Similarity Search Over Time-Series Data Using Wavelets

Ivan Popivanov
University of Toronto
Department of Computer Science
10 King's College Road
Toronto, Ontario
M5S 3G4, Canada
(416) 978 6986
popivan@cs.toronto.edu

Renée J. Miller
University of Toronto
Department of Computer Science
6 King's College Road
Toronto, Ontario
M5S 3H5, Canada
(416) 946 3621
miller@cs.toronto.edu

Abstract

We consider the use of wavelet transformations as a dimensionality reduction technique to permit efficient similarity search over high-dimensional time-series data. While numerous transformations have been proposed and studied, the only wavelet that has been shown to be effective for this application is the Haar wavelet. In this work, we observe that a large class of wavelet transformations (not only orthonormal wavelets but also bi-orthonormal wavelets) can be used to support similarity search. This class includes the most popular and most effective wavelets being used in image compression. We present a detailed performance study of the effects of using different wavelets on the performance of similarity search for time-series data. We include several wavelets that outperform both the Haar wavelet and the best known non-wavelet transformations for this application. To ensure our results are usable by an application engineer, we also characterize the best wavelet to use based on the characteristics of the data set. In addition to addressing the specific problem of similarity search, our results shed light on how more robust compression algorithms can be exploited within data management.

1 Introduction

The quantity of data stored in computers is growing rapidly. Much of this data, particularly data collected automatically by sensing or monitoring applications, is time-series data. A time series is a real-valued sequence, which represents the status of a single variable over time. The monitored activity can be a process defined by some human activity, like the fluctuations in Microsoft stock closing prices, or a natural process, like Lake Huron historical water levels. The presence of a time component in data is what unifies such diverse data sets and classifies them as time series. Therefore, it is hardly surprising that much research has been devoted recently to the efficient management of time-series data [AFS93, RM97, KAS98, KCPM00, et al].

Analysis of time-series data is rooted in the ability to find similar series. Similarity is defined in terms of a distance metric, most often Euclidean distance or relatives of the Euclidean distance

[AFS93]. Other distance metrics, including the $\mathcal{L}_p$ Norms may also be used [YF00]. Because of the high dimensionality of most time series, the direct indexing of time series is prohibitive. As a result dimensionality reduction appears to be the most promising method for overcoming this problem.

Agrawal, Faloutsos and Swami first proposed the use of distance preserving transformations for this task (specifically orthonormal transformations which preserve the Euclidean distance) [AFS93]. The transformations are applied to the original data and a few coefficients (or features) of the transformed data are then indexed. Queries on the data are transformed into queries on these features that can be efficiently answered using the index. The answer in the feature space, when converted back to the data space, must be a superset of the original query answer. This property has been referred to as the *contractive* property of the transform which ensures *no false dismissals*. The intuition is that the transformed query may have false positives (data that is not in the query result) but no false negatives or false dismissals. Hence, the results of the transformed query may be scanned to eliminate false positives and arrive at the correct answer to the original query. The correctness of this technique, referred to as an *F-index*, is predicated on having no false dismissals. The efficiency is determined largely by two factors: the precision and the number of features used in the index. The precision is the ratio of the size of the answer of the original query divided by the size of the answer of the transformed query. The closer the precision is to one, the more efficient the technique will be. The precision is a measure of how well the transformation is able to capture the information or energy of the data in the indexed features. The number of features used in the index will also effect performance. The use of too many features will render the index search less effective as the performance of even the best multidimensional index strategies decreases in high dimensions [WJ96, BKK96, KS97, CPZ97, CM99].

Given this basic framework, there are still many open questions. We will address the following questions in this work.

1. Which transformations are effective for similarity search over time series? The original work by Agrawal *et al*, as well as subsequent research [FRM94, RM98], used the Discrete Fourier Transform (DFT) for feature extraction. Later on, the Singular Value Decomposition (SVD) transform was suggested as a very accurate (high precision), but computationally expensive, alternative [WAA^{+}96]. More recently, the Haar Wavelet [CF99, WAA00] and other similar techniques [KCPM00, CM99, YF00] have been used to improve various aspects of the similarity search process.

