Lossless Image Compression using Wavelets over Finite Rings and Related Architectures

A. Klappenecker, F. U. May, and A. Nückel

Universität Karlsruhe, IAKS, Am Fasanengarten 5, D-76128 Karlsruhe, Germany
waveletQira.uka.de

ABSTRACT

In this paper we give a brief introduction to filter banks over commutative rings. In contrast to filter banks over the real numbers, we employ finite ring arithmetic to control the number of bits in the signal representations. This way we avoid the coefficient swell problem that is preeminent in rings of characteristic zero. We derive decompositions for images that are tailored to dedicated hardware implementations. These decompositions reduce the size of line-buffers which dominate the silicon area in integrated circuit implementations. As an application, we derive a lossless compression scheme for 8 bit monochrome images using wavelet filters with values in the ring $\mathbf{Z}/256\mathbf{Z}$.

Keywords: wavelets, commutative rings, lossless image compression, integrated circuits

1. INTRODUCTION

We investigate wavelet based methods for lossless compression of images that are well-suited for realization in dedicated hardware. There is a natural demand for such compression methods. For example, in Germany it is legally obliged to store radiological images without loss over several years.

Most known lossy wavelet based compression schemes can be subdivided into the three steps *transformation, quantization,* and *entropy coding,* cf. [1, chapter 7]. The transformation and the entropy coding steps are both invertible, but the quantization has to be omitted to achieve lossless compression. Certainly, the implementation of the wavelet transform has to ensure that the invertibility is not affected by round-off errors. This suggests to consider rings and fields that differ from the real or complex numbers allowing fixed-point arithmetic. Thus, a natural choice are filters with dyadic rational or integer coefficients.

However, a serious disadvantage of filter banks over rings of characteristic zero is that more and more bits are needed to represent the signal values in successive decomposition steps, a phenomenon known in computer algebra as *intermediate coefficient swell.* For example, applying a dyadic rational biorthogonal Coifman wavelet filter pair[2] of degree 3 to a gray-scale image with 8 bits per pixel requires about 32 bits per coefficient after a few decomposition steps.

One way to get around the coefficient swell problem is to employ finite field or finite ring arithmetic. These computational structures allow to control the "amplitudes" of all signals and are extensively used for example in error correcting codes[3,4] and in cryptographic schemes.[5,6] The use of filter banks over finite fields in error control code applications were proposed by Sarkar and Poor.[7,8] Vaidyanathan suggested to use filter banks over finite fields in data encryption schemes.[9] Lossless compression of binary images with the help of filter banks over the finite field with two elements were proposed by Swanson and Tewfik[10] and by Johnston.[11] These applications motivate the study of filter banks over commutative rings.

2. FILTERBANKS OVER COMMUTATIVE RINGS

In this section we discuss two-channel filter banks for signals with values in a commutative ring A (all rings are assumed to be associative and to contain an identity element; moreover, we assume that A is non-trivial, that is, $1 \neq 0$). Later we will impose that A is finite, but the theory sketched in this section can be developed without this restriction.[12] Filter banks over commutative rings are a natural generalization of filter banks over the real numbers. Therefore, many statements below are reminiscent of the special case $A = \mathbf{R}$, the real numbers.

Signals. We assume that the analyzed signals are elements of the free A-module of finitely supported sequences

$$M = \{f\colon \mathbf{Z} \to A \mid |\operatorname{supp}(f)| < \infty\}.$$

The standard basis $B = \{\delta_k \mid k \in \mathbf{Z}\}$ of this module is given by the delta functions $\delta_k\colon \mathbf{Z} \to A$, where

$$\delta_k(x) := \begin{cases} 1 & \text{if } x = k, \\ 0 & \text{if } x \neq k. \end{cases}$$

Thus, any signal $s \in M$ can be expressed formally as $s(x) = \sum_{k\in\mathbf{Z}} s(k)\delta_k(x)$, where all but finitely many coefficients $s(k)$ are zero.

z-Transform. Let $A[z, 1/z]$ be the A-algebra of Laurent polynomials in the indeterminate z. Recall that there is an A-module isomorphism from M onto $A[z, 1/z]$ induced by $\delta_k \mapsto z^{-k}$. We refer to this isomorphism as the z-transform. Thus, the z-transform $s(z)$ of a signal $s \in M$ is explicitly given by

$$\sum_{k\in\mathbf{Z}} s(k)\,\delta_k(x) \longmapsto s(z) = \sum_{k\in\mathbf{Z}} s(k)\, z^{-k}.$$

We interpret signals and filters freely as elements of M or $A[z, 1/z]$, whatever viewpoint is more convenient.

