**JOINT OPTIMIZATION OF RUN-LENGTH CODING, HUFFMAN CODING AND QUANTIZATION TABLE WITH COMPLETE BASELINE JPEG COMPATIBILITY**

*En-hui Yang* and *Longji Wang*

* Dept. of ECE, University of Waterloo, Ontario, Canada, ehyang@uwaterloo.ca
** Research In Motion, Waterloo, Ontario, Canada, lowang@rim.com

**ABSTRACT**

JPEG optimization strives to maximize the best rate distortion performance while remaining faithful to the JPEG syntax. Given an image, if soft decision quantization (SDQ) is applied to its DCT coefficients, then Huffman table, quantization step sizes and SDQ coefficients are three free parameters over which a JPEG encoder can optimize. In this paper, we first propose a novel algorithm to find the optimal SDQ coefficient indices in the form of run-size pairs among all possible candidates given that the other two parameters are fixed. Based on this algorithm, we then formulate an iterative algorithm to jointly optimize the run-length coding, Huffman coding and quantization step sizes. The proposed iterative algorithm achieves a compression performance better than any previously known JPEG compression results and even exceeds the quoted PSNR results of some state-of-the-art wavelet-based image coders like Shapiro’s embedded zerotree wavelet algorithm at the common bit rates under comparison.

**Index Terms**— Image coding, optimization methods, rate distortion theory, dynamic programming.

**1. INTRODUCTION**

JPEG [1] is a popular DCT-based still image compression standard. The popularity of the JPEG coding system has motivated the study of JPEG optimization schemes [2]-[4] which remain faithful to the JPEG syntax. This kind of decoder-compatible JPEG optimization is of great commercial value because the optimized JPEG images take less memory to store and less time to transmit while the JPEG decoders keep unchanged; it will find more and more applications in wireless communications.

A JPEG encoder consists of three basic steps – DCT transform, quantization and entropy coding, where the entropy coding consists of the run-length coding and Huffman coding. This framework offers significant opportunity to apply rate-distortion (R-D) consideration at the encoder side. It is evident that one can optimize JPEG encoding by finding a good hard-decision quantization table and Huffman coding tables. What less obvious is that one can also optimize JPEG encoding by optimizing the image data. Depending on the stage where the image data are during the whole JPEG encoding process, the image data take different forms as shown in Figure 1. Before hard decision quantization, they take the form of DCT coefficients; after hard decision quantization, they take the form of DCT indices, i.e., quantized DCT coefficients; after zig-zag sequencing and run-length coding, they take the form of run-size pairs followed by integers specifying the exact amplitude of DCT indices within respective categories (such integers are called in-category indices in this paper). Although the JPEG syntax allows the quantization tables to be customized at the encoder, typically some scaled versions of the example quantization tables given in the standard [1] (called default tables in this paper) are used. The scaling of the default tables is suboptimal because the default tables are image-independent. Even with an image-adaptive quantization table, JPEG must apply the same table for every image block, indicating that potential gain remains from optimizing the coefficient indices, i.e., DCT indices. Since DCT indices can be equivalently represented as run-size pairs followed by in-category indices through run-length coding, we shall simply refer to coefficient index optimization as run-length coding optimization in parallel with step size and Huffman coding optimization. In this paper, we not only propose a very neat, graph-based run-length code optimization scheme, but also present an iterative optimization scheme which jointly optimizes the run-length coding, Huffman coding and quantization step sizes as shown in Figure 1.

The rest of this paper is organized as follows. We formulate our joint optimization problem in Section 2 and give the solutions in Section 3. Detailed experimental results are given in Section 4 and Section 5 concludes this paper.

**2. FORMAL PROBLEM DEFINITION**

We now formulate our joint optimization problem, where the minimization is done over all the three free parameters in baseline JPEG. We only consider the optimization of AC coefficients in this paper and the optimization of DC coefficients can be considered separately using a trellis structure.

Given an input image $I_0$ and a fixed quantization table $Q$ in the JPEG encoding, the coefficient indices completely determine a sequence of run-size pair followed by in-category indices for each
8x8 block through run-length coding, and vice versa. Our problem is posed as a constrained optimization over all possible sequences of run-size pairs \((R,S)\) followed by in-category indices \(ID\), all possible Huffman tables \(H\), and all possible quantization tables \(Q\).