The incorporation of new and better transformations has not yet taken advantage of the rapidly growing suite of sophisticated wavelet transformations being produced in the data compression community. These transformations are quickly been adopted for numerous applications. For example, the *JPEG 2000* [JPE00] standard replaces the Discrete Cosine Transform (a relative of DFT) used by its predecessor, with one of the Daubechies transforms (with filter lengths 9/7) [ABMD92] and a symmetric 5/3 transform [TlG88], for lossy and lossless compression, respectively. Inspired by the success of these transforms in the closely related area of compression, we present a study of their use in dimensionality reduction in time series. The only wavelet that has been studied for this latter application is the Haar wavelet [CF99, SS99a, SS99b]. The Haar wavelet is the simplest wavelet. In the data compression community, it is often used as a teaching example, but rarely in applications [Dau92].

Note that the question of which transformation to use is not simply a matter of search performance. In particular, we are not just looking for the best transformation in terms of search time. Different transformations may have different properties appropriate for different applications. Hence, different transformation may be preferred if they are consistent with the application semantics.

Our first contribution is the observation that not only orthonormal, but also bi-orthonormal wavelets, can be used for similarity search. This observation permits us to use a large class of wavelets that arguably have found the most success in compression. For example, note that both the wavelets used in the JPEG 2000 standard are bi-orthonormal, though neither is orthonormal.

Our second contribution is an empirical study that details the performance of a large number of wavelets in similarity search. We show that some of these more sophisticated wavelets can outperform both Haar and the standard transforms used in similarity search.

2. How do different transformations interact with index-based similarity search? To use wavelets or any transformation in practice for this application, we must know how to best configure our indexing technique. For wavelets, both the wavelet function and the filter length can greatly effect the performance of the index. The way in which each transformation concentrates the energy or information of the data into features can be quite different. As a result, different transformations may be sensitive to the number of features used in the similarity search. This sensitivity and the interaction of this sensitivity with the index must be understood. In general, if more features are indexed, the precision will be higher, but the index search less effective. But the specifics of this trade-off will vary with the transform and must be understood.

Our third contribution is a characterization of how to configure an indexing strategy for the best performing transformations. That is, we empirically determine the number of features and the filter lengths that optimize these transformations.

3. What transformations are best for which classes of data? Different transformations are effective at capturing different properties of the data. As a results, we expect that different transformations will be suitable for different classes of data. An important observation is that not all time-series datasets are alike. Consider some of the applications that have served as motivations for much of this work.

- *"Find if a musical score is similar to one of the copyrighted scores" [AFS93].*
- *"Find all currencies whose prices w.r.t. [the] US Dollar have changed similarly to the price of gold for a specific period of time" [YF00].*
- *"Find past days in which the solar magnetic wind showed patterns similar to today's pattern" [FRM94].*

Acoustic signals may behave very differently from currency or stock patterns which in turn may differ dramatically from measurements of natural phenomenon like water levels or wind speeds [Sch91]. Most work on similarity search has focussed on data that can be classified as Brown noise (with special attention paid to stock prices) [AFS93, CF99, RM98, KCPM00, YF00]. Furthermore, most have evaluated the technique on synthetically generated "pure" Brown noise (generated using random walks). However, not all time-series data is Brown noise, and not all real Brown noise data is "pure" Brown noise. Important (and common) classes of time-series data may also be classified as Black or Pink noise or may fall in between these classes [Sch91].

Our final contribution is a study including different classes of time-series data (ranging from Black, to Brown, to Pink and in between). We consider whether each class can in fact be indexed effectively using an F-index. For each indexable class, we show the performance of our proposed wavelet transformations. In addition to addressing the specific problem of similarity search, our results shed light on how these more robust compression algorithms can be used within data management. Specifically, we show that wavelet techniques can be exploited in managing and analyzing not only highly correlated Black and Brown noise data, but also data that lies in the area between Brown and Pink. This is an important result as many real time-series data sets cannot be modeled by pure random walks (that is, they are not pure Brown noise) and hence fall into this region.

