# REMOVAL OF TILE ARTIFACTS USING PROJECTION ONTO SCALING FUNCTIONS FOR JPEG 2000

**Kathrin Berkner and Edward L. Schwartz**

Ricoh Innovations, Inc., 2882 Sand Hill Road, Suite 115, Menlo Park, CA 94025, email: [berkner, schwartz]@rii.ricoh.com

***Abstract**: For practical image compression systems, handling images divided into independent tiles in the pixel domain is important. However, tiling introduces artifacts when transform coefficients are quantized. This paper presents a "detiling" solution applicable to wavelet transform coefficients. The solution, called Projection Onto Scaling Functions, eliminates the blocking artifacts by computing approximations that match the smoothness of the compressed image inside of a tile. This solution is applicable to any wavelet filter – extending previous work that applied only to a limited class of wavelet filters – and specifically targets use in decoding of JPEG2000 compressed images.*

## 1 INTRODUCTION

Image compression systems used in real world applications have to incorporate memory and bandwidth constraints into their architecture. A typical approach to reduce memory cost and/or to allow parallel processing is to divide images into tiles and compress each tile independently. Such a tiling is standard in the DCT-based JPEG compression system where tiles consists of blocks of 8x8 pixels. Naive decoding for independent tiles can create visible tile boundary artifacts. For DCT-based compression they are commonly called block-artifacts. Significant work has been done in block-artifact removal in DCT-compressed images. Approaches include simply smoothing the decompressed image at the boundaries with a lowpass filter, computing a polynomial fit through the given coefficients (Section K.8 of [1]), Projection Onto Convex Sets [2] and denoising via wavelet shrinkage [3]. Those methods rely on assuming an image model for describing smoothness across block boundaries.

Compression artifacts in wavelet-based compressed images have very different characteristics than DCT artifacts. Without any tiling, in so-called *full-frame compression*, a DCT system would produce severe Gibbs artifacts over the entire image. Wavelet-based systems would produce some ringing around edges and some "wavelet noise" in smooth regions by reconstruction of isolated wavelet basis vectors. Both artifacts are determined by the smoothness of the wavelet system. The Gibbs artifacts in DCT compression are reduced by choosing very small tiles. As a consequence, tile artifacts overcome the Gibbs artifacts. Wavelet-based systems allow for much larger tiles, and both types of artifacts, full-frame and tile boundary artifacts are clearly visible.

In this paper a detiling solution for wavelet-based compressed images is proposed with the goal to modify an image at the tile boundaries such that the image quality near those boundaries matches the quality inside of a tile. That way a homogeneous image quality is obtained, i.e. the detiling solution adapts automatically to the full-frame image quality determined by the encoder. It takes into account the four components that influence tile boundary artifacts: wavelet filters, maximal decomposition level, quantization and location of tile boundaries. The first three are responsible for full-frame compression artifacts.

First, the general Projection Onto Scaling Functions (POSF) "detiling" solution is described for arbitrary wavelet filters, resolution levels, and quantization. Then the details of how to apply the solution to the decoding of a JPEG2000 (J2K) [5] bit stream for a specific J2K transform and quantization are given and demonstrated in examples.

## 2 DETILING

The Projection Onto Scaling Functions detiling solution consists of matching both the smoothness of the wavelet system and quantization bounds.

### 2.1 Matching smoothness of wavelet systems

The full-frame wavelet decomposition of a signal *s* is given by $s = \sum_k c_{J,k}\varphi_{J,k} + \sum_{j \ge J}\sum_k d_{j,k}\psi_{j,k}$ where $\varphi$ and $\psi$ are scaling and wavelet function of the inverse transform, and c and d are scaling and wavelet coefficients. In terms of J2K, $c_{J,k}$ are lowpass coefficients at the coarsest resolution, $d_{j,k}$ are the high pass coefficients at all finer resolutions, and $\varphi$ and $\psi$ are the basis vectors of the system's inverse wavelet transform.

