Incremental Refinement of Computation for the Discrete Wavelet Transform

Yiannis Andreopoulos, *Member, IEEE*, and Mihaela van der Schaar, *Senior Member, IEEE*

***Abstract*—Contrary to the conventional paradigm of transform decomposition followed by quantization, we investigate the computation of two-dimensional (2-D) discrete wavelet transforms (DWT) under quantized representations of the input source. The proposed method builds upon previous research on approximate signal processing and revisits the concept of *incremental refinement of computation*: Under a refinement of the source description (with the use of an embedded quantizer), the computation of the forward and inverse transform refines the previously computed result, thereby leading to incremental computation of the output. In the first part of this paper, we study both the forward and inverse DWT under state-of-the-art 2-D lifting-based formulations. By focusing on conventional bitplane-based (double-deadzone) embedded quantization, we propose schemes that achieve incremental refinement of computation for the multilevel DWT decomposition or reconstruction based on a bitplane-by-bitplane calculation approach. In the second part, based on stochastic modeling of typical 2-D DWT coefficients, we derive an analytical model to estimate the arithmetic complexity of the proposed incremental refinement of computation. The model is parameterized with respect to i) operational settings, such as the total number of decomposition levels and the terminating bitplane; ii) input source and algorithm-related settings, e.g., the source variance, the complexity related to the choice of wavelet, etc. Based on the derived formulations, we study for which subsets of these model parameters the proposed framework derives identical reconstruction accuracy to the conventional approach without any incurring computational overhead. This is termed *successive refinement of computation*, since all representation accuracies are produced incrementally under a single (continuous) computation of the refined input source with no overhead in comparison to the conventional calculation approach that specifically targets each accuracy level and is not refinable. Our results, as well as the derived model estimates for incremental refinement, are validated with real video sequences compressed with a scalable coder.**

***Index Terms*—Approximate signal processing, computational complexity, discrete wavelet transform, incremental refinement of computation.**

Manuscript received October 25, 2006; revised May 9, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ilya Pollak. This work was supported by the NSF: CCF 0541867 (Career Award), CCF 0541453, and CNS 0509522. A portion of this paper was presented at the 2007 IEEE International Conference on Image Processing (ICIP).

Y. Andreopoulos is with the Department of Electronic Engineering, Queen Mary University of London, London E1 4NS, U.K. (e-mail: yiannis.a@elec.qmul.ac.uk).

M. van der Schaar is with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: mihaela@ee.ucla.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2007.906727

I. Introduction

A common use of multidimensional transform representations is in decorrelating input data and aiding the processing or compression of multimedia information, such as images or video sequences. For this purpose, conventional approaches used in all mainstream applications or algorithms based on industry standards for image and video coding [1], [2] produce the transform representation to a maximum degree of precision needed and then quantize and process (or compress) the transform coefficients. Even though this approach follows the conceptual design of such systems in a straightforward manner, it wastes system resources in many cases where lossy representations of the input are required, e.g., at medium to low-bitrate coding, as noted by several authors [3]–[7]. Notice that new adaptive transforms for images [8] or improved motion estimation schemes for video [9] tend to produce very sparse representations that are coarsely quantized for medium and low-rate coding applications. Hence valuable resources are wasted in transforming sparse matrices of residual data. This realization led to schemes for computational optimization of transforms [7], [10]–[12], or approximate signal processing schemes [13], where the computational resources increase monotonically under increased precision requirements [13]. These studies focused mainly on the mainstream case of the discrete cosine transform (DCT) [7], [10], [11], but similar studies can be envisaged for other popular transforms [4] such as the discrete wavelet transform (DWT) or the Karhunen–Loève transform.

There are two main drawbacks for these cases. First, as noted by Lengwehasatit and Ortega [6] and Winograd and Nawab [5], it is hard to precisely estimate in advance the exact required quantization precision for the coding or processing of a certain source. Although there have been theoretical and *ad hoc* proposals for this purpose [6], [12], [15], the authors typically introduce rate-distortion tradeoffs in the problem and do not achieve the same representation accuracy as the original (albeit wasteful) computation of the transform. Second, even though frameworks that are optimized to predict and avoid computations in sparse (or coarsely quantized) areas of the transform representation do achieve a scalable increase in complexity with increased quality requirements, there is no incremental refinement of the computation from one operational point to the next. From a communication and coding perspective, a practical example is the popular case of embedded decoding of images or video [14]: even though an embedded coder may provide a refinement of the source information, this refinement cannot be used if the receiver has initiated the computation of the inverse transform, or, alternatively, the computation has to be restarted. Perhaps more importantly, from a systems perspective, even though recent advances in progressive image sensors [16] and also progressive (bitplane-based) image displays [17] could enable progressive sensing, processing, and display of visual data, current transform computation schemes do not allow for this functionality. In general, current schemes produce an "all or nothing" representation: the computation cannot be interrupted when resources become unavailable and retrieve a meaningful approximation of the final result.

One exception to this rule is the work of Nawab, Winograd, and Dorken [5], [15], [18], [19] on incremental refinement of discrete Fourier transforms (DFT), where they display a first realization of incremental refinement of computation. However, apart from the DFT, their work did not attempt to extend these schemes to other transforms,[1] such as the DWT, where one faces different challenges. In addition, no source-dependent closed-form model was proposed for the complexity analysis of the derived schemes in order to examine theoretically the properties of sources and algorithms that enable incremental refinement in an efficient manner.

We propose a novel approach that tackles these issues. Although we focus on the case of the DWT, as it is a popular choice for number of applications such as compression [1], [14], denoising, or edge detection [20], similar approaches can be derived for other multiscale 2-D transforms. The paper makes the following contributions.

- We propose a bitplane-based computation for the 2-D DWT and analyze its design for the multilevel decomposition and reconstruction. Two schemes are proposed, the first being the straightforward approach that computes all the required bitplanes of each decomposition level incrementally before moving to the next level ("quality-first"); the second approach performs a "frequency-first" computation, where all the intermediate results of all decomposition levels (i.e., all frequency bands) are computed for each input bitplane, prior to the processing of the next bitplane.
- In order to analyze the results, a stochastic modeling framework of the input source data is used. By focusing on the inverse DWT (where resource optimization is of paramount importance and lossy representations of the input transform coefficients are typically provided) we propose an analytical framework for the estimation of the required computations of the proposed incremental refinement approach. We use the derived model to establish the minimum bitplane above which the proposed scheme requires the identical or less amount of computation in comparison to the conventional approach.
- The derived incremental refinement of computation for the DWT is tested with real video data produced by a motion-compensated wavelet-based video coder in order to: i) practically validate the feasibility and the benefits of incremental refinement of computation for real-world multimedia data; ii) assess the accuracy of the proposed modeling framework for the estimation of complexity in incremental refinement; iii) establish the potential of the proposed approach for real-world media systems.

Concerning the system-related aspects of our work, we remark that there has been a flurry of research on novel sampling and capturing devices that merge successive approximation-based analog-to-digital converters with image sensors at the pixel or sample level [16], [21]. This enables the sample-based [21], or bitplane-based [16] capturing of the input multimedia data. Very recent results demonstrated that image displays enabling the incremental refinement of a large number of luminance shades without flicker are also possible [17]. This enables the incrementally produced output to be directly consumed by the display monitor. These novel developments in circuit theory and design seem very promising in solving the capturing and display aspects for systems that process the input data incrementally.

This paper is organized as follows. Section II presents the proposed framework for incremental refinement of the DWT. Section III discusses the proposed modeling for the estimation of computational resources required by the new approach and performs a theoretical comparison to the conventional approach. Section IV presents results for the proposed framework as well as validation of the proposed model for resource estimation and, finally, Section V concludes the paper. A short version of this work without the theoretical analysis appears in our corresponding conference paper [22].

II. Lifting-Based DWT Under Incremental Refinement of Computation

A. *The Discrete Wavelet Transform—Direct 2-D Computation Using the Lifting Scheme*

The fast discrete wavelet transform was initially proposed as a critically sampled time-frequency representation realized by filtering the C-sample input signal $\mathbf{x} = [x[0] \ \cdots \ x[C-1]]^T$ with a T_{low}-tap low-pass filter $\mathbf{h} = [h[T_{\text{low}}-1] \ \cdots \ h[0]]$ and a T_{high}-tap high-pass filter $\mathbf{g} = [g[T_{\text{high}}-1] \ \cdots \ h[0]]$ followed by downsampling by two [20]. Based on the first Noble identity, the signal analysis can be expressed in the time domain as $\mathbf{s} = \mathbf{E} \cdot \mathbf{x}$, where $\mathbf{E}$ is the $C \times C$ analysis matrix. Assuming (for illustration purposes) that $\mathbf{h}$ and $\mathbf{g}$ have equal maximum degrees in the Z domain, the analysis matrix is (block Toeplitz form)[2] [see (1) at the bottom of the next page]. The output of the decomposition $\mathbf{s}$ is

$$\text{s} = [x_{\text{low}}[0] \quad x_{\text{high}}[0] \quad x_{\text{low}}[1] \quad x_{\text{high}}[1] \quad \cdots \quad x_{\text{low}}[C/2-1] \quad x_{\text{high}}[C/2-1]]$$

consisting of alternating low- and high-frequency coefficients $x_{\text{low}}[i]$ and $x_{\text{high}}[i]$, $0 \le i < C/2$. The decomposition can it-

[1] Even though Winograd's Ph.D. thesis contains an example of incremental refinement for the DCT [19], this is based on a rather straightforward summation of precomputed results using lookup tables. This requires a significant amount of additional memory and cannot be generalized to global transforms such as the DWT since the memory requirements grow exponentially with the transform length [19].

[2] In order to treat the signal borders, the border coefficients of $\mathbf{E}$ are modified to perform symmetric extension, as appropriate.

erate on low-frequency coefficients $x_{\mathrm{low}}[i]$ of $\mathbf{s}$ thereby forming a multiresolution transform decomposition of the input signal $\mathbf{x}$ [20].

Daubechies and Sweldens demonstrate in their tutorial paper [23] that any analysis matrix $\mathbf{E}$ allowing for perfect reconstruction of $\mathbf{x}$ from $\mathbf{s}$ (i.e., $\mathbf{E}$ has to be invertible) can be factorized in a succession of liftings (updates) and dual-liftings (predictions), up to shifting and multiplicative constants

$$\mathbf{E} = \boldsymbol{\Upsilon} \cdot \prod_{k=1}^{K} \mathbf{U}_k \mathbf{P}_k \tag{2}$$

where matrices $\mathbf{P}_k$, $\mathbf{U}_k$ are breaking the realization of the high-pass and low-pass filters, respectively, into a series of lifting steps. The factorization of (2) is scaled by the diagonal matrix $\boldsymbol{\Upsilon} = \mathrm{diag}(\Upsilon\ 1/\Upsilon\ \cdots\ \Upsilon\ 1/\Upsilon)$ with alternating scaling factors Υ and $1/\Upsilon$. The conceptual explanation of the factoring algorithm is based on the notion that a high-pass filter "predicts" its current input signal value from its neighbouring context, while a low-pass filter "updates" its current input signal value based on the prediction error [23]. The respective lifting steps are expressed as the progressively shifted row of filter coefficients alternating with one row consisting of zeros and the unity sample, which propagates the output of the previous step [see (3) at the bottom of the page and (4) at the bottom of the next page] with $a_{\mathrm{p}(k)}[l]$, $a_{\mathrm{u}(k)}[l]$ the lth nonzero (and nonunity) predict and update filter coefficients respectively out of $T_{\mathrm{p}(k)} + 1$, $T_{\mathrm{u}(k)} + 1$ nonzero taps, and $P_{\mathrm{p}(k)}$, $P_{\mathrm{u}(k)}$ the respective maximum degrees in the Z-domain for the predict and update filters. Since the predict and update filters are applied to the even and odd polyphase components of the input, $P_{\mathrm{p}(k)}$ is an even number (or zero) and $P_{\mathrm{u}(k)}$ is an odd number. A unity tap is always placed at the position of the "current" input sample[3] in order to enable "prediction" or "update" of the sample based on its context. The corresponding signal flow graph is given in Fig. 1.

As an example, for the 9/7 filter-pair we have: $K = 2$, $T_{\mathrm{p}(k)} = T_{\mathrm{u}(k)} = 2$, and: $P_{\mathrm{p}(k)} = 0$, $P_{\mathrm{u}(k)} = 1$, $a_{\mathrm{p}(k)}[0] = a_{\mathrm{p}(k)}[1]$, $a_{\mathrm{u}(k)}[0] =, a_{\mathrm{u}(k)}[1]$ for each $k = \{1, 2\}$. For the 5/3 filter-pair we have the same settings but with $K = 1$ (one predict and one update step). Since these filter-pairs will be used in our experiments, their parameters are given in Table I. We provide the filter coefficients in a representation suitable for fixed point computations.

Since the factorization algorithm is based on polynomial division [23], there are many possible factorizations that allow for the realization of a given filter-bank with the lifting scheme.