\[
\min_{(R,S,0)} \left\{ d[I_0,(R,S,0)] \right\}
\]

subject to
\[
r[(R,S), H] \leq t_{\text{budget}}
\]

or equivalently
\[
\min_{(R,S,0)} \left\{ r[(R,S), H] \right\}
\]

subject to
\[
d[I_0,(R,S,0)] \leq d_{\text{budget}}
\]

where \(d[I_0,(R,S,0)]\) denotes the distortion between the original image \(I_0\) and the reconstructed image determined by \((R,S,0)\) and \(Q\) over all AC coefficients, and \(r[(R,S), H]\) denotes the compression rate for all AC coefficients resulting from the chosen sequence \((R,S,0)\) and the Huffman table \(H\). \(t_{\text{budget}}\) and \(d_{\text{budget}}\) are respectively the rate constraint and distortion constraint. With the help of the Lagrange multiplier, we may consider the rate-constrained problem or distortion constrained problem into the following unconstrained problem

\[
\min_{(R,S,0)} \left\{ J(\lambda) = d[I_0,(R,S,0)] + \lambda \cdot [r[(R,S), H]] \right\}
\]

where the Lagrangian multiplier \(\lambda\) is a parameter that represents the tradeoff of rate for distortion, and \(J(\lambda)\) is the Lagrangian cost. This type of optimization falls into the category of so-called fixed slope coding scheme advocated in [5].

It is informative to compare our joint optimization problem with the joint thresholding and quantizer selection in [4]. On one hand, both of them are an iterative process aiming to optimize the three parameters jointly. On the other hand, our scheme differs from that considered in [4] in two distinct aspects. First, we consider the full optimization of the coefficient indices or the sequence \((R,S,0)\) instead of a partial optimization represented by dropping only insignificant coefficient indices as considered in [4]. As we shall see in the next section, it turns out that the full optimization has a very neat, computationally effective solution. Second, we do not apply any time-consuming quantizer selection schemes to find the R-D optimal step sizes in each iteration. Instead, we use the default quantization table or an initial optimized quantization table and update the step sizes efficiently in each iteration for local optimization of the step sizes.

### 3. PROBLEM SOLUTIONS

The rate-distortion optimization problem (3) is a joint optimization of the distortion, rate, Huffman table, quantization table, and sequence \((R,S,0)\). To make the optimization problem tractable, we propose an iterative algorithm that chooses the sequence \((R,S,0)\), Huffman table, and quantization table iteratively to minimize the Lagrangian cost of (3), given that the other two parameters are fixed. Since a run-size probability distribution \(P\) completely determines a Huffman table, we use \(P\) to replace the Huffman table \(H\) in the optimization process. The iteration algorithm can be described as

1. Initialize the original distribution \(P_0\) from the given image
   \(I_0\) and a quantization table \(Q_0\). Set \(t = 0\), and specify a tolerance \(\varepsilon\) as the convergence criterion.

2. Fix \(Q\) and \(P\) for any \(t \geq 0\). Find an optimal sequence \((R,S,0)\) that achieves the following minimum
   \[
   \min_{(R,S,0)} \{ J(\lambda) = d[I_0,(R,S,0)] + \lambda \cdot r[(R,S), P]\}
   \]
   Denote \(d[I_0,(R,S,0)] + \lambda \cdot r[(R,S), P]\) by \(J'(\lambda)\).

3. Fix \((R,S,0)\). Update \(Q_t\) and \(P_t\) into \(Q_{t+1}\) and \(P_{t+1}\), respectively so that \(Q_{t+1}\) and \(P_{t+1}\) together achieve the following minimum
   \[
   \min_{Q,P} \{ J(\lambda) = d[I_0,(R,S,0)] + \lambda \cdot r[(R,S), P]\}
   \]

4. Repeat Steps 2) and 3) for \(t=0,1,2,\ldots\) until \(J'(\lambda) - J'^{t+1}(\lambda) \leq \varepsilon\). Then, output \((R_{t+1},S_{t+1},ID_{t+1})\), \(Q_{t+1}\), and \(P_{t+1}\).

Since the Lagrangian cost function is non-increasing at each step, convergence is guaranteed. The core of the iteration algorithm is Step 2) and Step 3), i.e., finding the sequence \((R,S,0)ID\) to minimize the Lagrangian cost \(J(\lambda)\) given \(Q\) and \(P\), and updating the quantization step sizes with the new indices of the image. These two steps are addressed separately as follows.

