Application of Reversible Denoising and Lifting Steps to LDgEb and RCT Color Space Transforms for Improved Lossless Compression

Roman Starosolski(✉)

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

Abstract. The lifting step of a reversible color space transform employed during image compression may increase the total amount of noise that has to be encoded. Previously, to alleviate this problem in the case of a simple color space transform RDgDb, we replaced transform lifting steps with reversible denoising and lifting steps (RDLS), which are lifting steps integrated with denoising filters. In this study, we apply RDLS to more complex color space transforms LDgEb and RCT and evaluate RDLS effects on bitrates of lossless JPEG-LS, JPEG 2000, and JPEG XR coding for a diverse image test-set. We find that RDLS effects differ among transforms, yet are similar for different algorithms; for the employed denoising filter selection method, on average the bitrate improvements of RDLS-modified LDgEb and RCT are not as high as of the simpler transform. The RDLS applicability reaches beyond image data storage; due to its general nature it may be exploited in other lifting-based transforms, e.g., during the image analysis for data mining.

Keywords: Image processing · Lossless image compression · Lifting technique · Denoising · Reversible denoising and lifting step · RDgDb · LDgEb · RCT · JPEG-LS · JPEG 2000 · JPEG XR

1 Introduction

Most color image compression algorithms independently compress the image components; since components in the RGB space are correlated, the compression is performed after transforming image data to a less correlated color space. For the lossless compression, the reversible color space transforms are employed, that are built using lifting steps [6]. In [9], we noticed that such step may increase the total amount of noise that must be encoded during compression. We replaced lifting steps with reversible denoising and lifting steps (RDLS), which are lifting steps integrated with denoising filters. We applied RDLS to a simple RDgDb [11] transform (also known as $A_{2,1}$ [14]) and found that RDLS improved bitrates of images in optical resolutions of acquisition devices. Experiments were performed for 3 significantly different standard image compression algorithms in lossless

S. Kozielski et al. (Eds.): BDAS 2016, CCIS 613, pp. 623–632, 2016.
DOI: 10.1007/978-3-319-34099-9_48

mode (LOCO-I/JPEG-LS[1] [16], JPEG 2000[2] [15], and HD-Photo/JPEG XR[3] [4,8]) and for a simple denoising filter. We found that the memoryless entropy of the component prediction error obtained with the nonlinear edge-detecting predictor MED [7,16] was a very efficient estimator of image component transform effects, suitable for selecting a filter for given image component. In this study, we apply RDLS to more complex color space transforms LDgEb [11] (denoted $A_{4,10}$ in [14]) and RCT (among others, used in JPEG 2000 standard) and evaluate RDLS effects using the same denoising filters, compression algorithms, and test images, as previously. Entropy estimation employing MED is used for selecting the denoising filter and deciding whether to exploit denoising.

The remainder of this paper is organized as follows. In Sect. 2 we briefly characterize RDLS, present the proposed RDLS-modified transforms along with the filter selection method and the denoising filters used, and describe the implementations and test data. Results are presented and discussed in Sect. 3; Sect. 4 summarizes the research.

2 Materials and Methods

2.1 RDLS

In [9] we proposed to replace color space transform lifting steps (Eq. 1) with RDLS (Eq. 2):

$$C_x \leftarrow C_x \oplus \mathrm{f}(C_1, \ldots, C_{x-1}, C_{x+1}, \ldots, C_n) \tag{1}$$

$$C_x \leftarrow C_x \oplus \mathrm{f}(C_1^d, \ldots, C_{x-1}^d, C_{x+1}^d, \ldots, C_n^d) \tag{2}$$

where C_i is the i-th component of the pixel, C_x is the component which is modified by the step, C_i^d is the denoised i-th component of the pixel, n is the number of components, and the operation $\oplus$ is reversible. Different denoising filters may be used for different components. For denoising of arguments of function f we may use any component of any pixel, but the C_x of the pixel to which the RDLS is being applied; in this study for denoising of C_i of specific pixel we use C_i of this pixel and of its neighbors. RDLS, like the lifting step it originates from, is trivially and perfectly invertible and may be computed in-place. However note, that denoising exploited in RDLS is irreversible and it is not an in-place operation—computing the function f argument C_i^d does not alter C_i. For more detailed characteristics of RDLS and examples of its application we refer the Reader to [9,12] and to the next subsection.