Decimation Operator. The decimation operator $[\downarrow 2]$ is defined (as usual) by

$$[\downarrow 2] : \begin{cases} A[z, 1/z] & \longrightarrow & A[z, 1/z], \\ a(z) & \longmapsto & a_e(z), \end{cases} \qquad \text{with} \qquad a(z) = a_e(z^2) + z a_o(z^2).$$

Two-Channel Filter Banks. Consider the two-channel filter bank as illustrated in Figure 1. We assume that the analysis filters $\widetilde{\alpha}, \widetilde{\beta}$ and the synthesis filters α, β are all elements of the A-module M, thus all filters have finite impulse response. We want to derive now necessary and sufficient conditions on these filters that ensure perfect reconstruction [in the sense that on input of an arbitrary signal $s \in M$ the output of the filter bank $\hat{s} \in M$ coincides with s].

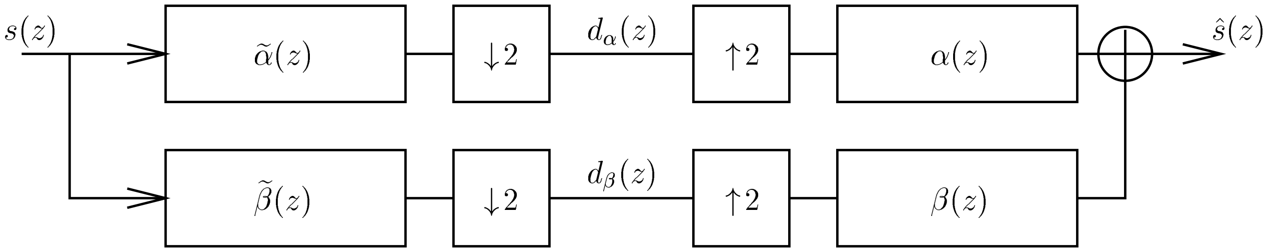

Figure 1. Two-channel filter bank for signals with values in a commutative ring A.

Analysis. Let $s(z) \in A[z, 1/z]$ be an arbitrary signal. The output of the analysis step may be described by the convolutions $\widetilde{\alpha}(z)s(z)$ and $\widetilde{\beta}(z)s(z)$, followed by decimations. Thereby we obtain two signals

$$d_\alpha(z) = [\downarrow 2]\,\widetilde{\alpha}(z)s(z) \qquad \text{and} \qquad d_\beta(z) = [\downarrow 2]\,\widetilde{\beta}(z)s(z). \tag{1}$$

The intermediate results $d_\alpha(z)$ and $d_\beta(z)$ in equation (1) can be expressed more explicitly with the help of the polyphase components of the signal and of the filters. Let

$$s(z) = s_e(z^2) + z s_o(z^2), \qquad \widetilde{\alpha}(z) = \widetilde{\alpha}_e(z^2) + z^{-1}\widetilde{\alpha}_o(z^2), \qquad \widetilde{\beta}(z) = \widetilde{\beta}_e(z^2) + z^{-1}\widetilde{\beta}_o(z^2)$$

be the decomposition according to cosets of $A[z^2, 1/z^2]$ in $A[z, 1/z]$ (note that we used different coset representatives for the signal and for the filters for later convenience). The analysis step can now be formulated as follows:

$$(s_e(z), s_o(z))\, H_p(z) = (d_\alpha(z), d_\beta(z)), \quad \text{where} \quad H_p(z) = \begin{pmatrix} \widetilde{\alpha}_e(z) & \widetilde{\beta}_e(z) \\ \widetilde{\alpha}_o(z) & \widetilde{\beta}_o(z) \end{pmatrix}. \tag{2}$$

The matrix $H_p(z)$ is called the *polyphase matrix* associated with the filter pair $\widetilde{\alpha}, \widetilde{\beta}$. Polyphase matrices are of fundamental importance in the analysis of multirate systems.

Synthesis. The analysis stage produces the signals d_α, d_β. The rate of these signals is expanded by inserting zero-valued samples between adjacent samples. The result of the rate expansion can be described by $[\uparrow 2]d_\alpha(z) = d_\alpha(z^2)$ and $[\uparrow 2]d_\beta(z) = d_\beta(z^2)$. These sequences $d_\alpha(z^2), d_\beta(z^2)$ are filtered with α and β respectively. The synthesis step thus yields a signal $\hat{s}$ given by

$$\hat{s}(z) := \alpha(z)\, d_\alpha(z^2) + \beta(z)\, d_\beta(z^2). \tag{3}$$

Let

$$\alpha(z) = \alpha_e(z^2) + z\alpha_o(z^2), \qquad \beta(z) = \beta_e(z^2) + z\beta_o(z^2), \qquad \hat{s}(z) = \hat{s}_e(z^2) + z\hat{s}_o(z^2).$$

Using these decompositions, one can express equation (3) as follows:

$$\hat{s}_e(z^2) + z\hat{s}_o(z^2) = \left(\alpha_e(z^2)d_\alpha(z^2) + \beta_e(z^2)d_\beta(z^2)\right) \; + z\left(\alpha_o(z^2)d_\alpha(z^2) + \beta_o(z^2)d_\beta(z^2)\right).$$

