A practical application of skipped steps DWT in JPEG 2000 part 2-compliant compressor

Roman Starosolski

Institute of Informatics, Silesian University of Technology,
Akademicka 16, 44-100 Gliwice, Poland
roman.starosolski@polsl.pl

Abstract. In this paper, we evaluate effects of applying the fixed skipped steps discrete wavelet transform (fixed SS-DWT) variants in the lossless compression that is compliant with part 2 of the JPEG 2000 standard. Compared to results obtained previously using a modified JPEG 2000 part 1 compressor, for a large and diverse set of test images, we found that extensions of part 2 of the standard allow further bitrate improvements. We experimentally confirmed that the fixed SS-DWT variants may be obtained in compliance with the standard and we identified practical JPEG 2000 part 2-compliant compression schemes with various trade-offs between the bitrate improvement and the compression process complexity.

Keywords: image processing, image compression standards, lossless image compression, JPEG 2000, DWT, RDLS, step skipping, SS-DWT

1 Introduction

In lossless JPEG 2000, the reversible discrete wavelet transform (DWT) decomposes an image into subbands of different characteristics, that are then independently entropy coded [22]. In [14] we noticed, that the lifting steps (LS) [3, 21] employed by DWT may propagate noise between subbands and worsen compression effects and we proposed to replace LS with reversible denoising and lifting steps (RDLS). RDLS are LS integrated with denoising filters in such a way that the perfect transform reversibility is preserved despite the inherently lossy denoising. RDLS resulted in the greatest bitrate improvements when the noise filtering was applied during computing of some subbands only, but for some images, the best bitrates were obtained when the entire DWT stage of JPEG 2000 was skipped—we suspected that similarly to denoising, the optimum might be in-between skipping and applying DWT. Therefore, in [17] we proposed the skipped steps DWT (SS-DWT) obtained from DWT by skipping selected steps of its computation and, among others, defined two simple fixed SS-DWT variants FIX1 and FIX2 of properties especially interesting from a practical standpoint.

NOTICE: this is the author's version of a work that was subsequently published in S. Kozielski et al. (Eds.): BDAS 2018, CCIS 928, pp. 334–348, 2018. The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-99987-6_26.

These variants allow obtaining good bitrate improvements and are compliant to part 2 of the JPEG 2000 standard [6], as opposed to the general SS-DWT case that is neither part 1 [5] nor part 2-compliant. The research was done using a JPEG 2000 part 1 implementation modified to obtain SS-DWT and we suspected that the fixed variants might be more effective in the part 2-compliant compressor. Therefore in this work, from a practical standpoint, we evaluate effects of applying fixed SS-DWT variants in JPEG 2000 part 2-compliant compressor.

The remainder of this paper is organized as follows. In subsections of the next Section, we briefly describe DWT (Sec. 2.1), RDLS and it's application to DWT (Sec. 2.2), and SS-DWT (Sec. 2.3); then in Sec. 2.4 the fixed SS-DWT variants are presented and their compliance with JPEG 2000 part 2 is discussed; Section 2.5 contains the experimental procedure including the description of the test images and the used implementations. The experimental results are presented and discussed in Sec. 3. Section 4 summarizes the findings.

2 Materials and methods

2.1 Lifting-based discrete wavelet transform

DWT employed in image compression algorithms decomposes an image into subbands of different characteristics that are independently entropy coded. Subbands represent image details of different orientations and sizes. For brevity, as in the previous work [17], we describe here only the lifting-based reversible DWT with Cohen Daubechies Faveau (5,3) wavelet filter, reduced to essentials. Among others, it is exploited in lossless JPEG 2000 compression of images. For further details as well as for more general characteristics of various variants of DWT, their application in JPEG 2000, and the JPEG 2000 standard the reader is referred to [1, 2, 5, 6, 22].

The one-dimensional DWT (1D-DWT) transforms in-place a discrete signal $S = s_0 s_1 s_2 \ldots s_{l-1}$ of finite length l into two subbands:

- a low-pass filtered signal L, that represents the low-frequency features of S;
- a high-pass filtered signal H containing high-frequency features that, along with the low-pass signal, allows the perfect reconstruction of the original signal.

S is transformed in 3 steps. First, in the prediction step, we perform the high-pass filtering of odd samples (hereafter, the parity of the sample or pixel is determined by its location and not its value) by applying to each of them the LS presented in below Eq.1:

$$s_x \leftarrow s_x - \lfloor (s_{x-1} + s_{x+1})/2 \rfloor. \tag{1}$$

In each LS a single signal sample is modified by adding to it a linear combination of other samples (in a general case the sum may be negated). A transform performed as a sequence of LS has advantageous properties: it may be computed in-place and it is easily and perfectly invertible.

Another LS is then, during the update step, applied to each even sample:

$$s_x \leftarrow s_x + \lfloor (s_{x-1} + s_{x+1} + 2)/4 \rfloor. \tag{2}$$

Finally, in the reorder step, we reposition even samples to the lower half of the original signal, preserving their ordering (sample s_x is moved to $s_{x/2}$), and odd samples are moved to the upper half. We obtain separate subbands L and H, respectively. The reorder step is not an LS.