[1] Information technology–Lossless and near-lossless compression of continuous-tone still images–Baseline, ISO/IEC International Standard 14495-1 and ITU-T Recommendation T.87 (2006).

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

[3] Information technology–JPEG XR image coding system–Image coding specification, ISO/IEC International Standard 29199-2 and ITU-T Recommendation T.832 (2012).

2.2 RDLS-Modified Transforms

Here we show the application of RDLS to RDgDb and, for brevity, present only the RDLS-modified variants of LDgEb and RCT. Equation 3 shows the RDgDb transform, as it was presented in [11], where the definitions of other unmodified transforms may also be found. To obtain RDLS-modified RDgDb (RDLS-RDgDb) we first rewrite RDgDb using the notation as in Eqs. 1 and 2, where the same symbol denotes the pixel's component both before and after modifying its value by the lifting step or by the RDLS. Equation 4 persents RDgDb (both forward [left-hand side] and inverse) as a sequence of lifting steps; generally, the steps must be performed in a specified order. C_1, C_2, and C_3 denote R, G, and B components of the untransformed image, respectively, and R, Dg, and Db components of the transformed image, respectively. Next, we simply replace lifting steps (Eq. 1) in Eq. 4 with RDLS (Eq. 2) constructed based on them and obtain RDLS-RDgDb (Eq. 5). The employed denoising filters and method of selecting the filter for a given image component is described in the following subsection.

$$\begin{aligned} R &= R \\ Dg &= R - G \\ Db &= G - B \end{aligned} \iff \begin{aligned} R &= R \\ G &= R - Dg \\ B &= G - Db \end{aligned} \tag{3}$$

$$\begin{aligned} &\text{step 1: } C_3 \leftarrow -C_3 + C_2 \\ &\text{step 2: } C_2 \leftarrow -C_2 + C_1 \\ &\text{step 3: } C_1 \leftarrow C_1 \end{aligned} \iff \begin{aligned} &\text{step 1: } C_1 \leftarrow C_1 \\ &\text{step 2: } C_2 \leftarrow -C_2 + C_1 \\ &\text{step 3: } C_3 \leftarrow -C_3 + C_2 \end{aligned} \tag{4}$$

$$\begin{aligned} &\text{step 1: } C_3 \leftarrow -C_3 + C_2^d \\ &\text{step 2: } C_2 \leftarrow -C_2 + C_1^d \\ &\text{step 3: } C_1 \leftarrow C_1 \end{aligned} \iff \begin{aligned} &\text{step 1: } C_1 \leftarrow C_1 \\ &\text{step 2: } C_2 \leftarrow -C_2 + C_1^d \\ &\text{step 3: } C_3 \leftarrow -C_3 + C_2^d \end{aligned} \tag{5}$$

The RDLS-RDgDb transform (Eq. 5) may be presented using component symbols like in standard definitions of color space transforms (Eq. 6). Note, that the same symbols denote components of regular transform and its RDLS-modified variant.

$$\begin{aligned} &\text{step 1: } Db = -B + G^d \\ &\text{step 2: } Dg = -G + R^d \\ &\text{step 3: } R = R \end{aligned} \iff \begin{aligned} &\text{step 1: } R = R \\ &\text{step 2: } G = -Dg + R^d \\ &\text{step 3: } B = -Db + G^d \end{aligned} \tag{6}$$

In Eq. 7 we present the RDLS-LDgEb transform and in Eq. 8 the RDLS-RCT transform, that were obtained from LDgEb and RCT, respectively. The floor of division by integer power of 2 is computed using the arithmetic right shift.

$$\begin{aligned} &\text{step 1: } Dg = -G + R^d \\ &\text{step 2: } L = R - \lfloor Dg^d/2 \rfloor \\ &\text{step 3: } Eb = B - L^d \end{aligned} \iff \begin{aligned} &\text{step 1: } B = Eb + L^d \\ &\text{step 2: } R = L + \lfloor Dg^d/2 \rfloor \\ &\text{step 3: } G = -Dg + R^d \end{aligned} \tag{7}$$

$$\begin{array}{ll} \text{step 1:} & Cv = R - G^d \\ \text{step 2:} & Cu = B - G^d \\ \text{step 3:} & Y = G + \lfloor (Cv^d + Cu^d)/4 \rfloor \end{array} \iff \begin{array}{ll} \text{step 1:} & G = Y - \lfloor (Cv^d + Cu^d)/4 \rfloor \\ \text{step 2:} & B = Cu + G^d \\ \text{step 3:} & R = Cv + G^d \end{array} \tag{8}$$

It is worth noting, that components of RDLS-LDgEb and RDLS-RCT, as opposed to RDLS-RDgDb, may require greater bit depths, than their non-RDLS equivalents. In research reported herein, we used increased component depth only when component pixels actually exceeded original depth; in all such cases extending the depth by 1 bit was sufficient.