The wavelet decomposition with tiles is given by $s = \sum_k \tilde{c}_{J,k}\tilde{\varphi}_{J,k} + \sum_{j \ge J}\sum_k \tilde{d}_{j,k}\tilde{\psi}_{j,k}$, where $\tilde{\varphi}, \tilde{\psi}, \tilde{c}, \tilde{d}$ differ from $\varphi, \psi, c, d$ only at the tile boundaries. After compression the scaling and wavelet coefficients are replaced by the reconstructed quantized coefficients $c^{(q)}$ and $d^{(q)}$. Typically, the scaling coefficients are quantized much less than the wavelet coefficients.

In order to match the smoothness of the wavelet system in the reconstruction the first step for "detiling" the image is to try to find wavelet coefficients $\hat{d}$, such that the tile inverse transform at each level j yields the same result as the full-frame transform projected onto the scaling functions, i.e.

$$\sum_k c^{(q)}_{j,k}\varphi_{j,k} = \sum_k \tilde{c}^{(q)}_{j,k}\tilde{\varphi}_{j,k} + \sum_k \hat{d}_{j,k}\tilde{\psi}_{j,k}.$$

In general, the coefficients $c^{(q)}_{j,k}$ are not available. But for many wavelet systems used in practice when applying mirroring techniques at the boundaries the differences $|\tilde{c}_{j,k} - c_{j,k}|$ in lowpass coefficients are very small compared to the differences $|\tilde{d}_{j,k} - d_{j,k}|$ in the detail coefficients. Therefore, the coefficients $\tilde{c}^{(q)}_{j,k}$ are used as estimates for $c^{q)}_{j,k}$. The problem is now reduced to solving the set of linear equations

$$\sum_k \tilde{c}^{(q)}_{j,k}\varphi_{j,k}(t) = \sum_k \tilde{c}^{(q)}_{j,k}\tilde{\varphi}_{j,k}(t) + \sum_k \hat{d}_{j,k}\tilde{\psi}_{j,k}(t) \qquad (1)$$

for samples at positions t that are being influenced by tile boundary detail coefficients. Depending on the length of the wavelet and scaling filters, the set of linear equations that have to be solved could be overdetermined. In order to avoid dealing with an overcomplete system, the linear equations are solved only for a number of reconstructed samples close to the tile boundary that is equal to the number of detail coefficients $\tilde{d}_{j,k}$ that differ from the full-frame reconstructed coefficients $d_{j,k}$. This result can also be considered a FIR filter that is applied to both tiles near the boundary. A prior paper [4] describes a similar approach for wavelet systems that apply two-tap lowpass filter in the forward transform. In that case $\tilde{c}_{j,k} = c_{j,k}$ and the set of equation from Eq.(1) reduces to a single equation for one detail coefficient.

### 2.2 Matching quantization

In smooth image regions matching only smoothness yields correct results. However, in cases where an edge is aligned at a tile boundary the reconstruction will be too smooth. Following the approach in [4] of incorporating the quantization into the detiling solution it is possible to maintain the full-frame characteristics around edges by clipping the corrected coefficients $\hat{d}$ back into their original quantization bounds. The details are explained in the following.

A commonly used quantization scheme is scalar quantization. For a quantizer Q the quantized wavelet coefficient is given by $q(d) = \text{sign}(d) \cdot \lfloor |d|/Q \rfloor$. Inverse scalar quantization provides a reconstructed value $q^{-1}(q(d)) = Qq(d) + \text{sign}(q(d)) \cdot r$ for $q(d) \neq 0$, where r is any integer between 0 and Q-1. For q(d) = 0, $q^{-1}(q(d)) = r$, where r is any value between -(Q-1) and Q-1. The value of r determines the exact reconstruction value of the possible values. Usual inverse scalar quantization picks r based on statistical assumptions. For example, r might be chosen such that the inverse quantization provides the midpoint of the valid range of values. The *matching the smoothness of wavelet systems* part of the detiling solution picks reconstruction values that are valid values based *only* on smoothness. If a value $\hat{d}$ does not correspond to a valid value for r, the $\hat{d}$ value is clipped to the closest valid r value or $\hat{d}$ is replaced by a value from a statistically based choice of r. With this step *matching the quantization* is added to the POSF detiling solution.

