Optimized discrete wavelet transform to real-time digital signal processing

Jan Vlach[1], Pavel Rajmic[1],
Jiri Prinosil[1], Josef Vyoral[1], Ivan Mica[1]

[1] Brno University of Technology, Faculty Electrical Engineering and Communication,
Department of Telecomunications, Purkynova 118,
612 00 Brno, Czech Republic
jan.vlach@phd.feec.vutbr.cz, rajmic@feec.vutbr.cz,
{xprino01,xvyora01,xmicai00}@stud.feec.vutbr.cz

Abstract. In this paper, we propose optimized method of discrete wavelet transform. There is many use of wavelet transform in digital signal processing (compression, wireless sensor networks, etc.). In those fields, it is necessary to have digital signal processing as fast as it possible. The new segmented discrete wavelet transform (SegWT) has been developed to process in real-time. It is possible to process the signal part-by-part with low memory costs by the new method. In the paper, the principle and benefits if the segmented wavelet transform is explained.

1 Introduction

If we use wavelet transform in real applications we handle signals of finite length. It is not usually possible to process whole signal at a time, we must process the signal segment-by-segment. To calculation of sufficient amount of no redundant wavelet coefficients, it is necessary to know the signal behind the segment borders. Generally in this case, we determine signal behaviors behind the borders. Typical examples are extension techniques like zero-padding, smooth, symmetric, asymmetric and period extension. Zero-padding assumes samples outside the segment boundary are zero [1], [2]; periodic extension assumes that the signal is periodic [1], [2]. Symmetric extension assumes the signal is reflected at the segment boundaries [1]–[3]. These basic types of extension cause signal distortion at the boundaries. The sort of distortion depends on estimation range of signal specification behind boundaries. The more level of wavelet decomposition is chosen the much amount of distortion is caused. Such distortion happens to be unacceptable.

The way how to circumvent problem is adoption the segment techniques based upon overlap-save and overlap-add methods. In the case of fast discrete wavelet transform [5], we use overlap-save (OSC) and overlap-add convolution (OAC). Conditions are more complicated with increasing level of wavelet decomposition. These problems solved new method SegWT, which is optimized to discrete wavelet transform for segment processing with various segment lengths. It has a great potential application also in cases when it is necessary to process a long signal off-line and no sufficient memory capability is available. It is then possible to use this new method for equivalent segment wise processing of the signal and thus save the storage place.

2 Discrete-time wavelet transform

In digital signal processing we use finite discrete (or discrete-time) wavelet transform, abbreviated DTWT, which can be represented by an orthogonal matrix **W** of size *n*×*n* [7].

Let **x** be a vector of length *n*. Its wavelet transform is vector **y**, obtained as $\mathbf{y} = \mathbf{W}\mathbf{x}$. Due to the ortogonality of **W**, the inverse wavelet transform is $\mathbf{x} =$ $= \mathbf{W}^{-1}\mathbf{y} = \mathbf{W}^{\mathrm{T}}\mathbf{y}$.

In fact, instead of multiplying vector **x** by the matrix **W**, more effective Mallat's pyramid algorithm [5] is used for computing the transform. Each step of this algorithm corresponds to:

1. extending the input vector
2. filtering this vector by specific low-pass and high-pass filters
3. cropping the central part of the results
4. decimation the results

The coefficients from low-pass branch are called "approximations" and those from the high-pass branch are called "details". We can repeat this single transformation step with the approximations standing for the input. The number of such repetitions, *d*, is called transformation depth. Scheme of one step of this algorithm is depicted in Fig. 1. This way the input is divided into number of frequency subbands [1].

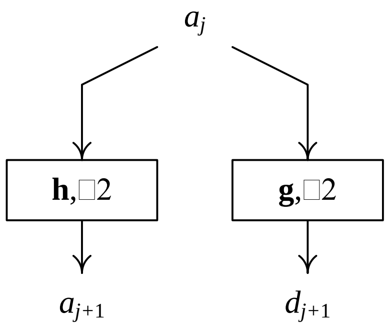

Fig. 1. One step of the forward wavelet transform – decomposition into details and approximations.

3 Segmented wavelet transform

The task for the segmented wavelet transform based on wavelet over-lap save convolution (WOSC) and wavelet overlap-add convolution (WOAC) techniques [6], SegWT, is naturally to allow signal processing by its segment, so that in this manner we get the same result (same wavelet coefficients) as in the ordinary DTWT case. In this problem, the transform depth *d*, wavelet filter length *m* and the segment length *s* play a crucial role.