2.3 Denoising Filters and Filter Selection

For denoising we employed 11 low-pass linear averaging filters (smoothing filters) with 3×3 pixel windows. The filtered pixel component C_i^d was calculated as a weighted arithmetic mean of the C_i components of pixels from the window. The weight of the window center point was different for different filters—from 1 to 1024 (integer powers of 2 only), while its neighbors' weights were fixed to 1.

In each RDLS-modified forward transform step s for all pixels of a given image a filter was selected individually for each component requiring denoising. E.g., in step 3 of forward RDLS-RCT (Eq. 8), 2 filters were selected for denoising of Cv and Cu and applied to all image pixels. All filter combinations were tested and we also allowed to not use the filtering. The combination resulting in the best estimated bitrate of the component being modified by the step s was used for actual compression. Thus we performed an exhaustive search of filters in a given step, however only step 3 of RDLS-RCT requires denoising of 2 components. Filter(s) selection for a given step was not revised based on compression effects of components transformed in other steps, therefore even assuming the perfect estimation of compression algorithm bitrate, modifying this way transform with RDLS may result in bitrate worsening. Filter selection must be passed to the decoder along with the compressed data, but its cost is negligible.

The memoryless entropy of the component prediction error obtained using MED predictor was used as an estimator of compressed component bitrate, overall 3-component image bitrate was estimated as sum of computed this way entropies of 3 components. We denote this estimation method as H0_pMED. The compression ratio or bitrate r, expressed in bits per pixel (bpp), is calculated as $r = 8l/m$, where m is the number of pixels in the image component, l is the length in Bytes of the file containing the compressed component, including compressed file header; smaller r denotes better compression. The memoryless entropy of the image component $H_0 = -\sum_{i=0}^{N-1} p_i \log_2 p_i$, where N is the alphabet size and p_i is the probability of occurrence of pixel component value i in the image component, is also expressed in bpp.

2.4 Implementations and Test Data Used

We used the following sets of images:

- Waterloo—Set ("Colour set") of images from the University of Waterloo[4];
- Kodak—Image set from the Kodak corporation[5];
- EPFL—Image set from the École polytechnique fédérale de Lausanne[6] [3];
- A1, A2, and A3—Image sets from the Silesian University of Technology[7];
- A1-red.3, A2-red.3, and A3-red.3—reduced size (3×) sets A1, A2, and A3.

Sets A1, A2, and A3 contain unprocessed photographic images in optical resolutions of acquisition devices, or (A3) as close to such resolution, as possible without interpolation of all components. Except for Waterloo, all images may be characterized as continuous-tone photographic. The most widely-known Waterloo set contains both photographic and artificial images; some of them are dithered, sharpened, have sparse histograms of intensity levels [10], are computer-generated or composed of others. The same image sets were used for experiments in the previous study, their detailed characteristics may be found in [9].

We used the Signal Processing and Multimedia Group, Univ. of British Columbia JPEG-LS implementation, version 2.2[8], JasPer implementation of JPEG 2000 by M. Adams, version 1.900[9] [1], and JPEG XR standard reference software[10].

3 Results and Discussion

In Tables 1, 2, and 3, for RDLS-RDgDb, RDLS-LDgEb, and RDLS-RCT, respectively, we report average entropies obtained using H0_pMED estimator and average bitrates for JPEG-LS, JPEG 2000, and JPEG XR compression algorithms in lossless mode, summed for all 3 image components. Entropy and bitrate changes due to RDLS with respect to non-RDLS transform variants are also reported—in columns labeled ΔH_0 for H0_pMED and Δr for compression algorithms. Figure 1 for the examined transforms presents average entropy and bitrate changes due to RDLS of individual transformed components and of the 3-component image; in electronic version of the paper components are presented using colors of RGB components from which they were transformed.

