<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Adaptive Lagrange Multiplier for Low Bit Rates in H.264</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Authors(s)</strong></td>
<td>Ivanov, Yuri; Bleakley, Chris J.</td>
</tr>
<tr>
<td><strong>Publication date</strong></td>
<td>2005-08-31</td>
</tr>
<tr>
<td><strong>Conference details</strong></td>
<td>9th Irish Machine Vision and Image Processing Conference (IMVIP), Belfast, Northern Ireland, 30 - 31 August, 2005</td>
</tr>
<tr>
<td><strong>Item record/more information</strong></td>
<td><a href="http://hdl.handle.net/10197/7102">http://hdl.handle.net/10197/7102</a></td>
</tr>
</tbody>
</table>
ADAPTIVE LAGRANGE MULTIPLIER FOR LOW BIT RATES IN H.264

Authors:
Yury V. Ivanov, Dr. C.J. Bleakley
Dept. of Computer Science, University College Dublin, Belfield, Dublin 4, Ireland
e-mail: {yury.ivanov, chris.bleakley}@ucd.ie

ABSTRACT

This paper describes a novel adaptive Lagrange multiplier algorithm based on the general Rate-Distortion model with dynamic lambda calculation for H.264 video coding. When experimentally tested with the JM 8.1a encoder on QCIF video sequences the algorithm provides 2.8% average bit rate saving with insignificant PSNR loss (around 0.13%) without computational complexity increase. Experiments indicate that the algorithm may also be adopted for higher bit rates.

Keywords: H.264, Lagrange multiplier, low bit rate video coding.
ADAPTIVE LAGRANGE MULTIPLIER FOR LOW BIT RATES IN H.264

ABSTRACT

This paper describes a novel adaptive Lagrange multiplier algorithm based on the general Rate-Distortion model with dynamic lambda calculation for H.264 video coding. When experimentally tested with the JM 8.1a encoder on QCIF video sequences the algorithm provides 2.8% average bit rate saving with insignificant PSNR loss (around 0.13%) without computational complexity increase. Experiments indicate that the algorithm may also be adopted for higher bit rates.

Keywords: H.264, Lagrange multiplier, low bit rate video coding.

1. INTRODUCTION

The new 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 [2] and H.263 [3] at lower bit rates. However, these advantages come at the cost of increased computational complexity.

Rate-Distortion Optimization (RDO) [4] is an optional tool specified within the H.264 standard. It achieves improved compression and visual quality. Unfortunately its utilization leads to a computational complexity increase of 100-200% [5].

This paper proposes a technique, based on the general RD model for adaptive Lagrange multiplier calculation. It achieves similar compression results at low bit rates to conventional RD with insignificant drop in visual quality with almost no computational complexity increase. The proposed algorithm can be used instead of the RDO tool for encoding low bit rate video sequences.

The rest of this paper is organized as follows. Section 2 explains the background theory on Rate-Distortion and mode decision. Section 3 describes the experimental results of applying the RD technique to low bit rate video sequences. In Section 4 the adaptive Lagrange multiplier algorithm is proposed and experimental results are presented. Section 5 holds a discussion of the achieved results and, finally, conclusions are given in Section 6.

2. RATE-DISTORTION IN H.264 MODE DECISION

Rate-Distortion is used in H.264 for making mode decisions. The Rate-Distortion model is based on the following Lagrangian formula [6]:

\[ \min \{ J \}, \quad \text{where} \quad J = D + \lambda \cdot R \]  

where \( D \) is a distortion measure (usually Sum of Absolute Differences) and \( R \) is the number of bits used. The Lagrangian rate-distortion function \( J \) is minimized for a particular value of the Lagrange multiplier \( \lambda \).

This technique allows effective evaluation of a large number of possible coding choices but, in practice, computation time is the limiting factor and some interactions between coding decisions are neglected. Thus, \( D, \lambda \) and \( R \) are approximations.

In the H.264 reference encoder JM 8.1a [7], developed by the JVT, the mode decision algorithm uses equation (1). An optional RDO tool is also provided, allowing more precise estimation of \( J \) since \( \lambda \) and \( R \) are optimized in a rate-distortion sense.

In general, the Lagrange multiplier \( \lambda \) can be different for the Intra and Inter mode decisions. For the Intra decision it is referred as \( \lambda_{\text{MODE}} \) and for the Inter decision as \( \lambda_{\text{MOTION}} \).