[3] In our analysis, the "current" input sample is defined as the filter position corresponding to zero degree in the Z-domain representation of the update filter and to -1 degree in the Z-domain representation of the predict filter (because we assume that the high-pass filter is applied to the odd polyphase components of the input signal).

$$\mathbf{E} = \begin{bmatrix} \ddots & & & \ddots & & & \ddots & & & \ddots \\ \cdots & h[0] & h[1] & h[2] & \cdots & h[T_{\mathrm{low}}-2] & h[T_{\mathrm{low}}-1] & 0 & 0 & \cdots \\ \cdots & g[0] & \cdots & g[T_{\mathrm{high}}-1] & 0 & \cdots & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 0 & h[0] & h[1] & h[2] & \cdots & h[T_{\mathrm{low}}-2] & h[T_{\mathrm{low}}-1] & \cdots \\ \cdots & 0 & 0 & g[0] & \cdots & g[T_{\mathrm{high}}-1] & 0 & \cdots & 0 & \cdots \\ \ddots & & & \ddots & & & \ddots & & & \ddots \end{bmatrix}. \tag{1}$$

$$\mathbf{P}_k = \left[\begin{matrix} \ddots & & & & \ddots & & & & \ddots & \\ \cdots & 0 & 0 & \cdots & 0 & 0 & 1 & 0 & & 0 \\ \cdots & a_{\mathrm{p}(k)}[0] & 0 & \cdots & a_{\mathrm{p}(k)}\left[\frac{P_{\mathrm{p}(k)}}{2}-1\right] & 0 & a_{\mathrm{p}(k)}\left[\frac{P_{\mathrm{p}(k)}}{2}\right] & 1 & a_{\mathrm{p}(k)}\left[\frac{P_{\mathrm{p}(k)}}{2}+1\right] & 0 \\ \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\ \cdots & 0 & 0 & a_{\mathrm{p}(k)}[0] & 0 & \cdots & a_{\mathrm{p}(k)}\left[\frac{P_{\mathrm{p}(k)}}{2}-1\right] & 0 & a_{\mathrm{p}(k)}\left[\frac{P_{\mathrm{p}(k)}}{2}\right] & 1 \\ \ddots & & & & \ddots & & & & \ddots & \end{matrix} \right.$$

$$\left. \begin{matrix} & & \ddots & & & & \ddots \\ 0 & \cdots & 0 & 0 & 0 & 0 & \cdots \\ a_{\mathrm{p}(k)}\left[\frac{P_{\mathrm{p}(k)}}{2}+2\right] & \cdots & 0 & a_{\mathrm{p}(k)}\left[T_{\mathrm{p}(k)}-1\right] & 0 & 0 & \cdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots \\ a_{\mathrm{p}(k)}\left[\frac{P_{\mathrm{p}(k)}}{2}+1\right] & 0 & a_{\mathrm{p}(k)}\left[\frac{P_{\mathrm{p}(k)}}{2}+2\right] & \cdots & 0 & a_{\mathrm{p}(k)}\left[T_{\mathrm{p}(k)}-1\right] & \cdots \\ & & \ddots & & & & \ddots \end{matrix} \right] \tag{3}$$

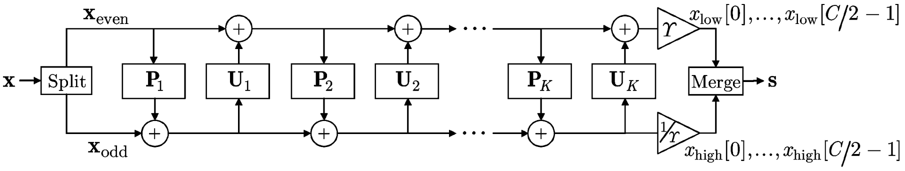

$\mathbf{x}_{\text{even}}$
$\mathbf{x}$
Split
$\mathbf{P}_1$
$\mathbf{U}_1$
$\mathbf{P}_2$
$\mathbf{U}_2$
$\mathbf{P}_K$
$\mathbf{U}_K$
Υ
$1/\Upsilon$
Merge
$\mathbf{S}$
$\mathbf{x}_{\text{odd}}$
$x_{\text{low}}[0],\ldots,x_{\text{low}}[C/2-1]$
$x_{\text{high}}[0],\ldots,x_{\text{high}}[C/2-1]$

Fig. 1. Signal flow graph for the lifting decomposition. Reconstruction is performed symmetrically by switching the split and merge steps and the scaling factors, traversing the graph from right to left (reversed signal flow) and flipping the signs of the predict and update coefficients.

TABLE I
FILTER COEFFICIENTS FOR THE PREDICTION AND UPDATE FILTERS OF THE 9/7 AND 5/3 FILTER-PAIRS. EACH COEFFICIENT IS REPRESENTED AS $a_{\text{step}(k)}[l] = a^{\text{quant}}_{\text{step}(k)}[l] \cdot 2^{-s_{\text{step}(k)}[l]}$. DUE TO THE FILTERS' SYMMETRY, $a_{\text{p}(k)}[0] = a_{\text{p}(k)}[1]$ AND $a_{\text{u}(k)}[0] = a_{\text{u}(k)}[1]$

Integer representation of lifting coefficients	9/7 filter-pair, $l = \{0,1\}$		5/3 filter-pair, $l = \{0,1\}$	
	$a^{\text{quant}}_{\text{step}(k)}[0]$	$s_{\text{step}(k)}[l]$	$a^{\text{quant}}_{\text{step}(k)}[l]$	$s_{\text{step}(k)}[l]$
1st predict step: $\text{step}(1) = \text{p}(1)$	-406	8	-8192	14
1st update step: $\text{step}(1) = \text{u}(1)$	-434	13	8192	14
2nd predict step: $\text{step}(2) = \text{p}(2)$	226	8	–	–
2nd update step: $\text{step}(2) = \text{u}(2)$	3633	13	–	–

In addition, even though the factorization presented in (2) starts with a prediction matrix, the order of steps ($\mathbf{P}_k$ and $\mathbf{U}_k$) can be interchanged. In the remainder of the paper, without loss of generality, we focus on factorizations beginning with a prediction step. The lifting scheme combines many advantages, such as low implementation complexity by reducing the required number of multiplications and additions, integer-to-integer transforms, and complete reversibility even if nonlinear or adaptive filters are used in the prediction and update steps.

Since the DWT is a separable transform, it is conventionally implemented in two dimensions via rowwise filtering followed by columnwise filtering of the intermediate results. However, this approach is not suitable for incremental refinement structures, as incremental refinement of the intermediate results does not represent a successive approximation of the final transform coefficients [19]. Hence, we need to focus on the direct 2-D computation of the DWT. Consider the 2-D DWT decomposition via K 2-D "predict" and "update" lifting steps performed for a 2-D image block $\mathbf{X}$ consisting of $R \times C$ pixels:

$$\mathbf{S} = (\boldsymbol{\Upsilon} \cdot \mathbf{U}_K \cdot \mathbf{P}_K \cdots \mathbf{U}_1 \cdot \mathbf{P}_1) \cdot \mathbf{X} \cdot (\boldsymbol{\Upsilon} \cdot \mathbf{U}_K \cdot \mathbf{P}_K \cdots \mathbf{U}_1 \cdot \mathbf{P}_1)^T \tag{5}$$

where $\mathbf{P}_k$, $\mathbf{U}_k$ and $\boldsymbol{\Upsilon}$ were defined previously, and $\mathbf{S}$ the 2-D array of output wavelet coefficients. Meng and Wang [24] proposed a reformulation in hierarchical decompositions, which, in its generalized form, is

$$\mathbf{S} = \boldsymbol{\Upsilon}\big(\mathbf{U}_K\left(\mathbf{P}_K(\cdots(\mathbf{U}_1(\mathbf{P}_1 \cdot \mathbf{X} \cdot \mathbf{P}_1^T)\mathbf{U}_1^T)\cdots)\mathbf{P}_K^T\right)\mathbf{U}_K^T\big)\boldsymbol{\Upsilon}^T. \tag{6}$$

The last equation is computing the 2-D DWT as a series of basic computation steps in two dimensions, starting from the

$$\mathbf{U}_k = \left[\begin{matrix} \ddots & & & & \ddots & & & & \ddots & \\ \cdots & a_{\text{u}(k)}[0] & 0 & \cdots & a_{\text{u}(k)}\left[\frac{P_{\text{u}(k)}+1}{2}-2\right] & 0 & a_{\text{u}(k)}\left[\frac{P_{\text{u}(k)}+1}{2}-1\right] & 1 & a_{\text{u}(k)}\left[\frac{P_{\text{u}(k)}+1}{2}\right] & 0 \\ \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 & 0 \\ \cdots & 0 & 0 & a_{\text{u}(k)}[0] & 0 & \cdots & a_{\text{u}(k)}\left[\frac{P_{\text{u}(k)}+1}{2}-2\right] & 0 & a_{\text{u}(k)}\left[\frac{P_{\text{u}(k)}+1}{2}-1\right] & 1 \\ \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ \ddots & & & & \ddots & & & & \ddots & \end{matrix}\right.$$

$$\left.\begin{matrix} & & \ddots & & & & \ddots \\ a_{\text{u}(k)}\left[\frac{P_{\text{u}(k)}+1}{2}+1\right] & \cdots & 0 & a_{\text{u}(k)}\left[T_{\text{u}(k)}-1\right] & 0 & 0 & \cdots \\ 0 & \cdots & 0 & 0 & 0 & 0 & \cdots \\ a_{\text{u}(k)}\left[\frac{P_{\text{u}(k)}+1}{2}\right] & 0 & a_{\text{u}(k)}\left[\frac{P_{\text{u}(k)}+1}{2}+1\right] & \cdots & 0 & a_{\text{u}(k)}\left[T_{\text{u}(k)}-1\right] & \cdots \\ 1 & 0 & 0 & \cdots & 0 & 0 & \cdots \\ & & \ddots & & & & \ddots \end{matrix}\right] \tag{4}$$

1	2	1	2	1	2	1	2
4	3	4	3	4	3	4	3
1	2	1	2	1	2	1	2
4	3	4	3	4	3	4	3
1	2	1	2	1	2	1	2
4	3	4	3	4	3	4	3
1	2	1	2	1	2	1	2
4	3	4	3	4	3	4	3

$\mathbf{P}_k$

3	4	3	4	3	4	3	4
2	1	2	1	2	1	2	1
3	4	3	4	3	4	3	4
2	1	2	1	2	1	2	1
3	4	3	4	3	4	3	4
2	1	2	1	2	1	2	1
3	4	3	4	3	4	3	4
2	1	2	1	2	1	2	1

$\mathbf{U}_k$

Fig. 2. Sequence of steps in the 2-D polyphase components of the input of each predict and update step [24].

inner part of (6), i.e., $\mathbf{P}_1 \cdot \mathbf{X} \cdot \mathbf{P}_1^T$, and working outwards to the final result $\mathbf{S}$. The computation is performed in squares of 2 × 2 input samples (2-D polyphase components). The computation order of predict and update steps of (3) and (4) can be seen in Fig. 2. The sequence of steps shown in the figure originates from the matrix row containing only the unity tap (placed at the position of the "current" input column sample) because this requires a simple assignment operation. It then proceeds clockwise to complete the computation for each square of 2 × 2 input samples. The analysis of this paper fits the sequence of steps seen in Fig. 2.

Based on this calculation and the predict matrix defined in (3), the prediction part $\mathbf{M}_k^{\mathrm{p}} = \mathbf{P}_k \cdot \mathbf{M}_{k-1}^{\mathrm{u}} \cdot \mathbf{P}_k^T$ of the kth lifting step of (6), where $\mathbf{M}_{k-1}^{\mathrm{u}}$ is the previously computed result (with $\mathbf{M}_0^{\mathrm{u}} \equiv \mathbf{X}$), can be written for all output coefficients $p_k[2i + \{0,1\}, 2j + \{0,1\}]$ $(0 \le i < R/2, 0 \le j < C/2)$ of $\mathbf{M}_k^{\mathrm{p}}$ as

$$\begin{aligned} p_k[2i,2j] & \\ &= u_{k-1}[2i,2j] \end{aligned} \tag{7}$$

$$\begin{aligned} p_k[2i,2j+1] & \\ &= u_{k-1}[2i,2j+1] + \sum_{l=0}^{T_{\mathrm{p}(k)}-1} a_{\mathrm{p}(k)}[l] \cdot \\ &\quad u_{k-1}\left[2i, 2(j+l) - P_{\mathrm{p}(k)}\right] \end{aligned} \tag{8}$$

$$\begin{aligned} p_k[2i+1,2j+1] & \\ &= u_{k-1}[2i+1,2j+1] + \sum_{l=0}^{T_{\mathrm{p}(k)}-1} a_{\mathrm{p}(k)}[l] \\ &\quad \times \big(u_{k-1}\left[2i+1, 2(j+l) - P_{\mathrm{p}(k)}\right] \\ &\qquad + p_k\left[2(i+l) - P_{\mathrm{p}(k)}, 2j+1\right]\big) \end{aligned} \tag{9}$$

$$\begin{aligned} p_k[2i+1,2j] & \\ &= u_{k-1}[2i+1,2j] + \sum_{l=0}^{T_{\mathrm{p}(k)}-1} a_{\mathrm{p}(k)}[l] \cdot \\ &\quad u_{k-1}\left[2(i+l) - P_{\mathrm{p}(k)}, 2j\right] \end{aligned} \tag{10}$$

where $u_{k-1}[r,c]$ are the input coefficients of matrix $\mathbf{M}_{k-1}^{\mathrm{u}}$ $(0 \le r < R, 0 \le c < C)$, with $u_0[r,c] \equiv x[r,c]$. Each of the above calculations in (7)–(10) is performed in the order presented (and illustrated in the left side of Fig. 2) and for all coefficients of the output, before initiating the calculation of the subsequent step. It is important to remark that this is the case for all subsequent calculations illustrated in this section. For the kth update step, i.e., $\mathbf{M}_k^{\mathrm{u}} = \mathbf{U}_k \cdot \mathbf{M}_k^{\mathrm{p}} \cdot \mathbf{U}_k^T$, we have

$$\begin{aligned} u_k[2i+1,2j+1] & \\ &= p_k[2i+1,2j+1] \end{aligned} \tag{11}$$

$$\begin{aligned} u_k[2i+1,2j] & \\ &= p_k[2i+1,2j] + \sum_{l=0}^{T_{\mathrm{u}(k)}-1} a_{\mathrm{u}(k)}[l] \\ &\quad p_k\left[2i+1, 2(j+l) - P_{\mathrm{u}(k)}\right] \end{aligned} \tag{12}$$

$$\begin{aligned} u_k[2i,2j] & \\ &= p_k[2i,2j] + \sum_{l=0}^{T_{\mathrm{u}(k)}-1} a_{\mathrm{u}(k)}[l] \\ &\quad \times \big(p_k\left[2i, 2(j+l) - P_{\mathrm{u}(k)}\right] \\ &\qquad + u_k\left[2(i+l) - P_{\mathrm{u}(k)}, 2j\right]\big) \end{aligned} \tag{13}$$

$$\begin{aligned} u_k[2i,2j+1] & \\ &= p_k[2i,2j+1] + \sum_{l=0}^{T_{\mathrm{u}(k)}-1} a_{\mathrm{u}(k)}[l] \\ &\quad p_k\left[2(i+l) - P_{\mathrm{u}(k)}, 2j+1\right]. \end{aligned} \tag{14}$$

Each of the calculations of (11)–(14) is performed in the order illustrated in the right side of Fig. 2.

B. Description of the Basic Algorithm for Incremental Refinement of Computation of the DWT

If we use embedded double-deadzone quantization of the input $\mathbf{X}$ with basic partition cell $\Delta = 1$ [1], each quantized coefficient $x^{\mathrm{quant}}[r,c]$ of the input $\mathbf{X}$ is represented by

$$x^{\mathrm{quant}}[r,c] = (-1)^{x^N[r,c]} \sum_{n=0}^{N-1} x^n[r,c] \cdot 2^n \tag{15}$$

with $x^n[r,c]$ the nth bit of quantized coefficient $x^{\mathrm{quant}}[r,c]$ (where $x^0[r,c]$ is the least-significant bit), and $x^N[r,c]$ is the sign bit. This is the popular case of successive approximation quantization (SAQ) [1], where, starting from the sign bit, each additional bitplane corresponds to increased precision in the approximation of $x[r,c]$.

