The Haar Wavelet Transform in the Time Series Similarity Paradigm

Zbigniew R. Struzik, Arno Siebes

Centre for Mathematics and Computer Science (CWI)
Kruislaan 413, 1098 SJ Amsterdam
The Netherlands
email: Zbigniew.Struzik@cwi.nl

Abstract. Similarity measures play an important role in many data mining algorithms. To allow the use of such algorithms on non-standard databases, such as databases of financial time series, their similarity measure has to be defined. We present a simple and powerful technique which allows for the rapid evaluation of similarity between time series in large data bases. It is based on the orthonormal decomposition of the time series into the Haar basis. We demonstrate that this approach is capable of providing estimates of the local slope of the time series in the sequence of multi-resolution steps. The Haar representation and a number of related represenations derived from it are suitable for direct comparison, e.g. evaluation of the correlation product. We demonstrate that the distance between such representations closely corresponds to the subjective feeling of similarity between the time series. In order to test the validity of subjective criteria, we test the records of currency exchanges, finding convincing levels of correlation.

1 Introduction

Explicitly or implicitly, record similarity is a fundamental aspect of most data mining algorithms. For traditional, tabular data the similarity is often measured by attribute-value similarity or even attribute-value equality. For more complex data, e.g., financial time series, such simple similarity measures do not perform very well. For example, assume we have three time series A, B, and C, where B is constantly 5 points below A, whereas C is randomly 2 points below or above A. Such a simple similarity measure would rate C as far more similar to A than B, whereas a human expert would rate A and B as very similar because they have the same shape.

This example illustrates that the similarity of time series data should be based on certain characteristics of the data rather than on the raw data itself. Ideally, these characteristics are such that the similarity of the time series is simply given by the (traditional) similarity of the characteristics. In that case, mining a database of time series is reduced to mining the database of characteristics using the traditional algorithms. This observation is not new, but can also (implicitly) be found in papers such as [1-7].

Which characteristics are computed depends very much on the application one has in mind. For example, many models and paradigms of similarity introduced to date are unnecesarily complex because they are designed to suit too large a spectrum of applications. The context of data mining applications in which matching time series are required often involves a smaller number of degrees of freedom than assumed. For example, in comparing simultanous financial time series, the time variable is explicitly known and time and scale shift are not applicable. In addition, there are strong heuristics which can be applied to these time series. For example, the concern in trading is usually to reach a certain level of index or currency exchange within a certain time. This is nothing else than increase rate or simply slope of the time series in question.

Consider a financial record over one year which we would like to compare with another such record from another source. The values of both are unrelated, the sampling density may be different or vary with time. Nevertheless, it ought to be possible to state how closely the two are related. If we were to do it in as few steps as possible, the first to ask would probably be about the increase/decrease in (log)value over the year. In fact, just a sign of a change over the year may be sufficient, showing whether there has been a decrease or an increase in the stock value. Given this information the next question might be what the increase/decrease was in the first half of the year and what it was in the second half. The reader will not be surprised if we suggest that perhaps the next question might be related to the increase/decrease in each quarter of the year.

This is exactly the strategy we are going to follow. The wavelet transform using the Haar wavelet (the Haar WT for short) will provide exactly the kind of information we have used in the above example, through the decomposition of the time series in the Haar basis. In section 2, we will focus on the relevant aspects of the wavelet transformation with the Haar wavelet. From the hierarchical scale-wise decomposition provided by the wavelet transform, we will next select a number of interesting representations of the time series in section 3. In section 4, these time series' representations will be subject to evaluation of their correlation products. Section 5 gives a few details on the computational efficiency of the convolution product. This is followed by several test cases of correlating examples of currency exchange rates in section 6. Section 7 closes the paper with conclusions and suggestions for future developments.

2 The Haar Wavelet Transform

As already mentioned above, the recently introduced Wavelet Transform (WT), see e.g. Ref. [9, 10], provides a way of analysing local behaviour of functions. In this, it fundamentally differs from global transforms like the Fourier Transform. In addition to locality, it possesses the often very desirable ability of filtering the polynomial behaviour to some predefined degree. Therefore, correct characterisation of time series is possible, in particular in the presence of *non-stationarities* like global or local trends or biases.