The rest of the paper is organized as follows. In Section 2, we survey the related work. In Section 3, we discuss the wavelet transform and its application as a dimensionality reduction technique. We show how bi-orthonormal wavelets may be used in similarity search. Our experimental results are reported and discussed in Section 4. Finally, Section 5 contains our conclusions and the direction of our current work.

2 Related Work

Similarity matching of time-series data was first examined by Agrawal *et al* [AFS93]. They suggest the use of the DFT for feature extraction, arguing that that most real signals need only the first few

Fourier coefficients to approximate them. They introduce an indexing mechanism called an *F-Index* which has been used as the framework for much of the subsequent work in this area. Faloutsos *et al* generalized the F-Index for subsequence matching [FRM94]. This early work proved that in order to guarantee no false dismissal, for a particular transform T, the distance measure in the feature space (for any two object x and y) must satisfy the following lower bounding lemma or contractive property.

$$D_{\text{feature}}(T(x), T(y)) \leq D_{\text{object}}(x, y), \tag{1}$$

Rafiei and Mendelzon showed how to handle moving averages in an F-Index framework [RM97]. In follow on work, they suggest the use of the symmetric property of the DFT to increase the precision of the distance measure without increasing the number of features stored in index [RM98]. Time warping distance has also been considered [RM97, KP00a, YJF98]. Chan and Fu used wavelets for dimensionality reduction in this application. In their work, they used the simplest wavelet, known as Haar wavelet, and showed performance improvements over DFT [CF99]. Struzik and Siebes have also applied the Haar wavelet in this domain [SS99a, SS99b]. The Piecewise Aggregate Approximation (PAA) transform, which is similar to the Haar transform, has also be used for similarity search [KP00b, KCPM00]. This work also extended the F-index framework to support weighted Euclidean distance [KCPM00]. Yi and Faloutsos used a similar transform based on segmented means and showed that it supported similarity search under any of the $\mathcal{L}_p$ norms [YF00].

3 Dimensionality Reduction Using Wavelets

The input time series (a finite sequence of real values) are alternatively called objects, sequences or signals in the literature. We will also interchange the terms depending on context and the aspect of the data we are discussing. The process of dimensionality reduction can be described as follows. The original times series or signal is a finite sequence of real values, or *coefficients*, recorded over time in some *object space*. This signal is transformed (using a specific transformation function) into a signal in a *transformed space*. To achieve dimensionality reduction some subset of the transformed coefficients are selected as *features*. These features form a *feature space* which is simply a projection of the transformed space.

The Fourier transform is based on the simple observation that every signal can be represented by a superposition of sine and cosine waves. The Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT) are efficient forms of the Fourier transform often used in applications.

Wavelets can be thought of as a generalization of this idea to a much larger family of functions than sine and cosine [Dau92, VK95]. Mathematically, a "wavelet" denotes a function $\psi_{j,k}$ defined on the real numbers R, which includes an integer translation by k, also called a shift, and a dyadic dilation (a product by the powers of two), often referred to as stretching. The following set of functions where j and k are integers, form a complete orthonormal system for $L^2(R)$.

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k), \tag{2}$$

Using these functions, we can uniquely represent any signal $f \in L^2(R)$ by the following series.

$$f = \sum_{j,k \in Z} \langle f, \psi_{j,k} \rangle \psi_{j,k}. \tag{3}$$

Here $\langle f, g \rangle := \int_R f\bar{g}dx$ is the usual inner product of two $L_2(R)$ functions. The functions $\psi_{j,k}(t)$ are referred as the *basis functions* of the wavelets. Note that while the Fourier transform has a single basis function (the exponential function), wavelets make use of an infinity family of basis functions.