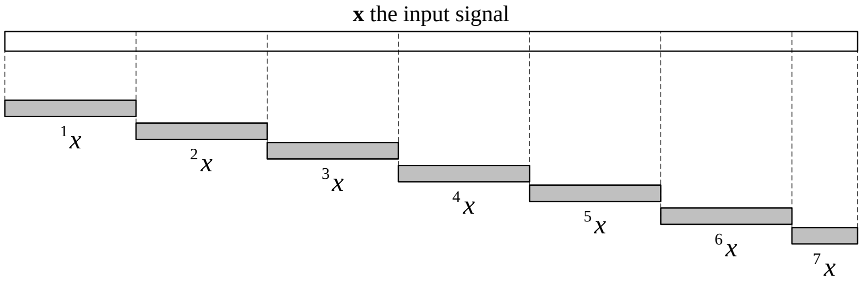

Fig. 2. Segmentation of the input signal. The last segment can be shorter than the others.

Derivation of the SegWT algorithm requires a very detailed knowledge of the DTWT and IDTWT [8] algorithms. Thanks to this it is possible to deduce fairly sophisticated rules how to handle the signal segments. We found out that, in dependence on *d*, *m*, *s* it is necessary to extend every segment from left by exact number of samples from the preceding segment and from right by another number of samples from the subsequent one. Fig. 3 illustrates the principle of segment extending.

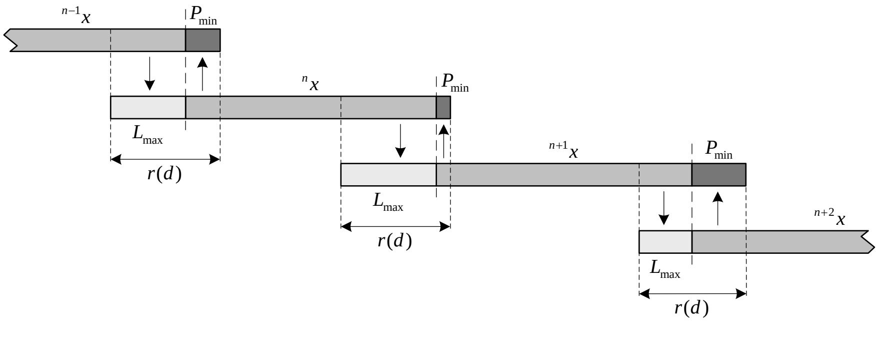

Fig. 3. Scheme of extending input segments. Each segment has to be extended different length form left and right ($L_{\max}$, $R_{\min}$) and the length can also differ from segment to segment. The sum of actual $L_{\max}$ and previous $R_{\min}$ is always constant ($r(d)$).

After using segmented forward wavelet transform which includes extending, cropping and decimation steps we gain wavelet coefficients ready to application process. Then we must processed coefficients reconstruct to source form. Segmented inverse wavelet transform includes similar step like in wavelet decomposition. The hardest part is to add overlapped parts of neighboring segments. The example of WOAC technique is show on Fig. 4.

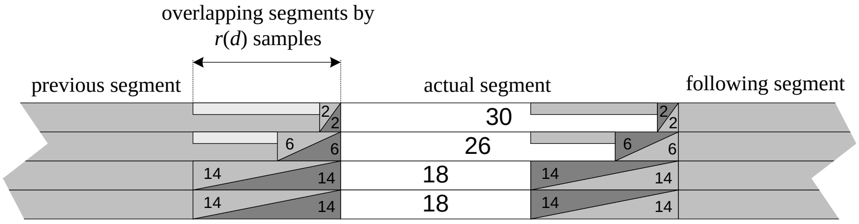

Fig. 4. Example of segmented wavelet reconstruction, show as superposition of all details and aproximations.

4 Experiments

To testing and checking the method we have used MATLAB, because of comfortable implementation the algorithm, also for availability build-in functions like fast convolution algorithm using FFT, downsampling, upsampling, etc. Testing experiment was based on comparison wavelet reconstruction process (SegIWT) of new method with standard IDTWT algorithm. Indeed, it was necessary to adjust the standard algorithm with the view to comparison, otherwise results would be incomparable.

