

Implementation of Orthogonal Wavelet Transforms and their Applications

Peter Rieder and Josef A. Nossek
Institute of Network Theory & Circuit Design
Technical University of Munich

Abstract

In this paper the efficient implementation of different types of orthogonal wavelet transforms with respect to practical applications is discussed. Orthogonal singlewavelet transforms being based on one scaling function and one wavelet function are used for denosing of signals. Orthogonal multiwavelets are based on several scaling functions and several wavelets. Since they allow properties like regularity, orthogonality and symmetry being impossible in the singlewavelet case, multiwavelets are well suited bases for image compression applications. With respect to an efficient implementation of these orthogonal wavelet transforms approximating the exact rotation angles of the corresponding orthogonal wavelet lattice filters by using very few CORDIC-based elementary rotations reduces the number of shift and add operations significantly. The performance of the resulting, computationally cheap, approximated wavelet transforms with respect to practical applications is discussed in this paper.

1 Introduction

In recent years wavelet transforms have gained a lot of interest in many application fields, whereby orthogonal wavelet bases have been introduced by examining different possibilities for their design [1]. Orthogonal singlewavelet transforms using one scaling function and one wavelet function were designed. Suitable architectures for implementing wavelet transforms are octavfilterbanks, whereby each stage of the filterbank is composed of a lowpass filter $G(z)$ and the complementary highpass filter $H(z)$. Orthogonal lattice filters can be used to implement these stages, whereby only orthogonal 2×2 -rotations are required [7, 8]. Orthogonal singlewavelets are suitable bases for different types of signals, e.g. images, since they analyse high frequencies with bases of small compact support and low frequencies with bases of large support. An important application is the denoising of images [3]. Thereby, the noisy image is transformed into the wavelet domain. Then, the small coefficients being dominated by the noise are eliminated by setting them to zero. Only the remaining large coefficients are used for reconstructing the image. The result is a denoised image mainly containing the pure image information.

Symmetry of the bases is a desired property for image compression applications [10], as symmetric bases allow to preserve the signal's phase in the transform domain and an efficient processing at borders. However, with exception of the trivial Haar bases, it is impossible to design orthogonal and symmetric singlewavelets. Multiwavelet systems based on 2 scaling functions and 2 multiwavelets allow the properties regularity, orthogonality and symmetry, simultaneously [9, 6]. This makes multiwavelets to appropriate bases for image compression algorithms. Also for implementing multiwavelet systems filterbank structures can be used. Thereby, with each stage of the transform the signal is divided by two lowpass filters $G_1(z), G_2(z)$ concerning the scaling functions and two highpass filters $H_1(z), H_2(z)$ concerning the multiwavelets. For implementing these stages of the transform lattice structures being composed of orthogonal 2×2 -rotations were presented in [6].

With respect to image compression two different approaches are possible: With the first approach a multiresolution of the image is executed and the coefficients of different scale are coded [10]. With the second approach after an 8- or 16-channel filtering (DCT, LOT) of the image the coefficients of the different channels are coded [5]. The big advantage of multiwavelet systems is, that they offer good bases for both approaches.

The computational complexity plays an important role not only for image denoising and compression, but also for many other algorithms where wavelet transforms are used. Therefore, an efficient implementation of orthogonal wavelet transforms is desired. Since the basic modules of the wavelet lattice filters are orthogonal 2×2 -rotations, implementing the orthogonal rotations by a few shift and add operations is equivalent to an efficient implementation of the whole transform. The CORDIC-algorithm offers one possibility to execute elementary rotations, whereby a sequence of elementary rotations being implementable with a few shift and add operations is used. CORDIC-based approximate rotations [4, 8, 2] renounce on the full sequence of elementary rotations by using only a few elementary rotations such that the computational complexity is significantly reduced. In this paper the influence of this approximation on the performance of the wavelet transform with respect to the practical applications is discussed. Approximated, orthogonal wavelet systems showing a very simple implementation are presented. With respect to practical applications, like image denoising or compression they perform in the same way as exact transforms.