The two-dimensional DWT (2D-DWT) of an image is obtained by first applying 1D-DWT to each image column, which results in L and H subbands of the image (Fig. 1.A-B). Then, by applying 1D-DWT to each row, we obtain the 1-level DWT consisting of LL and HL subbands (transformed from L subband) and LH and HH subbands (from H subband)—see Fig. 1.C. The higher-level DWT, that provides multiresolution image representation, is obtained by Mallat decomposition [9]. The $t+1$-level transform is obtained by applying the 1-level transform to the LL subband of the t-level transform (Fig. 1.D).

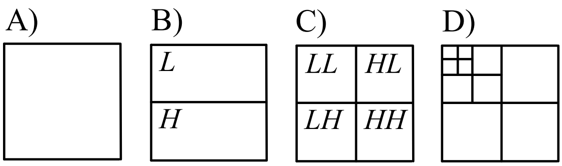


Fig. 1. 1-level 2D-DWT (A–C) and 3-level 2D-DWT (D).

2.2 Reversible denoising and lifting steps

The idea of skipping selected steps of a lifting-based transform originates from RDLS. An unwanted side effect of LS is that the sample being modified by LS (filtered by LS in the case of DWT) gets contaminated by noise from other samples. RDLS is a modification of LS (and consequently of the lifting-based transform) that integrates LS with denoising filters in order to avoid noise propagation while preserving other properties of LS (and of the lifting-based transform). Despite exploiting the inherently irreversible denoising, RDLS is perfectly reversible. RDLS was initially applied [12] to a simple reversible color space transform RDgDb [13] and then [16] to more complex color space transforms: LDgEb [13], RCT [5], and YCoCg-R [10] as well as to DWT [14]. Both in the case of reversible color space transforms (for three standard algorithms: JPEG-LS [23], JPEG 2000, and JPEG XR [4]) and the DWT exploited in lossless JPEG 2000, the application of RDLS resulted in practically useful improvements of image compression ratios of certain types of images. For example, in the latter case the lossless bitrates of non-photographic images were improved by about 14% [14].

For details and properties of the RDLS approach we refer the reader to [14, 15, 16] and describe below concisely the RDLS-modified DWT (RDLS-DWT). In RDLS-DWT the prediction (Eq. 1) and update (Eq. 2) steps are replaced by RDLS constructed based on them, i.e., by:

$$s_x \leftarrow s_x - \lfloor (s^d_{x-1} + s^d_{x+1})/2 \rfloor \text{ and} \tag{3}$$

$$s_x \leftarrow s_x + \lfloor (s^d_{x-1} + s^d_{x+1} + 2)/4 \rfloor, \tag{4}$$

respectively, where s^d_i denotes the denoised sample s_i. The reorder step remains unchanged.

The RDLS-modified transform is more general than the original one. Among others, by employing special denoising filters we may obtain, as a special case of the RDLS-modified transform, the unmodified transform. Such a special filter case denoted None, for which $s^d_i = s_i$, was proposed and investigated in [14]. Another special filter, the Null filter proposed in [16] (for which $s^d_i = 0$), allows to practically skip the lifting step.

2.3 Skipping steps of discrete wavelet transform

In [14] we found that the noise filtering in RDLS-DWT was the most effective in improving lossless JPEG 2000 bitrates when applied during computing of some RDLS-DWT subbands only and since in some cases the best bitrates were obtained when the DWT stage of JPEG 2000 was skipped, we suspected that similarly to denoising, the optimum might be in-between skipping and applying DWT. As opposed to RDLS-modified color space transforms, employing the None filter in RDLS-DWT does not allow to skip the entire transform. Color space transforms consist of nothing but the lifting steps, whereas DWT and RDLS-DWT, besides the lifting prediction and update steps (that may be turned into $s_x \leftarrow s_x$ by employing the Null filter in RDLS-DWT), perform the sample reordering steps.

Therefore in [17] we proposed SS-DWT obtained from DWT by skipping selected steps of its computation. We employed a heuristic for selecting steps to be skipped in an image-adaptive way and defined two simple fixed SS-DWT variants described in the next Section. The most interesting results, from a practical standpoint, were obtained by applying entropy estimation of JPEG 2000 coding effects for selecting among the fixed SS-DWT variants, the unmodified DWT, and the skipping of the DWT stage. The average bitrate improvement due to selecting among the above-mentioned fixed variants was similar to that of RDLS-DWT (roughly 14% for non-photographic images), but it was obtained at a significantly smaller cost; the overall compression time was only 3% greater than that of the unmodified JPEG 2000. By combining SS-DWT and RDLS-DWT even greater bitrate improvements (of up to about 17.5% for non-photographic images) might be obtained at a significantly increased cost of heuristic-based selecting, based on the actual bitrate instead of an estimated one, of the steps to be skipped and the denoising filters. For brevity, we refer the reader to [17] for a detailed description and properties of the general SS-DWT case and describe in the next Section only the fixed variants.

2.4 Fixed variants of skipped steps discrete wavelet transform

In [17], for experiments we used a JPEG 2000 implementation compliant to parts 1 and 10 of the JPEG 2000 standard (core coding system [5] and extensions for three-dimensional data [7], respectively) modified by replacing DWT with SS-DWT. The modification made the compression process not compliant with JPEG 2000. Furthermore, the characteristics of images transformed by SS-DWT were different to what was expected by the entropy coder of JPEG 2000 (parts 1 and 10), and therefore the transformed image was not encoded in the most efficient way. For instance, if the reorder step for LH and HH subbands got skipped, then these subbands were not created, but the core JPEG 2000 entropy coder unaware of the employed SS-DWT still encoded such data as 2 separate subbands.