In the case where the RDO tool is not used, the equation for the \( \lambda \) calculation is quite simple [7]:

\[ \lambda_{\text{MODE}} = \lambda_{\text{MOTION}} = f (QP) \]  

where \( QP \) is the quantization parameter.
where \( QP \) is quantizer step size. Thus, \( \lambda_{\text{MODE}} \) and \( \lambda_{\text{MOTION}} \) are fixed for the entire encoding process.

In the case where the RDO tool is used, the macroblock \( QP \) selected by the minimization routine is dependent on \( \lambda_{\text{MODE}} \). Therefore, \( \lambda_{\text{MODE}} \) and \( \lambda_{\text{MOTION}} \) are calculated differently:

\[
\lambda_{\text{MODE}} = 0.85 \cdot 2^{(QP/3)} \quad (3)
\]

\[
\lambda_{\text{MOTION}} = \sqrt{\lambda_{\text{MODE}}} \quad (4)
\]

In basic operation (RDO off), the encoder performs mode selection based on the parameter \( \lambda_{\text{MODE}} \), which is calculated from the \( QP \) setting as shown in equations (2-4). In fact, the relationship between \( \lambda_{\text{MODE}} \) and \( QP \) was derived experimentally by determining the rate-distortion optimal values of \( QP \) for a given set of \( \lambda_{\text{MODE}} \) values [4]. This relationship has a second order dependency on the bit rate of the sequence, namely that \( \lambda_{\text{MODE}} \) should be higher for higher bit rates.

In the next Section we will investigate the effect of RDO tool when applied to the encoding of low bit rate video sequences.

3. RATE-DISTORTION APPLIED TO LOW BIT RATE VIDEO SEQUENCES

For the purpose of RD investigation we used the reference JM 8.1a encoder. Different QCIF video sequences of 298 frames each were encoded with and without the RDO tool. The optimal encoder configuration for low bit rates is based on the experimental results from [8]:

- All 7 block modes in Variable Block Sizes
- Hadamard Transform and CABAC is on
- No B-frames
- Full search motion estimation algorithm
- Search Range Size = 8
- \( QP = 28 \)
- Picture frequency is 10 frames per second

The results of the experiment are given in Tables 1 and 4. It can be seen from Table 4 that bit rate savings for RD depend on the video sequence (4\% in average) while the average PSNR drop is insignificant (around 0.1\%).

The encoding statistics given in Table 1 indicate that the RDO tool uses less Intra coding in favor of more Inter coding. It seems that, for the tested video sequences, the RDO tool alters Intra/Inter decisions so that the bit rate is reduced without impacting visual quality and that most of the bit rate savings are achieved by the use of less Intra coding.

<table>
<thead>
<tr>
<th>Video sequence</th>
<th>No RD Intra MBs (I-blocks)</th>
<th>RD Intra MBs (I-blocks)</th>
<th>Opt. Intra MBs (I-blocks)</th>
<th>No RD Inter MBs (P-blocks)</th>
<th>RD Inter MBs (P-blocks)</th>
<th>Opt. Inter MBs (P-blocks)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coast</td>
<td>318</td>
<td>249</td>
<td>288</td>
<td>29184</td>
<td>29253</td>
<td>29214</td>
</tr>
<tr>
<td>Foreman</td>
<td>839</td>
<td>728</td>
<td>767</td>
<td>28663</td>
<td>28774</td>
<td>28735</td>
</tr>
<tr>
<td>Hall</td>
<td>241</td>
<td>178</td>
<td>227</td>
<td>29261</td>
<td>29324</td>
<td>29225</td>
</tr>
<tr>
<td>Mother</td>
<td>156</td>
<td>144</td>
<td>148</td>
<td>29346</td>
<td>29358</td>
<td>29502</td>
</tr>
</tbody>
</table>

Table 1. Intra/Inter MBs count for no RD, RD and optimal adaptive \( C'_\lambda \) lambda algorithms.

For the no RD case it is possible to vary \( \lambda_{\text{MODE}} \) in order to force the encoder to make more Inter decisions. In this way, it may be possible to reduce the bit rate without impacting a visual quality, in a similar manner to applying the RDO tool. This idea is investigated in the next Section.