2 Orthogonal Singlewavelets

Orthogonal singlewavelet systems are based on one scaling function $\Phi(t)$ and one wavelet function $\Psi(t)$, which meet the following dilation equations:

$$\Phi(t) = \sum_{i=0}^{n-1} g_i \Phi(2t - i) \quad \Psi(t) = \sum_{i=0}^{n-1} h_i \Phi(2t - i)$$

The discrete coefficients g_i and h_i define the discrete wavelet transform and the wavelet filters $G(z) = \sum_{i=0}^{n-1} g_i z^{-i}$ (lowpass) and $H(z) = \sum_{i=0}^{n-1} h_i z^{-i}$ (highpass). Wavelet transforms can be implemented by the filterbank structure of Figure 1, whereby each stage is composed of the complementary wavelet filters. Suitable architectures for the implementation of these orthogonal

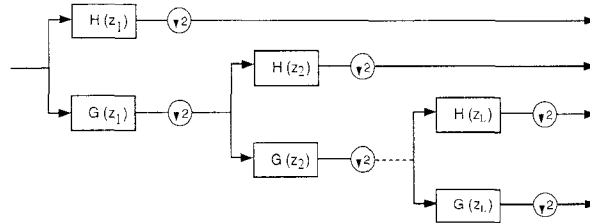


Figure 1: Filterbank structure implementing a discrete wavelet transform

filters are lattice structures. Figure 2 shows a lattice filter implementing one stage of Daubechies' wavelet transform of length $n = 4$. Obviously, the basic modules are 2×2 -rotations. By only using orthogonal rotations, orthogonality of the transform is structurally imposed. For the lattice filter to perform an orthogonal wavelet transform, another property is necessary. This property ensures, that the wavelet function is zero mean, what is equivalent with the wavelet having at least one vanishing moment and the transfer functions $G(z)$ and $H(z)$ having at least one zero at $z = 1$ and $z = -1$, respectively. In [7, 8] it was used, that these conditions are fulfilled, if the sum of rotation angles

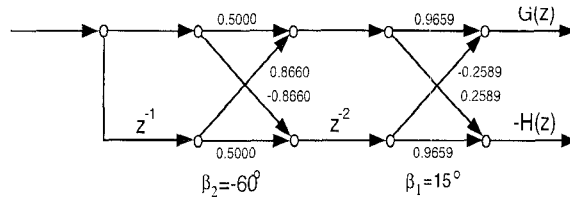


Figure 2: Lattice filter implementing one stage of Daubechies' wavelet transform of length $n = 4$

β_k is constant:

$$\sum_k \beta_k = -45^\circ.$$

Therefore, a lattice filter whose sum of all rotation angles is -45° performs an orthogonal wavelet transform, independent of the angles β_k .

3 Orthogonal Multiwavelets

Multiwavelet systems using 2 scaling functions and 2 wavelets are based on 4 dilation equations, that are also represented by the basis matrix \mathbf{W} of size $4 \times 4m$.

$$\Phi_v(t) = \sum_{l=1}^2 \sum_{k=0}^{2m-1} g_{v,2k-1+l} \Phi_l(2t-k); \quad v \in \{1, 2\};$$

$$\Psi_v(t) = \sum_{l=1}^2 \sum_{k=0}^{2m-1} h_{v,2k-1+l} \Phi_l(2t-k); \quad v \in \{1, 2\};$$

$$\mathbf{W} = \begin{pmatrix} \mathbf{W}^U \\ \mathbf{W}^L \end{pmatrix} = \begin{pmatrix} g_{1,0} & g_{1,1} & \dots & g_{1,4m-1} \\ g_{2,0} & g_{2,1} & \dots & g_{2,4m-1} \\ h_{1,0} & h_{1,1} & \dots & h_{1,4m-1} \\ h_{2,0} & h_{2,1} & \dots & h_{2,4m-1} \end{pmatrix}$$

In [9] multiwavelets were designed that allow the properties regularity, orthogonality and symmetry, simultaneously. Thereby, Φ_1 (1st row of \mathbf{W}) and Ψ_1 (3rd row of \mathbf{W}) being symmetric, and Φ_2 (2nd row of \mathbf{W}) and Ψ_2 (4th row of \mathbf{W}) being antisymmetric, requires a specially structured wavelet basismatrix \mathbf{W} .

$$\mathbf{W} = \begin{pmatrix} a_0 & b_0 & a_1 & b_1 & a_1 & -b_1 & a_0 & -b_0 \\ -a_0 & b_0 & a_1 & -b_1 & -a_1 & -b_1 & a_0 & b_0 \\ b_0 & -a_0 & -b_1 & a_1 & -b_1 & -a_1 & b_0 & a_0 \\ b_0 & a_0 & b_1 & a_1 & -b_1 & a_1 & -b_0 & a_0 \end{pmatrix};$$

Setting $a_0 = 0.009977, a_1 = 0.697129, b_0 = b_1 = -0.083399$ results in the bases plotted in Figure 3.