The Haar wavelets are the most elementary example of wavelets. Although they have many drawbacks, they still illustrate in a very direct way some of the main features of wavelets and so we present them as an example. From Equation (2), we see that each wavelet is built upon a particular function ψ. This function is often referred to as *mother wavelet*. The mother wavelet for the Haar wavelets is the following function.

$$\psi_{Haar}(t) = \begin{cases} 1, & \text{if } 0 < t < 0.5 \\ -1, & \text{if } 0.5 < t < 1 \\ 0, & \text{otherwise} \end{cases}$$

The Haar family of wavelets is produced using this mother wavelet and varying j and k in Equation 2. Daubechies [Dau92] discovered that wavelet transforms can be implemented using a pair of Finite Impulse Response (FIR) filters, called a Quadrature Mirror Filter (QMF) pair. These filters are often used in the area of signal processing as they lend themselves to efficient implementation. Each filter is represented as a sequence of numbers. The *filter length* is the length of this sequence. The output of a QMF pair consists of two separate components: a high-pass and a low-pass filter output, which correspond to high-frequency and low-frequency output, respectively. Wavelet transforms are considered to be hierarchical since they operate stepwise. The input on each step is passed through the QMF pair. Both the high-pass and low-pass component of the QMF output are half the length of the input. The high-pass component is naturally associated with details while the low-pass component concentrates most of the energy or information of the data. The low-pass component is used as further input, thus, reducing the length of the input by a factor of 2 at each step.

Example 1 *As an example, suppose we have the following one-dimensional signal with* 8 *coefficients:* $X = [8, 2, 9, 3, 4, 6, -1, 7]$. *Applying the Haar Wavelet transform to X, we obtain the* *following low-pass output:* $X' = [5, 6, 5, 3]$.

For the Haar wavelet, the transformed coefficients at different resolution levels are meaningful. The low-pass filter output is actually the pairwise average of the input. That is, the first two values of the original signal (8 *and* 2*) average to* 5*. Similarly,* 9 *and* 3 *average to* 6*,* 4 *and* 6 *average to* 5*. Notice that having the average it is not possible to reconstruct the original values, since an average value of* 5 *is the result of both pairs* $(8, 2)$ *and* $(-3, 13)$*. The high-pass filter corresponding to* X' *is given in the table below. The high-pass filter stores the information needed to exactly reconstruct the original signal. Specifically, it stores the distance between the average and the two values in the original signal. The first value of* -3 *in the high-pass filter indicates that the original pair is* $(8, 2) = (5-(-3), 5+(-3)$*. Notice that the sign controls whether it is* $(8, 2)$ *or* $(2, 8) = (5-3, 5+3)$*.*

The sequence X' *is used as the input for the next level of the decomposition process. Since the input has length* 8 *it takes two more steps to finish the decomposition process, and we get the following high-pass and low-pass components.*

Resolution	*low-pass*	*high-pass*
8	*[8, 2, 9, 3, 4, 6, -1, 7]*	
4	*[5, 6, 5, 3]*	*[-3, -3, 1, 4]*
2	*[5.5, 4]*	*[0.5, -1]*
1	*[4.75]*	*[-0.75]*

The original signal is represented by the last low-pass features and all the detail (high-pass) features.

$$X_{DWT} = [4.75, -0.75, 0.5, -1, -3, -3, 1, 4].$$

Notice that we can reconstruct uniquely the original signal by applying the reverse process. For dimensionality reduction, we would choose some prefix of this transformed signal to approximate the original signal. The more features we use, the closer will be the approximation of the original signal.

3.1 Benefits of Wavelet Transforms

Practical experience has shown that for many applications wavelet transforms are as powerful and versatile as the Fourier transform, yet without some of the limitations of the latter. Wavelets have numerous properties that can be exploited in data management [Kei01]. We briefly present a few of the most commonly cited properties

- Some wavelet transforms have *compact support*. This means that the basis functions are non-zero only on a finite interval. For comparison, the sine and cosine waves of the Fourier transform are infinite in extent. As a result, unlike the Fourier transform, which maps signals from the time domain into the frequency domain, wavelet transforms maps signals from time domain into time-frequency domain. What this means for an application is that wavelets are able to capture local (time dependent) properties of data, whereas Fourier transforms can only capture global properties.