On the other hand, the fixed variants of SS-DWT that were proposed in [17] and were found the most useful from a practical standpoint are, as opposed to the general SS-DWT case, compliant with the JPEG 2000 part 2 standard. Therefore in this work, we investigate the effects of applying the fixed SS-DWT variants in a JPEG 2000 implementation compliant to part 2 of the standard. We show how the fixed SS-DWT variants may in practice be easily obtained using an unaltered JPEG 2000 part 2 implementation and check if the SS-DWT effects get improved by exploiting extensions of part 2 of the standard.

When selecting a fixed SS-DWT variant for a specific image, besides two new variants named FIX1 and FIX2, we allowed choosing the unmodified DWT and skipping the entire DWT stage of JPEG 2000 (both being special SS-DWT cases). All in all, the following variants of fixed transforms for JPEG 2000 compression were exploited in the previous research:

- FIX1 – a variant of SS-DWT, in which we skip all the update steps;
- FIX2 – a variant of SS-DWT, in which we skip all the update steps (as in FIX1) and additionally skip the prediction step for the HH subband as well as the reorder step for HH and LH;
- NO_DWT – skipping the entire DWT (obtained practically by applying the 0-level DWT);
- DWT – compression with the unmodified part-1-compliant DWT.

Noteworthy, FIX2 results in a decomposition of an image into fewer subbands than the regular DWT or FIX1 because HH and LH subbands are not created from the H subband that remains unchanged after the transform—see Fig. 2.

As we noticed in [17], the FIX1 and FIX2 variants of SS-DWT are compliant with the JPEG 2000 part 2 standard because they may be obtained using an unmodified part 2-compliant compressor. The extension defined in Annex H of JPEG 2000 part 2 standard allows specifying arbitrary wavelet kernels. The FIX1 variant may be obtained by simply defining a kernel, that uses regular prediction step of the DWT (Eq.1) and skips the update by defining it to be $s_x \leftarrow s_x$. Since the arbitrary kernel is defined for a given tile (of a given component), the more complex variants of SS-DWT cannot be obtained this way.

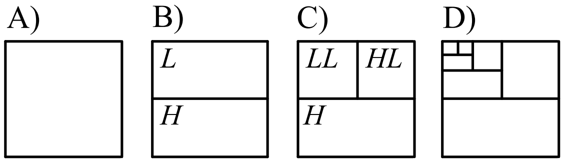


Fig. 2. 1-level FIX2 variant of SS-DWT (A–C) and 3-level FIX2 (D).

The extension defined in Annex F of JPEG 2000 part 2 describes arbitrary decompositions of tile-components. Among others, at each decomposition level, we may skip performing 1D-DWT in horizontal or vertical direction. As opposed to regular DWT that results in 4 subbands, performing 1D-DWT in just 1 direction results in 2 subbands. The JPEG 2000 part 2 entropy coding of such subbands may be more efficient than that of part 1 applied to SS-DWT subbands because the coder is aware of the actual decomposition applied. The transform at level $t+1$ is applied to the low-pass subband obtained with a t-level transform and we may specify decomposition structure individually for each level. At each decomposition level of FIX2 we need to skip 1D-DWT for LH and HH subbands and perform it for LL and HL. This is not directly supported by Annex H, but the 1-level FIX2 may be substituted by a 2-level part 2-compliant decomposition that at level 1 is performed only in vertical and at level 2 only in horizontal direction. Therefore, the 3-level FIX2 SS-DWT variant may be obtained by using an arbitrary DWT kernel with the skipped update step and a 6-level arbitrary decomposition that at odd levels is performed only in vertical direction and at even levels only in horizontal direction.

2.5 Procedure

In experiments, we used the green components of images from a CT2 set[1]. We chose this test data set in order to make the results obtained herein directly comparable to results of our earlier research on RDLS-DWT and SS-DWT, which were reported in [14] and [17], respectively. The CT2 is a recent, large set of color images that besides our earlier research was used in the research on lifting-based color space transforms [18, 19]. It contains 746 images taken from different sources; image sizes are from 180×117 to 6600×5100. The set was divided into subsets: Photo, consisting of 499 natural continuous tone photographs, and No-photo, consisting of 247 non-photographic images (e.g., computer-generated, composed from others including natural ones, or being screenshots). It is worth mentioning, that there currently is a growing interest in compression of the latter type of images, that are known as the screen content images; the most recent standard of video and image compression HEVC [8, 20] includes an extension for screen content image compression [11, 24].

[1] http://www1.hft-leipzig.de/strutz/Papers/Testimages/

In this study, we compare SS-DWT effects on the JPEG 2000 compressors compliant to parts 1 and 2 of the standard. Part 1 results are extracted from [17]. They were obtained using the RDLS-SS-DWT version 0.9[2]—our implementation of SS-DWT (and of RDLS-DWT). RDLS-SS-DWT is available as a patch to the IRIS-JP3D version 1.1.1[3]—a JPEG 2000 part 10 [2, 7] reference software, downward compatible with the core JPEG 2000 standard, developed by Tim Bruylants from Vrije Universiteit Brussel (VUB) and the Interdisciplinary Institute for BroadBand Technology (IBBT). For JPEG 2000 part 2-compliant compression we used the popular Kakadu implementation, version 7.8[4] by Kakadu Software. In the experiments, the entire image was compressed as a single tile, and we used the 3-level decomposition (0-level for the NO_DWT variant). In order to make our research easily reproducible and directly applicable in practice, in Table 1 we present the command-line options of the kdu_compress tool, from the Kakadu package, we used to obtain the FIX1, FIX2, NO_DWT, and DWT transform variants.