Comparing coefficients on the left and right hand side thus yields the following description of the synthesis filter bank in polyphase form:

$$(d_\alpha(z), d_\beta(z))\, G_p^t(z) = (\hat{s}_e(z), \hat{s}_o(z)), \quad \text{where} \quad G_p(z) = \begin{pmatrix} \alpha_e(z) & \beta_e(z) \\ \alpha_o(z) & \beta_o(z) \end{pmatrix}. \tag{4}$$

Perfect Reconstruction. The analysis of the decomposition and recomposition steps did not reveal any conditions on the characteristic of the ring so far. This characteristic free approach via polyphase matrices yields the following result on perfect reconstruction filter banks:

THEOREM 2.1. *Let $\widetilde{\alpha}, \widetilde{\beta} \in M$ be analysis filters of a two-channel filter bank with polyphase matrix $H_p(z)$. A perfect reconstructing synthesis filter pair $\alpha, \beta \in M$ exists for $\widetilde{\alpha}, \widetilde{\beta} \in M$ iff the polyphase matrix $H_p(z)$ is an element of the general linear group* $\mathrm{GL}(2, A[z, 1/z])$.

Proof. If we combine the equations (2) and (4), then the perfect reconstruction requirement $\hat{s}(z) = s(z)$ for all signals $s \in A[z, z^{-1}]$ is equivalent to $H_p(z)G_p^t(z) = I$. Thus $H_p(z)$ has a right inverse. Since $A[z, z^{-1}]$ is a commutative ring, this shows that the determinant $\det H_p(z)$ is a unit in $A[z, z^{-1}]$. A matrix over a commutative ring is invertible iff its determinant is a unit, cf. [13, Korollar 48.3] or [14, III §8.3 Prop. 5]. This implies that $H_p(z)$ is an element of $\mathrm{GL}(2, A[z, z^{-1}])$. The other direction is trivial. □

REMARK 2.2. *The same style of argument can be used for the M-channel case. An M-channel filter bank allows perfect reconstruction (with FIR filters) iff $H_p(z) \in \mathrm{GL}(M, A[z, 1/z])$.*

REMARK 2.3. *Another approach to perfect reconstruction filter banks makes use of modulation matrices. These matrices are obtained from the polyphase matrix by multiplication with a "diagonal delay matrix" and with a Fourier matrix, cf. [15, p. 310]. This approach can only be used over rings that support this Fourier matrix. In the two-channel case this means that the modulation matrix approach is restricted to rings where 2 is a unit.*

Much more can be said about filter banks over commutative rings. We have a good structural knowledge in the case where A is a field (finite or not).[12] In particular, there exists a complete parameterization. Moreover, we know how to factor wavelet filters into lifting (or ladder) steps.[16] Matters get more difficult when we consider commutative rings that fail to be fields. For example, it can be shown that a factorization into lifting steps is not possible over the integers.[17] However, the basic principles of filter banks over some commutative ring are as simple as in the special case of real numbers.

3. SOME FILTER EXAMPLES

In this section we illustrate the generation of filters. We choose the dyadic rationals and the integers modulo 256 as examples. The ring of dyadic rationals is obtained from the integers by localizing at 2. Thus, the elements of this ring are given by quotients $a/2^b$, $a, b \in \mathbf{Z}$, $b \geq 0$. Filters with coefficients in the dyadic rationals $\mathbf{Z}[1/2]$ allow implementations with fixed-point arithmetic, which explains the considerable interest of circuit designers in these filters. The ring of integers modulo 256 is a "natural" candidate for the representation of pixels in 8 bit monochrome images. We will use this ring in the lossless image compression application.

Dyadic Rationals $\mathbf{Z}[1/2]$. A simple way to generate numerous examples of these filters is as follows. Consider the group $\mathrm{SL}(2, \mathbf{Z}[1/2])$. This group has the presentation (cf. Serre [18, p. 81]):

$$\mathrm{SL}(2, \mathbf{Z}[1/2]) \cong \langle\, S, T, S_2 \mid S^4 = 1,\ S^2 = S_2^2 = (ST)^3 = (S_2T^2)^3,\ STS^{-1} = S_2T^4S_2^{-1} \,\rangle,$$

with

$$S = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad S_2 = \begin{pmatrix} 0 & 1/2 \\ -2 & 0 \end{pmatrix}.$$

The general linear group $\mathrm{GL}(2, \mathbf{Z}[1/2])$ is given by the semidirect product of $\mathrm{SL}(2, \mathbf{Z}[1/2])$ with the group $H = \langle U, V\rangle$ generated by the diagonal matrices

$$U = \begin{pmatrix} 1/2 & 0 \\ 0 & 1 \end{pmatrix}, \qquad V = \begin{pmatrix} -1/2 & 0 \\ 0 & 1 \end{pmatrix}.$$