- The efficiency of the wavelet transform is superior even when compared with the Fast Fourier transform. The Fourier transform is $O(n^2)$, where n is the length (number of attributes) of the data and Fast Fourier transform is $O(nlogn)$. In general, the speed of wavelet transforms is *linear* in the length of the data.[1]

- The Fourier transform gives the set of frequency components, which exist in our signal. On the other hand, wavelet transforms give gradually refined representation of the signal of different scales, which correspond to basis functions of different length. Hence, the wavelet transform is hierarchical and allows much finer tuning for a variety of applications [LYC96].

- Unlike the Fourier transform, wavelet transforms have an infinite set of possible basis functions. Thus, they provide access to information that can be obscured by other methods.

These properties and others have been exploited extensively for managing images [JWW97, JWW98, and others] and for a variety of data compression applications including selectivity estimate [MVW00], approximate query answering [CGRS00, VW99] and clustering [SCZ00]. However, their use as a scalable dimensionality reduction technique for time-series data has not yet been fully appreciated. In the next section, we will show how these properties can be exploited to improve the existing methods for similarity search.

3.2 Wavelet Comparison

We distinguish wavelet *families* which are named after their creator. The wavelets are implemented using a pair of FIR filters. Since the filters are of different lengths, we use the length to distinguish among the wavelets within each family. For example, Coiflet 4 (or the Coiflet wavelet of length 4), refers to the wavelet of the Coiflet family, which is implemented using a filter of length 4. The one exception to this is the Haar wavelet which has a fixed filter length of 2, and thus we do not indicate the filter length.

Several researchers have noted that wavelets can be used in similarity search. Chan & Fu [CF99] proposed and studied the use of the Haar wavelet transform as a dimensionality reduction technique for this application. Both this study and a similar study by Wu *et al* showed that the Haar wavelet captures the shape of time series better than Fourier transform within the context

[1] Another performance benefit of using wavelet transform is that it is real to real transform, hence, it is much easier to implement, and it reduces the pre-processing and post-processing of data. The Fourier transform is a complex transform, although its close cousin the Discrete Cosine transform is real to real.

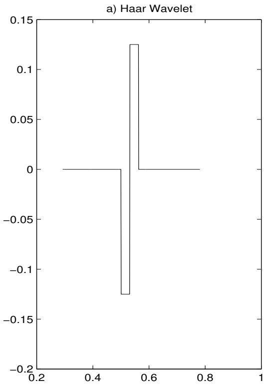

Figure 1: Haar and Daubechies 12 wavelets

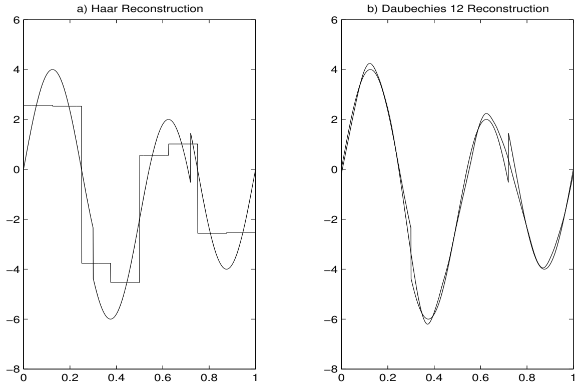

Figure 2: Haar and Daubechies 12 reconstruction

of similarity search [WAA00]. Although, the Haar wavelet shares most of the properties of other wavelets, there is a major drawback. The basis functions for the Haar wavelet, are not smooth (*i.e.*, they are not continuously differentiable). Rather, they have the shape shown in the Figure 1a. Hence, the Haar wavelet approximates any signal by a ladder-like structure. This undesirable effect when approximating close to smooth functions is illustrated in Figure 2a. Using 8 features of the Haar transform, we get the step-like signal depicted in this figure.

Hence, the Haar wavelet is not likely to approximate a smooth function using few features. The number of features we must use is high (this property is referred to as slow convergence). For comparison, we approximated the same function using the Daubechies wavelet with a filter length of 12. Figure 1b depicts the shape of the Daubechies basis functions and Figure 2b gives the reconstructed signal (again using 8 features).