For incremental refinement under the SAQ approximation of each input sample $x[r,c]$, the proposed computation of the first predict step for each bitplane n, $0 \le n < N$, is

$$\begin{aligned} p_1^{\mathrm{quant}}[2i,2j] & \\ &= (-1)^{x^N[2i,2j]} x^n[2i,2j] \end{aligned} \tag{16}$$

$$\begin{aligned} p_1^{\mathrm{quant}}[2i,2j+1] & \\ &= 2^n (-1)^{x^N[2i,2j+1]} x^n[2i,2j+1] \\ &\quad + \sum_{l=0}^{T_{\mathrm{p}(1)}-1} \Big(a_{\mathrm{p}(1)}^{\mathrm{quant}}[l] \cdot 2^n (-1)^{x^N\left[2i,2(j+l)-P_{\mathrm{p}(1)}\right]} \\ &\qquad x^n\left[2i, 2(j+l) - P_{\mathrm{p}(1)}\right]\Big) 2^{-s_{\mathrm{p}(1)}[l]} \end{aligned} \tag{17}$$

$$\begin{aligned}p_1^{\mathrm{quant}}&[2i+1,2j+1]\\ &= 2^n(-1)^{x^N[2i+1,2j+1]}x^n[2i+1,2j+1]\\ &\quad+\sum_{l=0}^{T_{\mathrm{p}(1)}-1}\Big[a_{\mathrm{p}(1)}^{\mathrm{quant}}[l]\\ &\qquad\times\Big(2^n(-1)^{x^N[2i+1,2(j+l)-P_{\mathrm{p}(1)}]}\\ &\qquad\quad x^n\left[2i+1,2(j+l)-P_{\mathrm{p}(1)}\right]\\ &\qquad\quad+p_1^{\mathrm{quant}}\left[2(i+l)-P_{\mathrm{p}(1)},\right.\\ &\qquad\quad\left.2j+1\right]\Big)\Big]\,2^{-s_{\mathrm{p}(1)}[l]}\end{aligned}\tag{18}$$

$$\begin{aligned}p_1^{\mathrm{quant}}&[2i+1,2j]\\ &= 2^n(-1)^{x^N[2i+1,2j]}x^n[2i+1,2j]\\ &\quad+\sum_{l=0}^{T_{\mathrm{p}(1)}-1}\Big(a_{\mathrm{p}(1)}^{\mathrm{quant}}[l]\cdot 2^n(-1)^{x^N[2(i+l)-P_{\mathrm{p}(1)},2j]}\\ &\qquad x^n\left[2(i+l)-P_{\mathrm{p}(1)},2j\right]\Big)\,2^{-s_{\mathrm{p}(1)}[l]}\end{aligned}\tag{19}$$

where $p_1^{\mathrm{quant}}[2i+\{0,1\},2j+\{0,1\}]$ is the output quadrant of coefficients, as computed from the signed quantized values (signed bitplanes) $(-1)^{x^N[r,c]}x^n[r,c]$ of the input. Equation (16) is a simple copy operation between input and output, while (17)–(19) predict the input coefficient bit of each case by an addition with a factor that depends on: i) the nth signed bit values of the coefficients in the spatial neighborhood of $(-1)^{x^N[2i+\{0,1\},2j+\{0,1\}]}x^n[2i+\{0,1\},2j+\{0,1\}]$; ii) the quantized representation of the lifting taps corresponding to matrix $\mathbf{A}_{\mathrm{p}(1)}$, which, for the case of the 9/7 filter-pair, can be found in Table I. All scalings with 2^w, $w\in\mathbb{Z}$, are implemented with bit-shifting operations and amount to negligible complexity in relation to multiplications or additions. The straightforward computation of (17)–(19) amounts to $4T_{\mathrm{p}(1)}N$ integer additions and $3(T_{\mathrm{p}(1)}/2)N$ multiplications for each set of four coefficients of $\mathbf{M}_1^{\mathrm{P}}$ (assuming that the filters of the prediction step are point symmetric, as in the case of the 9/7 filter-pair). Notice however that there are only a limited number of different inputs possible for the input coefficients of (17) and (19), and consequently only a limited number of distinct values can be produced. Similarly, for (18) that uses the previously computed values $p_1^{\mathrm{quant}}[2(i+l)-P_{\mathrm{p}}(1),2j+1]$, only a limited number of different inputs is possible and consequently only a limited number of distinct values can be produced. For the case of the 9/7 filter-bank for example, after a few straightforward calculations we find that only 15 input and output values are possible for (17) and (19), and only 99 possible input/output values are possible for (18). These values can be precomputed in advance and stored in a small lookup table with negligible memory cost. As a result, the entire calculation required for the first prediction step given by (16)–(19) is performed with lookup tables containing the precomputed result for each possible value of a coefficient $p_1^{\mathrm{quant}}[2i+\{0,1\},2j+\{0,1\}]$ of $\mathbf{M}_1^{\mathrm{P}}$. Once the possible values of both lookup tables have been precomputed, there is no further computational overhead for the calculation of (16)–(19).

The second matrix product $\mathbf{U}_1\cdot\mathbf{M}_1^{\mathrm{p}}\cdot\mathbf{U}_1^T$ is produced by reusing the results of (16)–(19)

$$\begin{aligned}u_1^{\mathrm{quant}}&[2i+1,2j+1]\\ &= p_1^{\mathrm{quant}}[2i+1,2j+1]\end{aligned}\tag{20}$$

$$\begin{aligned}u_1^{\mathrm{quant}}&[2i+1,2j]\\ &= p_1^{\mathrm{quant}}[2i+1,2j]\\ &\quad+\sum_{l=0}^{T_{\mathrm{u}(1)}-1}\left(a_{\mathrm{u}(1)}^{\mathrm{quant}}[l]\cdot p_1^{\mathrm{quant}}\left[2i+1,2(j+l)-P_{\mathrm{u}(1)}\right]\right)\\ &\quad\times 2^{-s_{\mathrm{u}(1)}[l]}\end{aligned}\tag{21}$$

$$\begin{aligned}u_1^{\mathrm{quant}}&[2i,2j]\\ &= p_1^{\mathrm{quant}}[2i,2j]\\ &\quad+\sum_{l=0}^{T_{\mathrm{u}(1)}-1}\Big[a_{\mathrm{u}(1)}^{\mathrm{quant}}[l]\left(p_1^{\mathrm{quant}}\left[2i,2(j+l)-P_{\mathrm{u}(1)}\right]\right.\\ &\qquad\qquad\left.+u_1^{\mathrm{quant}}\left[2(i+l)-P_{\mathrm{u}(1)},2j\right]\right)\Big]\\ &\quad\times 2^{-s_{\mathrm{u}(1)}[l]}\end{aligned}\tag{22}$$

$$\begin{aligned}u_1^{\mathrm{quant}}&[2i,2j+1]\\ &= p_1^{\mathrm{quant}}[2i,2j+1]\\ &\quad+\sum_{l=0}^{T_{\mathrm{u}(1)}-1}\left(a_{\mathrm{u}(1)}^{\mathrm{quant}}[l]\cdot p_1^{\mathrm{quant}}\left[2(i+l)-P_{\mathrm{u}(1)},2j+1\right]\right)\\ &\quad\times 2^{-s_{\mathrm{u}(1)}[l]}\end{aligned}\tag{23}$$

where $u_1^{\mathrm{quant}}[2i+\{0,1\},2j+\{0,1\}]$ is the output quadrant of coefficients, as computed for the output coefficients $p_1^{\mathrm{quant}}[r,c]$ of the prediction step of (16)–(19). Again, (20) is a simple copy operation, while for (21)–(23) the update filter coefficients $a_{\mathrm{u}(1)}^{\mathrm{quant}}[l]$ update the output (see Table I for an example instantiation of these coefficients). Notice that even though for (21)–(23) we could again use lookup tables to avoid computations since each $p_1^{\mathrm{quant}}[r,c]$ coefficient has a small range of possible values, the dynamic range grows larger with each matrix product. Hence, we do not follow this approach and instead perform the computations (additions, multiplications) directly. The remaining steps $k=2,\ldots,K$ to complete the single-level decomposition using the 2-D lifting formulation of (6) are performed as in (16)–(19) and (20)–(23) with the replacement of the input with the output of each previous step and the replacement of the coefficients $a_{\mathrm{p}(1)}^{\mathrm{quant}}[l]$, $a_{\mathrm{u}(1)}^{\mathrm{quant}}[l]$ by $a_{\mathrm{p}(k)}^{\mathrm{quant}}[l]$, $a_{\mathrm{u}(k)}^{\mathrm{quant}}[l]$, respectively. All steps can be performed in-place as in the conventional lifting decomposition, with the reuse of the memory for the $\mathbf{M}_1^{\mathrm{p}}$ and $\mathbf{M}_1^{\mathrm{u}}$ arrays. Once all the steps have been completed for the current decomposition level, the final computation is reordered in the output matrix $\mathbf{S}$ in binary-tree (“Mallat”) form [20] and the produced results of each bitplane are added to the results of the previous (more significant) bitplanes, if existing. Finally, the scaling performed at the end of each decomposition level in the conventional decomposition can be skipped altogether [25], or incorporated into the subsequent encoding or processing stage of each bitplane and as a result it is not explicitly considered in this paper.

C. Extension to Multiple Decomposition Levels and Transform Inversion

The multilevel extension of the derived bitwise computation can be performed in two ways. The straightforward approach performs a "quality-first" incremental refinement of computation: all the requested bitplanes of each decomposition level are performed as described before. Once completed, the same process occurs in the low-frequency subband of the decomposition for as many bitplanes as needed for the second level. The calculation continues in a similar fashion until all the necessary decomposition levels (defined as $L_{\max}$) are computed under the requested accuracy (number of bitplanes per level). The "quality-first" approach is straightforward and fits certain coding applications where the requested quantization precision is known *a priori*. However, it fails to provide incremental refinement of computation of the *final* multilevel DWT coefficients, since each level is computed to a requested precision before initiating the computation of the next level. Instead, this approach provides only band-limited incremental refinement, i.e., within each decomposition level of the transform.

For this reason, we propose a "frequency-first" incremental refinement of computation by continuing the bitwise lifting-scheme computations for all subsequent levels after the termination of one bitplane at the first level. This is achieved by reformulating the first prediction step of (16)–(19) for all levels beyond one as $(0 \le i < R/2^l, 0 \le j < C/2^l, 2 \le l \le L_{\max})$

$$p_1^{\text{quant}}[2i,2j] = u_K^{\text{quant}}[4i,4j] \tag{24}$$

$$\begin{aligned}p_1^{\text{quant}}[2i,2j+1] &= u_K^{\text{quant}}[4i,4j+2] \\ &\quad + \sum_{l=0}^{T_{\text{p}(1)}-1}\left(a_{\text{p}(1)}^{\text{quant}}[l]\cdot u_K^{\text{quant}}\left[4i,4(j+l)-2P_{\text{p}(1)}\right]\right) \\ &\quad \times 2^{-s_{\text{p}(1)}[l]}\end{aligned} \tag{25}$$

$$\begin{aligned}p_1^{\text{quant}}[2i+1,2j+1] &= u_K^{\text{quant}}[4i+2,4j+2] \\ &\quad + \sum_{l=0}^{T_{\text{p}(1)}-1}\Big[a_{\text{p}(1)}^{\text{quant}}[l] \\ &\qquad \times \big(u_K^{\text{quant}}\left[4i+4,4(j+l)-2P_{\text{p}(1)}\right] \\ &\qquad + p_1^{\text{quant}}\left[2(i+l)-P_{\text{p}(1)},2j+1\right]\big)\Big] \\ &\quad \times 2^{-s_{\text{p}(1)}[l]}\end{aligned} \tag{26}$$

$$\begin{aligned}p_1^{\text{quant}}[2i+1,2j] &= u_K^{\text{quant}}[4i+2,4j] \\ &\quad + \sum_{l=0}^{T_{\text{p}(1)}-1}\left(a_{\text{p}(1)}^{\text{quant}}[l]\cdot u_K^{\text{quant}}\left[4(i+l)-2P_{\text{p}(1)},4j\right]\right) \\ &\quad \times 2^{-s_{\text{p}(1)}[l]}\end{aligned} \tag{27}$$

where we use the outputs $\mathbf{M}_K^{\text{u}}$ of the last update step of the previous level and, importantly, at the end of the previous level we only perform the reordering and addition to the previous results for the high-frequency coefficients ($u_K^{\text{quant}}[2r,2c+1]$, $u_K^{\text{quant}}[2r+1,2c]$, $u_K^{\text{quant}}[2r+1,2c+1]$, $0 \le r < R/2^{l-1}$, $0 \le c < C/2^{l-1}$) and do the reordering of the low-frequency coefficients ($u_K^{\text{quant}}[2r,2c]$) only for the final (coarsest) decomposition level $L_{\max}$. This is due to the fact that each intermediate level receives the necessary low-frequency coefficients of the previous level with the change of indexing performed in (24)–(27) in comparison to (16)–(19).

Notice that, under the changed computation proposed in (24)–(27), the use of lookup tables becomes cumbersome for the first prediction step of all levels beyond one, as the inputs $u_K^{\text{quant}}[2r,2c]$ may have high dynamic range. Hence the computation is performed directly as shown in (24)–(27) without the use of lookup tables for these cases.

The precise algorithm steps for the "frequency-first" case, which is the multilevel incremental refinement of the DWT computation, are presented in pseudocode in Fig. 3.