Table 1. Commandline options of kdu_compress used to obtain fixed SS-DWT variants exploited in experiments.

Transform variant	Command-line options
FIX1	-i infile.pgm -o outfile_FIX1.jpx Creversible=yes Clevels=3 Catk=2 Kextension:I2=SYM Kreversible:I2=yes Ksteps:I2=2,0,1,1,1,0,0,0 Kcoeffs:I2=-0.5,-0.5,0
FIX2	-i infile.pgm -o outfile_FIX2.jpx Creversible=yes Clevels=6 Cdecomp=V(-),H(-),V(-),H(-),V(-),H(-) Catk=2 Kextension:I2=SYM Kreversible:I2=yes Ksteps:I2=2,0,1,1,1,0,0,0 Kcoeffs:I2=-0.5,-0.5,0
NO_DWT	-i infile.pgm -o outfile_NO_DWT.jpx Creversible=yes Clevels=0
DWT	-i infile.pgm -o outfile_DWT.jpx Creversible=yes Clevels=3

The compression ratio or bitrate r, expressed in bits per pixel (bpp), is calculated using the total size in Bytes of the compressed image including the compressed file format header. The bitrate is directly proportional to the compressed file size, hence smaller bitrate means better compression result. The effects of selection of SS-DWT variants on JPEG 2000 bitrate were analyzed based on bitrate changes with respect to the bitrate of the reference method. For part 1 results the unmodified IRIS-JP3D was the reference method, and Kakadu for part 2 results—in both cases reference bitrates were obtained by invoking coder with default parameters, except for setting the 3-level decomposition. The bitrate

[2] http://sun.aei.polsl.pl/~rstaros/rdls-ss-dwt/
[3] http://www.irissoftware.be/
[4] http://kakadusoftware.com/downloads/

change Δr was expressed in percentage of the reference method bitrate. Due to the large size of our test-set, we report averaged bitrates and averaged bitrate changes for a set and for its specific subsets rather than results for individual images.

For estimation of JPEG 2000 coding effects, in order to quickly select a fixed SS-DWT variant, we used an estimator that was found effective in the previous research, i.e., the memoryless entropy $H0$ of the t-level DWT or SS-DWT transformed image. $H0$ was computed as a sum of memoryless entropies of all subbands that would be independently encoded by core JPEG 2000 with an unmodified t-level DWT, i.e., 10 subbands for 3-level transform, regardless of the actual decomposition applied (recall Fig. 1.D). Subband entropy, calculated as $-\sum_{i=0}^{N-1} p_i \log_2 p_i$, where N is the alphabet size, and p_i is the probability of occurrence of a sample value i in the subband, was weighted with the size (number of contained samples) of the subband.

3 Results and discussion

In Fig. 3 we report bitrate changes, the average for all the images, due to applying fixed SS-DWT variants in JPEG 2000 part 1 and part 2-compliant compressors. Results of fixed variants FIX1 and FIX2 are presented and NO_DWT results are not included because the latter variant resulted in significant data expansion. Finally, Fig. 3 presents effects of selecting, for each image, the best variant out of some or all fixed SS-DWT variants described in Section 2.4 (i.e., FIX1, FIX2, DWT, and NO_DWT); we report effects of selecting the fixed variant based on the actual JPEG 2000 bitrates obtained using respective JPEG 2000 compressor (described in Section 2.5) and based on the entropy estimation using $H0$.

In Table 2 we present average bitrate changes, due to applying fixed SS-DWT variants in JPEG 2000 part 1 and part 2-compliant compressors, obtained for all images and for Photo and No-photo image groups. Presented are results for individual FIX1, FIX2 and NO_DWT variants of SS-DWT as well as effects of image-adaptive selecting, based on actual bitrates or on entropy estimation, of the best variant out of either FIX1, FIX2, DWT, and NO_DWT or FIX1, FIX2, and DWT. For brevity, we include only the sets of variants that resulted in the best bitrate improvements for various image groups (All, Photo, No-photo) or variant selection methods (bitrate or entropy estimation-based). Presented also are average absolute bitrates obtained for the unmodified DWT.