Thus $\mathrm{GL}(2, \mathbf{Z}[1/2])$ is generated by the matrices $S, T, U,$ and V (note that S_2 can be expressed as $S_2 = USU^{-1}$). Finally, take the matrices of $\mathrm{GL}(2, \mathbf{Z}[1/2])$ and the delay matrix

$$D = \begin{pmatrix} 0 & z^{-1} \\ 1 & 0 \end{pmatrix}.$$

These matrices generate a subgroup $G = \langle S, T, U, V, D\rangle$ of the group $\mathrm{GL}(2, \mathbf{Z}[\tfrac{1}{2}][z, 1/z])$. Any matrix in G gives a polyphase matrix and hence a filter with dyadic rational coefficients. For example, the polyphase matrix $H_p(z)$ given by $H_p(z) = U^{-1}T^{-1}DTU$ leads to the filter pair $\widetilde{\alpha}(z), \widetilde{\beta}(z)$:

$$\left(\widetilde{\alpha}(z), \widetilde{\beta}(z)\right) = (1, z^{-1}) H_p(z^2) = (1, z^{-1}) \begin{pmatrix} z^{-2} & 2z^{-2} \\ (1 - z^{-2})/2 & -z^{-2} \end{pmatrix} = \left(\frac{1}{2}z^{-1} + z^{-2} + \frac{1}{2}z^{-2},\ 2z^{-2} - z^{-3}\right).$$

Finite Ring $\mathbf{Z}/256\mathbf{Z}$. The canonical group homomorphism $\mathrm{SL}(2, \mathbf{Z}) \to \mathrm{SL}(2, \mathbf{Z}/256\mathbf{Z})$, induced by $\mathbf{Z} \to \mathbf{Z}/256\mathbf{Z}$, $x \mapsto (x \bmod 256)$, is surjective. The group $\mathrm{SL}(2, \mathbf{Z})$ is generated by the matrices S, T given in the previous example. Hence, $\mathrm{SL}(2, \mathbf{Z}/256\mathbf{Z})$ is generated by the image of the generators S and T under the canonical homomorphism. Thus, the general linear group $\mathrm{GL}(2, \mathbf{Z}/256\mathbf{Z})$ is generated by S, T and the diagonal matrices

$$\left\{ \begin{pmatrix} d & 0 \\ 0 & 1 \end{pmatrix} \,\middle|\, d \in U(\mathbf{Z}/256\mathbf{Z}) \right\}, \qquad \text{where } U(\mathbf{Z}/256\mathbf{Z}) \text{ denotes the units of the ring } \mathbf{Z}/256\mathbf{Z}.$$

Again, a simple way to generate polyphase matrices in $\mathrm{GL}(2, \mathbf{Z}/256\mathbf{Z}[z, 1/z])$ is to take products of matrices in $\mathrm{GL}(2, \mathbf{Z}/256\mathbf{Z})$ and powers of the delay matrix D (notation being as in the previous example). For example, the product of T^{-1}, S, and the diagonal matrix $d = \mathrm{diag}(-1, 1)$ gives the polyphase matrix

$$H_p(z) = T^{-1}\, S\, \mathrm{diag}(-1, 1) = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix} \in \mathrm{GL}(2, \mathbf{Z}/256\mathbf{Z}[z, 1/z]).$$

The filter pair $\widetilde{a}(z), \widetilde{b}(z)$ corresponding to this polyphase matrix is given by

$$\left(\widetilde{a}(z), \widetilde{b}(z)\right) = (1, z^{-1}) H_p(z^2) = (1, z^{-1}) \begin{pmatrix} 0 & 1 \\ 1 & -1 \end{pmatrix} = \left(z^{-1}, 1 - z^{-1}\right).$$

We will use this particularly simple filter pair in our compression application.

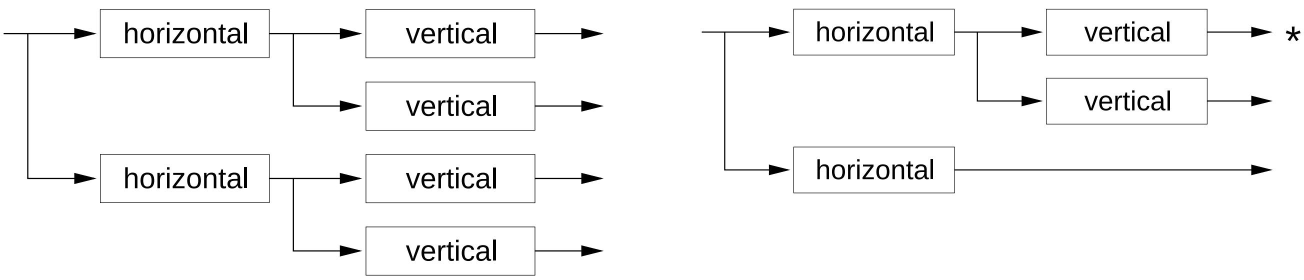

Figure 2. This figure shows the usual tensor filter approach on the left side. A hardware optimized approach is shown on the right.