Concerning the inverse transform, the process is exactly anti-symmetric as in the conventional lifting computation: all lifting steps are performed in reverse order by solving the forward bitwise lifting equations for the coefficients being predicted or updated during the forward transform. This means that all the steps presented in the pseudocode of Fig. 3 are executed in reverse order and from the coarsest level to the finest one. As an illustrative example, the inversion of (20)–(23) is

$$\begin{aligned}p_1^{\text{quant}}[2i,2j+1] &= u_1^{\text{quant}}[2i,2j+1] \\ &\quad - \sum_{l=0}^{T_{\text{u}(1)}-1}\left(a_{\text{u}(1)}^{\text{quant}}[l]p_1^{\text{quant}}\left[2(i+l)-P_{\text{u}(1)},2j+1\right]\right) \\ &\quad \times 2^{-s_{\text{u}(1)}[l]}.\end{aligned} \tag{28}$$

$$\begin{aligned}p_1^{\text{quant}}[2i,2j] &= u_1^{\text{quant}}[2i,2j] \\ &\quad - \sum_{l=0}^{T_{\text{u}(1)}-1}\Big[a_{\text{u}(1)}^{\text{quant}}[l]\big(p_1^{\text{quant}}\left[2i,2(j+l)-P_{\text{u}(1)}\right] \\ &\qquad + u_1^{\text{quant}}\left[2(i+l)-P_{\text{u}(1)},2j\right]\big)\Big] \\ &\quad \times 2^{-s_{\text{u}(1)}[l]}\end{aligned} \tag{29}$$

$$\begin{aligned}p_1^{\text{quant}}[2i+1,2j] &= u_1^{\text{quant}}[2i+1,2j] \\ &\quad - \sum_{l=0}^{T_{\text{u}(1)}-1}\left(a_{\text{u}(1)}^{\text{quant}}[l]p_1^{\text{quant}}\left[2i+1,2(j+l)-P_{\text{u}(1)}\right]\right) \\ &\quad \times 2^{-s_{\text{u}(1)}[l]}\end{aligned} \tag{30}$$

$$p_1^{\text{quant}}[2i+1,2j+1] = u_1^{\text{quant}}[2i+1,2j+1]. \tag{31}$$

The only difference in the anti-symmetry rule between forward and inverse transform occurs in the use of bitwise computation. In particular, in the forward transform we use the bitwise representation of the input image samples only in the first predict step of the first decomposition level ($(-1)^{x^N[r,c]}x^n[r,c]$ of

```
For each bitplane $n$, $n = \{N-1, N-2, \mathrm{K}, 0\}$
  For each decomposition level $l$, $l = \{1, \mathrm{K}, L_{\max}\}$
    For each lifting step $k$, $k = \{1, 2, \mathrm{K}, K\}$
      If $k = 1$ and $l = 1$        // 1st prediction step of the first level
        Apply each of the (16)-(19) - use of a precalculated lookup table for all possible combinations of inputs
      Else if $k = 1$               // 1st prediction step of an intermediate level
        Apply each of the (24)-(27)
      Else                          // kth prediction step of an intermediate level
        Apply each of the (16)-(19) with $\mathrm{p}(1)$ replaced by $\mathrm{p}(k)$ and each $(-1)^{x^N[r,c]} x^n[r,c]$ replaced by $u_{k-1}^{\text{quant}}[r,c]$, $0 \le r < R/2^l$, $0 \le c < C/2^l$
      If $k = 1$                    // 1st update step
        Apply each of the (20)-(23) for all coefficients $p_1^{\text{quant}}[r,c]$ (result of the 1st predict step of the current level), $0 \le r < R/2^l$, $0 \le c < C/2^l$
      Else                          // kth update step
        Apply each of the (20)-(23) with with $\mathrm{u}(1)$ replaced by $\mathrm{u}(k)$ and each coefficient $p_1^{\text{quant}}[r,c]$ replaced by $p_K^{\text{quant}}[r,c]$, $0 \le r < R/2^l$, $0 \le c < C/2^l$
    If $l < L_{\max}$               // reordering of an intermediate level
      Reorder the high-frequency coefficients $u_K^{\text{quant}}[2r, 2c+1]$, $u_K^{\text{quant}}[2r+1, 2c]$, $u_K^{\text{quant}}[2r+1, 2c+1]$, $0 \le r < R/2^l$, $0 \le c < C/2^l$ in binary-tree ("Mallat")form and add them to the results of the previously-computed bitplane
    Else                            // reordering of the last level
      Reorder all the coefficients $u_K^{\text{quant}}[r,c]$, $0 \le r < R/2^{l-1}$, $0 \le c < C/2^{l-1}$ in binary-tree ("Mallat") form and add them to the results of the previously-computed bitplane
```

Fig. 3. Pseudocode for "frequency-first" incremental refinement of computation of the DWT.

(16)–(19) with $0 \le n \le N-1$) and then the intermediately produced results are used as inputs. However, in the inverse transform the bitwise representation of the input wavelet coefficients is used for all the decomposition levels, since at each decomposition level we get new coefficients from the high-frequency bands, which are processed bitplane-by-bitplane under the proposed incremental refinement of computation.

III. Modeling of Incremental Refinement of Computation

The proposed algorithm for incremental refinement of computation of the DWT presents a data-adaptive computation where each nonzero input bit causes a certain amount of computation towards the completion of the transform decomposition. In order to understand the behavior of such approaches better, we analyze the expected performance using stochastic source models. Coupled with assumptions on the implementation complexity of each operation (addition or multiplication), we derive a parametric estimation of the expected complexity that depends on: i) the source statistics; ii) the operational parameters (number of decomposition levels, terminating bitplane); and iii) the algorithm-dependent parameters for the complexity estimation (expressing the filtering complexity). Based on these, we can investigate whether incremental refinement of the transform computation can achieve comparable or even superior performance to the conventional computation. The presentation of this section remains rather generic and extensions to other filters, other quantization settings, or even other transforms can be envisaged. Finally, we are currently focusing on modeling the inverse DWT (IDWT) operations since the IDWT is part of multimedia decoders that are the typical resource-constrained systems. For example, scalable coding may encode the entire transform representation incrementally (bitplane-by-bitplane), but it typically enables decoding at multiple qualities corresponding to various bitplanes in the transform domain.

The following subsection introduces the implementation complexity assumptions for the performance of the required arithmetic operations, while Section III-B introduces the utilized stochastic source models and the necessary derivations for our analytical approach. Finally Section III-C analyzes the performance of the proposed scheme as compared to the conventional computation of the IDWT.

A. *Implementation Complexity of Variable Bit-Width Arithmetic Operations*

In this paper, we are quantifying the benefits of conventional transform calculation versus incremental refinement in terms of the computational effort required to complete the decomposition or reconstruction task, whether it is for a single bitplane or for the entire set of bitplanes. In this respect, a metric used commonly in the literature is the required number of additions and multiplications. However, arithmetic operations in the proposed incremental refinement approaches deal with data with significantly reduced bitwidth in comparison to the conventional computation of the DWT that processes all bitplanes at once. To distinguish this effect, we propose the following metrics inspired by classic research work in the area-time complexity of binary multiplication and addition [26], [27].

Definition 1: Addition of two numbers represented with N_1 and N_2 bits, each having an additional bit as the sign bit as in (15), requires the following number of operations:

$$\mathrm{Cost}_{\mathrm{add}} = \begin{cases} \max\{N_1, N_2\}+1, & \text{if both numbers are nonzero} \\ 0, & \text{otherwise.} \end{cases} \tag{32}$$

■

Definition 2: Multiplication of two numbers represented with N_1 and N_2 bits, each having an additional bit as the sign bit as in (15), requires the following number of operations:

$$\mathrm{Cost}_{\mathrm{mult}} = \begin{cases} (\max\{N_1, N_2\}+1) \\ \quad \cdot (\min\{N_1, N_2\})^{1+\xi}, & \text{if both numbers are nonzero} \\ 0, & \text{otherwise.} \end{cases} \tag{33}$$

with $\xi \geq 0$ a system parameter indicating how "hard" is binary multiplication in comparison to binary addition. ■

These definitions can be intuitively viewed as follows: Assume a virtual processing element (PE) able to perform signed addition between two bits and the carry information. Addition is (maximally) requiring $\max\{N_1, N_2\} + 1$ activations of the PE for two numbers with N_1 and N_2 bits. Similarly, by viewing multiplication as cumulative additions, the number of activations of the PE is given by (33), with the system parameter ξ indicating the cost of accumulating the intermediate results. If any of the two operands is zero, no operations are required (apart from a minimal "zero detection" effort), since the result is trivial.

Notice that the aim of Definition 1 and Definition 2 is not in providing the exact arithmetic complexity relating to a particular processor or custom hardware (which depends on a number of system factors), but rather to parametrically characterize in a generic manner the relative importance of addition and multiplication under different bitwidths. This will enable generic but useful comparisons of different algorithms for the transform computation to indicate the potential of the proposed incremental refinement approaches.

Based on the complexity definitions presented here, in the following subsection we estimate the expected number of arithmetic operations based on stochastic source modeling.

B. Operations for Incremental Refinement of Computation of the DWT

1) Stochastic Source Model for Low-frequency Wavelet Coefficients: The rate-distortion characteristics of low-frequency wavelet coefficients x_{low} are typically modeled using the high-rate uniform quantization assumption [1] for independent Gaussian random variables X_{low} with mean μ_{low} and variance σ^2_{low}. In our recent work [30] we validated that the Gaussian i.i.d. assumption for low-frequency spatio–temporal subbands turns out to be accurate when one performs more than two spatial decomposition levels in the input image or video sequence, since in this case the correlation between neighboring coefficients in the low-frequency spatio–temporal subband is often small enough such that the variance of the noise between adjacent coefficients is close to the coefficient variances themselves. While low-frequency spatial (or spatio-temporal in the case of 3-D video coding [31]) subbands account for a large percentage of the image and video transform computation, the high-frequency subbands also contribute a significant amount to the overall complexity, since they represent the majority of the multilevel DWT decomposition. Thus, accurate modeling of the high-frequency subband statistics is also very important for precise modeling of the DWT operations.

2) Stochastic Source Model for High-frequency Wavelet Coefficients: Each high-frequency subband of resolution (decomposition) level l, $1 \leq l \leq L_{\max}$, consists of $R/2^l \times C/2^1$ wavelet coefficients $x_{\mathrm{high}}[i]$, $0 \leq i < R \cdot C \cdot 2^{-2l}$. These coefficients are modeled as a doubly stochastic process [28], [29], i.e., as conditionally independent zero mean Gaussian random variables $X_{\mathrm{high}}[i] \sim N(0, \theta_l)$ given their variances. The variances of the high-frequency wavelet coefficients of each level l are modeled by Θ_l, which is given by an exponential distribution[4]

$$\Theta_l \sim \mathrm{P}(\theta_l) = \frac{1}{\sigma_l^2} e^{-\frac{1}{\sigma_l^2}\theta_l}. \tag{34}$$

In this case, the marginal distribution of each high-frequency wavelet coefficients $x_{\mathrm{high}}[i]$ of decomposition level l turns out to be Laplacian [28]

$$\begin{aligned} \mathrm{P}_{\mathrm{marginal}}\left(x_{\mathrm{high}}[i]\right) &= \int_0^\infty \mathrm{P}\left(x_{\mathrm{high}}[i] | \theta_l\right) \mathrm{P}(\theta_l) d\theta_l \\ &= \int_0^\infty \frac{1}{\sqrt{2\pi\theta_l}} e^{-\frac{1}{2\theta_l}(x_{\mathrm{high}}[i])^2} \frac{1}{\sigma_l^2} e^{-\frac{1}{\sigma_l^2}\theta_l} d\theta_l \\ &= \frac{1}{\sqrt{2}\sigma_l} e^{-\frac{\sqrt{2}}{\sigma_l}|x_{\mathrm{high}}[i]|}. \end{aligned} \tag{35}$$

We organize coefficients $x_{\mathrm{high}}[i]$ into vector $x_{\mathrm{high}} = [x_{\mathrm{high}}[0] \cdots x_{\mathrm{high}}[R \cdot C \cdot 2^{-2l} - 1]]$. In addition, for each bitplane n, we denote the bitplane threshold level as $T_n = 2^n$.

3) Expected Number of Operations: In this subsection we focus on the inverse DWT and attempt to derive the expected number of operations for the transform realization via the proposed approach.

For each input bitplane n of each subband (which corresponds to threshold T_n), the wavelet coefficients that have a significant bit at bitplane n will correspond to multiplications and additions as presented in Section II. If we quantify this cost for each coefficient as $\mathrm{Cost}_{\mathrm{ops}} = \mathrm{Cost}_{\mathrm{mult}} + \mathrm{Cost}_{\mathrm{add}}$, where the cost for multiplications and additions is derived based on Definition 1 and Definition 2, the operations for the significant coefficients of bitplane n across $L_{\max}$ decomposition levels are

$$\begin{aligned} \mathrm{Cost}_{\mathrm{ops}}^{\mathrm{refine}}(L_{\max}, n) &= C_{\mathrm{lifting}}^{\mathrm{low}} \cdot \frac{R \cdot C}{2^{2L_{\max}}} \cdot \beta_{n,L_{\max}}^{\mathrm{low}} \cdot \mathrm{Cost}_{\mathrm{ops}} \\ &\quad + 3 \cdot C_{\mathrm{lifting}}^{\mathrm{high}} \sum_{l=1}^{L_{\max}} \left(\frac{R \cdot C}{2^{2l}} \cdot \beta_{n,l}^{\mathrm{high}} \cdot \mathrm{Cost}_{\mathrm{ops}} \right) \end{aligned} \tag{36}$$

[4] Random variables are indicated by the capitalized symbol of their corresponding quantities, while probability density functions are indicated by $\mathrm{P}(\cdot)$.

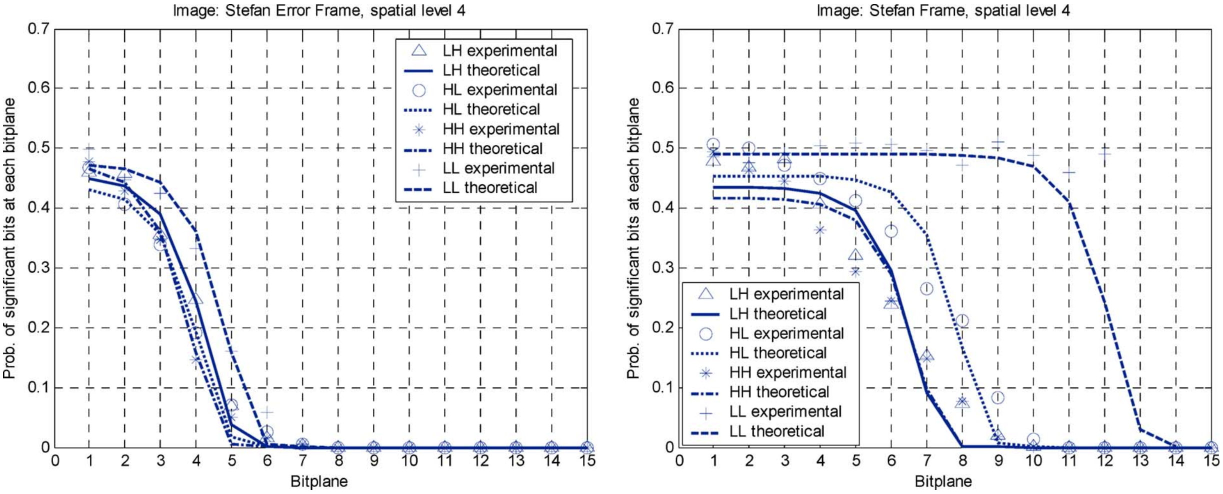


Fig. 4. Representative examples of probability of significance for different wavelet subbands of an error-frame (left) and an original video frame (right) from the video sequence "Stefan."

where $\beta_{n,l}^{\text{band}}$, $\text{band} = \{\text{low}, \text{high}\}$, indicates the probability of the wavelet coefficients of a low or high-frequency subband of level l having a nonzero bit at bitplane n and $C_{\text{lifting}}^{\text{band}}$ are scaling factors relating to the particular lifting decomposition of interest. These factors express the overhead corresponding to the size of the lifting filters and the number of lifting steps and they will be derived in the experimental section. Our goal in this subsection is to establish an estimate for the probabilities $\beta_{n,l}^{\text{band}}$ in order to derive the final complexity estimate of (36). This is achieved in the following two propositions.