In order to get valuation the new algorithm we defined a quantity, called *percentage velocity gain* (Δt), which indicates how much faster SegIWT process is in comparison to IDTWT algorithm. For example, $\Delta t = 50\%$ means, it is possible to process whole signal in two units of time with new method and in three units of time with standard algorithm; in other words, new algorithm is three over two times faster. The percentage velocity gain is defined

$$\Delta t = \frac{t_{old}}{t_{new}} \cdot 100 - 100 \quad [\%], \tag{1}$$

where t_{old} is time necessary to signal reconstruction with IDTWT and t_{new} is time necessary to signal reconstruction with SegIWT. Simulation parameters were chosen:

- decomposition depth $d = 4$,
- Daubechies wavelet type 4, filter length $m = 8$,
- vector length in depth level $x_d = 4$

These parameters were chosen with respect to acceptable simulation duration and also for showing model behavior of new method.

4.1 Dependence on signal length

First experiment was percentage velocity gain dependence on whole signal length (number of samples).

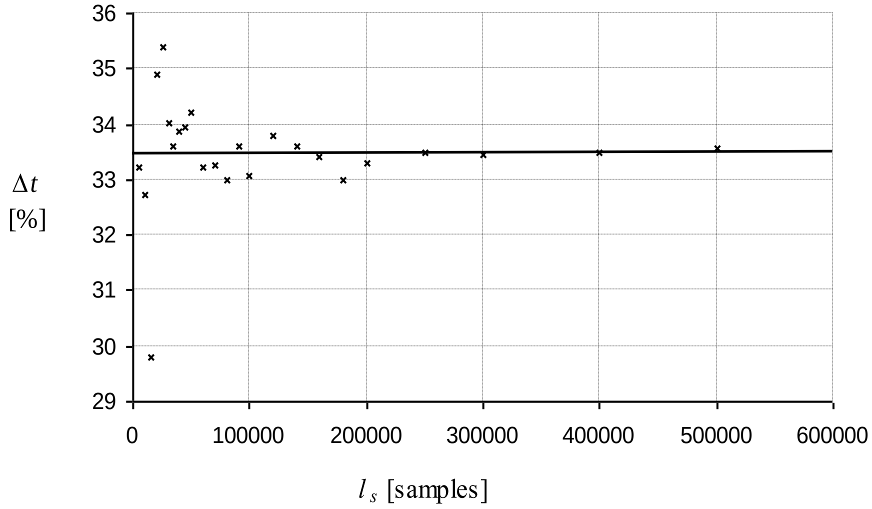

Fig. 5. Percentage velocity gain Δt in dependence on whole signal length l_s. It is evident that mean value of Δt is not dependent on l_s. The mean value of Δt is circa 33% for above-mentioned experiment. Considerable variation of Δt values is caused by short simulation time for short length l_s. These durations are not possible to gauge precisely.

4.2 Dependence on decomposition depth

This experiment was based on changing decomposition depth in signal with constant length l_s = 100 000 samples (Fig. 6). When we set simulation parameters in order to with increasing decomposition depth the number of segments is constant, we obtain the similar dependence (Fig. 7). The descent of Δt is caused by immense time cost to overlap segments (lots of segments are being overlapped). If we choose the decomposition depth greater than 10, the time cost of the new algorithm is approximately the same. Indeed, it is important to notice the IDTWT standard algorithm needs to read much more data from the computer memory by same time cost which means if the access to the memory is slow it could affect simulation results markedly.

In those experiments in which there are another filters, the results are almost the same like in Fig. 6. With increasing filter length the length of overlapped parts also increases. However, this not affects the simulation process as well, because wavelet filters length is very short in comparison to segment length.

