-pixel accuracy in ME reduces complexity further at the cost of a significant bit rate increase and noticeable quality degradation. Utilization of UVLC instead of CABAC for low VBS also has a high impact on the bit rate and leads to an increase in complexity. Switching off Hadamard scales complexity to 28% with minimal bit rate increase and quality degradation.

After encoder profiling it can be concluded that the computational complexity of the standard H.264 encoder can be scaled to 28% by adjusting only the standard encoding tools. The simulation results provide information on the combination of encoding tools and parameters that are optimal for setting H.264 encoder complexity to a specific operating point  $C_i$ .

Using the results obtained during the previous experiments,  $W_i$  was calculated for each complexity point  $C_i$  as given in equation (3) with  $\mu=13$ . For each  $(C_i, W_i)$  calculated,  $(C_j, W_j)$  where  $C_i \leq C_j$  and  $W_i \leq W_j$  we leave only  $(C_i, W_i)$  and discard  $(C_j, W_j)$ . Therefore the CHIM of the Pareto surface is isolated. The variation of  $W$  with complexity averaged across all sequences is plotted in Figure 2 with optimal points marked as crossed squares and non-optimal points as hollow

diamonds. Values for  $\Delta R$ ,  $\Delta D$  and  $W$  from Figure 2 and H.264 parameter settings are given in Tables 3 and 4, respectively.



**Figure 2. Variation of W-metric with Complexity of Encoder**

As can be observed from Table 3, the function  $f_d$  has  $\Delta D_{max} = 0.37$  dB and, at the same time, complexity is gradually scaled down to 28% from its highest settings, resulting in up to a 17% of a bit rate increase  $\Delta R$ . Perceptual evaluation of visual quality reveals no anomalies in the encoded sequences. The slight quality degradation measured between full and minimal complexity modes only can be noticed on still images.

**Table 3. Optimal Points on Complexity Curve**

Value	Complexity, %													
	79	73	64	58	54	47	42	41	39	38	37	33	32	28
$\Delta R$ , %	0.5	0.8	1	1.4	2.4	2.9	4.3	6.4	7.4	9.3	12.3	13.4	14.9	16.9
$\Delta D$ , dB	0.01	0.12	0.12	0.13	0.14	0.14	0.15	0.16	0.2	0.21	0.27	0.27	0.28	0.37
$W$	0.69	2.48	2.7	3.16	4.26	4.88	6.49	8.61	10.1	12.2	15.9	17	18.6	21.8

**Table 4. H.264 Encoding Tools Settings for Reference Curve**

Encoding tool	Complexity, %														
	100	79	73	64	58	54	47	42	41	39	38	37	33	32	28
VBS	7	7	4	4	4	3	3	3	3	2	2	1	1	1	1
Search range	8	4	8	6	4	6	4	2	1	2	1	4	2	1	1
Hadamard	on	on	on	on	on	on	On	on	on	on	on	on	on	on	off

In order to investigate the behavior of the Pareto curve further from the ‘knee’, additional experiments were performed with the same configuration as for the 100% complexity point, except search range size was increased to 32.  $W$  averaged across all sequences was  $-0.07$  while complexity increase is 345% from reference point (or about 4.5 times). Thus, increasing the search

range further than 8 for QCIF and CIF images provides little PSNR or bit rate advantage, but does lead to a significant complexity increase. This result is consistent with [2, 3]. It is recommended to avoid large search ranges for low video resolutions.

The Rate-Distortion Optimization (RDO) tool [6] was also tested for the same reason. When switched on, it gives around a 295% complexity increase, while  $W$  drops to  $-2.83$ . Since this complexity point is so far from the Pareto ‘knee’, again, it is not included in the graphs. However, when working on optimizing of RDO algorithm (e.g. [12]), developers may want to extend the complexity curve with this point for assessment purposes.

Finally, it can be concluded that the optimal set of H.264 parameters given in Table 4 is generally suitable for the purposes of H.264 complexity scaling. The complexity curve can be refined further by running more experiments in the desired range.

## 4 Assessment of Published Algorithms

For the purpose of illustration, several recently published algorithms were investigated [10-15]. The method proposed in [10] utilizes motion vector cost and previous frame information for adaptive threshold cost based selection of macroblock (MB) mode. Fast multi-block selection [11] is based on the idea of detecting fast and slow moving areas of the frame and processing them differently. The algorithm proposed in [12] utilizes a special block matching order combined with SAD pre-calculation for reducing ME complexity and for skipping spatial predictive coding. The method [13] is based on the correlation of motion vectors across the various MB partitions. The block mode selection algorithm in [14] relies on two factors – complexity of macroblock and MB mode from previous frame. The low complexity encoding scheme described in [15] uses VBS prediction from the surrounding MBs.