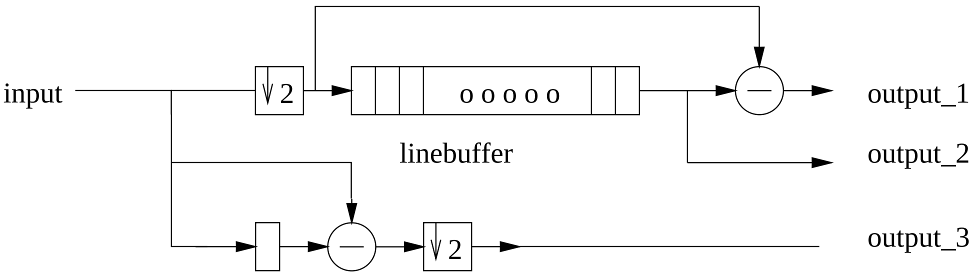

Figure 3. A realization of the right side of Figure 2 with filter pair $\widetilde{a}(z) = z^{-1}$ and $\widetilde{b}(z) = 1 - z^{-1}$.

4. ARCHITECTURES

In this section we derive architectures for the decomposition of images by successively filtering rows and columns respectively. Our main interest is to find a decomposition structure that is amenable to VLSI implementation.

We consider 8 bit monochrome images. Lossless compression is basically achieved by first transforming the image with a filter bank followed by arithmetic encoding of the individual subbands. A natural choice for 8 bit images is to employ $\mathbf{Z}/256\mathbf{Z}$ arithmetic. We use the simple filters $(\widetilde{a}(z), \widetilde{b}(z)) = (z^{-1}, 1 - z^{-1})$ in order to minimize the wrap-around effects due to modular arithmetic. Note that in two's complement number representation the subtraction modulo 256 can be realized by a single subtractor, cf. [19].

Assume that the image signal is fed into the circuit row-by-row. A horizontal decomposition is realized as follows. The first filter, a simple dirac sequence, and the first decimation step just select every other sample. The second filter subtracts neighboring samples and the corresponding decimation step selects every other of these differences. A vertical decomposition with these filters is more expensive as it requires one linebuffer to compute the difference of vertically neighoring samples.

In the usual tensor product approach the horizontal filtering is followed by a vertical filtering of both subbands, as is shown in Figure 2 (left). Unfortunately, this requires two linebuffers in order to compute the vertical differences. Only one linebuffer is necessary in the decompostion shown in Figure 2 (right). As the linebuffers occupy most of the silicon area, this leads to considerable savings. The simulation results in Table 2 show that both decomposition types yield comparable compression efficieny. The output of the dirac "low-pass" filter (marked with an asterisk in Figure 2) is transformed recursively using the same scheme. The output of the filter bank after a single decomposition step is illustrated in Figure 6.

Figure 3 sketches the structure of one decomposition step. The circuit consists of two modulo 256 subtracters and a linebuffer. The linebuffer stores one line of the downsampled image (output of filter $\widetilde{a}$). `Output_2` is used as input signal for the next recursive decomposition. The size of the linebuffers decreases logarithmically with each decomposition step. Figure 7 shows the final layout for three decomposition steps. The layout is designed for a 1μ CMOS process. The linebuffers are composed of static 8-bit registers (each 8-bit register is of size $80 \times 250\mu^2$). A decomposition step can be realized with two subtractors ($250 \times 150\mu^2$), a line buffer, and one additional register.

The architecture is fully scaleable and all wireing channels are integrated in the base-cells mentioned above. In many applications it may be advantageous to use dynamic registers, reducing the area consumption considerably.

5. LOSSLESS COMPRESSION

The first step employs a filter bank to reduce the entropy of the image signal. We use the filter pair described in the previous section. The distribution of the coefficients in a typical subband is shown in Figure 4. One observes that most of the coefficients are "near" zero and many possible values do not occur at all. The individual subbands are entropy encoded using a technique similar to lossless JPEG.[20]

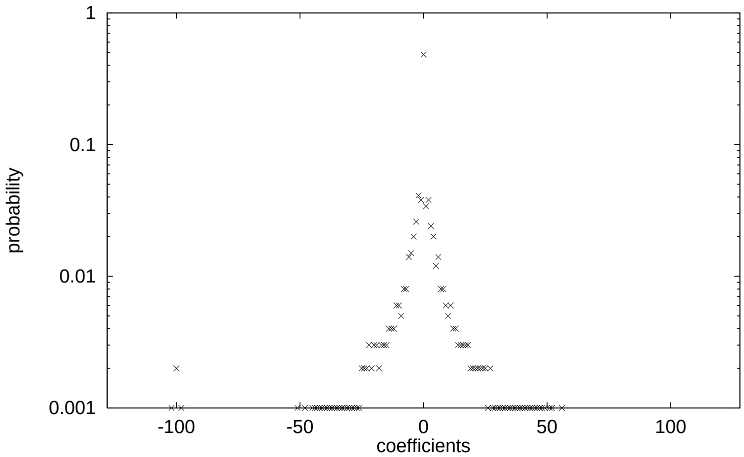

Figure 4. This figure shows the distribution of coefficients in a typical subband.