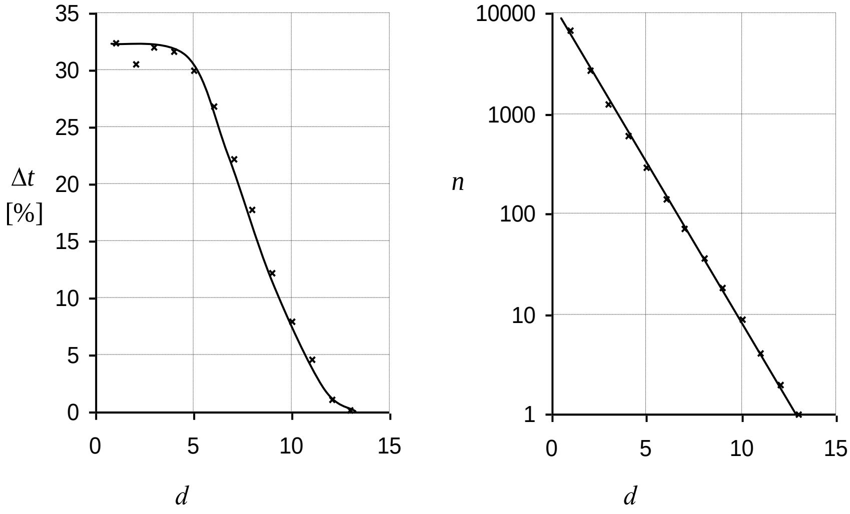

Fig. 5. Percentage velocity gain Δ*t* in dependence on *d* the constant decomposition depth. The value *n* is the number of segments.

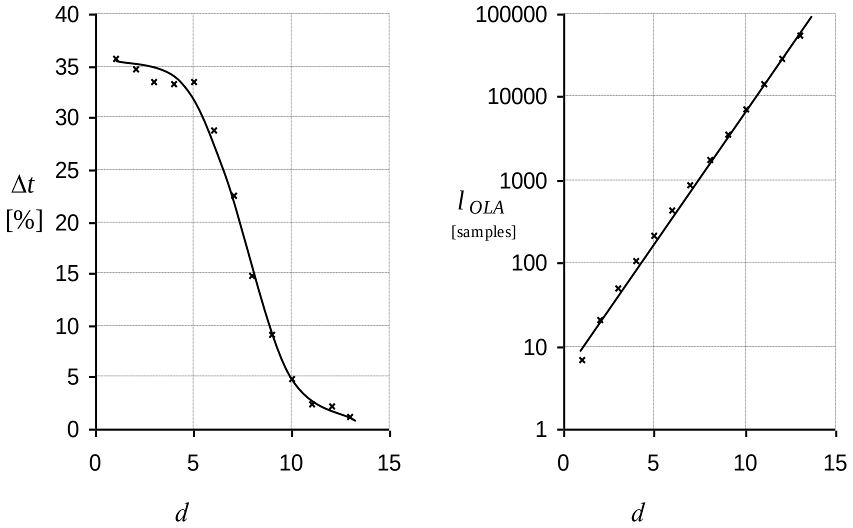

Fig. 6. Percentage velocity gain Δ*t* in dependence on *d* the decomposition depth with constant number of segments. The value l_{OLA} is the number of overlapped samples.

5 Conclusions

In the paper, a novel SegWT method was described. The method can be used in real-time applications, in which the signal segment-by-segment processing is required. As it was proved in presented experiments the computational time was decreased up to about 33% in comparison to standard fast wavelet transform algorithm. Thus, the method would be suitable e.g. for implementation on digital processors and the range of applications of the new algorithms is very wide – from noise cancelation in speech signals by using tresholding of wavelet coefficients to image processing (compression JPEG2000, pattern recognition). These areas are two typically subjects of telecommunications research.

Acknowledgements. The paper was prepared within the framework of No. 102/06/P407 and No. 102/07/1303 projects of the Grant Agency of the Czech Republic and No. 1ET301710509 project of the Czech Academy of Sciences.

References

1. Strang, G., Nguyen, T.: Wavelets and Filter Banks. Cambridge, MA: Wellesley-Cambridge.(1996)
2. Mertins, A.: Signal Analysis Wavelets, Filter Banks, Time–Frequency Transforms and Applications. New York: Wiley (1999)
3. Bradley, J. N., Brislawn, C. M., Faber, V.: Reflected boundary conditions for multirate filter banks, In Proc. IEEE Int. Symp. Time–Frequency and Time–Scale Analysis. (1992)
4. Muramatsu, S., Kiya, H.: Extended overlap-add and -save methods for multirate signal processing," IEEE Trans. Signal Processing, Vol. 45. (1997) 2376–2380
5. Mallat, S.: A Wavelet Tour of Signal Processing. 2nd edition, Academic Press. (1999)
6. Nealand, J. H., Bradley, A. B., Lech, M.: Overlap-Save Convolution Applied to Wavelet Analysis. IEEE Signal Processing Letters, Vol. 15, 2. (2003)
7. Vidakoci, B.: Statistical Modeling by Wavelets (Wiley Series in Probability and Statistics), John Wiley & Sons, New York. (1999)
8. Misiti, M., Misiti, Y., Oppenheim, G.; Poggi J.-M.: Wavelet Toolbox User's Guide, The MathWorks, Inc. (2001)