Based on the simulation results provided by the authors, the  $W$ -metric was calculated for each algorithm and plotted along with the reference complexity curve, as shown in the Figure 3.

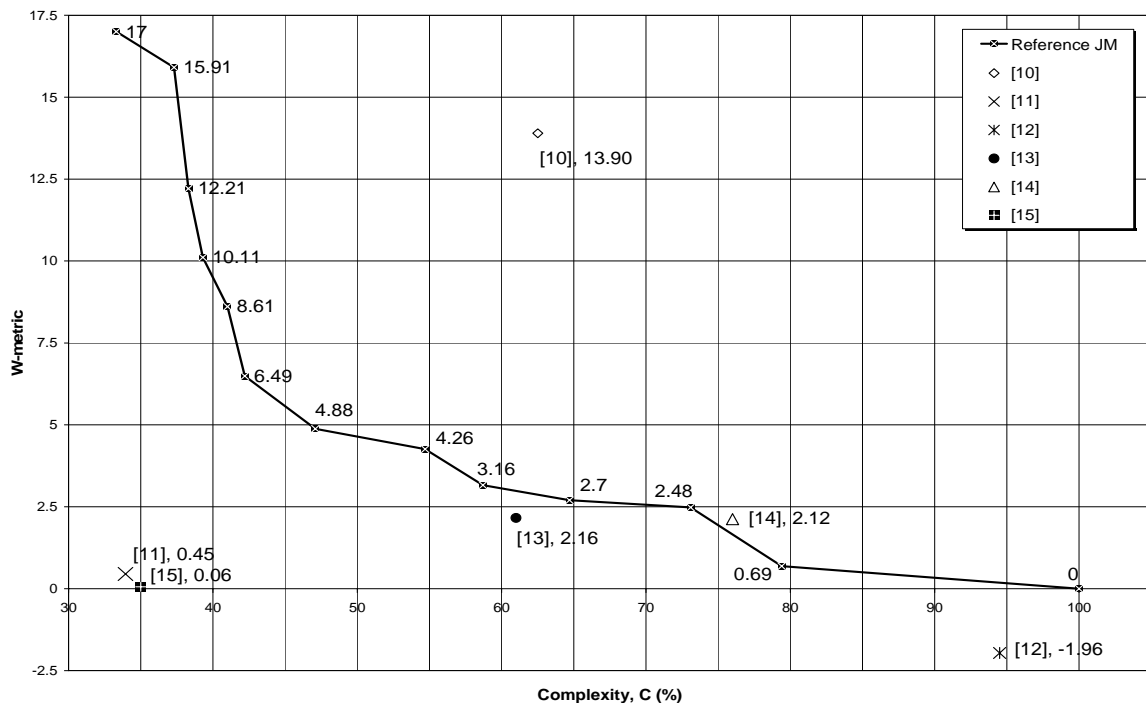


Figure 3. Comparison of Published Algorithms with Reference Complexity Curve

It can be seen that algorithms [10] and [13] provide complexity reduction around 40%, but [10] at the cost of significant bit rate increase (resulted high  $W$ ). Better results than [10] can be achieved by running H.264 with the parameters given at Table 4 for the complexity point of 58%.

Alternatively, to achieve  $W$  around 13.9 as in [10], the 38% complexity point can be chosen, thus, the same bit rate and quality results will be achieved by simply reducing VBS number and search range size. The algorithm in [14] is also not optimal – with only 25% of computational reduction the resulting  $W$  lies almost on the Pareto convex hull. It is preferable to use the configuration associated with the 79% complexity point.

In contrast, algorithms in [11] and [15] perform much better than the scaled reference encoder. Both provide quite significant complexity reduction of around 65% and have relatively low  $W$ . This can only be obtained by scaling the reference encoder to a complexity point of 70–80%. Thus, these algorithms are about 2.3 times more efficient than reference JM encoder.

The algorithm in [12] uses an improved rate-distortion technique and reports a significant complexity reduction, but calculated relative to a reference encoder with RDO on (i.e. relative to 295% instead of 100%). Plotted on the curve, it results of  $W = -1.96$  without reducing complexity significantly relative to the reference encoder with RDO off (only 10%). However, since similar  $W$  can only be achieved by running JM with RDO tool, the algorithm provides better bit rate than the reference encoding configuration.