Proposition 1: An approximation result to the probability that the wavelet coefficients of a high-frequency subband have a nonzero bit at bitplane n is

$$\beta_{n,l}^{\text{high}} \cong \exp\left\{\frac{-T_n^2}{12.5\sigma_l^2}\right\} - \left(\exp\left\{\frac{-T_n^2}{12.5\sigma_l^2}\right\}\right)^4 \tag{37}$$

and σ_l^2 the variance of the high-frequency subbands of decomposition level l.

Derivation: See Appendix I. ■

Proposition 2: An approximation result to the probability that the wavelet coefficients of a low-frequency subband have a nonzero bit at bitplane n is

$$\beta_{n,L_{\max}}^{\text{low}} \cong 0.490 \exp\left\{\frac{-T_n^2}{39590 \cdot 4^{L_{\max}}}\right\}. \tag{38}$$

Derivation: See Appendix II. ■

Notice that, even though we shall be using conventional dyadic thresholds corresponding to bitplane-based SAQ, both Proposition 1 and Proposition 2 can be formulated with any set of thresholds, thereby accommodating a broad family of embedded quantizers.

Typical examples of the probability of nonzero bits for several bitplanes of SAQ are presented in Fig. 4 for both regular images (video frames), as well as error images produced by motion compensated prediction. The model prediction based on Proposition 1 and Proposition 2 is presented as well, where we apply (37) and (38) individually for each subband, i.e., σ_l^2 is the variance of the doubly stochastic model (high-frequency coefficients) fitted to the experimentally measured variance of each subband. The figure demonstrates that the proposed model appears to predict rather accurately the probability of nonzero bits in each bitplane of the different wavelet subbands. Notice that the probability for nonzero bits of the low-frequency subband of an error frame (left side of Fig. 4) is modeled by (37) because it corresponds to a high-frequency spatio-temporal subband, since the error frame is produced by motion-compensation. In addition, for the example of the low-frequency subband of images presented in the right side of Fig. 4, the model of tends to predict accurately for the majority of coefficient bitplanes, with the exception of the maximum bitplane. Overall, the derived models of (37) and (38) are capturing the different trends of the various subbands in function of the source statistics. Notice from Fig. 4 that the probability of nonzero bits decreases faster in function of the bitplane for the case of error frames, indicating the sparsity of the representation.

The derived framework can be used as a theoretical basis to study the conditions under which incremental refinement of computation for the inverse DWT can achieve the same performance as the conventional calculation.

C. Theoretical Comparison of Incremental Refinement of Computation With Conventional Computation—Achieving Successive Refinement of the IDWT Computation

We begin this section by establishing the equivalent theoretical estimation of computational requirements for the conventional (nonrefinable) computation of the inverse DWT. This is achieved by the following propositions.

Proposition 3: An approximation result to the probability that the wavelet coefficients of a high-frequency subband have nonzero bits between bitplane $N-1$ (maximum bitplane) and n is

$$\chi_{n,l}^{\text{high}} \cong \exp\left\{-\frac{T_n^2}{12.5\sigma_l^2}\right\}. \tag{39}$$

Derivation: The result can be reached by the integration over each (doubly stochastic) component of vector x_{high} between $(-\infty, -T_n]$ and $[T_n, \infty)$, since, due to the probabilistic nature of our model, coefficients are not strictly upper bounded by $[-T_N, T_N]$. This integration can be equivalently expressed as the difference between integrating the entire distribution of each component of x_{high} and integrating within the interval $[-T_n, T_n]$, i.e.,

$$\chi_{n,l}^{\text{high}} = 1 - \int_{-T_n}^{T_n} \int_{-T_n}^{T_n} \cdots \int_{-T_n}^{T_n} \mathrm{P}(\mathbf{x}_{\text{high}}) d\mathbf{x}_{\text{high}}. \tag{40}$$

We derived an approximation of the integration in (40) given by (50) contained in Appendix I, and this leads to the final result. ■

Proposition 4: An approximation result to the probability that the wavelet coefficients of a low-frequency subband have significant bits between bitplane $N-1$ (maximum bitplane) and n is

$$\chi_{n,L_{\max}}^{\text{low}} \cong \exp\left\{ \frac{-T_n^2}{39590 \cdot 4^{L_{\max}}} \right\}. \tag{41}$$

Derivation: Based on the central limit theorem and the same numerical estimation process as in the derivation of Proposition 2, we derive the parameters of the final Gaussian distribution: $(\mu_{\text{siglow}}^{\text{limit}\prime})^2 \cong 0$ and $(\sigma_{\text{siglow}}^{\text{limit}\prime})^2 \cong 19795 \cdot 4^{L_{\max}}$. With proper scaling to normalize the result to the range of the observed measurements we derive the final result of (41). ■

Under the derived results of Proposition 3 and Proposition 4, we can formulate the complexity estimate for the case of the direct computation of all bitplanes $N-1, \ldots, n$ following similar steps as in the case of (36), thereby deriving

$$\begin{aligned}\text{Cost}_{\text{ops}}^{\text{direct}}(L_{\max}, n) = {} & C_{\text{lifting}}^{\text{low}} \cdot \frac{R \cdot C}{2^{2L_{\max}}} \cdot \chi_{n,L_{\max}}^{\text{low}} \cdot \text{Cost}'_{\text{ops}} \\ & + 3 \cdot C_{\text{lifting}}^{\text{high}} \sum_{l=1}^{L_{\max}} \left(\frac{R \cdot C}{2^{2l}} \cdot \chi_{n,l}^{\text{high}} \cdot \text{Cost}'_{\text{ops}} \right).\end{aligned} \tag{42}$$

Notice that, similar to (36) where we are considering only the bitplane-by-bitplane computation, $\text{Cost}'_{\text{ops}}$ represents the expected number of operations for each coefficient. Since the theoretical estimation of this cost in function of the relevant cost in (36) is difficult, we set $\text{Cost}'_{\text{ops}} = \lambda \cdot \text{Cost}_{\text{ops}}$, with $\lambda > 0$, and treat the scaling λ as a system parameter that should be taken into account in our further exploration. This scaling represents the average overhead of each expected computation expressed in (42) as compared to the incremental refinement approach expressed by (36), with $\lambda = 1$ representing the case of no overhead.

We can now attempt to answer the following interesting question. Under the given source model and starting from bitplane N, what is the lowest bitplane n_{SRC} until which the proposed incremental refinement approach performs equal (or less) computations to the conventional (nonincremental) approach? This is the subset of operational settings where both schemes would obtain the equivalent approximation of the input source, with the proposed approach possessing also the property of incremental refinement. We term this as *successive refinement of computation* following the analogous information-theoretic definition of Equitz and Cover [32]. This can be derived by the following inequality for the computation between (36) and (42)

$$\text{Cost}_{\text{ops}}^{\text{direct}}(L_{\max}, n_{\text{SRC}}) \geq \sum_{m=n_{\text{SRC}}}^{N-1} \text{Cost}_{\text{ops}}^{\text{refine}}(L_{\max}, m) \tag{43}$$

which, after a few straightforward manipulations becomes

$$\begin{aligned}& \lambda \cdot \chi_{n_{\text{SRC}},L_{\max}}^{\text{low}} - \sum_{m=n_{\text{SRC}}}^{L_{\max}+8} \beta_{m,L_{\max}}^{\text{low}} \\ & \geq 3 \frac{C_{\text{lifting}}^{\text{high}}}{C_{\text{lifting}}^{\text{low}}} \sum_{l=1}^{L_{\max}} 4^{L_{\max}-l} \left[\sum_{m=n_{\text{SRC}}}^{L_{\max}+8} \beta_{m,l}^{\text{high}} - \lambda \cdot \chi_{n_{\text{SRC}},l}^{\text{high}} \right]\end{aligned} \tag{44}$$

where we replaced $N-1$ by $L_{\max}+8$ based on (54) derived in Appendix II (derivation of Proposition 2). Apart from n_{SRC}, the resulting expression of (44) is parametrical to the variance of the high-frequency subbands [due to (37), (39)], the maximum number of decomposition levels, the scaling λ, and, finally, the ratio $C_{\text{ratio}} = C_{\text{lifting}}^{\text{high}} / C_{\text{lifting}}^{\text{low}}$ of scaling factors relating to the particular lifting decomposition of interest. However, since we cannot determine the solutions for n_{SRC} analytically from (44), in the following section we present experimental results that derive n_{SRC} based on experiments with two filter-pairs applied to real images. We provide some practical values for the modeling parameters λ, C_{ratio} and ξ of (44), (33) that help us evaluate the accuracy of the derived model for predicting the performance of incremental refinement, as well as establishing the successive refinement region. The experimental results also validate the practical efficiency of incremental refinement of computation for the IDWT.

IV. Experimental Results

We first examine the performance of incremental refinement in individual cases of IDWT applied to video frames as well as error frames produced by motion-compensated prediction. We also establish the performance of the proposed approach experimentally, and via the proposed model of Section III. Subsequently, we test the performance of incremental refinement of computation for entire video sequences decompressed with a scalable video coder.

Fig. 5 and Fig. 6 present representative results from two video sequences in terms of computation [measured based on (32), (33) in conjunction with our experimental software implementation] versus distortion (measured in terms of peak-signal to noise ratio—PSNR) achieved by stopping at several bitplanes (from $n = 6$ to $n = 2$) and focusing directly at the multilevel decomposition with $L_{\max} = 4$. The results of the conventional (nonrefinable) approach are measured also based on (32), (33) and by performing multiple IDWTs. The 9/7 and 5/3 filter-pairs were used for these results and throughout this section. We set the precision for fixed-point software implementation to 14 bits for the fractional part and 13 bits for the integer part (including the sign bit). In the results of this section we focus on the "frequency-first" approach for the proposed incremental refinement since we determined experimentally that it

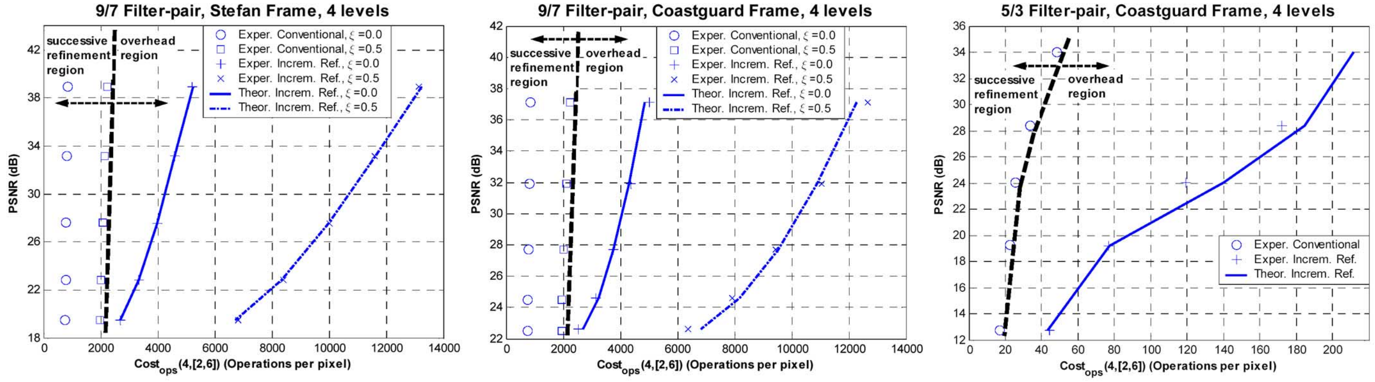


Fig. 5. Computation-Distortion results for the conventional and the proposed approach for the case of video frames: (two left plots) 9/7 filter-pair with $\xi = 0.0$ and $\xi = 0.5$ (for which we indicate the successive refinement region) and two representative video frames; (right plot) 5/3 filter-pair (results with one video frame, indicating the successive refinement region). For each case, the transform terminates at bitplane $n = \{2, \ldots, 6\}$ corresponding to low, medium and high distortion. The model results are also presented for each case.

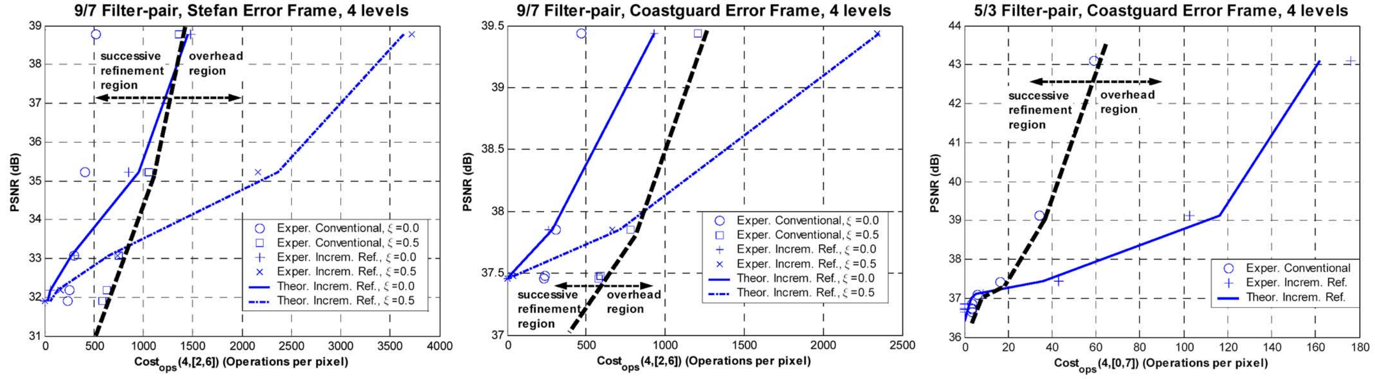
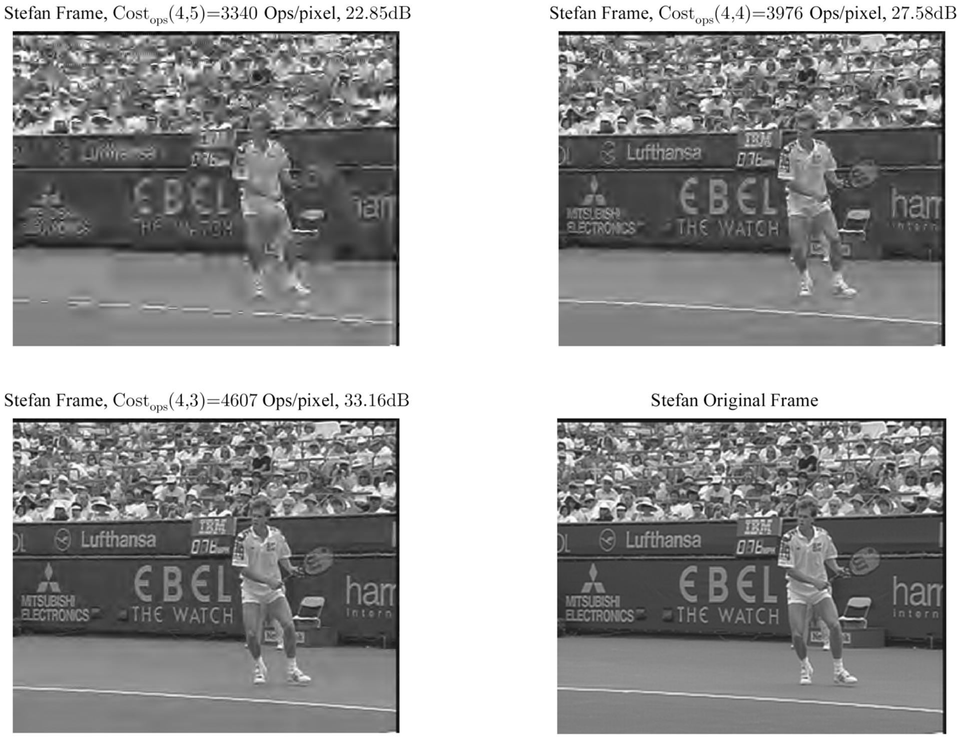