We subdivide the range of the coefficients into buckets of different size. A bucket is a finite ordered set of coefficient values. Each coefficient value can be represented uniquely by bucket number and position. More concretely, we use the buckets in Table 1. For instance, the value 22 is represented by bucket 11 and offset $3 = (011)_2$. Buckets are encoded with an adaptive model conditioned by several contexts.[21,22] The context is computed by averaging the bucket numbers of the neighboring pixels, as shown in Figure 5. The offsets are encoded bitwise using a binary model. Note that the number of these bits depends on the size of the bucket. A hardware implementation of an arithmetic coder was published in [23].

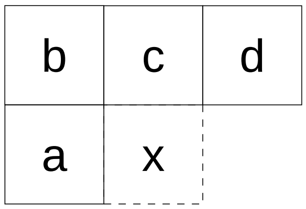

Figure 5. The context for encoding pixel x is computed by averaging the bucket numbers of pixels $a, \ldots, d$.

Table 2 shows compression results for various images. Our methods U3 and HW3 perform better than GIF and GNU ZIP for all test images. Lossless JPEG performs better on the images *clock2*, *modern*, and *cathedral*. Note that for some images the simplified hardware implementation HW3 performs even better than the usual decomposition scheme U3.

6. CONCLUSIONS

We applied filter banks over finite rings to lossless compression of monochrome images. Our method outperforms standard compression methods. We expect further improvements by introducing prediction as used in the S+P

bucket	ordered value set
0	0
1	1, -1
2	2, -2
3	3, -3
4	4, -4
5	5, -5, 6, -6
6	7, -7, 8, -8
7	9, -9, 10, -10
8	11, -11, 12, -12
9	13, -13, 14, -14, 15, -15, 16, -16
10	17, -17, 18, -18, 19, -19, 20, -20
11	21, -21, 22, -22, 23, -23, 24, -24
12	25, -25, 26, -26, 27, -27, 28, -28
13	29, -29, 30, -30, ..., 36, -36
14	37, -37, 38, -38, ..., 44, -44
15	45, -45, 46, -46, ..., 52, -52
16	53, -53, 54, -54, ..., 60, -60
17	61, -61, 62, -62, ..., 76, -76
18	77, -77, 78, -78, ..., 92, -92
19	93, -93, 94, -94, ..., 108, -108
20	109, -109, 110, -110, ..., 124, -124
21	125, -125, 126, -126, 127, -127, 128

Table 1. Buckets used for the encoding of coefficient values.

	LJPG	GNUZIP	GIF	U3	HW3
window	3.665	3.553	4.135	3.254	3.265
clock	4.047	3.718	4.357	3.694	3.679
clock2	4.814	5.466	6.504	4.894	4.905
balconies	4.004	4.096	4.655	4.040	3.983
gable	4.692	4.712	5.687	4.422	4.359
old window	4.691	4.556	5.192	4.372	4.325
modern	3.846	4.155	4.715	4.001	3.926
cathedral	5.743	6.403	7.997	5.859	5.852
windmill	4.395	4.394	5.188	4.158	4.057
mission	4.084	3.963	4.626	3.698	3.587

Table 2. A comparison of different lossless compression schemes for various images. The table entries denote the average number of bits per pixel in the compressed image. The left side shows the standard compression methods lossless JPEG (LJPG), GNU ZIP, and GIF. Our compression schemes are shown on the right side, namely the usual deomposition with three recursion steps (U3) and the hardware optimized scheme (HW3) as implemented.

