Optimizing the Deblocking Algorithm for H.264 Decoder Implementation

By Ken Kin-Hung Lam
klam@mail.ece.utexas.edu
What is “deblocking”?

- Remove the discontinuity across block boundaries resulted from block-based DCT and coarse quantization steps in low bit rate encoding.

[List, Joch, Lainema, Bjøntegaard and Karczewicz, 2003]
Technical Challenges

• Remove blocking artifacts without affecting the true edges in the picture so that the overall sharpness of the image is maintained.
  – A low pass filter is an intuitive choice, but the picture is blurred in general.

• Computationally intensive, which requires as much as 1/3 of resources of the decoder.

• High implementation complexity with the proposed algorithms based on complex mathematical operations.
Projection on Convex Sets (POCS) based Iterative Algorithm [Zakhor, 1992]

• One projection is on the set of signals that are bandlimited in both horizontal and vertical directions. Another projection is on the set of quantization intervals of the DCT coefficients.

• Iteratively mapping a pixel value back and forth between the two sets until the value is within both sets.

• There is no guarantee in the time to convergence due to the iterative nature of the algorithm.
Weighted Sums of Symmetrically Aligned Pixels [Averbuch, Schclar and Donoho, 2005]

- A pixel’s value is recomputed with a weighted sum of itself and the other 3 pixel values which are symmetrically aligned with respect to block boundaries.
- The weights can be linear or quadratic.
- Every pixel of the picture is going through a filtering process.
Adaptive Deblocking Filter
[List, Joch, Lainema, Bjøntegaard and Karczewicz, 2003]

- An edge is classified with a set of conditions on the data points normal to the edge.
- Filters with different strength are applied to different class of edge. For example, strong filter is applied on smooth blocks.
- The algorithm is adaptive because the conditions are based on the quantization parameters included in the relevant blocks.
Proposed Works

- A simple blocking artifact detection algorithm which is easier to be implemented.
  - Keeping power consumption minimization in mind.
- An algorithm identifying edges which will not result in a great improvement after filtering, and skip the filtering.
  - Trading off quality for a lower computation complexity.
- Implementing the new algorithm on an open source reference codec and comparing the quality results with some existing algorithms.