Fig. 6. Computation-Distortion results for the conventional and the proposed approach for the case of error frames: (two left plots) 9/7 filter-pair with $\xi = 0.0$ and $\xi = 0.5$ (for which we indicate the successive refinement region) and two representative error frames; (right plot) 5/3 filter-pair (results with one error frame, indicating the successive refinement region). For each case, the transform terminates at bitplane $n = \{2, \ldots, 6\}$ corresponding to low, medium and high distortion. The model results are also presented for each case.

performs almost identically to the "quality-first" approach and it also represents the generic multilevel incremental refinement paradigm where each input source bitplane refines the final result of the IDWT.

Starting with the case of regular (intra) video frames (Fig. 5), we observe that the proposed approach does not achieve performance close or inside the successive refinement region marked by the computation required by the conventional approach. To the contrary, it introduces significant computational overhead for both cases of ξ values tested for the 9/7 filter-pair. However, it appears that the incremental refinement case with $\xi = 0$ is close to the conventional case for $\xi = 0.5$ in the high-distortion (low PSNR) region. This is a scenario that could be feasible in practice, since the overhead of multiplication versus addition (expressed by increased values of ξ) increases with increased dynamic range, and this is indeed the case of the conventional approach versus the proposed incremental refinement of computation: the conventional computation performs less operations but with a higher dynamic range on average, in comparison to the proposed approach. The 5/3 filter-pair does not perform any multiplications because the filter taps seen in Table I can be applied with additions and bit shifts. Hence, ξ does not apply for this case. Notice that the lack of multiplications combined with the simplicity of the 5/3 filter-pair as compared to the 9/7 decreases the required operations almost by a factor of 50 for the same output distortion (PSNR). Of course, as typical JPEG-2000 experiments show [1], in image compression applications the 5/3 filter-pair leads to a 10%–20% rate increase in comparison to the 9/7 for the same output distortion. Nevertheless, for many resource-constrained environments this is only a small disadvantage compared to the significant complexity reduction exemplified by the comparative results of the two filter-pairs.

For the 9/7 filter-pair, the theoretical results produced by (36) agree very well with the experimental results for the case of intra frames seen in Fig. 5. The model is less accurate for the case of the 5/3 filter-pair. The two factors $C_{\text{lifting}}^{\text{band}}$ of (36) that scale the estimates $\beta_{n,l}^{\text{band}}$ (which are calculated by Proposition 1 and Proposition 2) can be estimated by a detailed analysis based on the length of the lifting filters, as well as the average increase in the dynamic range during the synthesis of the low-frequency component of each level. Since this approach is cumbersome and depends on several filter-specific factors, we estimate them based on curve fitting. Overall, the examples reported in Fig. 5 for the 9/7 filter-pair provided: $C_{\text{ratio}} = 0.5$ for the case of $\xi = 0.0$ and $C_{\text{ratio}} = 1.6$ for the case of $\xi = 0.5$. In addition,



Stefan Frame, $\mathrm{Cost}_{\mathrm{ops}}(4,5)=3340$ Ops/pixel, 22.85dB

Stefan Frame, $\mathrm{Cost}_{\mathrm{ops}}(4,4)=3976$ Ops/pixel, 27.58dB

Stefan Frame, $\mathrm{Cost}_{\mathrm{ops}}(4,3)=4607$ Ops/pixel, 33.16dB

Stefan Original Frame

Fig. 7. Examples of a video frame ("Stefan" sequence) when terminating the computation at $n_{\mathrm{src}} = \{5,4,3\}$ using the 9/7 filter-pair with $\xi = 0.0$. The computation/distortion results correspond to the proposed approach (see Fig. 5).

in order to experimentally quantify the value of λ, we measured the average ratio of the number of operations $(\mathrm{Cost}_{\mathrm{ops}})$ for each nonzero input in the proposed approach versus the conventional approach $(\mathrm{Cost}'_{\mathrm{ops}})$. This derived $\lambda = 0.95$ for the case where both methods have $\xi = 0.0$. Based on (44), we expect that successive refinement of computation is only marginally possible in this case, which agrees with the quantitative comparison of Fig. 5. Interestingly, when $\xi = 0.0$ in the proposed approach and $\xi = 0.5$ in the conventional approach, we derived $\lambda = 1.15$. This indicates [from (44)] that $n_{\mathrm{SRC}} = 6$. This seems to agree with the fact that $\mathrm{Cost}_{\mathrm{ops}}(4,6)$ of the incremental refinement approach with $\xi = 0.0$ (9/7 filter-pair) is close to the successive refinement region indicated in Fig. 5.

In the case of error frames (Fig. 6), the proposed approach performs within the successive refinement region for the medium to high-distortion regime. In general, error frame representations tend to be more suitable for incremental refinement due to the sparsity of the input transform representation. For example, only the result for $n = 2$ is outside the successive refinement region in the "Coastguard" error frame, whose variance is 30% lower than the variance of the "Stefan" error frame. Interestingly, the proposed approach for $\xi = 0$ performs entirely within the successive refinement region of the conventional approach with $\xi = 0.5$. Even though we do not indicate the performance for other intermediate values of ξ, we observed that the curves are "shifted" to the left linearly to the increase in value of ξ, hence it is easy for the reader to infer these cases from the provided experiments. Concerning the evaluation of the model of (36), we observe that the model follows the general trend of the experimental results. In this case, we again derived experimentally the values for the modeling parameters (9/7 filter-pair): $C_{\mathrm{ratio}} = 0.3$ for both $\xi = 0.0$ and $\xi = 0.5$, and $\lambda = 1.4$. By solving (44) with these parameters (using numerical methods), we find that these values correspond to the "good" region for successive refinement, where we have $n_{\mathrm{SRC}} = \{2,3\}$. This agrees with the derived experiments of Fig. 6.

Visual comparisons for video frames and error frames are presented in Fig. 7 and Fig. 8. Notice that both the conventional and the proposed approach produce virtually the same result (i.e., near identical PSNR performance) for each terminating bitplane, with some very minor discrepancy (in the order of 0.002 dB in PSNR) due to minor differences in the rounding errors produced by the fixed point computations for each case. Terminating the computation at a high bitplane produces ringing and blurring artifacts as shown in Fig. 7, effects that are well known from the image coding literature [1]. Nevertheless, in all cases a meaningful approximation of the final result can be obtained and the quality is progressively increased with increased computation.

As a concluding remark for the results of Figs. 5–8, it is important to note that the proposed approach has two fundamental properties that the conventional approach is lacking:

- Incremental refinement is computationally scalable with respect to quantization precision.
- Incremental refinement provides all the results reported in Figs. 5 and 6 by successively continuing the computation. On the other hand, the conventional approach has to invert the transform in its entirety for each experimental point. Consequently, if we consider a dynamic scenario where the source distortion is reduced after the initiation of the computation, the cumulative overhead of moving from one

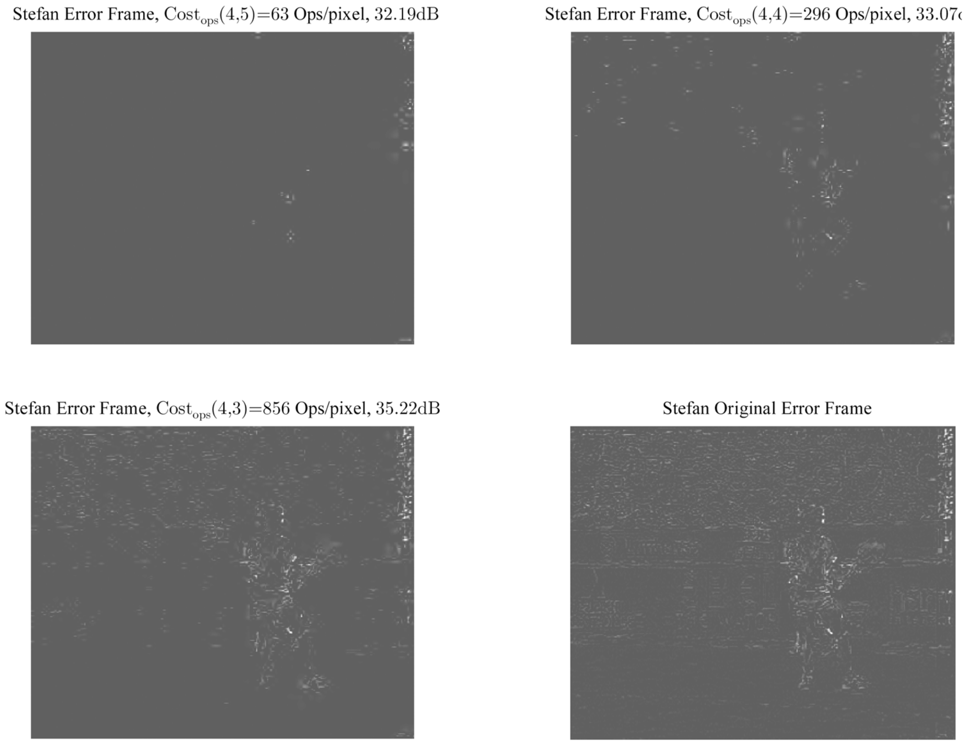

Stefan Error Frame, $\mathrm{Cost}_{\mathrm{ops}}(4,5)=63$ Ops/pixel, 32.19dB
Stefan Error Frame, $\mathrm{Cost}_{\mathrm{ops}}(4,4)=296$ Ops/pixel, 33.07
Stefan Error Frame, $\mathrm{Cost}_{\mathrm{ops}}(4,3)=856$ Ops/pixel, 35.22dB
Stefan Original Error Frame

Fig. 8. Examples of an error frame ("Stefan" sequence) when terminating the computation at $n_{\mathrm{src}} = \{5, 4, 3\}$ using the 9/7 filter-pair with $\xi = 0.0$. We have increased the brightness and contrast of the images to highlight the content. The computation/distortion results correspond to the proposed approach (see Fig. 6).

point to the next in the Distortion-Computation curves of Fig. 5 and Fig. 6, is significantly higher for the conventional approach.

The proposed approach appears to provide benefits for the case of error frames. As a result, it would be interesting to evaluate its performance in a coding framework using motion-compensated prediction. Representative results are presented in Fig. 9 in terms of mean PSNR for embedded coding of the Y, U, V channels of CIF-resolution video using a scalable video coder [33] with a group-of-pictures structure consisting of one intra-frame and 23 error frames. We have included as a reference in these results the computation required for the conventional separable (row-column) lifting computation [23], because it is commonly used in practical applications [1]. The results indicate that incremental refinement becomes beneficial for the medium to high distortion regime, corresponding to medium to low-rate (28–31 dB in PSNR, corresponding to 128 to 350 kbps), with the exception of the "Stefan" sequence where the proposed approach provides comparable computation only in the high distortion regime. Fig. 9 shows the striking difference in complexity scalability between the proposed approach and the conventional computation approaches.

Concerning the low-distortion (high-rate) regime, an overhead ranging up to three-fold increase in computation is imposed due to the incremental refinement property. We remark though that the comparison is carried out under the direct 2-D lifting-based realization, which is already reducing computation by approximately 40% in comparison to the conventional separable lifting-based DWT [24]. Hence, as shown by Fig. 9, the proposed approach performs relatively better when compared to the conventional separable lifting realization. Notice that all the results presented for each sequence in Fig. 9 are produced incrementally by the proposed approach (with respect to the IDWT part), while both conventional computations require reinitiating the computation for each experimental point reported in the figure. This indicates that under dynamic resource allocation, the proposed approach can refine the output representation. Moreover, when system resources become scarce, the proposed approach can still produce a high-distortion, albeit meaningful, result unlike the conventional approach that is constrained to a certain range of computation.