First, let us examine the average bitrate changes for all images. Looking at the effects of replacing standard DWT with fixed SS-DWT variants FIX1 and FIX2 in part 1 and part 2-compliant compressors we notice that FIX2 is noticeably more efficiently encoded by the part 2-compliant compressor (as these compressors obtained average bitrate improvements of 4.47% and 4.88%, respectively). The FIX1 variant, that results in the same decomposition of an image into subbands as the DWT (the decomposition is the same, the characteristics of subbands differs), is encoded similarly efficiently by both compressors. Consequently, when for each image we select a SS-DWT variant from a set of up to

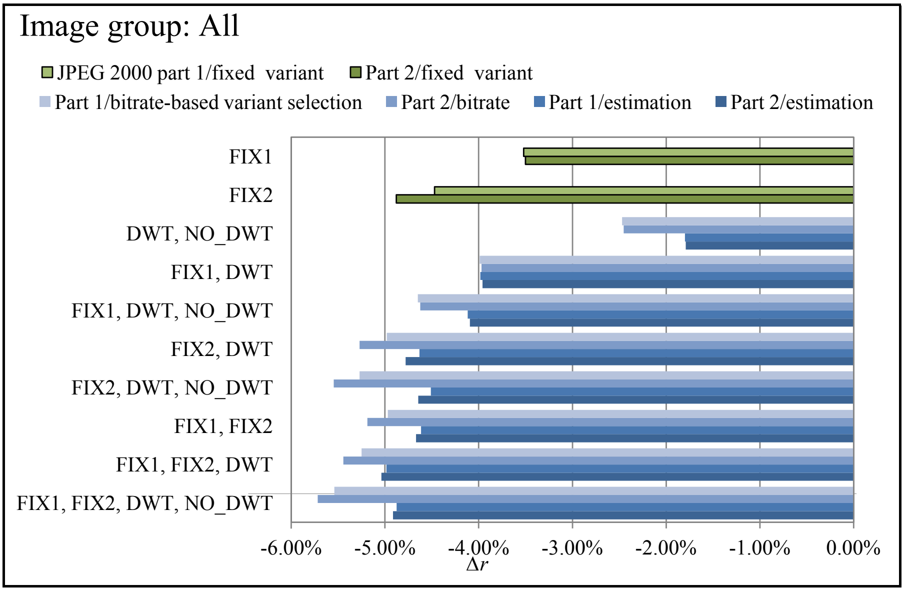


Fig. 3. Comparison of bitrate changes due to applying fixed SS-DWT variants in JPEG 2000 part 1 and 2-compliant compressors, averaged for all CT2 images. Results plotted for individual fixed variants FIX1 and FIX2 (NO_DWT not plotted) and for selecting the best out of several variants; selection of the variant based on the actual JPEG 2000 bitrates and the entropy-estimated ones.

Table 2. Effects of selected SS-DWT variants on bitrates of JPEG 2000 part 1 and 2.

Transform variant	JPEG 2000 part 1			JPEG 2000 part 2		
	All	Photo	No-photo	All	Photo	No-photo
r_{DWT}	3.6395	3.9975	2.9162	3.6452	4.0021	2.9241
Δr_{FIX1}	-3.52%	0.15%	-10.94%	-3.50%	0.15%	-10.88%
Δr_{FIX2}	-4.47%	0.30%	-14.11%	-4.88%	0.04%	-14.82%
$\Delta r_{\mathrm{NO\_DWT}}$	23.33%	25.59%	18.77%	23.30%	25.56%	18.72%
$\Delta r_{\mathrm{FIX1,FIX2,DWT/bitrate}}$	-5.25%	-0.62%	-14.61%	-5.44%	-0.68%	-15.07%
$\Delta r_{\mathrm{FIX1,FIX2,DWT/estimation}}$	-4.98%	-0.47%	-14.10%	-5.04%	-0.47%	-14.26%
$\Delta r_{\mathrm{FIX1,FIX2,DWT,NO\_DWT/bitrate}}$	-5.54%	-0.62%	-15.48%	-5.72%	-0.68%	-15.89%
$\Delta r_{\mathrm{FIX1,FIX2,DWT,NO\_DWT/estimation}}$	-4.87%	-0.47%	-13.77%	-4.91%	-0.47%	-13.89%

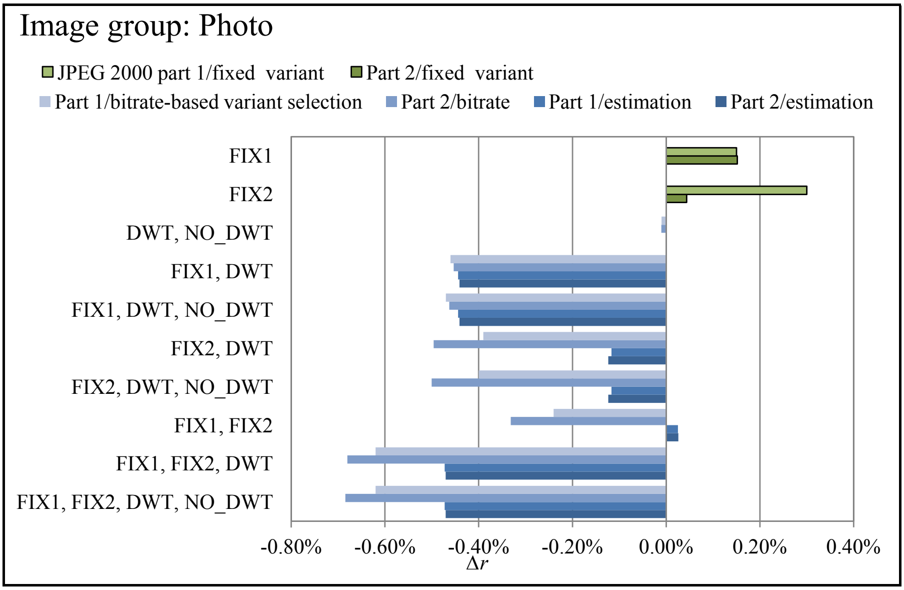
r_{DWT} – average JPEG 2000 bitrate obtained for the unmodified 3-level DWT (bpp), $\Delta r_{\mathrm{variant\_list/selection\_criterion}}$ – average bitrate change obtained by using for each image the one of the listed SS_DWT variants that was found the best based on the selection_criterion being the actual JPEG 2000 bitrate (bitrate) or entropy-estimated one (estimation).