Also for multiwavelet transforms, filterbanks (Figure 4) and lattice structures offer an efficient implementation of the multiwavelet filters $G_v(z) = \sum_{i=0}^{n-1} g_{v,i} z^{-i}, H_v(z) = \sum_{i=0}^{n-1} h_{v,i} z^{-i}, v \in \{1, 2\}$. Figure 5 shows a lattice structure implementing orthogonal multiwavelet filters. Again, orthogonality is ensured by using only orthogonal rotations, a wavelet transform (at least one vanishing moment) is guaranteed by the constant sum of rotation angles

$$\sum_{k=1}^{n/2} \omega_k = 0.$$

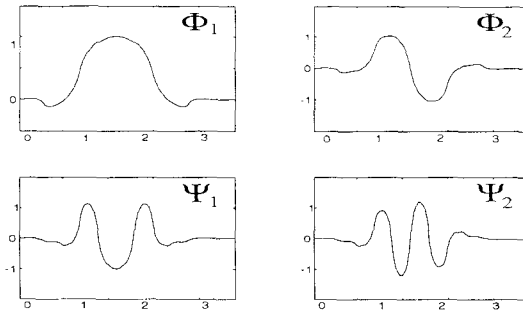


Figure 3: Multiwavelets of $n = 4$, $p = 2$ and the corresponding scaling functions

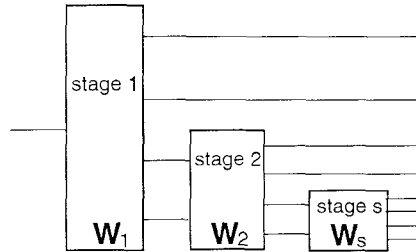


Figure 4: Filterbank for implementing multiwavelet transformationen

4 Efficient Implementation of Orthogonal Rotations

Since in the singlewavelet- as well as in the multiwavelet case the lattice filters are composed of orthogonal 2×2 -rotations only, in order to get a simple implementation of the filters, an efficient implementation of the orthogonal rotations is sufficient.

An orthogonal 2×2 -rotation $\mathbf{R}(\alpha)$ is defined as follows:

$$\mathbf{R}(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix}$$

The CORDIC algorithm is a common method to execute orthogonal rotations by using a sequence of $w + 1$ μ -rotations (w being the wordlength):

$$\mathbf{R}(\alpha) = \frac{1}{K_w} \prod_{k=0}^w \begin{bmatrix} 1 & -\sigma_k 2^{-k} \\ \sigma_k 2^{-k} & 1 \end{bmatrix}, \quad \sigma_k \in \{+1, -1\},$$

with $\frac{1}{K_w} = \prod_{k=0}^w \frac{1}{\sqrt{1+2^{-2k}}}$ being the scaling factor. This corresponds to the representation of α as

$$\alpha = \sum_k \sigma_k \alpha_k = \sum_k \sigma_k \arctan 2^{-k}.$$

This representation of an angle in the "arctan 2^{-k} " basis is also the basic idea of CORDIC-based approximate rotations [4], but there we have $\sigma_k \in \{-1, 0, +1\}$.