Wavelet compression systems commonly use a 2D separable wavelet transform. In this case the explicit quantization of the 1D inverse horizontal wavelet transform can be applied to LL and HL subbands after detiling based on the HL scalar quantization to result in L coefficients (assuming inverse horizontal transform is before vertical inverse transform as in J2K). The 1D inverse horizontal wavelet transform can be applied to LH and HH subbands without detiling to result in H coefficients. The 1D vertical inverse wavelet transform can be applied to each pair of L and H coefficients after detiling. However, to apply detiling the quantization for both H coefficients is needed but it is only given implicitly by the known quantization of the LH and HH coefficients. The proposed method for clipping to the H quantization is:

1) Use Eq. (1) to find $\hat{H}$ values.
2) Use the forward horizontal wavelet transform on the pair of L and $\hat{H}$ coefficients to compute $L\hat{H}$ and $H\hat{H}$
3) If $L\hat{H}$ and/or $H\hat{H}$ is out of the valid range, clip or make a statistically based valid choice.
4) Use the inverse horizontal wavelet transform on the now valid $L\hat{H}$ and $H\hat{H}$ to get a pair of valid $\hat{H}$ values.

The detiling solution does not require scalar quantization, any quantization that defines a valid range of reconstruction values maybe used.

### 2.3 Other methods for tile boundary artifact reduction

The proposed solution for wavelet compression systems incorporates smoothness of the wavelet system across tile boundaries into the detiling strategy. The alternative methods mentioned in the introduction, originally developed for a DCT compression systems, do not include any characteristics of smoothness of the transform and require an image model. Any of those methods could be applied with appropriate modifications to wavelet tile boundary artifacts, and the results would be good for images that matched a specific model, and bad for images that did not. The advantage of the detiling solution of this paper is that no image model is required because the full-frame wavelet system is modeled instead. The examples in the Section 4 demonstrate the power of not requiring an image model.

Another method for reducing tile boundary artifacts is to overlap tiles when encoding. Annex I of J2K Part II [5] provides several options for doing this (the type of overlap may be SSO, TSSO or both and the wavelet transform may modified at the boundary or not). Overlap has a rate cost, because redundant information at tile boundaries is being coded independently in multiple tiles. While these options are allowed, there are no compliance requirements for decoders to handle files with these options. The detiling solution of this paper has the advantage that as a decoder option, the encoder and the rate of encoded files are not effected and tile boundary artifacts can be reduced from files that all J2K decoders can decode.

## 3 APPLICATION TO JPEG 2000

The detiling solution can be applied to J2K decoding when decoding multiple tiles. In addition to the previously discussed advantages of using tiles, the compliance portion of the J2K standard encourages the use of tiles.

In order to support devices with limited capabilities, Part IV and Amendment 1 of Part I of [5] define “classes” of decoders and “profiles” of codestreams. A Class 0 decoder only is required to decode 128×128 reduced resolutions. Both Profiles 0 and 1 require a 128x128 subband or tiles with a 128x128 subband. Encoding images in a Profile 0 code stream so a minimally capable, Class 0, decoder (perhaps in a mobile and/or battery powered device) can fully decode them guarantees compatibility. For images bigger than 128×128 pixels, using 128×128 tiles allows decoding at full resolution.

### 3.1 Extracting transform and quantization from headers

Detiling uses information from J2K headers (see Annex A of [5]). The wavelet transform used for the compression is extracted from the SPcod or SPcoc "transformation" field in appropriate COC or COD tag for each tile-component. The number of wavelet transform levels is extracted from the SPcod or SPcoc "number of decomposition levels" field.