We examined RDLS effects for several transforms, compression algorithms, and test image sets. In majority of cases, the RDLS-RDgDb obtains the best bitrates among all RDLS-modified transforms. For all these transforms, the

[4] http://links.uwaterloo.ca/Repository.html.
[5] http://www.cipr.rpi.edu/resource/stills/kodak.html.
[6] http://documents.epfl.ch/groups/g/gr/gr-eb-unit/www/IQA/Original.zip.
[7] http://sun.aei.polsl.pl/~rstaros/optres/.
[8] http://www.stat.columbia.edu/~jakulin/jpeg-ls/mirror.htm.
[9] http://www.ece.uvic.ca/~mdadams/jasper/.
[10] Information technology–JPEG XR image coding system–Reference software, ISO/IEC International Standard 29199-5 and ITU-T Recommendation T.835 (2012).

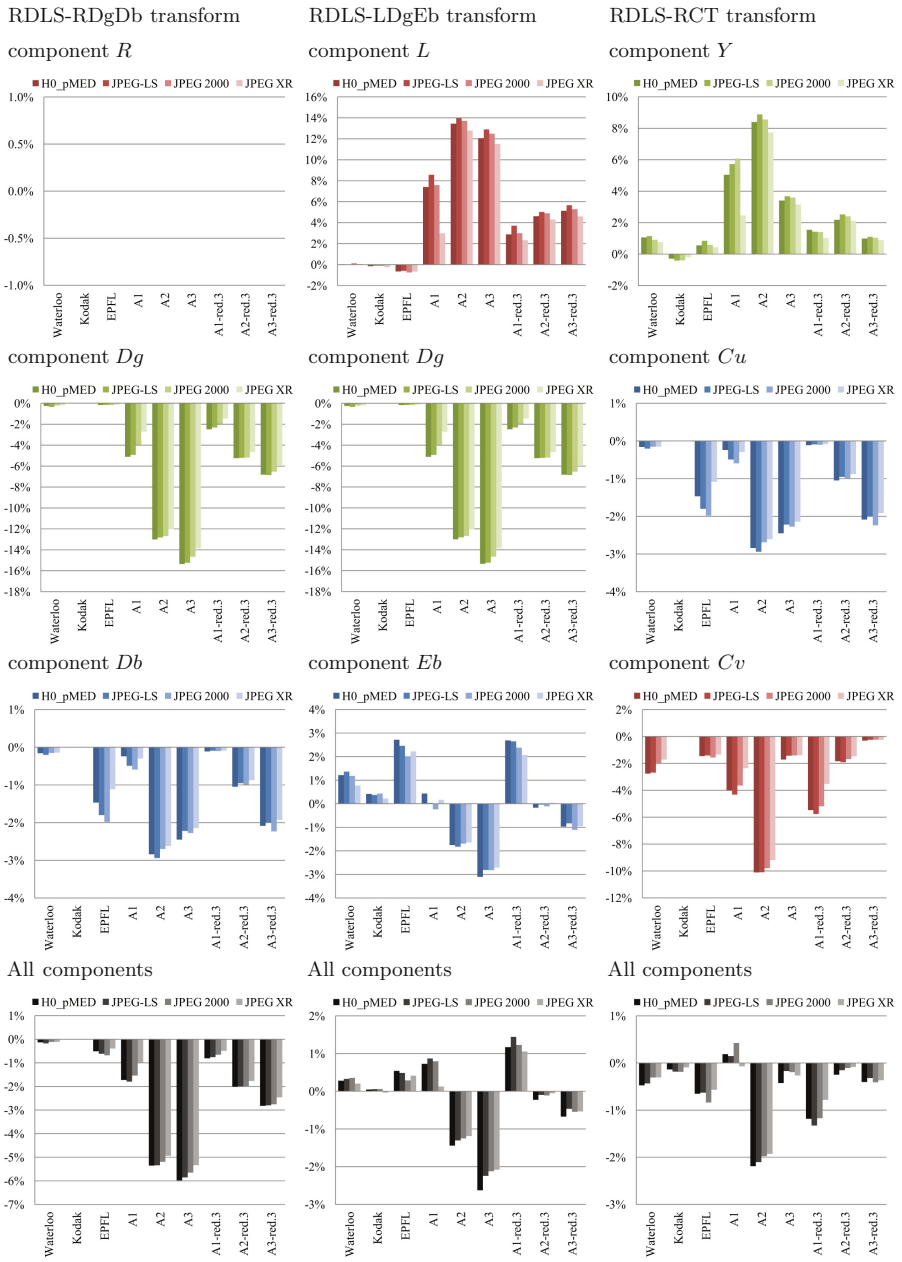

Fig. 1. Average entropy and bitrate changes due to RDLS for RDgDb (left-hand panels), LDgEb (middle panels), and RCT (right-hand panels) (Colour figure online).