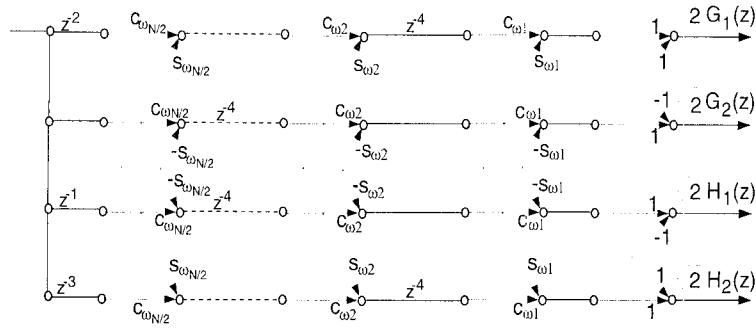


Figure 5: Lattice structure implementing multiwavelet filters

In [4] double rotations consisting of 2 equal CORDIC elementary rotations were used, which rotate by the angle $2\alpha_k$, i.e. $\mathbf{R}(2\alpha_k) = \mathbf{R}(\alpha_k)\mathbf{R}(\alpha_k)$ such that

$$\mathbf{R}(2\alpha_k) = \frac{1}{K_k^2} \begin{bmatrix} 1 & -\sigma 2^{-k} \\ \sigma 2^{-k} & 1 \end{bmatrix} \begin{bmatrix} 1 & -\sigma 2^{-k} \\ \sigma 2^{-k} & 1 \end{bmatrix}$$

Now, the scaling factor $\frac{1}{K_k^2}$ can be factorized into a sequence of shift & add operations:

$$\frac{1}{K_k^2} = \frac{1}{1 + 2^{-2k}} = (1 - 2^{-2k})(1 + 2^{-4k})(1 + 2^{-8k}) \dots \quad (1)$$

With one (or a few) of these double rotations an approximate rotation can be composed, that is simple to implement, approximates any rotation angle to a certain accuracy (increasing the number of double rotations increases the accuracy), and is always exactly orthogonal independent of the accuracy.

5 Efficiently implemented Singlewavelet transforms for image denoising

In order to get efficiently implemented singlewavelet transforms, instead of the complete sequence of elementary rotations only a few CORDIC-based elementary rotations are used in order to approximate the exact rotations quite well. By always using double rotations (with different signs) the constant sum of rotation angles is not violated and a simple implementation of the scaling factor is possible. For the presented example of $n = 4$ $\beta_1 = -60^\circ$ and $\beta_2 = 15^\circ$ are approximated by $\tilde{\beta}_1 = -45^\circ - \arctan 2^{-2} \approx -59.04^\circ$ and $\tilde{\beta}_2 = \arctan 2^{-2} \approx 14.04^\circ$. The corresponding scaling functions $\Phi(t)$ for the exact rotation angles (dotted line) and for the case of the approximation (solid line) are plotted in Figure 6 (upper part) together with the zeros of the transfer functions $G(z)$. The resulting lattice structure is shown in Figure 6 (lower part), whereby only very few shift and add operations are necessary.

The decisive question is, how the performance of the wavelet transform is influenced by the approximation that allows the very simple implementation. Denoising of images is an important application of orthogonal wavelet transforms. Thereby, according to Figure 7 the noisy image is transformed into the wavelet domain, the small coefficients being dominated by the noise are thresholded, before the image is reconstructed by the inverse wavelet transform. In order to improve the performance of the transform, redundant, time-invariant, undecimated, orthogonal wavelet transforms are often used instead of nonredundant, time-variant, decimated, orthogonal

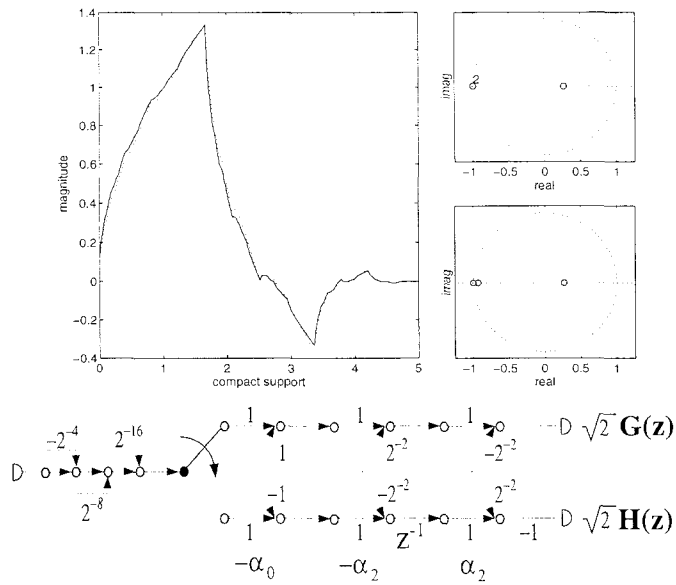


Figure 6: Comparison of Daubechies' standard scaling function of length $n = 4$ (dotted line) and the version showing a simple implementation (solid line)

wavelet transforms [3]. In any case, essential is that those orthogonal bases lead to the best results, which approximate the signal best. For images wavelet bases outperform e.g. Fourier bases, since compactly supported wavelets allow to analyse abrupt as well as slow changes of the signal.