4 variants (FIX1, FIX2, DWT, and NO_DWT) the greater differences between results of part 1 and part 2-compliant compressors appear when the set contains the FIX2 variant—both in the case when the selection is based on the actual bitrate of the compressor and on the entropy estimation. Differences in bitrate improvements are smaller when we employ entropy estimation, which may be attributed to using imperfect estimation based on standard DWT decomposition structure and memoryless entropy.

In Table 2 we see that skipping the entire DWT stage worsens bitrates significantly on average for the whole set (however, for some images skipping DWT is beneficial). By selecting DWT or NO_DWT for each image individually we get an average bitrate improvement of roughly 2% and this is the greatest improvement we obtained without SS-DWT (NO_DWT may be considered a special case 0-level DWT). Much greater improvement, roughly 2 to 2.5 times greater depending on the criterion used to select DWT or NO_DWT, is obtainable without the need to select one from several variants by just employing FIX2.

The FIX2 variant alone obtains the average bitrate improvement of 4.88%, which is a significant improvement as for lossless image compression and it is the majority of the bitrate improvement attainable with the more complex image-adaptive selection of fixed SS-DWT variants. In [17] the computational time cost of compression schemes exploiting SS-DWT and RDLS-DWT was expressed in comparison to the actual cost of unmodified JPEG 2000 part 1 compression with IRIS-JP3D. FIX2 is actually simpler than DWT, fewer steps are needed to compute it and its cost was 3% smaller than that of DWT; in our part 2-compliant compressor FIX2 is obtained by means of increasing the number of levels of transform that in turn is simplified compared to DWT—anyway the cost of FIX2 roughly equals the cost of DWT. The compression exploiting entropy estimation-based selection between FIX1, FIX2, and DWT is quick (was 3% slower than using just DWT), but results in an improvement greater by 0.16 percentage points than FIX2. Actual bitrate-based selection of the variant (from: FIX1, FIX2, DWT, and NO_DWT) allows average improvement of bitrate greater by 0.84 percentage points than FIX2, at the cost of 2.84 times longer compression time (or nearly 4 times longer if we naively run the compressor, treated as a black-box, four times for four SS-DWT variants). All in all, for the whole test image set, the following two variants seem practically useful: FIX2 gives significant bitrate improvements without increasing the compression time, bitrate-based selection of variant from FIX1, FIX2, DWT, and NO_DWT gives noticeable further improvements at an increased computational time cost. We remark that using the general SS-DWT case combined with a technique it originates from, i.e., RDLS, further bitrate improvements may be obtained at the cost of giving up the JPEG 2000 compliance and further complexity increase.

In Fig. 4 we report the average JPEG 2000 bitrate changes, due to applying fixed SS-DWT variants, for Photo (top panel) and No-Photo (bottom panel) images. Looking at the results for Photo images we see that applying to all Photo images a single fixed variant worsens bitrates, but in the case of FIX2 and the part 2-compliant compressor, the bitrate worsening is negligible (as opposed



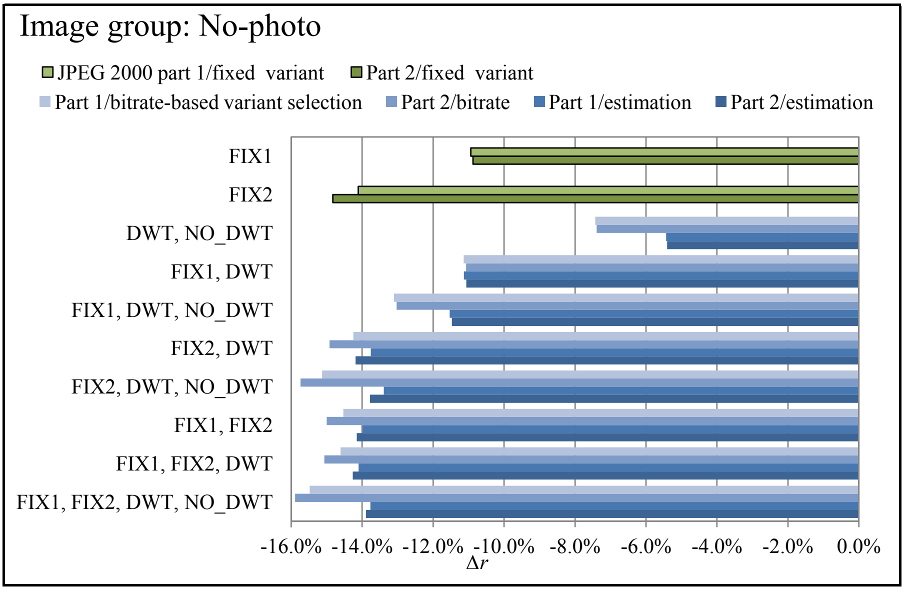


Fig. 4. Comparison of bitrate changes due to applying fixed SS-DWT variants in JPEG 2000 part 1 and 2-compliant compressors, averaged for Photo (top panel) and No-photo (bottom panel) images. Results plotted for individual fixed variants FIX1 and FIX2 (NO_DWT not plotted) and for selecting the best out of several variants; selection of the variant based on the actual JPEG 2000 bitrates and the entropy-estimated ones.