## 5 Conclusions

In this paper a method for accessing the effectiveness of low complexity H.264 video encoding algorithms was proposed. The method allows direct comparison of the results obtained for various previously published low complexity H.264 encoding schemes. It has been demonstrated that by introducing a coding efficiency metric as a single measure, the assessment of bit rate and perceptual quality can be unified.

To the author's knowledge there have been no publications providing an analysis of the optimal encoding parameters for a required complexity point. The computational complexity of the H.264 encoder was scaled by adjusting the encoding parameter configuration. Pareto analysis was introduced for identifying the optimum operating points. The obtained results not only demonstrate the general picture of H.264 complexity scaling, which was found to be consistent with other publications, but, more important, they allow systems designers to select the optimum encoder operating point for a given processor.

The effectiveness of a number of published low complexity H.264 video encoding schemes was assessed by projecting results provided by the authors onto the Pareto curve. It was found that a number of papers use sub optimal configurations for the reference encoder (i.e. [10], [14]). While some ambiguity may arise from utilization of different software and hardware platforms, the results suggest that these methods do not outperform a reference encoder with encoding parameters scaled in an optimal fashion.

## Acknowledgements

This research project is sponsored by University College Dublin, Ireland.

## References

- [1] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG. (2003). *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H264|ISO/IEC 14496-10 AVC)*. document JVT-G050d35.doc, 7<sup>th</sup> Meeting: Pattaya, Thailand.
- [2] Implementation Studies Group of ISO/IEC. (2002). *Main Results of the AVC Complexity Analysis*. ISO/IEC JTC1/SC29/WG11, Klagenfurt.



- [3] Saponara, S., Blanch, C., Denolf, K. and Bormans, J. (2002). *Data Transfer and Storage Complexity Analysis of the AVC Codec on a Tool-By-Tool Basis*. ISO/IEC JCT1/SC29/WG11, M8547, Klagenfurt.
- [4] Ostermann, J., Bormans, J., List, P., Marpe, D., Narroschke, M., Pereira, F., Stockhammer, T. and Wedi, T. (2004). Video Coding with H.264/AVC: Tools, Performance, and Complexity. *IEEE Circuits and Systems Magazine*, 4.1: 7-28.
- [5] JVT reference software JM 9.5, on the Web: <http://iphome.hhi.de/suehring/tml>.
- [6] Sullivan, G. J. and Wiegand, T. (1998). Rate-Distortion Optimization for Video Compression. *IEEE Signal Processing Magazine*, 15.6: 74-90.
- [7] Everett III, H. (1963). Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources. *Operations Research*, 11: 399-417.
- [8] Kuhn P. (1999). *Algorithms, Complexity Analysis and VLSI Architecture for MPEG-4 Motion Estimation*, Kluwer Academic Publishers: Boston.
- [9] Das I. (1999) On characterizing the ‘knee’ of the Pareto curve based on Normal-Boundary Intersection, *Structural and Multidisciplinary Optimization*, 18.3:107-115.
- [10] Ahmad A., Khan N., Masud S. and Maud M.A. (2004). *Selection of Variable Block Sizes in H.264*, in Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP’04), 3:173-176.
- [11] Chang A., Wong P.H.W., Yeung Y.M. and Au O.C. (2004). *Fast Multi-Block Selection for H.264 Video Coding*, in Proc. of IEEE Int. Symp. on Circuits and Systems (ISCAS’04), 3:817-820.
- [12] Han K. and Y. Lee Y. (2004). *Fast Macroblock Mode Decision in H.264*, in Proc. of IEEE Region 10 Conf. (TENCON’04), 1:347 – 350.
- [13] Kuo T. Y. and Chan C.H. (2004). *Fast Macroblock Partition Prediction for H.264/AVC*, in Proc. of IEEE Int. Conf. on Multimedia and Expo (ICME’04), 1: 675–678.
- [14] Yu A.C. (2004). *Efficient Block-Size Selection Algorithm for Inter-Frame Coding in H.264/MPEG-4 AVC.*, in Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP’04), 3: 169-172.
- [15] Li G. L., Chen M. J., Li H. J. and Hsu C. T. (2005). *Efficient Motion Search and Mode Prediction Algorithms for Motion Estimation in H.264/AVC*, in Proc. of IEEE Int. Symp. on Circuits and Systems (ISCAS’05), 6: 5481 – 5484.
- [16] Bjontegaard G. (2001). *Calculation of average PSNR differences between RD curves*, ITU-T VCEG Meeting, Austin: VCEG-M33.