To measure and compare the quality of different transforms, we will use the *mean relative error* between the original signal and the transformed signal.

Definition 3.1 *Given a signal x of length N, let $\bar{x}$ denote any approximation of x. The* mean relative error *is defined as the mean of the relative error for each feature.*

$$MeanRelativeError = \frac{1}{N}\sum_{i=1}^{N}\frac{|\bar{x}[i]-x[i]|}{|x[i]|}$$

In the example of Figure 2, the mean relative error for reconstruction using the Haar wavelet was 1.6576. The error improved to 0.4771 when we used the Daubechies wavelet to reconstruct the original signal. Note that the compression ratio is $8/1024 = 0.078$ in both cases. Using Daubechies, just a few features are enough to have a meaningful close representation of the original data. This property is extremely important for similarity search as the efficiency of the index search will deteriorate if too many features are used.

Schroeder suggests that real world time-series data are indeed smoother than random data [Sch91]. He goes on to characterize the smoothness using the energy spectra of the data which are captured by functions of the form $f^{-\beta}$. Data sets with a constant energy spectrum (that is $\beta = 0$) are called white noise. As β increases, so does the smoothness of the data. Note that the smoothness is a measure of the correlation in the data. A skewed energy spectrum means that the data is strongly correlated, and as a result the data looks more like a smooth function than like random data. Schroeder's classification is summarized in Table 1.

β	Name	Examples from the class
$\beta = 0$	*White noise*	Random data
$\beta = 1$	*Pink noise*	Acoustic signals
$\beta = 2$	*Brown noise*	Stock prices and exchange rates
$\beta > 2$	*Black noise*	Natural phenomenon such as wind speeds and water levels

Table 1: Classification of time-series data characteristics.

Different types of data can be (grossly) assigned a class. For example, acoustic signals are commonly regarded as Pink noise, many data sets recording long range activities such as stock prices are typically brown noise and many measurements of natural phenomenon are black noise. However, these semantic groupings are not absolute. Furthermore, data may lie in between these classes. Indeed, in our experiments (Section 4), we use a river data set (measurements of river flow) that has an energy spectrum around 1.4 making it a light ruddy tan, not black as might be assumed by its semantics. The importance of this classification though, is that it gives us a way of determining how smooth a data set is, and as a result of picking a good wavelet for approximating it. By using a wavelet which better captures the specific characteristics of time series data sets, the same number of features will contain a much better description of the original data.

Unlike Haar wavelet, most other wavelets are smooth. Throughout our experiments, we observed a number of wavelets outperforming the Haar wavelet. However, some wavelets with smooth basis functions, show inferior performance compared to the Haar wavelet, since their basis functions are not similar to the shape of time-series data. It is therefore important to empirically validate which wavelet functions perform well for this application (Section 4). Before we turn to this evaluation, we consider which classes of wavelet functions can be used for similarity search.

Many early wavelets have been developed based on function bases with certain properties that are adapted to particular applications [Str81, Mey98]. Development of wavelets with more continuous derivatives has attracted the interest of researchers in many areas and spawned the active area of wavelet theory [Dau88, Dau93]. The Daubechies wavelet families were among the first wavelets with compact support, and are still the most intensively used in practical applications. Wang argues that their properties are fundamental in image processing [Wan98]. These wavelets have been used extensively in the Stanford Image Database Project [JWW98, JWW98]. *Coiflet* wavelets [Dau93] and *Spline* wavelets [Chu92], to name just a few, are other families of practically used wavelets. Recently, the FBI introduced wavelet compression support for their fingerprint database [Bri96]. The emerging *JPEG 2000* [JPE00] standard replaces Discrete Cosine Transform used by its predecessor, with wavelet transforms.

3.3 Using Wavelets in Similarity Search

Before we can use general wavelet transforms for similarity search, we must show that they have the contractive property and thus can be used in a way that guarantees no false dismissals. Chan and Fu state that the contractive property is only known for the Haar wavelet [CF99]. In this section, we show that we can in fact use any bi-orthonormal wavelet for similarity search.