For detiling, a scalar quantizer Q is mapped to J2K by $Q = 2^{M_b - N_b} \cdot \Delta_b$. The value $\Delta_b$ is the traditional scalar quantizer defined by J2K in Annex E of [5] and the appropriate QCC or QCD header. For the 5-3 transform, $\Delta_b$ is 1, and quantization is performed by truncation of bitplanes (specifically, $N_b$-$M_b$ bitplanes) only. The maximum number of bitplanes, $M_b$, is computed from the appropriate QCC or QCD header according to Annex E. The number of bitplanes decoded for a specific coefficient, $N_b$, requires information from the packet header for each codeblock (see Annex B of [5]). $N_b$ may be different for different coefficients in the same codeblock and determined from the operation of the coefficient bit modeling (Annex D of [5]).

## 4 EXAMPLES

This section contains examples using the reversible 5-3 transform of J2K. The adaptation to the irreversible 9-7 transform is straightforward.

First those coefficients $\tilde{d}$ are determined that differ from the full-frame coefficients. They are shaded gray in the following table with tile boundary between position 7 and 8. For this step the filters used in the forward transform (lowpass: [-1/8 1/4 3/4 1/4 -1/8], highpass: [-1/2 1 -1/2]) are needed.

| pos. | (3) | (4) | (5) | (6) | (7) | (8) | (9) | (10) | (11) | (12) |
|---|---|---|---|---|---|---|---|---|---|---|
| $\tilde{c}_1$ | | (4) | | (6) | | (8) | | (10) | | (12) |
| $\tilde{d}_1$ | (3) | | (5) | | (7) | | (9) | | (11) | |
| $\tilde{c}_2$ | | (4) | | | | (8) | | | | (12) |
| $\tilde{d}_2$ | | | | (6) | | | | (10) | | |

The filters of the inverse transform needed in Eq.(1) are the two phases of inverse low (even: [1], odd: [1/2 1/2]) and highpass (odd: [-1/8 3/4 -1/8], even: [-1/4, -1/4]) filters. Since the odd inverse filter is longer, the position most effected by the tile boundary is the position adjacent to the boundary that is odd. Eq.(1) is solved at this position. Because the forward high pass filter is only three taps, the high pass coefficient closest to the other side of the boundary is unaffected by the boundary for the first level of the transform. For transform levels after the first, Eq.(1) is solved at the even position adjacent to the boundary.

Keeping in mind that in J2K filtering at boundaries is handled by symmetric extension Eq.(1) for the odd position at the first level results in

$$\frac{1}{2}\tilde{c}_1(6) + \frac{1}{2}\tilde{c}_1(8) = \frac{1}{2}\tilde{c}_1(6) + \frac{1}{2}\tilde{c}_1(6) - \frac{1}{8}\tilde{d}_1(5) + \frac{3}{4}\hat{d}_1(7) - \frac{1}{8}\tilde{d}_1(5).$$

Solving for the unknown $\hat{d}_1$ gives

$$\hat{d}_1(7) = (2\tilde{c}_1(8) - 2\tilde{c}_1(6) + \tilde{d}_1(5))/3. \quad (2)$$

For the odd position (6), at level 2 Eq.(1) yields

$$\frac{1}{2}\tilde{c}_2(4) + \frac{1}{2}\tilde{c}_2(8) = \frac{1}{2}\tilde{c}_2(4) + \frac{1}{2}\tilde{c}_2(4) - \frac{1}{8}\tilde{d}_2(2) + \frac{3}{4}\hat{d}_2(6) - \frac{1}{8}\tilde{d}_2(2)$$

Solving for the unknown $\hat{d}_2$ gives $\hat{d}_2(6) = (2\tilde{c}_2(8) - 2\tilde{c}_2(4) + \tilde{d}_2(5))/3$. For the even position (8) at level 3, Eq.(1) yields $\tilde{c}_2(8) = \tilde{c}_2(8) - \frac{1}{4}\hat{d}_2(10) - \frac{1}{4}\hat{d}_2(10)$, and solving for the unknown $\hat{d}_2$ gives $\hat{d}_2(10) = 0$.