Conceptually, the wavelet transform is an inner product of the time series with the scaled and translated wavelet $\psi(x)$, usually a n-th derivative of a smoothing kernel $\theta(x)$. The scaling and translation actions are performed by two parameters; the scale parameter s 'adapts' the width of the wavelet to the *microscopic resolution* required, thus changing its frequency contents, and the location of the analysing wavelet is determined by the parameter b:

$$Wf(s,b) = < f, \psi > (s,b) = \frac{1}{s} \int_{\Omega} dx \; f(x) \; \psi(\frac{x-b}{s}) \; , \qquad (1)$$

where $s, b \in \mathbf{R}$ and $s > 0$ for the continuous version (CWT), or are taken on a discrete, usually hierarchical (e.g. dyadic) grid of values s_i, b_j for discrete version (DWT, or just WT). Ω is the support of the $f(x)$ or the length of the time series.

The choice of the smoothing kernel $\theta(x)$ and the related wavelet $\psi(x)$ depends on the application and on the desired properties of the wavelet transform. In [6, 7, 11], we used the Gaussian as the smoothing kernel. The reason for this was the optimal localisation both in frequency and position of the related wavelets, and the existence of derivatives of any degree n. In this paper, for the reasons which will become apparent later, see section 3, we will use a different smoothing function, namely a simple block function:

$$\theta(x) = \begin{cases} 1 & \text{for} \quad 0 \leq x < 1 \\ 0 & \text{otherwise} . \end{cases} \tag{2}$$

The wavelets obtained from this kernel are defined on finite support and go by the name of their inventor Haar:

$$\psi(x) = \begin{cases} 1 & \text{for} \quad 0 \leq x < \frac{1}{2} \\ -1 & \text{for} \quad \frac{1}{2} \leq x < 1 \\ 0 & \text{otherwise} . \end{cases} \tag{3}$$

For a particular choice of rescaling and position shift parameters (dyadic pyramidal scheme), the Haar system constitutes an orthonormal basis:

$$\psi_{m,n}(x) = 2^{-m}\psi(2^{-m}x - n), \qquad m > 0, n = 0 \ldots 2^m . \tag{4}$$

Assume an arbitrary time series $f = \{f_i\}, i = 1 \ldots 2^N$ on the normalised support $\Omega(f) = [0, 1]$. Using the orthonormal basis just described, the function f can be represented with the linear combination of Haar wavelets:

$$f = f^0 + \sum_{m=0}^{N} \sum_{l=0}^{2^m} c_{m,l} \; \psi_{m,l}, \tag{5}$$

where f_0 is the most coarse approximation of the time series; $f^0 =< f, \theta >$, and each coefficient $c_{m,l}$ of the representation can be obtained as $c_{m,l} =< f, \psi_{m,l} >$.
In particular, the approximations f^j of the time series f with the smoothing kernel $\theta_{j,k}$ form a 'ladder' of multi-resolution approximations:

$$f^{j-1} = f^j + \sum_{k=0}^{2^j} < f, \psi_{j,k} > \psi_{j,k}, \tag{6}$$

where $f^j =< f, \theta_{j,k} >$ and $\theta_{j,k} = 2^{-j}\theta(2^{-j}x - k)$.
It is thus possible to 'move' from one approximation level $j - 1$ to another level j by simply adding (subtracting for j to $j - 1$ direction), the detail contained in the corresponding wavelet coefficients $c_{j,k}, k = 0 \ldots 2^j$.

In figure 1, we show an example decomposition and reconstruction with the Haar wavelet. The time series analysed is $f_{1..4} = \{9, 7, 3, 5\}$.