The contractive property for the relative distance in the object and feature spaces can be used to guarantee that an F-index does not result in any false dismissals.

$$D_{\text{feature}}(T(x), T(y)) \leq D_{\text{object}}(x, y) \tag{4}$$

Fukanaga includes a proof that the Euclidean distance is preserved for the class of orthonormal transforms [Fuk90]. The Haar wavelet as well as many other wavelets belong to the class of *orthonormal* wavelets. Intuitively, this means that the transform can be implemented using an orthonormal matrix. Hence, for a pair of objects x and y transformed using an orthonormal transformation T the following equality holds.

$$D_{\text{transform}}(T(x), T(y)) = D_{\text{object}}(x, y) \tag{5}$$

The feature space is generated by extracting a subset of F features from the transformed space. Typically, the subset is a prefix of the feature sequence but it may be any subset. We use $T_f(x)$ to denote this subset of features for the transform of a specific object x. A simple analysis of the Euclidean distance yields the following result.

$$D_{\text{feature}}(T_f(x), T_f(y)) \leq D_{\text{transform}}(T(x), T(y)) = D_{\text{object}}(x, y) \tag{6}$$

Hence, to use any orthonormal wavelet in an F-index method, we simply transform the query object q (where the query predicate is *find all objects within a distance* ϵ *of* q). We extract the indexed features from $T(q)$ and use the index to find all objects x such that $D_{\text{feature}}(T(q), T(x)) \leq \epsilon$. For indices that use hyper-rectangles rather than spheres (for example R^*-trees), we actually need to transform the query predicate to find all objects that lie in a hyper-rectangle of width 2ϵ centered at $T(q)$. The retrieved objects ($T(x)$) are then scanned to obtain the original data points and each is tested to see if $D_{\text{object}}(q, x) \leq \epsilon$.

However, many wavelet transforms used in practice are not orthonormal. Indeed, the majority of wavelets used in the area of image compression, belong to the class of *bi-orthonormal* wavelets (defined below). As we will see, the class of orthonormal wavelets, is a subset of the class of bi-orthonormal wavelets. The contractive property has not been shown for the class of bi-orthonormal wavelets.

There are certain requirements that a wavelet should satisfy. It has been proven that a necessary and sufficient condition for stable reconstruction is that the energy of the wavelet features lies between two positive bounds [Chu92]. That means that there exists constants A and B such that the following holds. Here x is a data object and X is the transformation of x under the wavelet, $X = T(x)$.[2]

$$A\|X\|^2 \leq \|x\|^2 \leq B\|X\|^2 \tag{7}$$

Note that for $A = B = 1$, Equation (7) turns into an equality: $\|x\|^2 = \|X\|^2$

This equality shows that the energy is preserved and, actually it holds for orthonormal transforms. Hence, the class of bi-orthonormal wavelets is a superset for the class of orthonormal wavelets. However, for bi-orthonormal transformations, the Euclidean distance is *not* in general preserved. However, Equation (7) is sufficient to let us determine a transformed query predicate that will guarantee no false dismissals.

Notice first that for ϵ-queries of radius ϵ with a query object q (and transformed query object $Q = T(q)$) we have the following predicate.

$$A\|X - Q\|^2 \leq \|x - q\|^2 < \epsilon^2 \tag{8}$$

The last inequality implies, that a necessary condition for the norm to be less than ϵ^2 is that every magnitude on the left be less than ϵ^2. Hence, all the qualifying points in the transform space must satisfy the following inequality.

$$\|X - Q\|^2 < \frac{\epsilon^2}{A} \tag{9}$$

So for each dimension k, we have the following.

$$|X[k] - Q[k]| < \frac{\epsilon}{\sqrt{A}} \tag{10}$$

As a result, using a search sphere of radius $\frac{1}{\sqrt{A}}\epsilon$ or a hyper-rectangle of width $\frac{2}{\sqrt{A}}\epsilon$ in each dimension in the feature space will guarantee that we obtain all the required points. Thus, using this transformed query predicate, we can guarantee there are 'no false dismissal' even for bi-orthogonal wavelets.