Table 1. Effects of the RDLS-RDgDb transform and comparison to RDgDb

Images	H0_pMED		JPEG-LS		JPEG 2000		JPEG XR	
	H_0	ΔH_0	r	Δr	r	Δr	r	Δr
Waterloo	8.9014	−0.13 %	8.8498	−0.18 %	11.1299	−0.12 %	13.2834	−0.11 %
Kodak	10.2876	0.00 %	9.5676	0.00 %	9.4755	0.00 %	10.8725	0.00 %
EPFL	11.1806	−0.51 %	10.2746	−0.62 %	10.6515	−0.68 %	11.6264	−0.39 %
A1	6.7779	−1.73 %	6.2722	−1.80 %	6.2938	−1.54 %	8.1445	−0.99 %
A2	14.6469	−5.35 %	14.0353	−5.34 %	14.2550	−5.19 %	14.6526	−4.94 %
A3	15.1662	−5.98 %	14.5854	−5.85 %	14.7919	−5.65 %	15.3186	−5.34 %
A1-red.3	9.0112	−0.81 %	8.2770	−0.76 %	8.4625	−0.65 %	9.5931	−0.49 %
A2-red.3	14.0530	−2.01 %	13.4235	−1.99 %	13.8597	−1.98 %	14.5830	−1.77 %
A3-red.3	13.3956	−2.82 %	12.7130	−2.80 %	13.0802	−2.76 %	14.0082	−2.46 %

Table 2. Effects of the RDLS-LDgEb transform and comparison to LDgEb

Images	H0_pMED		JPEG-LS		JPEG 2000		JPEG XR	
	H_0	ΔH_0	r	Δr	r	Δr	r	Δr
Waterloo	9.0756	0.28 %	8.9888	0.33 %	11.2588	0.35 %	13.3249	0.21 %
Kodak	10.1512	0.05 %	9.4370	0.06 %	9.4282	0.06 %	10.8499	−0.04 %
EPFL	11.3739	0.54 %	10.4858	0.49 %	10.8297	0.29 %	11.7821	0.41 %
A1	6.9416	0.73 %	6.4348	0.87 %	6.4421	0.80 %	8.2329	0.13 %
A2	14.6249	−1.44 %	14.0454	−1.30 %	14.2524	−1.25 %	14.6502	−1.19 %
A3	15.3025	−2.63 %	14.7217	−2.24 %	14.9125	−2.12 %	15.4212	−2.08 %
A1-red.3	9.2698	1.17 %	8.5435	1.44 %	8.7113	1.23 %	9.8088	1.05 %
A2-red.3	14.1779	−0.23 %	13.5386	−0.10 %	13.9815	−0.11 %	14.7211	−0.05 %
A3-red.3	13.6814	−0.67 %	12.9811	−0.46 %	13.3597	−0.54 %	14.2952	−0.53 %

Table 3. Effects of the RDLS-RCT transform and comparison to RCT

Images	H0_pMED		JPEG-LS		JPEG 2000		JPEG XR	
	H_0	ΔH_0	r	Δr	r	Δr	r	Δr
Waterloo	8.9756	−0.48 %	8.9236	−0.43 %	11.1792	−0.31 %	13.2790	−0.31 %
Kodak	10.2758	−0.14 %	9.5577	−0.18 %	9.4893	−0.19 %	10.9110	−0.09 %
EPFL	11.2681	−0.65 %	10.3995	−0.63 %	10.7430	−0.84 %	11.6942	−0.57 %
A1	6.9397	0.19 %	6.4536	0.15 %	6.4821	0.42 %	8.2503	−0.07 %
A2	14.6337	−2.19 %	14.0611	−2.10 %	14.2587	−1.98 %	14.6532	−1.93 %
A3	15.2361	−0.43 %	14.6042	−0.17 %	14.7557	−0.19 %	15.2918	−0.27 %
A1-red.3	9.1002	−1.18 %	8.4063	−1.33 %	8.5820	−1.17 %	9.6624	−0.78 %
A2-red.3	14.0771	−0.25 %	13.4646	−0.15 %	13.8917	−0.11 %	14.6175	−0.08 %
A3-red.3	13.2992	−0.40 %	12.5899	−0.32 %	12.9520	−0.41 %	13.9179	−0.37 %