to the part 1-compliant compressor, where FIX2 worsened bitrates noticeably). Both in the case of part 1 and part 2-compliant compressors, improvements of roughly 0.5% may be obtained by selecting among fixed variants (FIX1, FIX2, and DWT) based on fast entropy estimation—as opposed to results for all images, for Photo images the entropy estimation presents a good trade-off between the bitrate improvement and its cost. A smaller further improvement (of up to about 0.3 percentage points in the case of the part 2 compressor and selecting of variant form FIX1, FIX2, DWT, and NO_DWT) may be obtained at a significantly greater cost of the bitrate-based selection of fixed variant.

For No-photo images the bitrate improvements due to SS-DWT are much larger. Using FIX2 for all No-photo images results in average improvement of almost 15% in the case of the part 2-compliant compressor; all entropy estimation-based variants obtain lower improvements on these images. The greatest bitrate improvement of almost 16% may be obtained at a significantly greater cost of the bitrate-based selection of fixed variant between FIX1, FIX2, DWT, and NO_DWT.

Generally, besides the noticeably better effects of the FIX2 variant of SS-DWT, the bitrate improvements obtained by part 1 and part 2 compressors are similar to each other (if the difference is noticeable, then the part 2 is better). From the practical standpoint, however, part 2 results are much more important, as obtaining these improvements does not require modifying the compressor nor the decompressor and thus the SS-DWT-based compression scheme is compliant to the JPEG 2000 standard. The above statement is valid for both the individual fixed variants and the image-adaptive selecting of fixed variants. Noteworthy the JPEG 2000 standard, both part 1 [5] and part 2 [6], defines the syntax of the code stream with the compressed image and the decoding process—these parts of standards are called normative, there are also informative parts, that present the examples of standard-conforming encoding procedures. The compression algorithm, in order to be JPEG 2000-compliant, has to output a code stream that can be correctly decompressed by a JPEG 2000-compliant decompressor. Hence, we obtained significant bitrate improvements in compliance with the JPEG 2000 part 2 standard.

4 Conclusion

The crucial practical advantage of using the FIX1 and FIX2 fixed variants of SS-DWT in a compressor compliant to part 2 of the JPEG 2000 standard, as opposed to a modified JPEG 2000 part 1 compressor, is that such compression process remains compliant with the standard. That is, we get the compression scheme that does not require to modify the already existing compressors (or decompressors) and the compressed images can be decompressed with any JPEG 2000 part 2-compliant decompressor. Previous research was done for modified JPEG 2000 part 1 compressor. In this work, we compared effects of applying fixed SS-DWT variants on the bitrates obtained by JPEG 2000 part 1 and part 2 compressors. Besides FIX1 and FIX2 we allowed using standard transform or skipping the transform stage entirely; experiments were performed for a large and diverse test set of images divided into Photo and No-photo groups.

The greatest differences in SS-DWT effects were observed for FIX2 that either significantly improves bitrates (of No-photo images, by almost 15%) or results in a negligible bitrate change (an increase by 0.04% in the case of Photo images; for these images, FIX2 applied in JPEG 2000 part 1 compressor worsened bitrates noticeably). For the entire test image set, FIX2 resulted in the average bitrate improvement of 4.88%. Simply using FIX2 instead of DWT appears a practical and immediately applicable compression scheme compliant to the JPEG 2000 part 2, that without increasing the compression or decompression process complexity attains the majority of the bitrate improvement attainable by using the several times more complex non-standard schemes it originates from (i.e., in [17] by combining the general case SS-DWT and RDLS-DWT we obtained average improvement of bitrates of No-photo images of about 17.5%).

Further bitrate improvement requires an image-adaptive selection of the fixed SS-DWT variants; two such compression schemes compliant with JPEG 2000 part 2 may be practically interesting:

- selecting, based on entropy estimation, a variant from: FIX1, FIX2, and DWT—at a very small cost of performing the estimation-based variant selection we get scheme that improves bitrates of both kinds of images; bitrates of Photo images are improved by about 0.5%, however, improvements of No-photo images (of over 14%) are noticeably smaller than in the case of simply applying FIX2 to every image, the improvement for the entire set is 5.04%;
- selecting, based on actual JPEG 2000 bitrates, a variant from: FIX1, FIX2, DWT, and NO_DWT—this way we get the greatest bitrate improvements of almost 16% for No-photo and almost 0.7% for Photo images, however, the compression time gets a few times longer than in the case of the unmodified JPEG 2000, the improvement for the entire set is 5.72%.

Acknowledgment

This work was supported by the 02/020/BK_17/0105 grant from the Institute of Informatics, Silesian University of Technology and by the 02/020/RGJ18/0123 grant from the Silesian University of Technology.