It should now be obvious how to handle the third or higher level of the 5-3 transform and alternative alignment of the tile boundary.

### 4.1 Example for a smooth signal

As a numerical example consider the signal s to be a linear ramp. Without tiles *(full-frame)*, the lowpass coefficients $c_1$ and the highpass coefficients $d_1$ for one level of the 5-3 transform are shown in the next table. With tiles, the lowpass coefficients $\tilde{c}_1$ and the highpass coefficients $\tilde{d}_1$ are shown below. Also shown are the high pass coefficients after quantization, $q(\tilde{d}_1)$, with a quantizer Q > 2, and the reconstructed quantized signal s'. The shaded values differ from the corresponding full-frame values.

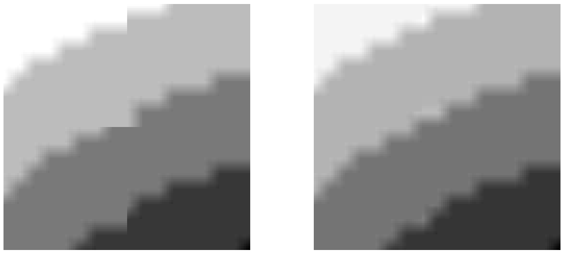
| pos. | (3) | (4) | (5) | (6) | (7) | (8) | (9) | (10) | (11) |
|---|---|---|---|---|---|---|---|---|---|
| s | 10 | 12 | 14 | 16 | 18 | 20 | 22 | 24 | 26 |
| $c_1$ | | 5 | | 7 | | 9 | | 11 | |
| $d_1$ | 0 | | 0 | | 0 | | 0 | | 0 |
| $\tilde{c}_1$ | | 12 | | 17 | | 20 | | 24 | |
| $\tilde{d}_1$ | 0 | | 0 | | 2 | | 0 | | 0 |
| $q^{-1}(q(\hat{d}_1))$ | 0 | | 0 | | 0 | | 0 | | 0 |
| s' | 10 | 12 | 14 | 17 | 17 | 20 | 22 | 24 | 26 |

Using POSF Eq.(2) results in $\hat{d}_1(7)$=2. Applying clipping has no effect. Since for this example, $\hat{d}_1(7)$ is equal to $\tilde{d}_1(7)$ the "detiled" signal will be equal to the original signal.

Figure 1 shows a magnified tile boundary with and without detiling in a smooth region of a test image for quantizer Q=16 (LL at level l=3) Q=32 (l=3), Q=64 (l=2), Q=128 (l=1), denoted as Q: 16 | 32 64 128.

### 4.2 Exa mple of keeping edges sharp

An example of a step edge shows how the quantization bounds of high pass coefficients are used to keep edges sharp. The following table displays the full-frame coefficients for one level of the 5/3 transform and reconstruction for two different quantizers Q=16 and Q=256.



tile boundary artifacts with detiling solution

**Figure 1** Detiling in smooth regions

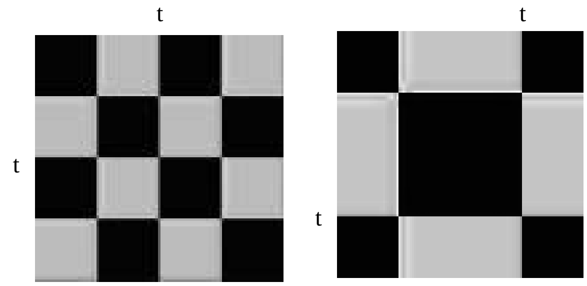
| pos. | (3) | (4) | (5) | (6) | (7) | (8) | (9) | (10) |
|---|---|---|---|---|---|---|---|---|
| s | 210 | 210 | 210 | 210 | 210 | 0 | 0 | 0 |
| $c_1$ | | 210 | | 236 | | 26 | | 0 |
| $d_1$ | 0 | | 0 | | 106 | | 0 | |
| $q^{-1}(q(d_1))$, Q=16 | 0 | | 0 | | 104 | | 0 | |
| s' (Q=16) | 210 | 210 | 210 | 210 | 209 | 0 | 0 | 0 |
| $q^{-1}q(d_1)$, Q=256 | 0 | | 0 | | 0 | | 0 | |
| s' (Q=256) | 210 | 210 | 223 | 236 | 131 | 26 | 13 | 0 |