obtained bitrates significantly differ among the compression algorithms. Constantly, except for the Kodak set, JPEG-LS obtains the best bitrates and JPEG-XR the worst; for Kodak the JPEG 2000 is the best. However, the bitrate improvements due to RDLS are similar for different compression algorithms and entropy estimated bitrate.

On average, RDLS improves bitrates of all the examined transforms. The bitrate improvements of RDLS-modified LDgEb and RCT are not as high as in the case of the simpler RDgDb transform and for the former transforms RDLS sometimes results in bitrate worsening of 3-component image and, to a greater extent, of individual components. For these transforms we did not find any objective image feature allowing to predict RDLS effectiveness—as opposed to RDgDb, that is the most effective for images in optical resolutions of acquisition devices.

LDgEb and RCT transforms contain steps, during which a given component C_i is modified based on another one C_a which has already been modified based on C_i (see step 2 of LDgEb and step 3 of RCT). For example, let's look at steps 1 and 2 of forward LDgEb (its RDLS version is presented in Eq. 7). Step 1 modifies the G component, that from this moment is denoted as Dg, step 2 modifies R, that is then denoted as L. Step 2 of the regular lifting transform may decrease in L the amount of information which was originally present in R, that is of both noise and of the noise-free signal, and insert to L a fraction of information originally present in G (also consisting of noise and noise-free signal). Actually, as the transform is reversible, the information from R is beforehand (in step 1) propagated to Dg, and then in step 2 a part of it (as a fraction of Dg) is subtracted from L. In step 1 components are subtracted—the use of the noise-free signal from R is supposed to result in decreasing the correlation between R and Dg whereas noise from R adds to noise in Dg. When we employ RDLS, step 2 applied to R reduces the noise-free signal originally contained in it, but not the noise (noise from R is not present in Dg because filtering during step 1 prohibited propagating it); on the other hand RDLS avoids transferring to L the noise originally present in G. In the case of our images the former effect, complemented by component distortions introduced by imperfect denoising we use, has greater impact on the component bitrate, than the later—see component L (and Y for RDLS-RCT) in Fig. 1. Also the Eb component bitrate is worsened for some sets, which may be attributed to using for its calculation the denoised component L, bitrates of which are for some sets worsened by RDLS.

The filter selection method we use may be responsible for worse RDLS effects in the case of more complicated transforms. Since we select the best filters for a given step by analyzing filtering effects on bitrate of the component C_i being modified by this step only, the selection may not be optimal if C_i is then used in calculation of components modified in further steps. Better bitrates of such components and of overall 3-component image could be obtained by selecting filters based on overall 3-component image bitrate. Checking all filter combinations for all the steps would be too complex in practice, but employing a heuristics that would search for the best filters in a given step based on compression effects estimated for the entire image (similarly to [12]) seems worthwhile. We also note, that the selection process

complexity might be substantially reduced by using only a certain number of image pixels for transform effect estimation—in [13,14] such optimization applied to a similar problem resulted in a close to optimum estimation.

4 Conclusions

RDLS effects differ among transforms, yet are similar for JPEG-LS, JPEG 2000, and JPEG XR algorithms as well as for entropy of the component prediction errors obtained using MED. On average, RDLS improves bitrates of all the examined transforms. The bitrate improvements of RDLS-modified LDgEb and RCT are not as high as in the case of the simpler RDgDb transform and for the former transforms RDLS sometimes results in bitrate worsening; both effects may be attributed to the employed method of selecting the denoising filters. As opposed to RDgDb, we did not identify an objective image feature to which the RDLS bitrate improvement could be linked. As the RDLS effects clearly depend on denoising filters used, we expect that the application of better filters may further improve bitrates of the RDLS-modified color space transforms. For some images the denoising filter parameters might be determined directly from the acquisition process parameters [2]. The bitrate improvements we obtained for some of the test-sets are useful from practical standpoint. In the ongoing research, we investigate other filters and filter selection methods as well as simplified compression effect estimators, that jointly are expected to result in greater bitrate improvements obtained at a significantly lower filter selection cost. RDLS recently was found effective for a much more complex, involving more interdependent steps, multi-level 2D DWT transform in lossless JPEG 2000 compression [12], however, its applicability reaches beyond image data storage. Due to its general nature it may be exploited in other lifting-based transforms, e.g., during the image analysis for data mining and skin segmentation [5].