4. METHOD OF ADAPTIVE LAGRANGE MULTIPLIER CALCULATION AND SIMULATION RESULTS

In the first series of experiments we chose to apply a fixed multiplier, \( C_\lambda \) to the conventional (no RD) value of \( \lambda_{\text{MODE}} \). Thus, equation (2) was transformed to:

\[
\lambda_{\text{MODE}} = \lambda_{\text{MOTION}} = C_\lambda \cdot f(QP) \quad (5)
\]

The experimental results below for video sequences with 200 frames each indicate that a bit rate reduction is achieved when \( 1 < C_\lambda < 2 \). For instance, Foreman with \( C_\lambda =1.75 \) shows a 5.39\% bit
savings, which is even higher than that achieved using the RDO tool.

The optimal value of $C_\lambda$ marked with gray in Table 2, is different for every video sequence and tends to be higher for the higher bit rate sequence.

Table 2. Bit rate for selected video sequences encoded with fixed value of $C_\lambda$.

<table>
<thead>
<tr>
<th>Video sequence</th>
<th>Orig. JM</th>
<th>0.5</th>
<th>1.25</th>
<th>1.5</th>
<th>1.75</th>
<th>2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreman</td>
<td>49.02</td>
<td>59.8</td>
<td>47.47</td>
<td>46.66</td>
<td>46.38</td>
<td>46.68</td>
</tr>
<tr>
<td>News</td>
<td>24.98</td>
<td>27.9</td>
<td>24.77</td>
<td>24.77</td>
<td>25.00</td>
<td>25.28</td>
</tr>
<tr>
<td>Mother</td>
<td>17.34</td>
<td>22.53</td>
<td>17.04</td>
<td>16.91</td>
<td>17.06</td>
<td>17.33</td>
</tr>
<tr>
<td>Akiyo</td>
<td>8.35</td>
<td>9.98</td>
<td>8.22</td>
<td>8.34</td>
<td>8.47</td>
<td>8.56</td>
</tr>
</tbody>
</table>

Table 3. Bit rate for selected video sequences encoded with different algorithms where $QP=24$.

<table>
<thead>
<tr>
<th>Video sequence</th>
<th>Orig. JM ($QP=24$)</th>
<th>Opt. $C'_\lambda$ &quot;as is&quot;</th>
<th>Opt. $C'_\lambda$, when $R'=R/2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coastguard</td>
<td>123.02</td>
<td>120.98</td>
<td>120.85</td>
</tr>
<tr>
<td>Foreman</td>
<td>76.07</td>
<td>73.97</td>
<td>73.66</td>
</tr>
<tr>
<td>Silent</td>
<td>45.87</td>
<td>44.85</td>
<td>44.81</td>
</tr>
<tr>
<td>Mother</td>
<td>26.58</td>
<td>26.02</td>
<td>25.99</td>
</tr>
</tbody>
</table>

For the given encoding parameters, the dependency of $C_\lambda$ on bit rate can be approximated using the MSE method as the following function:

$$C_\lambda = f(R) = -0.0002 \cdot R^2 + 0.023 \cdot R + 1.099$$  \hspace{1cm} (6)

where $R$ is the average bit rate.

Using equation (6) it is possible to calculate $C_\lambda$ adaptively based on the average bit rate as determined during the encoding process. In the next series of experiments the following algorithm for adaptive $C'_\lambda$ calculation was used.

```plaintext
start encoding with initial $C'_\lambda$;
bite sent = 0;
for every frame except Intra frames {
        bits sent += current frame bits;
        if ( (Frame number % S) == 0 )
                calculate average bit rate;
                calculate new $C'_\lambda$;
                bits sent = 0;
}
```

The experimental results are given in Table 4. The average bit rate saving around 2.8% with PSNR drop of 0.13% was achieved for the adaptive $C'_\lambda$ calculation algorithm, where initial $C'_\lambda=1.0$ and $S=5$.

Finally experiments with different $QP$ were performed. The selected video sequences were encoded using the optimal adaptive $C'_\lambda$ algorithm with $QP=24$. Two cases were tested, one using equation (6) "as is" and one using scaling of $R$ according to the ratio of the new bit rate ($QP=24$) to that of the previous version ($QP=28$). The results are shown in Table 3. In the case of scaled $R$, the percentage bit rate savings achieved were similar those obtained in the $QP=28$ case.

Since modifying the value of $\lambda$ in equation (2) leads to the change in the balance between rate and distortion, subjective evaluation of all video sequences was also performed. In fact, the visual difference between the pictures produced by original JM8.1a and adaptive Lambda method can be hardly found (see Figure 1.)

Figure 1. Image produced by JM (left) and proposed adaptive Lambda algorithm (right).