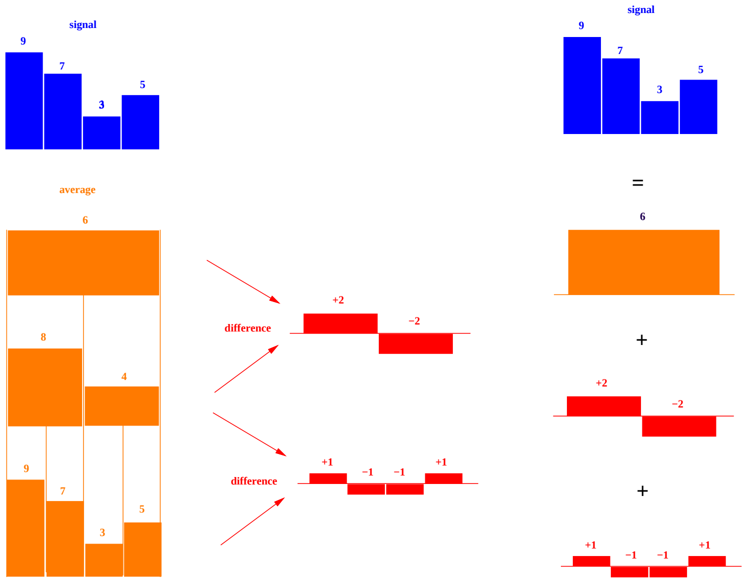

Fig. 1. Decomposition of the example time series into Haar components. Right: reconstruction of the time series from the Haar components.

Note that the set of wavelet coefficients can be represented in a hierarchical (dyadic) tree structure, through which it is obtained. In particular, the reconstruction of each single point f_i of the time series is possible (without reconstructing all the $f_j \neq f_i$), by following a single path along the tree, converging to the point f_i in question. This path determines a unique 'binary address' of the point f_i.

3 Time Series Representations with Haar Family

Note that the Haar wavelet implements the operation of derivation at the particular scale at which it operates. From the definition of the Haar wavelet ψ, (eq. 3, see also figure 2) we have:

$$\psi(x) =< D(2\;x), \theta(2\;x) > \;,$$

where D is the derivative operator

$$D(x) = \begin{cases} 1 & \text{for} \quad x = 0 \\ -1 & \text{for} \quad x = 1 \\ 0 & \text{otherwise} \;. \end{cases} \tag{7}$$

For the wavelet transform of f, we have the following:

$$
\begin{aligned}
< f(x), \psi_{l,n}(x) > = \\
&= < f(x), < D_{l,n}(2\;x), \theta_{l,n}(2\;x) >> \\
&= < f(x), 2^{-1} < D_{l-1,n}(x), \theta_{l-1,n}(x) >> \\
&= << 2^{-1} D_{m,n}(x), f(x) >, \theta_{m,n}(x) > \\
&= 2^{-1} < Df_{m,n}(x), \theta_{m,n}(x) > \;\; . \qquad (8)
\end{aligned}
$$

where Df is the derivative of the function f and θ is the smoothing kernel. The wavelet coefficients obtained with the Haar wavelet ψ are, therefore, proportional to the local averages of the derivative of the time series f at a given resolution. This is a particularly interesting property of our representation, which makes us think that the representations derived from the Haar representation will be quite useful in time series mining. Indeed, in the analysis of patterns in time series, local slope is probably the most appealing feature for many applications.

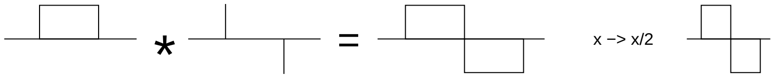

Fig. 2. Convolution of the block function with the derivative operator gives the Haar wavelet after rescaling the time axis $x \to x/2$. $*$ stands for the convolution product.

The most direct representation of the time series with the Haar decomposition scheme would be encoding a certain predefined, highest, i.e. most coarse, resolution level s_{max}, say one year resolution, and the details at the lower scales: half (a year), quarter (of a year) etc., down to the minimal (finest) resolution of interest s_{min}, which would often be defined by the lowest sampling rate of the signals. [1] The coefficients of the Haar decomposition between scales $s_{max}..s_{min}$ will be used for the representation:

$$Haar(f) = \{c_{i,j} \;\; : i = s_{max}..s_{min}, \;\; j = 1..2^i \;\} \; .$$

The Haar representation is directly suitable to serve for comparison purposes when the absolute (i.e. not relative) values of the time series (and the local slope) are relevant. In many applications one would, however, rather work with value independent, scale invariant representations. For that purpose, we will use a number of different, special representations derived from the Haar decomposition WT. To begin with, we will use the sign based representation. It uses only the sign of the wavelet coefficient and it has been shown to work in the CWT based decomposition, see [6].