V. Conclusion

We propose a method for incremental computation of the forward and inverse DWT under a bitplane-based formulation of the 2-D lifting decomposition. This results in a continuous computation for the output where, under any termination point, the representation corresponding to the provided input source accuracy can be retrieved. In addition, to granular and scalable computation, the proposed DWT calculation ensures that, should additional computational resources and source refinement bits be provided, the transform calculation can be enhanced from the previously computed result. In order to study the properties of the proposed approach, we proposed a stochastic model that predicts the expected computation in function of source statistics as well as operational and algorithm settings (decomposition levels, terminating bitplane, filtering complexity etc.). This

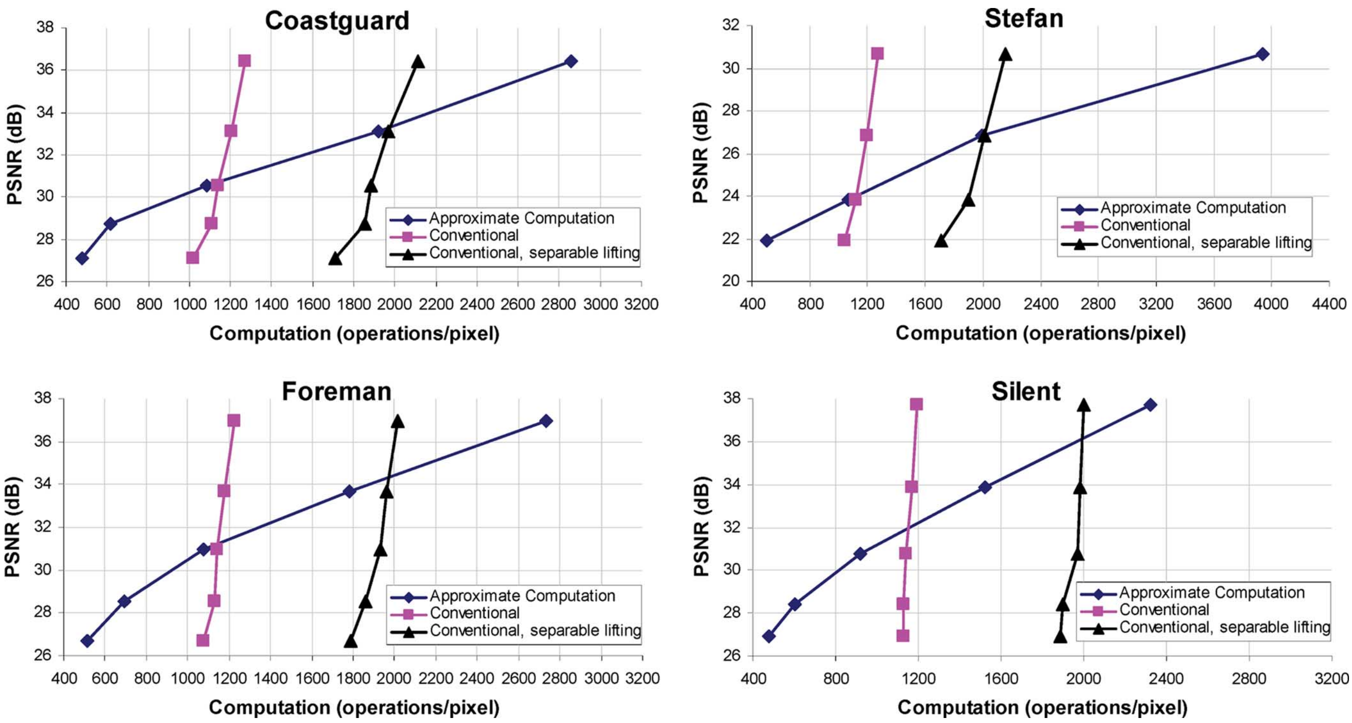


Fig. 9. Computation-Distortion results for the conventional approach versus the proposed approach in the case of video sequences. The computation required by the nonrefinable separable lifting is included as well as a reference.

enables a first exploration of the subspace of operational parameters for which incremental refinement provides comparable or even superior computational efficiency in comparison to the conventional (nonrefinable) computation. This property, termed successive refinement of computation in this paper, is tested in practice with real video sequences and an initial model validation is carried out with two popular wavelet filter-pairs. The results demonstrate that successive refinement of computation is feasible for the medium to high distortion regime.

Further exploration of the proposed approach could involve theoretical and experimental validation of successive refinement for the forward DWT. Note that applications of incremental refinement for the forward DWT exist, e.g., for progressive image denoising and edge detection using multiresolution decompositions [20], or for the residual image transformation of the enhancement layer in progressive FGS coding [34]. In addition, a detailed experimental study with a variety of filter-pairs, image sizes and decomposition levels would be of interest in order to ascertain the practical impact of several algorithmic features in the required computation. This can be coupled with an analytical study of the interrelationship between the input source model and the system parameters C_{ratio}, λ and ξ (in function of various wavelet filter-pairs), which would lead to a deeper understanding of incremental refinement properties. Higher-dimensional extensions of the proposed framework could also be investigated, e.g., for 3-D medical data or even 3-D motion-compensated wavelet video coding. Furthermore, extensions of the proposed bitplane-based calculation to other quantizers or other approaches that reduce computation by exploiting correlations in the input source, e.g., via the use of bitplane-based prediction could be envisaged. Finally, interesting extensions could involve the formulation of the proposed scheme in adaptive and nonlinear lifting scheme decompositions [8], [35] that produce sparse transform representations.

Appendix I

Derivation of Proposition 1: The probability that the coefficients of a high-frequency subband have a significant bit at bitplane n can be expressed by integrating over the density function of each component of x_{high} between $[-T_{n+1}, -T_n]$ and $[T_n, T_{n+1}]$, i.e., over the intervals corresponding to bitplane n. The key insight for reaching the result is that this integration can be expressed as the difference between integrating from $[-T_{n+1}, T_{n+1}]$ and $[-T_n, T_n]$

$$\begin{aligned} &\int\limits_{-T_{n+1}}^{-T_n} \int\limits_{-T_{n+1}}^{-T_n} \cdots \int\limits_{-T_{n+1}}^{-T_n} \mathrm{P}(\mathbf{x}_{\text{high}}) d\mathbf{x}_{\text{high}} \\ &+ \int\limits_{T_n}^{T_{n+1}} \int\limits_{T_n}^{T_{n+1}} \cdots \int\limits_{T_n}^{T_{n+1}} \mathrm{P}(\mathbf{x}_{\text{high}}) d\mathbf{x}_{\text{high}} \\ &= \int\limits_{-T_{n+1}}^{T_{n+1}} \int\limits_{-T_{n+1}}^{T_{n+1}} \cdots \int\limits_{-T_{n+1}}^{T_{n+1}} \mathrm{P}(\mathbf{x}_{\text{high}}) d\mathbf{x}_{\text{high}} \\ &- \int\limits_{-T_n}^{T_n} \int\limits_{-T_n}^{T_n} \cdots \int\limits_{-T_n}^{T_n} \mathrm{P}(\mathbf{x}_{\text{high}}) d\mathbf{x}_{\text{high}}. \end{aligned} \tag{45}$$

Replacing $\mathrm{P}(x_{\text{high}})$ as $R \cdot C \cdot 2^{-2l}$ Gaussian distributions parameterized by Θ_l which follows the distribution of (34) [see (46) and (47) at the bottom of the page] where (47) was reached by rearranging integrals and by the fact that the $R \cdot C \cdot 2^{-2l}$

resulting integrals of each of the two terms are equal, therefore variables $x_{\mathrm{high}}[i], 0 \leq i < R \cdot C \cdot 2^{-2l}$ where replaced by x_{high} in the final expression.

The remaining steps follow a similar methodology as in our previous work [30]. Recall the definition of the erf function $\mathrm{erf}(z) = (2/\sqrt{\pi}) \int_0^z e^{-x^2} dx$. Using substitution of variables, we get the following expression:

$$\begin{aligned} &\frac{1}{\sqrt{2\pi\theta_l}} \int_{-T}^{T} e^{-\frac{1}{2\theta_l}(x_{\mathrm{high}})^2} dx_{\mathrm{high}} \\ &= 2 \cdot \frac{1}{\sqrt{2\pi\theta_l}} \int_{0}^{T} e^{-\frac{1}{2\theta_l}(x_{\mathrm{high}})^2} dx_{\mathrm{high}} \\ &= \frac{2}{\sqrt{\pi}} \int_{0}^{\frac{T}{\sqrt{2\theta_l}}} e^{-(x')^2} dx'_{\mathrm{high}} = \mathrm{erf}\left(\frac{T}{\sqrt{2\theta_l}}\right). \qquad (48) \end{aligned}$$

Hence, substituting (48) into (45) with $T = \{T_n, T_{n+1}\}$ gives us

$$\begin{aligned} &\int_{-T_{n+1}}^{-T_n} \int_{-T_{n+1}}^{-T_n} \cdots \int_{-T_{n+1}}^{-T_n} \mathrm{P}(\mathbf{x}_{\mathrm{high}}) d\mathbf{x}_{\mathrm{high}} \\ &+ \int_{T_n}^{T_{n+1}} \int_{T_n}^{T_{n+1}} \cdots \int_{T_n}^{T_{n+1}} \mathrm{P}(\mathbf{x}_{\mathrm{high}}) d\mathbf{x}_{\mathrm{high}} \\ &= \int_0^{\infty} \frac{1}{\sigma_l^2} e^{-\frac{1}{\sigma_l^2}\theta_l} \left[\mathrm{erf}\left(\frac{T_{n+1}}{\sqrt{2\theta_l}}\right)\right]^{R \cdot C \cdot 2^{-2l}} d\theta_l \\ &\quad - \int_0^{\infty} \frac{1}{\sigma_l^2} e^{-\frac{1}{\sigma_l^2}\theta_l} \left[\mathrm{erf}\left(\frac{T_n}{\sqrt{2\theta_l}}\right)\right]^{R \cdot C \cdot 2^{-2l}} d\theta_l. \qquad (49) \end{aligned}$$

In lack of an analytical solution for (49), we use the following approximation [30]:

$$\begin{aligned} &\int_0^{\infty} \frac{1}{\sigma_l^2} e^{-\frac{1}{\sigma_l^2}\theta_l} \left[\mathrm{erf}\left(\frac{T}{\sqrt{2\theta_l}}\right)\right]^{R \cdot C \cdot 2^{-2l}} d\theta_l \\ &\cong \int_0^{\infty} \frac{1}{\sigma_l^2} e^{-\frac{1}{\sigma_l^2}\theta_l} I\left(\theta_l \geq \frac{T^2}{12.5}\right) d\theta_l \\ &= \int_0^{\frac{T^2}{12.5}} \frac{1}{\sigma_l^2} e^{-\frac{1}{\sigma_l^2}\theta_l} d\theta_l \\ &= 1 - \exp\left(-\frac{T_n^2}{12.5\sigma_l^2}\right) \qquad (50) \end{aligned}$$

with $I(\theta_l \geq T^2/12.5)$ the indicator function, which is used for the term $[\mathrm{erf}(T/\sqrt{2\theta_l})]^{R \cdot C \cdot 2^{-2l}}$. The justification of this replacement is based on the observation that we can approximate the error function raised to a large power based on the indicator function, by choosing the threshold of the indicator function appropriately. For images of 256 × 256 to 1024 × 1024 pixels and $l = 3$ or $l = 4$ decomposition levels, typical plots of $[\mathrm{erf}(T/\sqrt{2\theta_l})]^{R \cdot C \cdot 2^{-2l}}$ versus $I(T/\sqrt{2\theta_l} \geq 2.5)$ are seen in Fig. 10. Instead of using a fitting technique to derive the appropriate threshold value for the indicator function based on the plots of Fig. 10, we derived the utilized value based on the agreement of the derived approximation with the observed experimental results of probability of significance of high-frequency wavelet subbands. This derives the setting $\theta_l \geq T^2/12.5$ used in the indicator function in (50).

$$\begin{aligned} &\int_{-T_{n+1}}^{T_{n+1}} \int_{-T_{n+1}}^{T_{n+1}} \cdots \int_{-T_{n+1}}^{T_{n+1}} \mathrm{P}(\mathbf{x}_{\mathrm{high}}) d\mathbf{x}_{\mathrm{high}} - \int_{-T_n}^{T_n} \int_{-T_n}^{T_n} \cdots \int_{-T_n}^{T_n} \mathrm{P}(\mathbf{x}_{\mathrm{high}}) d\mathbf{x}_{\mathrm{high}} \\ &= \int_{-T_{n+1}}^{T_{n+1}} \int_{-T_{n+1}}^{T_{n+1}} \cdots \int_{-T_{n+1}}^{T_{n+1}} \left(\int_{0+}^{\infty} \frac{1}{\sigma_l^2} e^{-\frac{1}{\sigma_l^2}\theta_l} \cdot \frac{1}{(2\pi\theta_l)^{R \cdot C \cdot 2^{-2l}/2}} e^{-\frac{1}{2\theta_l}\left[(x_{\mathrm{high}}[0])^2 + \ldots + \left(x_{\mathrm{high}}[R \cdot C \cdot 2^{-2l} - 1]\right)^2\right]} d\theta_l \right) \\ &\quad \times dx_{\mathrm{high}}[0] \ldots dx_{\mathrm{high}}[R \cdot C \cdot 2^{-2l} - 1] \qquad (46) \\ &\quad - \int_{-T_n}^{T_n} \int_{-T_n}^{T_n} \cdots \int_{-T_n}^{T_n} \left(\int_{0+}^{\infty} \frac{1}{\sigma_l^2} e^{-\frac{1}{\sigma_l^2}\theta_l} \cdot \frac{1}{(2\pi\theta_l)^{R \cdot C \cdot 2^{-2l}/2}} e^{-\frac{1}{2\theta_l}\left[(x_{\mathrm{high}}[0])^2 + \ldots + \left(x_{\mathrm{high}}[R \cdot C \cdot 2^{-2l} - 1]\right)^2\right]} d\theta_l \right) \\ &\quad \times dx_{\mathrm{high}}[0] \ldots dx_{\mathrm{high}}[R \cdot C \cdot 2^{-2l} - 1] \\ &= \int_{0+}^{\infty} \frac{1}{\sigma_l^2} e^{-\frac{1}{\sigma_l^2}\theta_l} \left(\frac{1}{\sqrt{2\pi\theta_l}} \int_{-T_{n+1}}^{T_{n+1}} e^{-\frac{1}{2\theta_l}(x_{\mathrm{high}})^2} dx_{\mathrm{high}} \right)^{R \cdot C \cdot 2^{-2l}} \\ &\quad \times d\theta_l - \int_{0+}^{\infty} \frac{1}{\sigma_l^2} e^{-\frac{1}{\sigma_l^2}\theta_l} \left(\frac{1}{\sqrt{2\pi\theta_l}} \int_{-T_n}^{T_n} e^{-\frac{1}{2\theta_l}(x_{\mathrm{high}})^2} dx_{\mathrm{high}} \right)^{R \cdot C \cdot 2^{-2l}} d\theta_l. \qquad (47) \end{aligned}$$

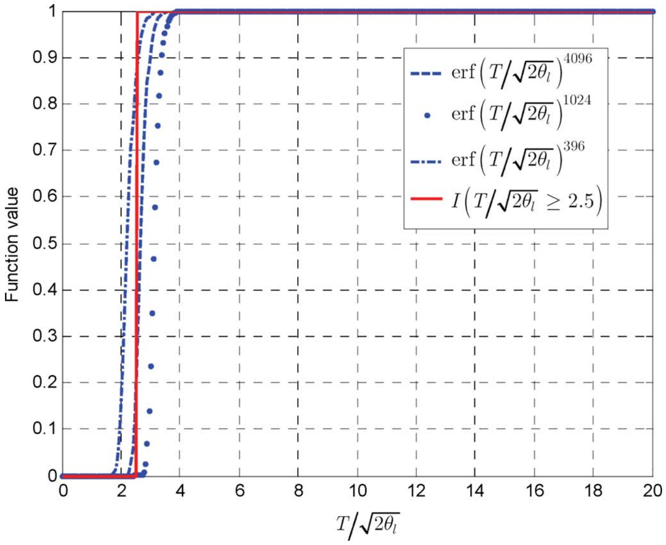


Fig. 10. Justification of the approximation of $\mathrm{erf}(T/\sqrt{2\theta_l})^n$ by $I(T/\sqrt{2\theta_l} \geq 2.5)$: the plot shows each function for typical values of n corresponding to image sizes and decomposition levels of interest. The range of $T/\sqrt{2\theta_l}$ shown in the plot covers the experimentally derived range of values.