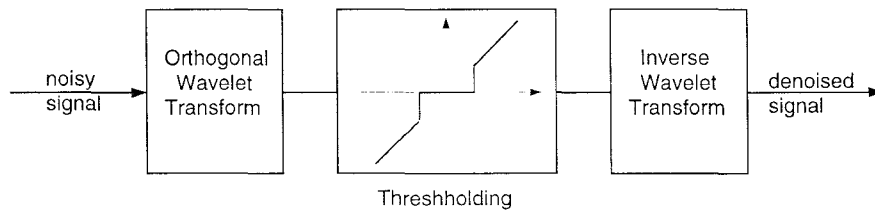


Figure 7: Denoising of signals via wavelet transform

Figure 8 (left) shows a Lena image, whose quality is affected by white Gaussian noise of 21.2 dB. By wavelet denoising the quality of the image can be improved to 28.4 dB what is documented in Figure 8 (right). Note, that this application is quite robust with respect to the accuracy of the wavelet transform: There is almost no difference in the performance of the algorithm using the exact or the approximated wavelet filters. Only exact orthogonality -this is guaranteed in both cases- is important for this application. Note also, that wavelet denoising is not limited to a special kind of noise, different kinds of disturbances, e.g. blockartefacts, can be filtered out of the images.



Figure 8: Noisy (left) and denoised (right) version of Lena

6 Efficiently implemented Multiwavelet transforms for image compression

In order to get efficiently implemented multiwavelet transforms, again, we approximate the exact rotation angles by using only a few CORDIC-based elementary rotations. The example of Figure 3 requires the angles $\omega_1 = -\omega_2 = 6.8^\circ$, which are approximated by using only one elementary rotation by $\tilde{\omega}_1 = -\tilde{\omega}_2 = \arctan 2^{-3} \approx 7.1^\circ$. This results in the lattice structure of Figure 9 and in continuous bases that cannot be distinguished from those of Figure 3. With respect to image

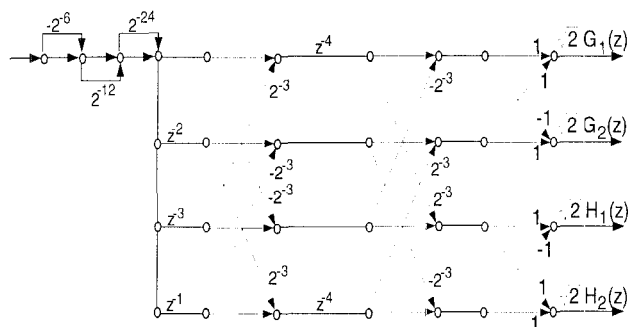


Figure 9: Lattice structure approximating the multiwavelet filters of $n = 4$ and $p = 2$

compression one has to choose between two wide-spread algorithms:

- Wavelet coder encode the different scales of the image in the multiresolving wavelet domain.
- Classical subband coder use the DCT or LOT [5] in order to get the 8 (16) frequency channels of the image being encoded and compressed.

Though both approaches are possible with the presented multiwavelets, here we focus on the subband coder showing how multiwavelet packets work as lapped orthogonal transforms. In order to get an

8 channel filterbank, the wavelet packet transform with 2 stages of Figure 10 must be used. Note, that both stages (\mathbf{W}_1 , \mathbf{W}_2) are slightly different because a wavelet-like transform being necessary [9]. Implementing this wavelet packet transform with 2 stages using approximate rotations leads to

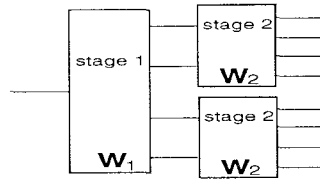


Figure 10: 8-channel wavelet packet structure

the very simple structure of Figure 11. The computational costs are reduced to a few shift and add operations (Note, that only the scaling is not considered in Figure 11. Thereby, the analysis and synthesis can be combined).