### 3.1 Graph-based run-length coding optimization

As mentioned in Section 2, JPEG quantization lacks local adaptivity even with an image-adaptive quantization table, which indicates that potential gain remains from the optimization of the coefficient indices themselves. This gain is exploited in Step 2). Optimal thresholding in [3],[4] only considers a partial optimization of the coefficient indices, i.e., dropping less significant coefficients in the R-D sense. In this paper, we propose an efficient graph-based optimal path searching algorithm to optimize the coefficient indices fully in the R-D sense. It can not only drop the less significant coefficients, but also can change them from one category to another - even changing a zero coefficient to a small nonzero coefficient is possible if needed in the R-D sense. Since given the Lagrangian cost \(J(\lambda)\) is block-wise additive given \(Q\) and \(P\), the minimization in Step 2) can be solved in a block by block manner. That is, the optimal sequence \((R,S,0)ID\) can be determined independently for every 8x8 image block. Thus, in the following, we limit our discussion to only one 8x8 image.

Let us define a graph with 65 nodes (or states). As shown in Figure 2, the first 64 states, numbered as \(i=0,1,\ldots,63\), correspond to the 64 coefficients of an 8x8 image block in zigzag order. The last state is a special state called the end state, and will be used to take care of EOB (end-of-block). Each state \(i(i \leq 63)\) may have incoming connections from its previous 16 states \(j(j < i)\), which correspond to the run, \(R\), in an \((R,S)\) pair (in JPEG syntax, \(R\) takes value from 0 to 15). The end state may have incoming connections from all the other states with each connection from state \(i(i \leq 62)\) representing the EOB code after the \(i^{th}\) coefficient. State 63 goes to state end without EOB code. For a given state \(i(i \leq 63)\) and its predecessor \(i-1\) \((0 \leq r \leq 15)\), there are 10 parallel transitions between them which correspond to the size group, \(S\), in an \((R,S)\) pair. For simplicity, we only draw one transition in the graph shown in Figure 2; the complete graph needs the expansion of \(S\). For each state \(i(i > 15)\), there is one more transition from state \(i-15\) to state \(i\) which corresponds to the pair \((15, 0)\), i.e., ZRL (zero run length) code. We assign a cost for each transition \((r, s)\) from
state $i-r-1$ to state $i$ as the incremental Lagrangian cost of going from state $i-r-1$ to state $i$ when the $j^\text{th}$ DCT coefficient is quantized to size group $s$ (i.e., the coefficient index needs $s$ bits to represent its amplitude) and all the $r$ DCT coefficients are quantized to zero. Specifically, this incremental cost is equal to

$$\sum_{j=r}^{s} C_{ij}^{r} \left[ C_{ij} - q_{ij} \cdot ID_{ij} \right]^{2} + \lambda \cdot \left( -\log_{2} P(r,s) + s \right)$$

where $C_{ij} = 1, 2, \ldots, 63$ is the $j^\text{th}$ DCT coefficient, $ID_{ij}$ is the in-category index corresponding to the size group $s$ that gives rise to the minimum distortion to $C_{ij}$ among all in-category indices within the category specified by the size group $s$, and $q_{ij}$ is the $i^\text{th}$ quantization step size. With these definitions, every possible run-size pairs of an 8x8 block corresponds to a path from state 0 to the end state with a Lagrangian cost. Therefore, we may employ a fast dynamic programming algorithm to find the optimal path from state 0 to state end among all possible paths which results in the minimum Lagrangian cost. The readers are referred to [6] for more details.

The above procedure is a full dynamic programming method, and always gives us the optimal solution. To further reduce its computational complexity, we can modify it slightly. In particular, we do not have to compare the incremental costs among the 10 or 11 parallel transitions from one state to another state. Instead, it is sufficient for us to compare only the incremental costs among the 10 or 11 parallel transitions from one state to another state. Instead, it is sufficient for us to compare only the incremental costs among the 10 or 11 parallel transitions from one state to another state.