Acknowledgments. This work was supported by BK-263/RAU2/2015 grant from the Institute of Informatics, Silesian University of Technology.

References

1. Adams, M.D., Ward, R.K.: JasPer: a portable flexible open-source software tool kit for image coding/processing. In: 2004 Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004), vol. 5, pp. 241–244 (2004). doi:10.1109/ICASSP.2004.1327092
2. Bernas, T., Starosolski, R., Robinson, J.P., Rajwa, B.: Application of detector precision characteristics and histogram packing for compression of biological fluorescence micrographs. Comput. Methods Programs Biomed. **108**(2), 511–523 (2012). doi:10.1016/j.cmpb.2011.03.012
3. De Simone, F., Goldmann, L., Baroncini, V., Ebrahimi, T.: Subjective evaluation of JPEG XR image compression. In: Proceedings of the SPIE, Applications of Digital Image Processing XXXII, vol. 7443, p. 74430L (2009). doi:10.1117/12.830714

4. Dufaux, F., Sullivan, G.J., Ebrahimi, T.: The JPEG XR image coding standard. IEEE Sig. Process. Mag. **26**(6), 195–199, 204 (2009). doi:10.1109/MSP.2009.934187
5. Kawulok, M., Kawulok, J., Nalepa, J.: Spatial-based skin detection using discriminative skin-presence features. Pattern Recogn. Lett. **41**, 3–13 (2014). doi:10.1016/j.patrec.2013.08.028
6. Malvar, H.S., Sullivan, G.J., Srinivasan, S.: Lifting-based reversible color transformations for image compression. In: Proceedings of the SPIE, Applications of Digital Image Processing XXXI, vol. 7073, p. 707307 (2008). doi:10.1117/12.797091
7. Martucci, S.A.: Reversible compression of HDTV images using median adaptive prediction and arithmetic coding. In: Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 1310–1313 (1990)
8. Srinivasan, S., Tu, C., Regunathan, S.L., Sullivan, G.J.: HD Photo: a new image coding technology for digital photography. In: Proceedings of the SPIE, Applications of Digital Image Processing XXX, vol. 6696, p. 66960A (2007). doi:10.1117/12.767840
9. Starosolski, R.: Reversible denoising and lifting based color component transformation for lossless image compression (2015). arXiv:1508.06106 [cs.MM]
10. Starosolski, R.: Compressing high bit depth images of sparse histograms. In: Simos, T.E., Psihoyios, G. (eds.) International Electronic Conference on Computer Science. AIP Conference Proceedings, vol. 1060, pp. 269–272. American Institute of Physics, USA (2008). doi:10.1063/1.3037069
11. Starosolski, R.: New simple and efficient color space transformations for lossless image compression. J. Vis. Commun. Image Represent. **25**(5), 1056–1063 (2014). doi:10.1016/j.jvcir.2014.03.003
12. Starosolski, R.: Application of reversible denoising and lifting steps to DWT in lossless JPEG 2000 for improved bitrates. Sig. Process. Image Commun. **39**(A), 249–263 (2015). doi:10.1016/j.image.2015.09.013
13. Strutz, T.: Adaptive selection of colour transformations for reversible image compression. In: Proceedings of the 20th European Signal Processing Conference (EUSIPCO 2012), pp. 1204–1208 (2012)
14. Strutz, T.: Multiplierless reversible colour transforms and their automatic selection for image data compression. IEEE Trans. Circuits Syst. Video Technol. **23**(7), 1249–1259 (2013). doi:10.1109/TCSVT.2013.2242612
15. Taubman, D.S., Marcellin, M.W.: JPEG2000 Image Compression Fundamentals, Standards and Practice. The Springer International Series in Engineering and Computer Science, vol. 642. Springer, New York (2004). doi:10.1007/978-1-4615-0799-4
16. Weinberger, M.J., Seroussi, G., Sapiro, G.: The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS. IEEE Trans. Image Process. **9**(8), 1309–1324 (2000). doi:10.1109/83.855427