For the quantization Q=16, $q(d_1(7))=6$. Inverting the quantization, $q^{-1}(q(d_1(7)))$ is allowed to be any value from 6×16=96 to 6×16+15=111. Picking the midpoint results in reconstruction coefficient 104. Contrasting reconstruction s' for Q=16 and Q=256, it is clear that appearance of edges in a full-frame lossy decompressed image depends on the quantization.

Now consider the case when there is a tile boundary at the same location as the step edge.

| pos. | (3) | (4) | (5) | (6) | (7) | (8) | (9) | (10) |
|---|---|---|---|---|---|---|---|---|
| f | 210 | 210 | 210 | 210 | 210 | 0 | 0 | 0 |
| $\tilde{c}_1$ | | 210 | | 210 | | 0 | | 0 |
| $\tilde{d}_1$ | 0 | | 0 | | 0 | | 0 | |
| clipped $\hat{d}_1$, Q=16 | 0 | | 0 | | -15 | | 0 | |
| s' (Q=16) | 210 | 210 | 212 | 214 | 199 | 0 | 0 | 0 |
| $\hat{d}_1$, Q=256 | 0 | | 0 | | -140 | | 0 | |
| s' (Q=256) | 210 | 210 | 227 | 245 | 105 | 0 | 0 | 0 |

With Eq. (2), $\hat{d}_1(7)$ results as $\hat{d}_1(7)$=-140. For Q=16, since $q(\tilde{d}_1(7))=0$, $\hat{d}_1(7)$ is constrained to the range -15…15, so -140 is not a permitted value and has to be clipped into the quantization range. Good options for a permitted value is the midpoint reconstruction 0 or to clip -140 to the valid range, resulting in -15. If 0 is chosen, the step edge is recovered exactly for this example. Choosing to clip, which gives the smoothest valid reconstruction gives still a good representation of a step edge for Q=256. Observe that for both the full-frame and the detiled reconstructions s'(7)-s'(8)=105. Figure 2 shows how edges are preserved for a checkerboard image where edges marked with "t" (only) coincide with tile boundaries.




with detiling solution close-up

**Figure 2** Detiling at sharp edges (Q:1 | 4 8 16).

tile boundary artifacts with detiling solution

**Figure 3** Part of "bike" image at compression ratio 60:1.

## 5 CONCLUSIONS

A detiling solution for wavelet compressed images is presented. Using Projections Onto Scaling Functions image quality characteristics inside of a tile, that are determined by the wavelet system, levels of resolution and quantization, are matched at the tile boundaries. The power of not requiring any image model is illustrated in synthetic examples and a J2K test image (Figure 3, Q: 16 | 32 64 128). Explicit details on how to apply POSF to J2K compressed images were described.

## BIBLIOGRAPHY


[1] W. B. Pennebaker and J. L. Mitchell, *JPEG - Still Image and Data Compression Standard*. Van Nostrand Reinhold, New York, 1993.

[2] Y. Yang, N.P. Galatsanos, A.K. Katsaggelos, "Projection-Based Spatially Adaptive Reconstruction of Block-Transform Compressed Images," IEEE Trans. Image Proc., pp. 896-908, 1995.

[3] A. Nosratinia, "Embedded Post-Processing for Enhancement of Compressed Images," Proceedings of DCC, pp. 62-71, 1999.

[4] E.L. Schwartz, K. Berkner, M.J. Gormish, "Optimal tile boundary artifact removal with CREW," Proc. of Picture Coding Symposium, Portland, OR, pp. 285-288, 1999.

[5] JPEG 2000 Image Coding Standard, Part I, II, IV, and Amendment 1 of Part I, http://www.JPEG.org/CDs15444.htm.