5. DISCUSSION

Experimental results from Table 4 show that despite the fact that coefficients for equation (6) were calculated empirically from a limited number of video sequences, the method in general is applicable for wide range of QCIF video sequences. For the $QP$
values rather than the 28, method also works, as can be seen from Table 3, but equation (6) should be extended to allow for the \( QP \) change: i.e. the calculation of an average bit rate \( R \) should be dependent on \( QP \).

Evidently the technique provides improved coding performance. The mean gradient of the RD curve calculated across several sequences for the adaptive lambda algorithm is 48% better than that obtained with simple scaling of \( QP \).

The encoding statistics for the optimal version of the adaptive lambda algorithm, shown in the Table 1, indicate that usage of equation (5) instead of equation (2) leads to Intra/Inter decision percentages similar to those obtained using the RDO tool, i.e. equations (3-4).

The fact that the bit rate saving of the RDO tool was only partially achieved with the adaptive lambda algorithm can be explained by noting that the RDO tool performs a \( J \) minimization routine in addition to applying equations (3-4). The adaptive lambda algorithm only relies on the tendency of the optimal \( \lambda_{MODE} \) to be higher for higher bit rate sequences.

The encoding statistics for the optimal version of the adaptive lambda algorithm, shown in the Table 1, indicate that usage of equation (5) instead of equation (2) leads to Intra/Inter decision percentages similar to those obtained using the RDO tool, i.e. equations (3-4).

The fact that the bit rate saving of the RDO tool was only partially achieved with the adaptive lambda algorithm can be explained by noting that the RDO tool performs a \( J \) minimization routine in addition to applying equations (3-4). The adaptive lambda algorithm only relies on the tendency of the optimal \( \lambda_{MODE} \) to be higher for higher bit rate sequences.

The encoding statistics for the optimal version of the adaptive lambda algorithm, shown in the Table 1, indicate that usage of equation (5) instead of equation (2) leads to Intra/Inter decision percentages similar to those obtained using the RDO tool, i.e. equations (3-4).

The encoding statistics for the optimal version of the adaptive lambda algorithm, shown in the Table 1, indicate that usage of equation (5) instead of equation (2) leads to Intra/Inter decision percentages similar to those obtained using the RDO tool, i.e. equations (3-4).

The encoding statistics for the optimal version of the adaptive lambda algorithm, shown in the Table 1, indicate that usage of equation (5) instead of equation (2) leads to Intra/Inter decision percentages similar to those obtained using the RDO tool, i.e. equations (3-4).

The encoding statistics for the optimal version of the adaptive lambda algorithm, shown in the Table 1, indicate that usage of equation (5) instead of equation (2) leads to Intra/Inter decision percentages similar to those obtained using the RDO tool, i.e. equations (3-4).

Table 4. Bit rate and PSNR for different algorithms.

<table>
<thead>
<tr>
<th>Video sequence</th>
<th>Bit rate, kbits/s</th>
<th>PSNR, dB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Orig. JM</td>
<td>RD</td>
</tr>
<tr>
<td>Coast</td>
<td>65.3</td>
<td>63.93</td>
</tr>
<tr>
<td>Carphon e</td>
<td>51.56</td>
<td>49.42</td>
</tr>
<tr>
<td>Foreman</td>
<td>44.12</td>
<td>41.94</td>
</tr>
<tr>
<td>Table</td>
<td>41.9</td>
<td>40.14</td>
</tr>
<tr>
<td>Silent</td>
<td>28.15</td>
<td>26.83</td>
</tr>
<tr>
<td>Mean difference with orig. JM, %</td>
<td>-4.01</td>
<td>-2.79</td>
</tr>
</tbody>
</table>

Table 4. Bit rate and PSNR for different algorithms.

It should also be noticed, that the bit rate saving of 2.8% for proposed method is achieved without computational cost increase. Usage of RD leads up to 200% computational complexity increase [5] and generally is not recommended for QCIF sequences for that reason [8].

6. CONCLUSIONS

The proposed adaptive Lagrange multiplier algorithm gives around 2.8% of average bit rate improvement for the H.264 encoder without applying the RDO tool while achieving nearly the same PSNR without computation complexity increase.

A method using a fixed multiplier was also tested. However, the optimum value varies with video sequence. While nearly the same results were achieved with the fixed multiplier, the dynamic version is preferred due to its robust nature.

Potential future work includes extension of the algorithm to higher bit rates (CIF) and refinement of the adaptation model based on the analysis of a greater number of test cases.

6. REFERENCES