$$s_{i,j} = sgn(c_{i,j})$$

[1] In practice one may need to interpolate and re-sample signals in order to arrive at a certain common or uniform sampling rate. This is, however, a problem of the implementation and not of the representation and it is related to how the convolution operation is implemented.

where

$$sgn(x) = \begin{cases} 1 & \text{for} \quad x \geq 0 \\ -1 & \text{for} \quad x < 0 \; . \end{cases}$$

The sign representation is an extreme case of discretisation representation since it reduces the range of coefficients in the representation to two discrete levels. For some purposes this may be too coarse. Another possibility to arrive at a scale invariant representation is to use the difference of the logarithms (DOL) of values of the wavelet coefficient at the highest scale and at the working scale:

$$v_{i,j}^{DOL} = log(|c_{i,j}|) - log(|c_{1,1}|) \; ,$$

where i, j are working scale and position respectively, and $c_{1,1}$ is the first coefficient of the corresponding Haar representation. Note that the sign representation $s_{i,j}$ of the time series is complementary/orthogonal to the DOL representation.

The DOL representation can be conveniently normalised to give the rate of increase of v^{DOL} with scale:

$$h_{i,j} = v_{i,j}^{DOL} / \log(2^{(i)}) \qquad \text{for } i > 0 \; .$$

This representation resembles the Hölder exponent approximation of time series local *roughness* at the particular scale of resolution i as introduced in [7].

4 Distance Evaluation with Haar Representations

The measure of the correlation between the components $c_g^{i,j}$ and $c_g^{k,l}$ of two respective time series f and g can be put as:

$$C(f,g) = \sum_{\{i,j,k,l\}=0}^{m,n} w_i c_f^{i,j} \; w_k c_g^{k,l} \; \delta_{i,j;k,l}$$

where

$$\delta_{i,j;k,l} = 1 \qquad \text{iff} \qquad i = k \;\&\; j = l$$

and the (optional) weights w_i and w_k depend on their respective scales i and k. In our experience the orthogonality of the coefficients is best employed without weighting.

Normalisation is necessary in order to arrive at the correlation product between $[0, 1]$ and will simply take the form of

