JPEG Quality Enhancer without Quantization Table

Presented by Yan-Yu Fu

Advisor : Frank Shih
What is Blocking, Blurring and Ringing?
In those Low-Bit Rate (High Compression Ratio) JPEG images, quality is degenerated due to artifacts like, blocking, and blurring.

How to enhance JPEG images with high compression ratio and low PSNR?

How to measure the Blurriness and Blockiness such that once the accumulated score is below certain threshold, there is a necessity for image enhancement?
**Previous Method : JPEG Transcoder**

**Problem : Need Both** Quantization table and Huffman Table **in JFIF header**

<table>
<thead>
<tr>
<th>Step</th>
<th>Compression</th>
<th>Step</th>
<th>Decompression</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Convert ColorSpace</td>
<td>5</td>
<td>Convert ColorSpace</td>
</tr>
<tr>
<td>2</td>
<td>DownSampling by 8x8 Blocks</td>
<td>4</td>
<td>UpSampling</td>
</tr>
<tr>
<td>3</td>
<td>Discrete Cosine Transform (DCT)</td>
<td>3</td>
<td>Inverse DCT</td>
</tr>
<tr>
<td>4</td>
<td>Quantization</td>
<td>2</td>
<td>De-Quantization</td>
</tr>
<tr>
<td>5</td>
<td>Huffman Encoding</td>
<td>1</td>
<td>Huffman Decoding</td>
</tr>
</tbody>
</table>
Problem:
In those Low-Bit Rate (High Compression Ratio), JPEG images, quality is degenerated due to artifacts like, blocking, and blurring.

Solution Part I:
JPEG Quality Estimator

Solution Part II:
JPEG Quality Enhancer without Quantization Table and Huffman Table
Let $X$ be the test JPEG image of size $M \times N$. Let $d_h(m, n)$ be the difference along each horizontal line.

$$d_h(m, n) = X(m, n+1) - X(m, n)$$

$$B_h = \frac{1}{M} \left( \left\lfloor \frac{N}{8} \right\rfloor - 1 \right) \sum_{i=1}^{M} \sum_{j=1}^{\left\lfloor \frac{N}{8} \right\rfloor - 1} |d_h(i, 8j)|$$

$$B = \frac{(B_h + B_v)}{2} \quad (This\ is\ Blockiness)$$
Blurriness : Edge Discontinuity

1:40

1:200
Metric of Blurriness

- **Idea:** the discontinuity of edges

- **Algorithm:**
  1. Use Sobel edge detector to get edges
  2. Calculate the gradient in the binary image.
  3. Sum the absolute value of column gradient.
  4. Sum up all the column gradient, and divide by the total # of column; this is the Blurriness.
Two Sets of Images:

Set 1 contains “Lena.jpg” with Compression Ratio ranging from 20 to 200 step 20.

Set 2 contains “Lauren.jpg” with Compression Ratio ranging from 20 to 200 step 20.

Each Set will be gone through the process of
I. Blockiness Test
II. Blurriness Test
Compressed Lena

1:40

1:100

1:200
Compressed Lauren Bacall

1:40  1:100  1:200
## Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
<th>120</th>
<th>140</th>
<th>160</th>
<th>180</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lena</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Lauren</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Blocking threshold for enhancement: 9
Blurring threshold for enhancement: 4
Algorithm I: **Wiener filter**

A noise-removal filter, using neighborhoods of size M-by-N to estimate the local image mean and standard deviation.

Algorithm II: **Disk Filter + Wiener**

1. design a disk filter with a variable size radius.
   So, it returns a circle averaging filter within the square matrix of side 2*RADIUS+1.
2. Use disk filter to operate on JPEG image
3. Use wiener filter
Algorithm III:

1. Disk Filter
2. Edge Taper
3. Wiener Filter

Edge Taper is used to tape the discontinuities along the image edges.
Algorithm IV: Wiener and Morphology

1. Wiener Filter
2. Morphology

Use `se = strel('Disk', 1);`
`Image2 = imclose(Image1, se);`

Since Morphology can only be applied to binary or gray-scale image, when I deal with color image, then deal Red Green Blue separately.
Experimental Result (I)

Algorithm I: (PSNR from 20.0221 to 20.8266)

Lena 1:200

Wiener Filter
Experimental Result (II)

Algorithm II: (PSNR from 20.0221 to 20.8889)

Lena 1:200

Disk Filter (radius=1)
Experimental Result (III)

Algorithm IV. (PSNR from 20.0221 to 20.9045)

Lena 1:200

Combo Filter
Experimental Result (IV)

Algorithm V. (PSNR from 20.0221 to 20.9269)

Lena 1:200

Morphology