Bibliography

[1] Addison, P.S.: The Illustrated Wavelet Transform Handbook: Introductory Theory and Applications in Science, Engineering, Medicine and Finance. CRC Press (2017)

[2] Bruylants, T., Munteanu, A., Schelkens, P.: Wavelet based volumetric medical image compression. Signal Processing: Image Communication 31, 112–133 (2015), http://dx.doi.org/10.1016/j.image.2014.12.007

[3] Daubechies, I., Sweldens, W.: Factoring wavelet transforms into lifting steps. Journal of Fourier Analysis and Applications 4(3), 247–269 (1998), http://dx.doi.org/10.1007/BF02476026

[4] Dufaux, F., Sullivan, G.J., Ebrahimi, T.: The JPEG XR image coding standard. IEEE Signal Processing Magazine 26(6), 195–199,204 (2009), http://dx.doi.org/10.1109/MSP.2009.934187

[5] ISO/IEC, ITU-T: Information technology – JPEG 2000 image coding system: Core coding system (2004), ISO/IEC International Standard 15444-1 and ITU-T Recommendation T.800.

[6] ISO/IEC, ITU-T: Information technology – JPEG 2000 image coding system: extensions (2004), ISO/IEC International Standard 15444-2 and ITU-T Recommendation T.801.

[7] ISO/IEC, ITU-T: Information technology – JPEG 2000 image coding system: Extensions for three-dimensional data (2011), ISO/IEC International Standard 15444-10 and ITU-T Recommendation T.809.

[8] ISO/IEC, ITU-T: Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 2: High efficiency video coding (2015), ISO/IEC International Standard 23008-2 and ITU-T Recommendation H.265.

[9] Mallat, S.: A theory for multiresolution signal decomposition: the wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 11, 674–693 (1998), http://dx.doi.org/10.1109/34.192463

[10] Malvar, H.S., Sullivan, G.J., Srinivasan, S.: Lifting-based reversible color transformations for image compression. In: Applications of Digital Image Processing XXXI. Proc. SPIE, vol. 7073, p. 707307 (2008), http://dx.doi.org/10.1117/12.797091

[11] Peng, W.H., Walls, F.G., Cohen, R.A., Xu, J., Ostermann, J., MacInnis, A., Lin, T.: Overview of screen content video coding: Technologies, standards, and beyond. IEEE Journal on Emerging and Selected Topics in Circuits and Systems 6(4), 393–408 (2016), http://dx.doi.org/10.1109/JETCAS.2016.2608971

[12] Starosolski, R.: Reversible denoising and lifting based color component transformation for lossless image compression, arXiv:1508.06106 [cs.MM] (2016), http://arxiv.org/abs/1508.06106

[13] Starosolski, R.: New simple and efficient color space transformations for lossless image compression. Journal of Visual Communication and Image

Representation 25(5), 1056–1063 (2014), http://dx.doi.org/10.1016/j.jvcir.2014.03.003

[14] Starosolski, R.: Application of reversible denoising and lifting steps to DWT in lossless JPEG 2000 for improved bitrates. Signal Processing Image Communication 39(A), 249–263 (2015), http://dx.doi.org/10.1016/j.image.2015.09.013

[15] Starosolski, R.: Application of reversible denoising and lifting steps to LDgEb and RCT color space transforms for improved lossless compression. In: Proc. BDAS 2016. Springer CCIS, vol. 613, pp. 623–632 (2016), http://dx.doi.org/10.1007/978-3-319-34099-9_48

[16] Starosolski, R.: Application of reversible denoising and lifting steps with step skipping to color space transforms for improved lossless compression. Journal of Electronic Imaging 25(4), 043025 (2016), http://dx.doi.org/10.1117/1.JEI.25.4.043025

[17] Starosolski, R.: Skipping selected steps of DWT computation in lossless JPEG 2000 for improved bitrates. PLOS ONE 11(12), e0168704 (2016), http://dx.doi.org/10.1371/journal.pone.0168704

[18] Strutz, T.: Multiplierless reversible colour transforms and their automatic selection for image data compression. IEEE Transactions on Circuits and Systems for Video Technology 23(7), 1249–1259 (2013), http://dx.doi.org/10.1109/TCSVT.2013.2242612

[19] Strutz, T., Leipnitz, A.: Reversible colour spaces without increased bit depth and their adaptive selection. IEEE Signal Processing Letters 22(9), 1269–73 (2015), http://dx.doi.org/10.1109/LSP.2015.2397034

[20] Sullivan, G., Ohm, J., Han, W., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. IEEE Transactions on Circuits and Systems for Video Technology 22(12), 674–693 (2012), http://dx.doi.org/10.1109/TCSVT.2012.2221191

[21] Sweldens, W.: The lifting scheme: a custom-design construction of biorthogonal wavelets. Applied and Computational Harmonic Analysis 3, 186–200 (1996), http://dx.doi.org/10.1006/acha.1996.0015

[22] Taubman, D.S., Marcellin, M.W.: JPEG2000 Image Compression Fundamentals, Standards and Practice. Springer US (2004), http://dx.doi.org/10.1007/978-1-4615-0799-4

[23] Weinberger, M.J., Seroussi, G., Sapiro, G.: The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. IEEE Transactions on Image Processing 9(8), 1309–1324 (2000), http://dx.doi.org/10.1109/83.855427

[24] Xu, J., Joshi, R., Cohen, R.A.: Overview of the emerging hevc screen content coding extension. IEEE Transactions on Circuits and Systems for Video Technology 26(1), 50–62 (2016), http://dx.doi.org/10.1109/TCSVT.2015.2478706