$$C_{\text{normalised}}(f,g) = \frac{C(f,g)}{\sqrt{C(f,f)\; C(g,g)}} \; .$$

The distance of two representations can be easily obtained as

$$\text{Distance}(f,g) = -\log(|C_{\text{normalised}}(f,g)|) \; .$$

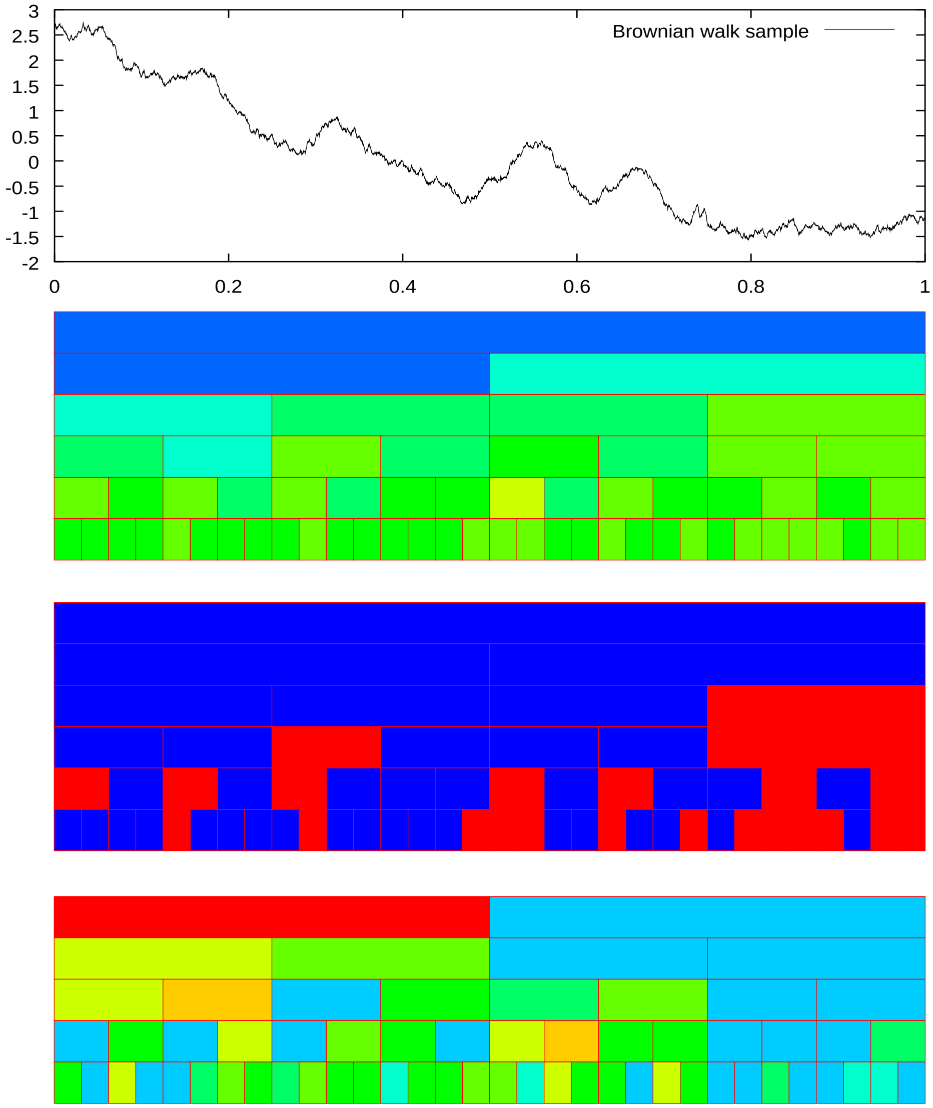

Fig. 3. Top plot contains the input signal. The top colour (gray-scale) panel contains the Haar decomposition with six scale levels from $i = 1$ to $i = 6$, the smoothed component is not shown. The colour (gray shade) encodes the value of the decomposition from dark blue (white) for -1 to dark red (black) for 1. The centre panel shows the sign of the decomposition coefficients, i.e. dark blue (white) for $c_{i,j} \geq 0$ and dark red (black) for $c_{i,j} < 0$. The bottom colour (gray-scale) panel contains the Hölder decomposition with five scale levels $i = 2 \ldots 6$.

5 Incremental calculation of the decomposition coefficients and the correlation product

One of the severe disadvantages of the Haar WT is the lack of translation invariance; when the input signal shifts by $\varDelta t$ (e.g. as the result of acquiring some additional input samples), the coefficients of the Haar wavelet transform need to be recalculated. This is rather impractical when one considers systematically updated inputs like financial records.

When the representation is to be updated on each new sample, little can be done other than to recalculate the coefficients. The cost of this resides mainly in the cost of calculating the inner product. Direct calculation is of nm complexity, where $n = 2^N$ is the length of time series and m is the length of the wavelet. The cost of calculating the inner product therefore grows quickly with the length of the wavelet and for the largest scale it is n^2. The standard way to deal with this problem is to use the Fast Fourier Transform for calculating the inner product of two time series, which in case of equal length reduces the complexity to $n \log(n)$.

Additional savings can be obtained if the update of the WT does not have to be performed on every new input sample, but it can be done periodically on each new n samples (corresponding with some $\varDelta t$ time period). In this case, when the $\varDelta t$ coincides with the working scale of the wavelet at a given resolution, particular a situation arrises:

- only the coefficients at scales larger than $\varDelta t$ scale have to be recalculated;
- coefficients of $f|_{x_0}^{x_0+\varDelta x}$ must be calculated anew;
- other coefficients have to be re-indexed or removed.

This is also illustrated in figure 4.

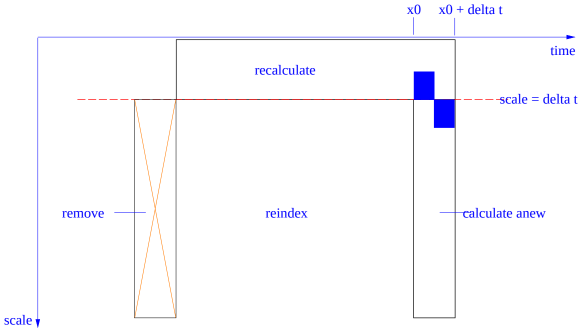

Fig. 4. Representation update scheme in the case of the shift of the input time series by $\varDelta t =$ working scale of the wavelet.