The replacement of (50) in (49) for $T = \{T_n, T_{n+1}\}$ along with the relationship:

$$\exp\left(-\frac{T_{n+1}^2}{12.5\sigma_l^2}\right) = \exp\left[-\frac{(2T_n)^2}{12.5\sigma_l^2}\right] = \exp\left(-\frac{T_n^2}{12.5\sigma_l^2}\right)^4 \tag{51}$$

derives the approximation of (37). ■

Appendix II

Derivation of Proposition 2: Under the i.i.d. Gaussian distribution $X_{\mathrm{low}} \sim \mathrm{P}(x_{\mathrm{low}}) = N(0, \sigma_{\mathrm{low}}^2)$ of each low frequency wavelet coefficient x_{low}, the probability that a low-frequency coefficient has a significant bit at bitplane n, denoted by $\beta_{n,L_{\max}}^{\mathrm{low,coeff}}$, is estimated by integrating the Gaussian distribution over $[-T_{n+1}, -T_n]$ and $[T_n, T_{n+1}]$. The desired probability amounts to the difference between integrating from $[-T_{n+1}, T_{n+1}]$ and $[-T_n, T_n]$

$$\begin{aligned}\beta_{n,L_{\max}}^{\mathrm{low,coeff}} &= \int_{-T_{n+1}}^{-T_n} \mathrm{P}(x_{\mathrm{low}})dx_{\mathrm{low}} + \int_{T_n}^{T_{n+1}} \mathrm{P}(x_{\mathrm{low}})dx_{\mathrm{low}} \\ &= \int_{-T_{n+1}}^{T_{n+1}} \mathrm{P}(x_{\mathrm{low}})dx_{\mathrm{low}} - \int_{-T_n}^{T_n} \mathrm{P}(x_{\mathrm{low}})dx_{\mathrm{low}}. \end{aligned} \tag{52}$$

Based on the property $T_{n+1} = 2T_n$ we have

$$\beta_{n,L_{\max}}^{\mathrm{low,coeff}} = \mathrm{erf}\left(T_n \frac{\sqrt{2}}{\sqrt{\sigma_{\mathrm{low}}}}\right) - \mathrm{erf}\left(T_n \frac{1}{\sqrt{2\sigma_{\mathrm{low}}}}\right). \tag{53}$$

If we model $\beta_{n,L_{\max}}^{\mathrm{low,coeff}}$ of each low-frequency wavelet coefficient x_{low} as i.i.d. random variables (since X_{low} are i.i.d.) and derive their distribution function, the probability that the wavelet coefficients of the low-frequency subband have a significant bit at bitplane n is the convolution of these distributions for all $R \cdot C \cdot 2^{-2L_{\max}}$ low-frequency wavelet coefficients. Under the central limit theorem, this is a Gaussian distribution with mean $\mu_{\mathrm{siglow}}^{\mathrm{limit}}$ and variance $(\sigma_{\mathrm{siglow}}^{\mathrm{limit}})^2$ defined based on the parameters of each individual distribution. However, instead of deriving the parameters of each individual distribution, we opt to fit (using least squares) the parameters $\mu_{\mathrm{siglow}}^{\mathrm{limit}}$ and $(\sigma_{\mathrm{siglow}}^{\mathrm{limit}})^2$ directly to the experimental histogram of probabilities of significance of low-frequency wavelet coefficients of several real images. A variety of image sizes (from 256 × 256 to 1024 × 1024) and decomposition levels (from three to five) was used. In addition, based on the gains of low-pass filters of popular wavelet filter-pairs [1] and by bounding the input signal values in the interval [0,255] (i.e., regular 8-bit images), the maximum bitplane threshold for which significant bits may occur for the low-frequency subband of typical images decomposed up to $L_{\max}$ levels is

$$\max\{T_n\} = 2^{L_{\max}+8}. \tag{54}$$

Hence, the set of dyadically increased thresholds: $T_n = \{2^0, 2^1, 2^2, \ldots, 2^{L_{\max}+8}\}$ was used during the experimental parameter fitting. This derived the following approximation for the parameters for the resulting Gaussian distribution:

$$\mu_{\mathrm{siglow}}^{\mathrm{limit}} \cong 0.088, \left(\sigma_{\mathrm{siglow}}^{\mathrm{limit}}\right)^2 \cong 19795 \cdot 4^{L_{\max}} \tag{55}$$

i.e., a near-zero mean Gaussian distribution with variance proportional to $4^{L_{\max}}$. The proportionality of the variance to this factor is justifiable because, based on the central limit theorem, $(\sigma_{\mathrm{siglow}}^{\mathrm{limit}})^2 = \sigma_{\mathrm{siglow}}^2/(R \cdot C \cdot 2^{-2L_{\max}})$, with $\sigma_{\mathrm{siglow}}^2$ the variance of the distribution of $\beta_{n,L_{\max}}^{\mathrm{low,coeff}}$. We set $\mu_{\mathrm{siglow}}^{\mathrm{limit}} = 0$ for simplicity since this had marginal effect in the model precision and also scale the final Gaussian distribution by a factor proportional to $2^{L_{\max}}$ (i.e., proportional to the subband dimensions) in order to normalize the result to the range of the observed measurements. The final probability of significance for low-frequency coefficients for every threshold T_n under these approximations, is given by (38). ■

Acknowledgment

The authors would like to thank the Associate Editor Prof. I. Pollak and the four anonymous reviewers for their helpful remarks. At the time of this work, Y. Andreopoulos was with the Department of Electrical Engineering, University of California, Los Angeles.

References

[1] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Boston, MA: Kluwer Academic, 2002.

[2] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[3] V. K. Goyal and M. Vetterli, "Computation-distortion characteristics of block transform coding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP) '97*, Apr. 1997, vol. 4, pp. 2729–2732.

[4] V. K. Goyal and M. Vetterli, "Manipulating rates, complexity and error-resilience with discrete transforms," in *Proc. IEEE Asilomar Conf. Signals, Syst., Comput.*, Nov. 1998, vol. 1, pp. 457–461.

[5] J. Winograd and S. H. Nawab, "Incremental refinement of DFT and STFT approximations," *IEEE Signal Process. Lett.*, vol. 2, no. 2, pp. 25–27, Feb. 1995.

[6] K. Lengwehasatit and A. Ortega, "Scalable variable complexity approximate forward DCT," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 11, pp. 1236–1248, Nov. 2004.

[7] Z. Wang, "Pruning the fast discrete cosine transform," *IEEE Trans. Commun.*, vol. 39, no. 5, pp. 640–643, May 1991.

[8] E. Le Pennec and S. Mallat, "Sparse geometric image representations with bandelets," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 423–438, Apr. 2005.

[9] M. Flierl, T. Wiegand, and B. Girod, "Rate-constrained multihypothesis prediction for motion-compensated video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 11, pp. 957–969, Nov. 2002.

[10] C. A. Christopoulos *et al.*, "Comparative performance evaluation of algorithms for fast computation of the two-dimensional DCT," in *Proc. IEEE Benelux Workshop on Circuits Syst. Signal Process.*, Mar. 1994, pp. 75–79.

[11] A. N. Skodras and A. G. Costantinides, "Efficient input-reordering algorithms for fast DCT," *Inst. Electr. Eng. Electron. Lett.*, vol. 27, no. 21, pp. 1973–1975, Oct. 1991.

[12] I. R. Ismaeil, A. Docef, F. Kossentini, and R. K. Ward, "A computation-distortion optimized framework for efficient DCT-based video coding," *IEEE Trans. Multimedia*, vol. 3, no. 3, pp. 210–298, Sep. 2001.

[13] J. T. Ludwig, S. H. Nawab, and A. Chandrakasan, "Low-power digital filtering using approximate processing," *IEEE J. Solid-State Circuits*, vol. 31, no. 3, pp. 340–395, Mar. 1996.

[14] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

[15] S. H. Nawab and E. Dorken, "A framework for quality versus efficiency tradeoffs in STFT analysis," *IEEE Trans. Signal Process.*, vol. 43, no. 4, pp. 998–1001, Apr. 1995.

[16] D. X. D. Yang, B. Fowler, and A. El Gamal, "A Nyquist-rate pixel-level ADC for CMOS image sensors," *IEEE J. Solid State Circuits*, vol. 34, no. 3, pp. 348–356, Mar. 1999.

[17] T. N. Ruckmongathan, "A successive approximation technique for displaying gray shades in liquid crystal displays (LCDs)," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 554–563, Feb. 2007.

[18] J. M. Winograd and S. Nawab, "Probabilistic complexity analysis of incremental DFT algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 1997, vol. 3, pp. 1985–1988.

[19] J. M. Winograd, "Incremental refinement structures for approximate signal processing," Ph.D. dissertation, Elect. Comput. Eng. Dept., Boston Univ., MA, Report No. ECE-97-003.

[20] S. Mallat, *A Wavelet Tour of Signal Processing*. San Diego, CA: Academic, 1998.

[21] M. B. Wakin, J. N. Laska, M. F. Duartem, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "An architecture for compressive imaging," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2006 , pp. 1273–1276.

[22] Y. Andreopoulos and M. van der Schaar, "Incremental refinement of computation for the discrete wavelet transform," presented at the IEEE Int. Conf. Image Process. (ICIP), San Antonio, TX, Sep. 2007.

[23] I. Daubechies and W. Sweldens, "Factorization wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, pp. 247–269, 1998.

[24] H. Meng and Z. Wang, "Fast spatial combinative lifting algorithm of wavelet transform using the 9/7 filter for image block compression," *Inst. Elect. Eng. Electron. Lett.*, vol. 36, no. 21, pp. 1766–1767, Oct. 2000.

[25] M. D. Adams and F. Kossentini, "Reversible integer-to-integer wavelet transforms for image compression: Performance evaluation and analysis," *IEEE Trans. Image Process.*, vol. 9, no. 6, pp. 1010–1024, Jun. 2000.

[26] R. P. Brent and H. T. Kung, "The area-time complexity of binary multiplication," *J. Assoc. Comp. Mach.*, vol. 28, no. 3, pp. 512–534, Jul. 1981.

[27] R. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Trans. Comput.*, vol. 35, no. 8, pp. 677–691, Aug. 1986.

[28] E. Y. Lam and J. W. Goodman, "A mathematical analysis of DCT coefficient distributions for images," *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1661–1666, Oct. 2000.

[29] S. Mallat and F. Falzon, "Analysis of low bit-rate image transform coding," *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 1027–1042, Apr. 1998.

[30] B. Foo, Y. Andreopoulos, and M. van der Schaar, "Analytical rate-distortion-complexity modeling of wavelet-based video coders," *IEEE Trans. Signal Process.*, accepted for publication.

[31] J.-R. Ohm, "Advances in scalable video coding," *Proc. IEEE*, vol. 93, pp. 42–56, Jan. 2005.

[32] W. H. R. Equitz and T. Cover, "Successive refinement of information," *IEEE Trans. Inf. Theory*, vol. 37, no. 2, pp. 269–275, Mar. 1991.

[33] Y. Andreopoulos, A. Munteanu, G. Van der Auwera, P. Schelkens, and J. Cornelis, "Scalable wavelet video-coding with in-band prediction—Implementation and experimental results," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2002, vol. 3, pp. 729–732.

[34] F. Wu, S. Li, and Y.-Q. Zhang, "A framework for efficient progressive fine granularity scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 332–344, 2001.

[35] G. Piella and H. J. A. M. Heijmans, "Adaptive lifting schemes with perfect reconstruction," *IEEE Trans. Signal Process.*, vol. 50, no. 7, pp. 1620–1630, Jul. 2002.

Yiannis Andreopoulos (M'00) received the electrical engineering Diploma and the M.Sc. degree in signal processing systems from the University of Patras, Greece, in 1999 and 2000, respectively. He receivedd the Ph.D. degree in applied sciences from the University of Brussels, Belgium, in May 2005, where he defended a thesis on scalable video coding and complexity modeling for multimedia systems. During his thesis work, he participated and was supported by the EU IST-project MASCOT, a Future and Emerging Technologies (FET) project.

During his Postdoctoral work at the University of California Los Angeles (UCLA) he performed research on cross-layer optimization of wireless media systems, video streaming, and theoretical aspects of rate-distortion-complexity modeling for multimedia systems. Since October 2006, he has been a Lecturer with the Queen Mary University of London, U.K.

Dr. Andreopoulos won the "Most-Cited Paper" award in 2007, from the Elsevier EURASIP *Journal Signal Processing: Image Communication*, based on the number of citations his 2004 article "In-band motion compensated temporal filtering" received within a three-year period. During 2002–2003, he made several decisive contributions to the ISO/IEC JTC1/SC29/WG11 (Motion Picture Experts Group—MPEG) committee in the early exploration on scalable video coding, which has now moved into the standardization phase.

Mihaela van der Schaar (SM'04) received the Ph.D. degree from the Eindhoven University of Technology, The Netherlands, in 2001.

She is currently an Assistant Professor with the Electrical Engineering Department, University of California, Los Angeles (UCLA). Prior to this (1996–June 2003), she was a Senior Researcher with the Philips Research, The Netherlands and USA, where she led a team of researchers working on multimedia coding, processing, networking, and streaming algorithms and architectures. She has published extensively on multimedia compression, processing, communications, networking, and architectures and holds 28 granted U.S. patents and several more pending.

Dr. van der Schaar was an active participant to the ISO Motion Picture Expert Group (MPEG) standard since 1999, to which she made more than 50 contributions and for which she received three ISO recognition awards. She also chaired the *ad hoc* group on MPEG21 Scalable Video Coding for three years, and also co-chaired the MPEG *ad hoc* group on Multimedia Test-bed. She was also elected a Member of the Technical Committee on Multimedia Signal Processing and of the Technical Committee on Image and Multiple Dimensional Signal Processing, both of the IEEE Signal Processing Society. She was an Associate Editor of the IEEE Transactions on Multimedia and SPIE *Electronic Imaging Journal*. Currently, she is an Associate Editor of the IEEE Transactions on Circuits and System for Video Technology (T-CSVT) and of the IEEE Signal Processing Letters. She received the NSF CAREER Award in 2004, IBM Faculty Award in 2005, Okawa Foundation Award in 2006, and the Best Paper Award for her paper published in 2005 in the IEEE Transactions on Circuits and Systems for Video Technology.