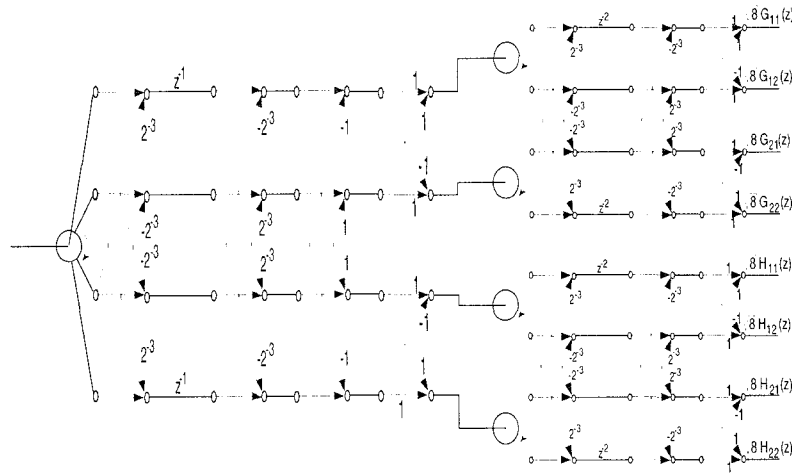


Figure 11: Architecture for efficiently implementing the multiwaveletpacket-based lapped orthogonal transform

The decisive question again is, how the multiwavelet-based lapped orthogonal transform showing the simple implementation performs in comparison to other well tried transforms with respect to image compression. Figure 12 compares the impulse responses of the transfer functions of the 8 channels, whereby the dotted line belongs to the classical LOT designed by Malvar and the solid line belongs to the multiwaveletpacket-based lapped orthogonal transform (there are no visible differences between exact and approximated case). Obviously, the curves are similar, in the multiwavelet case the behavior at the borders is even smoother.

In order to evaluate the quality of the multiwaveletpacket transform (MWT) in comparison to Malvar's LOT or the DCT, an image compression algorithm according to Figure 13 [11] was used. Figure 14 shows the results obtained for the Lena image, it confirms the similarity of the quality of Malvar's LOT and the multiwaveletpacket-based lapped orthogonal transform and the improvement

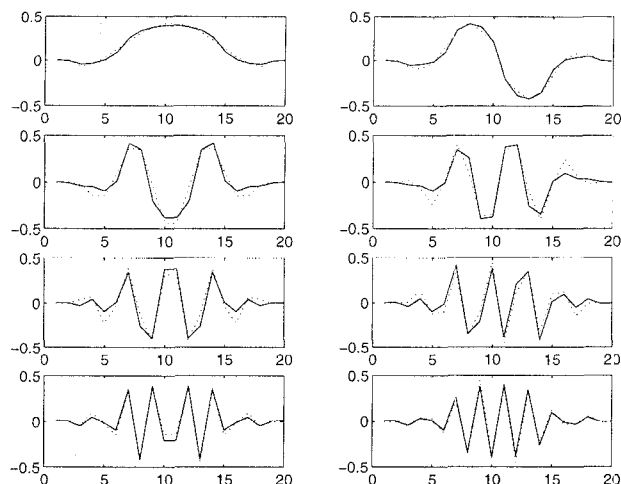


Figure 12: Impulse response of Malvar's LOT-filters (dotted line) compared to multiwaveletpacket-based filters (solid line)

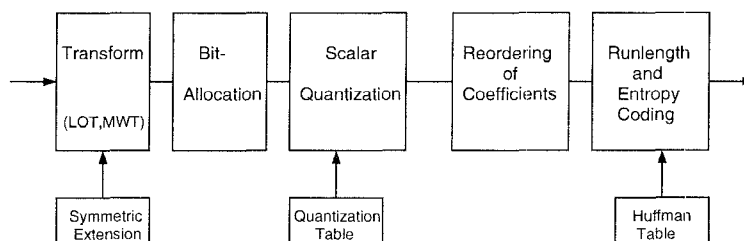


Figure 13: Coding procedure

of these transforms in comparison to the classical DCT: Depending on the compression (0.1–1.0 bpp) the SNR between reconstructed and original image is analyzed for the DCT, Malvar's LOT, and the multiwaveletpacket transform. For the most interesting range (0.5 bpp – 1.0 bpp) the multiwaveletpacket transform is even slightly better than Malvar's LOT.