As expected, the larger the time shift $\varDelta t$, the fewer the number of the coefficients which have to be recalculated and the larger the number of coefficients which have to be re-indexed (plus, of course, the number of coefficients which have to be calculated from

$f|_{x_0}^{x_0+\Delta t}$). For the full details of incremental calculation of coefficients the reader may wish to consult [8].

6 Experimental Results

We took the records of the exchange rate with respect to USD over the period 01/06/73 - 21/05/87. It contains daily records of the exchange rates of five currencies with respect to USD: Pound Sterling, Canadian Dollar, German Mark, Japanese Yen and Swiss Franc. (Some records were missing - we used the last known value to interpolate missing values.) Below, in figure 5 we show the plots of the records.

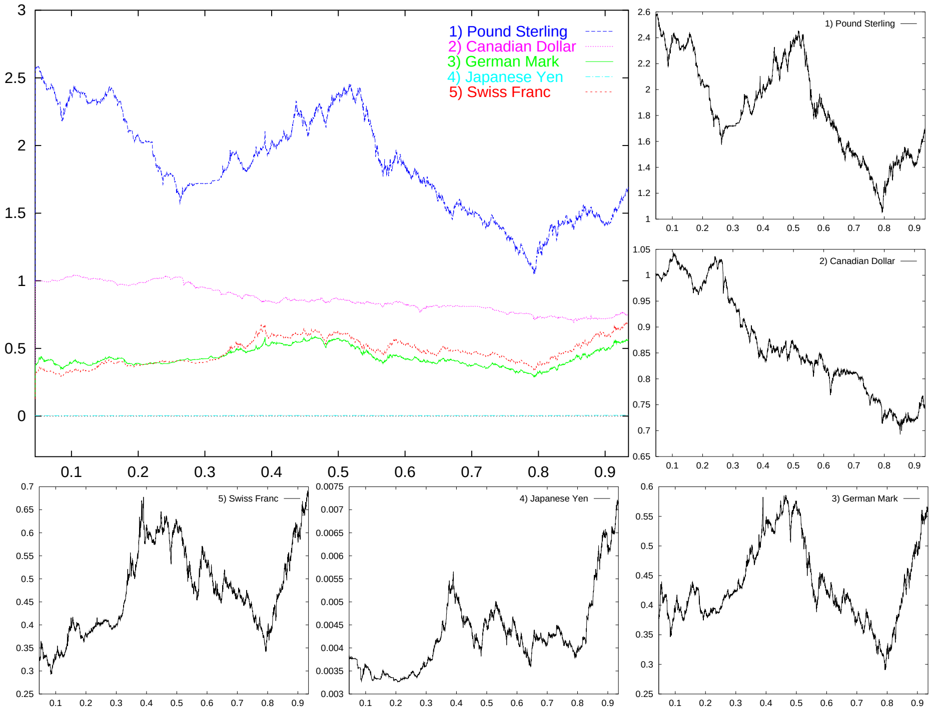

Fig. 5. Left above, all the records of the exchange rate used, with respect to USD over the period 01/06/73 - 21/05/87. In small inserts, single exchange rates renormalised, from top right to bottom left (clockwise), Pound Sterling, Canadian Dollar, German Mark, Japanese Yen and Swiss Franc, all with respect to USD.

All three representation types were made for each of the time series: the Haar, sign and Hölder representation. Only six scale levels (64 values) of the representation (five for Hölder, 63 points) were retained. These were next compared for each pair to give the correlation product.

In figure 6, we plot the values of the correlation for each of the pairs compared. The reader can visually compare the Haar representation results with his/her own ‘visual estimate’ of the degree of (anti-)correlation for pairs of plots in figure 5.