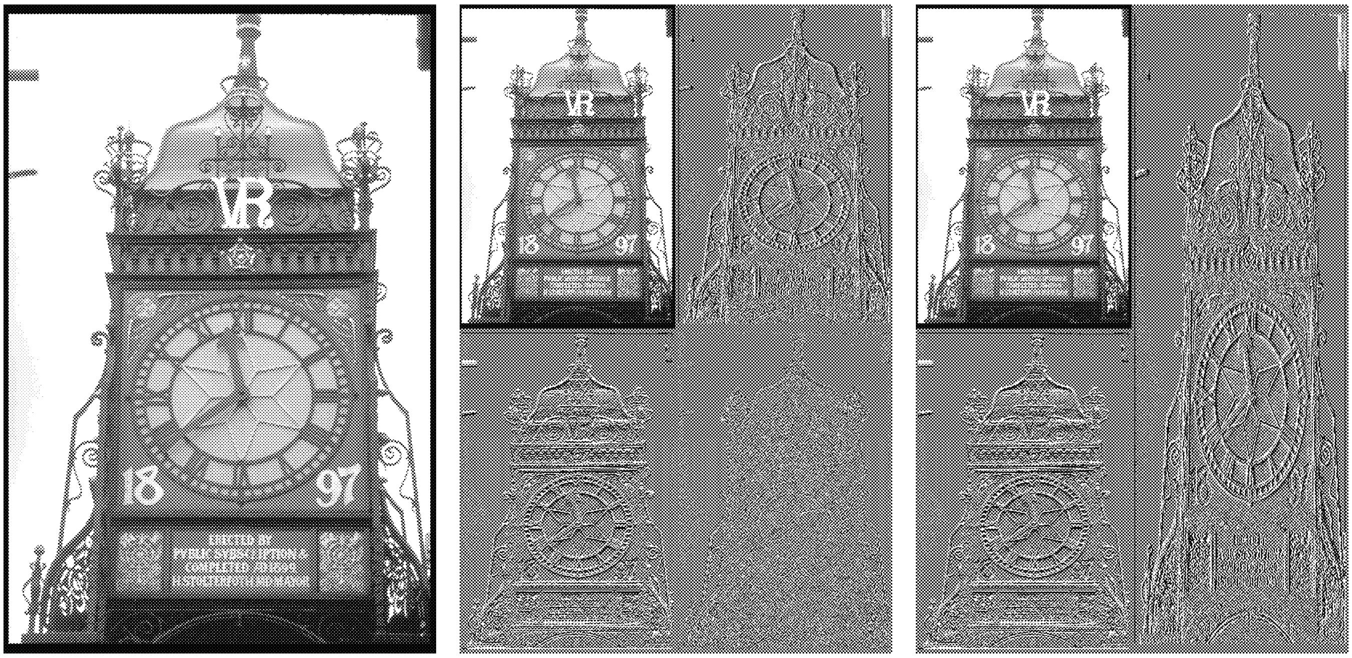

Figure 6. The original 'clock' image is shown on the left. The output of the usual decomposition in shown in the middle image. The right image shows the output of the hardware-optimized decomposition.

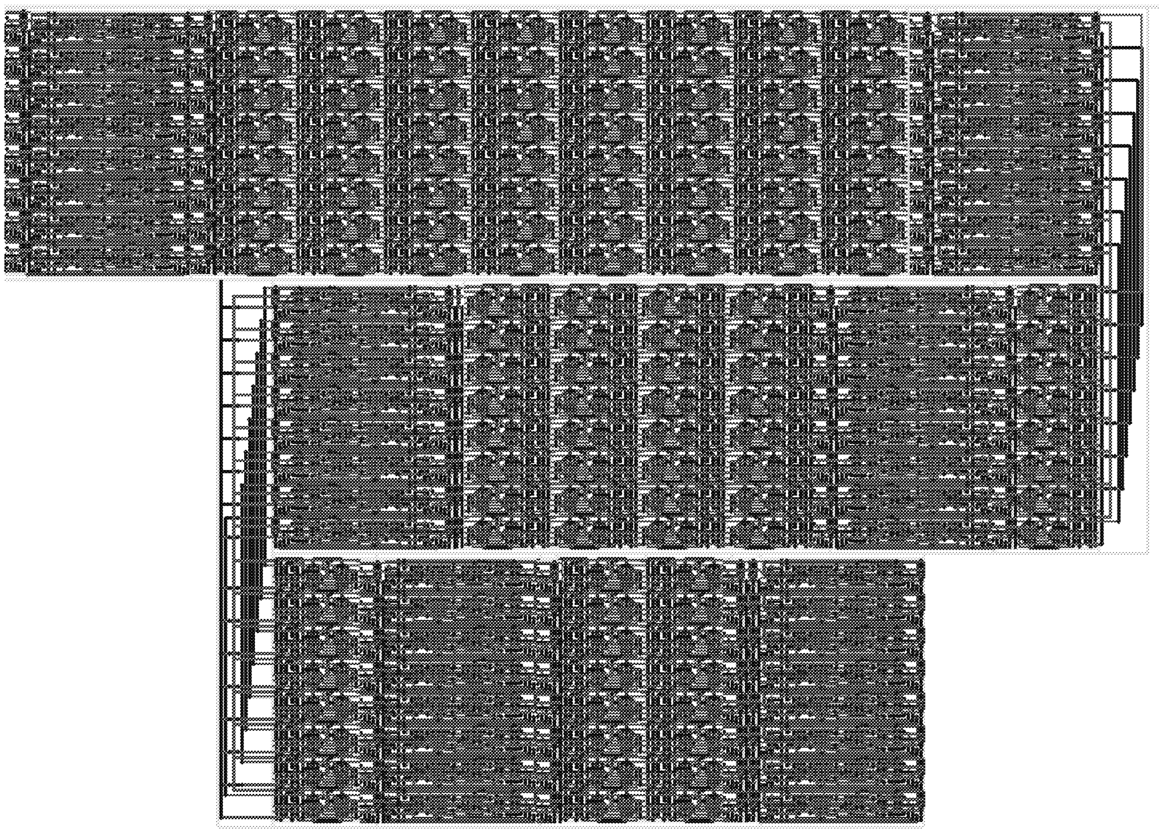

Figure 7. The layout for a decomposition of recursion depth three. The design is for a 1μ CMOS technology.

transform.[24] Generalizing the S+P transform, there is another promising research direction using lifting techniques to construct non-linear integer-to-integer wavelet transforms.[25] Further applications of filter banks over commutative rings[12] still need to be explored.