4. EXPERIMENTAL RESULTS

The proposed algorithm can be configured flexibly based on user’s requirement. We may optimize the run-size pairs only. Alternatively, we may run the joint optimization algorithm iteratively. Both configurations can start with the default quantization table or an initially optimized quantization table. In the latter case, we choose the fast algorithm in [2] to generate an initially optimized quantization table to start with. Table I compares the PSNR values of different settings of the proposed algorithm as well as the reference methods for 512x512 images Lena and Barbara. Figures 3 plots the PSNR against the bit rate for image Barbara. A customized Huffman table is used in the last entropy encoding stage like the optimal adaptive thresholding scheme in [4]. Several remarks are in order. First, the optimal adaptive thresholding scheme in [3], [4] is a subset of the proposed run-length coding optimization. Therefore, the proposed run-length coding optimization scheme outperforms the optimal adaptive thresholding scheme for both images under any bit rates as expected. Second, quantization table optimization plays a less role at low bit rates since more coefficients are quantized to zero at low bit rates. The proposed joint optimization scheme with an initial scaled default quantization table achieves better results that the joint optimization scheme in [4] at low bit rate(s), which obtained the best JPEG compression results before this paper. Third, the proposed algorithm with an initially optimized quantization table outperforms the joint optimization scheme in [4] for all bit rates under comparison and even exceeds the quoted PSNR results of some state-of-the-art wavelet-based image coders like Shapiro’s embedded zerotree wavelet algorithm [8] for some complicated image like Barbara at the bit rates under comparison.
Table I. Comparison of PSNR values with different optimization methods (512x512 Lena and Barbara)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>.25</td>
<td>31.63</td>
<td>32.1</td>
<td>32.21</td>
<td>32.37</td>
<td>32.47</td>
<td>32.3</td>
<td>33.17</td>
<td>33.17</td>
</tr>
<tr>
<td></td>
<td>.50</td>
<td>34.90</td>
<td>35.5</td>
<td>35.43</td>
<td>35.80</td>
<td>36.04</td>
<td>35.9</td>
<td>36.18</td>
<td>36.28</td>
</tr>
<tr>
<td></td>
<td>.75</td>
<td>36.62</td>
<td>37.2</td>
<td>37.32</td>
<td>37.68</td>
<td>38.14</td>
<td>38.1</td>
<td>38.02</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>37.91</td>
<td>38.4</td>
<td>38.68</td>
<td>39.26</td>
<td>39.63</td>
<td>39.6</td>
<td>39.42</td>
<td>39.55</td>
</tr>
<tr>
<td>Barbara</td>
<td>.25</td>
<td>25.31</td>
<td>25.9</td>
<td>26.09</td>
<td>26.93</td>
<td>27.04</td>
<td>26.7</td>
<td>26.64</td>
<td>26.77</td>
</tr>
<tr>
<td></td>
<td>.50</td>
<td>28.34</td>
<td>29.3</td>
<td>29.62</td>
<td>30.66</td>
<td>30.94</td>
<td>30.6</td>
<td>29.54</td>
<td>30.53</td>
</tr>
<tr>
<td></td>
<td>.75</td>
<td>31.02</td>
<td>31.9</td>
<td>32.30</td>
<td>33.14</td>
<td>33.82</td>
<td>33.6</td>
<td>32.55</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>33.16</td>
<td>34.1</td>
<td>34.52</td>
<td>35.23</td>
<td>36.07</td>
<td>35.9</td>
<td>34.56</td>
<td>35.14</td>
</tr>
</tbody>
</table>

We now present some computational complexity results of the proposed algorithm. As mentioned in Section 3, given a state and a predecessor, we may find the minimum incremental cost by comparing all the 10 size groups or 3 size groups (i.e., the size group from the hard-decision quantizer and its two neighboring groups). Our experiments show that these two schemes achieve the same performance in the region of interest. Only when λ is extremely large, we see that the results from comparing 10 size groups slightly outperform the results from comparing 3 size groups. These large values of λ are useless in practical situations. Therefore, all the experimental results in this paper are obtained by comparing 3 size groups. Table II tabulates the CPU time in second for the C code implementation of the proposed algorithm on a Pentium PC in one iteration with 512x512 Lena image. It can be seen that our algorithm is very efficient compared to the scheme in [4] (the scheme in [4] takes several dozens of seconds for one iteration). When the proposed algorithm is applied to web image acceleration, it takes around 0.2 second to optimize a typical size (300x200) JPEG color image with 2 iterations.

Table II. CPU time of the proposed algorithm on a Pentium PC (512x512 Lena)

<table>
<thead>
<tr>
<th>Settings</th>
<th>Float DCT</th>
<th>Fast integer DCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comparing 3 size groups</td>
<td>1.5 s</td>
<td>0.3 s</td>
</tr>
<tr>
<td>Comparing 10 size groups</td>
<td>2.0 s</td>
<td>0.7 s</td>
</tr>
</tbody>
</table>

5. CONCLUSIONS
In this work, we have presented a graph-based R-D optimal algorithm for JPEG run-length coding. It finds the optimal run-size pairs in the R-D sense among all the candidates. Based on this scheme, we have proposed an iterative algorithm to optimize run-length coding, Huffman coding and quantization table jointly. The proposed iterative joint optimization algorithm results in PSNR gain of up to 3 dB or alternatively up to 30% bit rate compression improvement for the test images, compared to baseline JPEG. Our algorithms are not only computationally effective but completely compatible with existing JPEG and MPEG decoders. They can be applied to the application areas such as web image acceleration, digital camera image compression, MPEG frame optimization and transcoding.

6. REFERENCES