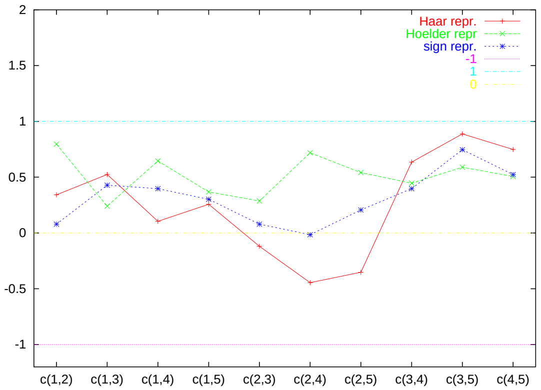

Fig. 6. The values of the correlation products for each of the pairs compared, obtained with the Haar representation, the sign representation, and the Hölder representation.

One can verify that the results obtained with the sign representation follow those obtained with the Haar representation but are weaker in their discriminating power (more flat plot). Also, the Hölder representation is practically independent of the sign representation. In terms of correlation product, its distance to sign representation approximately equals the distance of Haar represenation to the sign representation but with the oposite sign. This confirms the fact that the correlation in the Hölder exponent captures the value oriented, sign independent features (roughness exponent) of the time series.

7 Conclusions

We have demonstrated that the Haar representation and a number of related represenations derived from it are suitable for providing estimates of similarity between time series in a hierarchical fashion. In particular, the correlation obtained with the local slope of the time series (or its sign) in the sequence of multi-resolution steps closely corresponds to the subjective feeling of similarity between the example financial time series. Larger scale experiments with one of the major Dutch banks confirm these findings. The next step is the design and development of a module which will compute and update these representations for the 2.5 million time series which this bank maintains. Once this module is running, mining on the database of time series representations will be the next step.

References

1. R. Agrawal, C. Faloutsos, A. Swami. Efficient Similarity Search in Sequence Databases, In *Proc. of the Fourth International Conference on Foundations of Data Organization and*

Algorithms, Chicago, (1993).

2. R. Agrawal, K-I. Lin, H.S. Sawhney, K, Shim, Fast Similarity Search in the Presence of Noise, Scaling and Translation in Time Series Databases, in *Proceedings of the 21 VLDB Conference*, Zürich, (1995).
3. G. Das, D. Gunopulos, H. Mannila, Finding Similar Time Series, In *Principles of Data Mining and Knowledge Discovery*, Lecture Notes in Artificial intelligence 1263, Springer, (1997).
4. U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, Eds., *Advances in Knowledge Discovery and Data Mining*, AAAI Press/MIT Press, (1996).
5. C. Faloutsos, M. Ranganathan, Y. Manolopoulos, Fast Subsequence Matching in Time-Series Databases", in *Proc. ACM SIGMOD Int. Conf. on Management of Data*, (1994).
6. Z.R. Struzik, A. Siebes, Wavelet Transform in Similarity Paradigm I, *CWI Report*, **INS-R9802**, (1998), also in *Research and Development in Knowledge Discovery and Data Mining*, Xindong Wu, Ramamohanarao Kotagiri, Kevin B. Korb, Eds, *Lecture Notes in Artificial Intelligence* **1394**, 295-309, Springer (1998).
7. Z.R. Struzik, A. Siebes, Wavelet Transform in Similarity Paradigm II, *CWI Report*, **INS-R9815**, CWI, Amsterdam (1998), also in *Proc. 10th Int. Conf. on Database and Expert System Applications (DEXA'99)*, Florence, (1999).
8. Z.R. Struzik, A. Siebes, The Haar Wavelet Transform in Similarity Paradigm, *CWI Report*, **INS-R99xx**, CWI, Amsterdam (1999). http://www.cwi.nl/htbin/ins1/publications
9. I. Daubechies, *Ten Lectures on Wavelets*, S.I.A.M. (1992).
10. M. Holschneider, *Wavelets - An Analysis Tool*, Oxford Science Publications, (1995).
11. Z.R. Struzik, 'Local Effective Hölder Exponent Estimation on the Wavelet Transform Maxima Tree', *Fractals: Theory and Applications in Engineering*, Michel Dekking, Jacques Lévy Véhel, Evelyne Lutton, Claude Tricot, Eds, Springer (1999).

This article was processed using the LaTeX macro package with LLNCS style