AN IMPROVED DC RECOVERY METHOD FROM AC COEFFICIENTS OF DCT-TRANSFORMED IMAGES

Shujun Li[1], Junaid Jameel Ahmad[1], Dietmar Saupe[1], C.-C. Jay Kuo[2]

[1] Department of Computer and Information Science, University of Konstanz, Germany
[2] Ming Hsieh Department of Electrical Engineering, University of Southern California, USA

Quick Questions and Answers

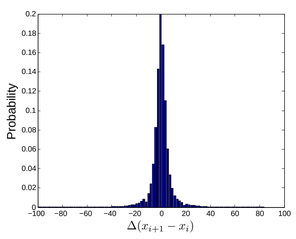
1. *What is it about?*
   AC2DC = Recovering DC coefficients from AC coefficients.
2. *What inspired your work?*
   The work of Uehara, Safavi-Naini and Ogunbona (USO) [IEEE Trans. Image Processing, 15(11): 3592-3596, 2006].
3. *How well does Uehara et al.'s work?*
   It is not bad, but cannot always produce good results.
4. *What have you done?*
   We proposed a new algorithm significantly better than USO.
5. *Does your algorithm have a name?*
   Yes! We call it FRM = "Under/Over-**F**low **R**ate **M**inimization".
6. *How did you make FRM better?*
   By reducing error propagation and introducing optimization.
7. *Can FRM be further improved?*
   Yes! There are plenty of possibilities to further improve FRM.
8. *Do you have a companion web page?*
   Yes: `http://www.hooklee.com/default.asp?t=AC2DC`

1. DC in DCT

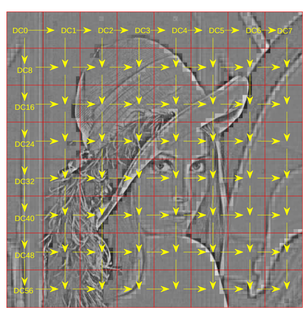
- Blockwise DCT is everywhere: JPEG, MPEG, VCD, DVD, ...
- DC coefficients contains more visual information and consume more bits when the image is encoded.
- Encrypting DC coefficients only is believed to be a good way of light-weight selective encryption (SE).
- DC encryption can also be combined with other SE methods.
- Unfortunately, in 2006 Uehara et al. proposed a method to recover DC coefficients from AC ones.

2. The USO Method

Property 1: *The difference between two neighboring pixels is a Laplacian variate with zero mean and a small variance.*



Property 2: *The range of pixel values calculated from AC coefficients constrains the DC coefficient:* $N(t_{\min} - \min(B^*)) \leq DC \leq N(t_{\max} - \max(B^*))$, *where* $[t_{\min}, t_{\max}]$ *is the pixel range and* B^* *is the DC-free block.*



Step 1: Reconstruct all DCs from DC(0) by scanning the image from a corner.
Step 2: Intersect valid DC ranges of all blocks to get a range that is used to globally adjust all DCs.
Step 3: Repeat Steps 1 and 2 by scanning the image from the four corners and average the results.
Step 4: Post-process the image to handle invalid pixel values.

3. Our FRM Method

3.1 Error propagation in the USO method

Original image:

Pixel value range: $[25, 245]$

Four scans:

Final result:

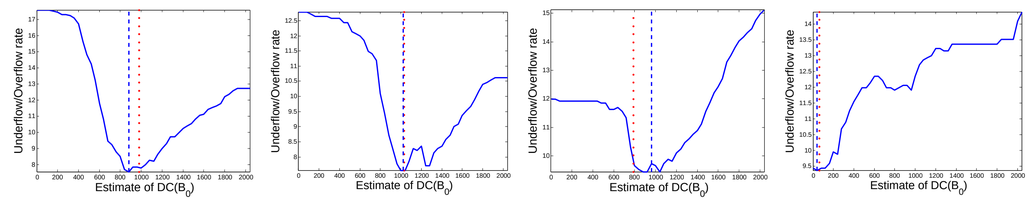
Pixel value range: $[-88, 303]$

Difference image:

IQA metrics: PSNR=14.3, SSIM=0.732, MS-SSIM=0.711

3.2 Our solution

Step 1: Do USO Step 1, but adjust (if necessary) the estimate DC of each block so that no under/over-flow pixel value exists.
Step 2: Repeat Step 1 for different values of DC(0) to minimize the blockwise under/over-flow rate.
Step 3: The same as the USO method.



Four scans:

Final result:

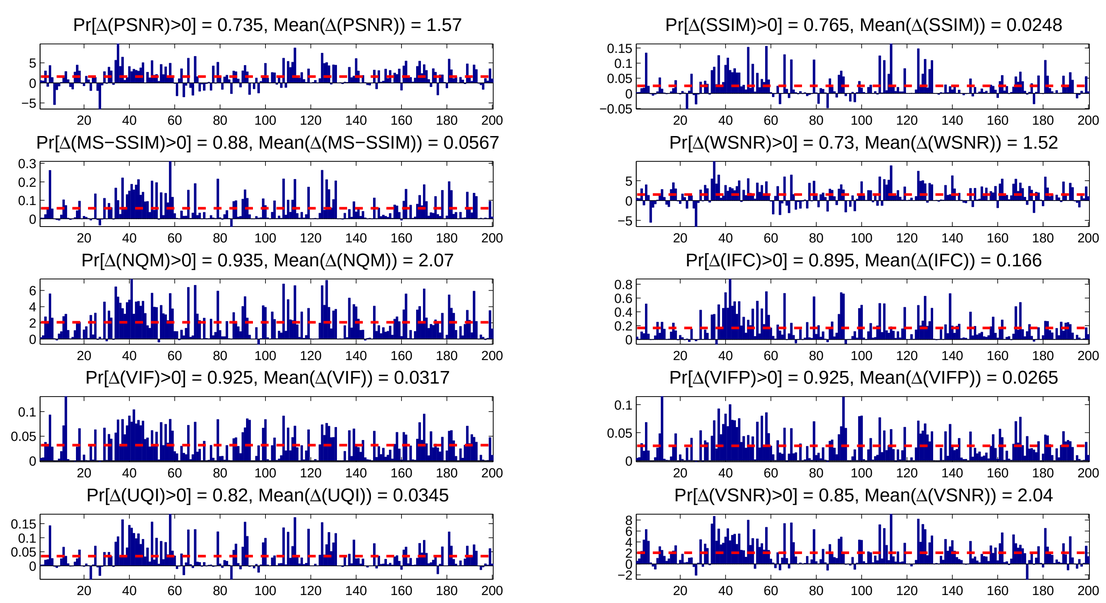
Pixel value range: $[9, 249]$

Difference image:

IQA metrics: PSNR=23.2, SSIM=0.9, MS-SSIM=0.924

3.3 More experimental results

Results on 200 test images: Δ = FRM − USO.



ICIP 2010 – 17th IEEE International Conference on Image Processing, Hong Kong SAR, China, September 26-29, 2010