Note, also DCT, LOT or generalized lapped orthogonal transforms can be efficiently implemented by using approximate rotations [2]. However, the presented architecture only requires computational cheap *orthogonal* rotations, whereby other structures use *orthonormal* rotations additionally requiring a scaling procedure per rotation.

7 Conclusion

In this paper the efficient implementation of orthogonal singlewavelet transforms and symmetric multiwavelet transforms based on 2 scaling functions was presented. Since both transforms are based on orthogonal 2×2 -rotations, their efficient implementation is achieved by using CORDIC-based approximate rotations being composed of very few shift and add operations. This approximation does not cause the loss of the quality of the transform with respect to its properties orthogonality, regularity, frequency behavior. For practical applications, like image denoising or compression,

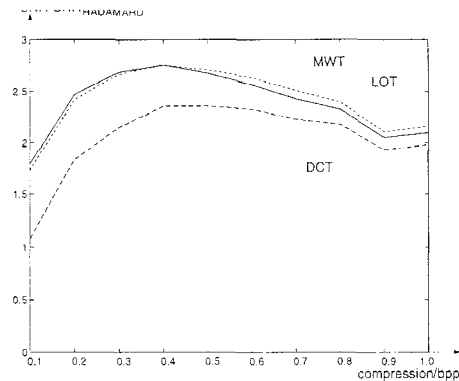


Figure 14: Comparison of different transforms with respect to image compression

where the computational complexity is important, the wavelet transforms showing a very simple implementation perform quite well.

Acknowledgement

The authors would like to thank S. Trautmann and T.Q. Nguyen for making their image compression algorithms being used for the presented work available on the internet.

References

- [1] I. Daubechies. *Ten Lectures on Wavelets*. Notes from the 1990 CBMS-NSF Conference on Wavelets and Applications at Lowell, MA. SIAM, Philadelphia, PA, 1992.
- [2] E.F. Deprettere, G. Hekstra, and R. Heusdens. Fast VLSI Overlapped Transform Kernel. Preprint.
- [3] D.L. Donoho. De-Noiseing by Soft-Thresholding. *IEEE Trans. Inform. Theory*, 41:613–627, 1995.
- [4] J. Götze and G.J. Hekstra. Adaptive Approximate Rotations for Computing the EVD. In M. Moonen and F. Cathoor, editors, *Algorithms and Parallel VLSI Architectures*. Elsevier Science Publishers, 1994.
- [5] H.S. Malvar and D.H. Staelin. The LOT: Transform Coding Without Blocking Effects. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 37(4):553–559, April 1989.
- [6] P.Rieder. Parameterization of Symmetric Multiwavelets. *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing, ICASSP 97, Munich*, April 1997.
- [7] P. Rieder, K. Gerganoff, J. Götze, and J.A. Nossek. Parameterization and Implementation of Orthogonal Wavelet transforms. *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing, ICASSP 96, Atlanta*, III:1515–1518, Mai 1996.
- [8] P. Rieder, J. Götze, J.A. Nossek, and C.S. Burrus. Parameterization of Orthogonal Wavelet Transforms and their Simple Implementation. *IEEE Trans. on Circuits and Systems II*, 1997.
- [9] P. Rieder and J.A. Nossek. Smooth Multiwavelets based on 2 Scaling Functions. *Proc. IEEE Int. Symp. on Time-Frequency and Time-Scale Analysis*, pages 309–312, June 1996. Paris.
- [10] J.M. Shapiro. Embedded Image Coding Using Zerotrees of Wavelet Coefficients. *IEEE Transaction on Signal Processing*, 41(12):3445–3462, December 1993.
- [11] S. Trautmann and T.Q. Nguyen. GenLOT: Design and Application for Transform-Based Image Coding. *Proc. of the Asilomar Conference*, Nov. 1995.