ACKNOWLEDGEMENTS

Andreas Klappenecker gratefully acknowledges support by DFG through Sonderforschungsbereich SFB414 "Rechner- und sensorgestützte Chirurgie". Frank May thanks DFG for partial support through project Be 887/6-4.

REFERENCES

1. M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*, Prentice Hall, 1995.
2. J. Tian and R. O. Wells, "Dyadic rational biorthogonal Coifman wavelet systems," Tech. Rep. CML TR 96-07, Department of Mathematics, Rice University, 1996.
3. F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*, Elsevier Science B. V., Amsterdam, 1977.
4. A. R. Hammons, P. V. Kumar, A. R. Calderbank, N. J. A. Sloane, and P. Solé, "The Z_4-linearity of Kerdock, Preparata, Goethals, and related codes," *IEEE Trans. on Information Theory* **40**, pp. 301–319, 1994.
5. W. Patterson, *Mathematical Cryptology*, Rowman & Littlefield, 1987.
6. G. Simmons, ed., *Contemporary Cryptology – The Science of Information Integrity*, IEEE Press, 1992.
7. S. Sarkar and H. V. Poor, "Finite field wavelet transforms and multilevel error protection," in *Proc. 1995 Intl. Symp. on Information Theory*, p. 428, IEEE, (Whistler, BC, Canada), 1995.
8. H. V. Poor, "Finite-field wavelet transforms," in *Information Theory and Applications II*, J. Chouinard, P. Fortier, and T. A. Gulliver, eds., LNCS 1133, pp. 225–238, Springer Verlag, 1996.
9. P. Vaidyanathan, "Unitary and paraunitary systems in finite fields," in *Proc. 1990 IEEE Int. Symp. on Circuits and Systems*, pp. 1189–1192, IEEE, 1990.
10. M. D. Swanson and A. H. Tewfik, "A binary wavelet decomposition of binary images," *IEEE Trans. on Image Processing* **5**(6), 1996.
11. C. P. Johnston, "The lifting scheme and finite-precision-error-free filter banks," in *Wavelet Applications in Signal and Image Processing IV*, M. Unser, A. Aldroubi, and A. Laine, eds., vol. 2825, pp. 307–316, SPIE, 1996.
12. A. Klappenecker, M. Holschneider, and K. Flornes, "Two-channel perfect reconstruction filter banks over commutative rings." Preprint, IAKS, Universität Karlsruhe, (http://iaks-www.ira.uka.de/home/klappi/), 1997.
13. G. Scheja and U. Storch, *Lehrbuch der Algebra*, vol. 1, B. G. Teubner Verlag, Stuttgart, 2nd ed., 1994.
14. N. Bourbaki, *Algebra I, Chap. 1-3*, Springer-Verlag, 1989.
15. G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, 1996.
16. I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps." Preprint, available through `http://cm.bell-labs.com/who/wim/papers/`, 1996.
17. P. M. Cohn, "On the structure of the GL_2 of a ring," *Publ. math. I.H.E.S.* **30**, pp. 365–413, 1966.
18. J.-P. Serre, *Trees*, Springer-Verlag, 1980. Translation of "Arbres, Amalgames, SL_2", Astérisque no. 46, Soc. Math. France, 1977.
19. G. W. Reitwiesner, "Binary arithmetic," in *Advances in Computers*, F. L. Alt, ed., vol. 1, pp. 231–308, Academic Press, 1960.
20. K. R. Rao and J. J. Hwang, *Techniques and standards for image, video, and audio coding*, Prentice Hall, 1996.
21. I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Comm. ACM* **30**, pp. 520–540, June 1987.
22. I. H. Witten and T. C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *IEEE Trans. Inf. Theory* **37**, pp. 1085–1094, July 1991.
23. F. May, A. Klappenecker, V. Baumgarte, A. Nückel, and T. Beth, "A high throughput multiplication free approximation to arithmetic coding," in *Proc. 1996 International Symposium on Information Theory and Its Applications*, pp. 845–847, IEEE, 1996.
24. A. Said and W. A. Pearlman, "An image multiresolution for lossless and lossy image compression," *IEEE Trans. on Image Proc.* , Sept. 1996.
25. A. R. Calderbank, I. Daubechies, W. Sweldens, and B. L. Yeo, "Wavelet transforms that map integer to integers." Preprint, 1996.