[2] Note that we are using standard vector notation to denote the Euclidean distance.

4 Experimental Results

To verify the effectiveness of our proposed method, we performed an extensive set of experiments on both real and synthetic data. Each experiment is designed to answer one of the following questions.

1. Do any wavelets outperform the Haar wavelet and the best known non-wavelet transformations? We first wished to validate our hypothesis that some of the more robust wavelets can be effective for similarity search. We also wished to understand which families of wavelets show good performance for this application.

2. How does the number of dimensions or features used in the indexing effect the search performance? The answer to this question is important to being able to configure the F-index.

3. Within a given family of wavelets (that is, for a fixed wavelet function), how does the filter length effect the precision of the query?

4. Are wavelets effective for different data classes?

4.1 Experimental Setup

All the experiments use the same multidimensional index structure, namely Norbert Beckman's Version 2 implementation of the R^*-tree [BKSS90]. For the wavelet transformations, we used the "Imager Wavelet Library" which is a research library [Fou95]. This library, while perhaps not containing the fastest implementations of specific wavelets, did permit us to experiment with a large suite of dozens of wavelet functions. For comparison with DFT, we used one of the best known implementations [FJ97]. Because of this choice, comparison of CPU times for the transformations would be very biased and unreliable. While the complexity of DFT is $O(nlogn)$ compared to $O(n)$ for wavelets, we actually observed CPU times that were about 6 times faster for DFT than for the Haar wavelet. We do not believe this is a fundamental constraint of wavelets but merely an artifact of these implementations. Furthermore, with the tremendous success of wavelets in numerous applications, we expect to see optimized libraries for wavelets emerging soon.

Different approaches have been proposed to compare the effectiveness of different transformations. We will use the query *precision* which we define as follows.

$$Precision(Q) = \frac{\text{Number of sequences in answer of Q}}{\text{Number of sequences retrieved using the index}} \tag{11}$$

Dimensionality reduction implies information loss. Therefore, the size of the data set retrieved using the index is always greater or equal to the size of the actual result set and the range for the precision is $[0, 1]$.

Parameter	Symbol	Ranges	Default
Number Features	F	3 - 21	9
Number Sequences	S	36,000	36,000
Length Sequences	D	128 - 512	128
Number Data Pages	P	9,000	9,000
Query Selectivity	s	.01 - .1	.01

Table 2: Summary of experimental parameters

The precision, however, only compares the pruning power of a particular technique, and it does not measure overall performance. To ensure our results are not biased by implementation details (such as our choice of transformation libraries), we report the performance in terms of the number of physical page accesses. Both for the index and for the database we use $4,096$ bytes as the default page size. The parameters for our experiments are summarized in Table 2. We follow the convention in this field of reporting results for relatively small databases as the improvements for larger datasets will be even more pronounced [AFS93, KCPM00]. We use sequences of length 128 as a default. Many of the experiments were conducted with a range of different input sequence lengths and we found the qualitative results unaffected by the sequence length. Where this was not the case, it is noted in the text.

4.2 Wavelet Study

Our first experiment is designed to determine whether any of the more robust wavelets can be effective for similarity search. To address this question, we copied an experimental set up from Wu *et al* which was designed to compare the use of the DFT (using the symmetry optimization [RM98]) with the Haar wavelet [WAA00].

The data set consists of $1,647$ stocks and their historical quotes in several time frames (daily, weekly, etc.)[3] We selected a set of 100 stocks and for each stock, we extracted the closing prices for the last 360 days. Then we used a 128-day sliding window, starting at the beginning and taking a sample at each data point. When the window reached the end of the 360 days sequence, we warped the beginning of the 360-day sequence to the end. Since we started with 100 stocks and for each sequence we have 360 subsamples, we ended up with 36000 time-series, each of which is 128 days long. The size of the database is approximately 37Mbs, stored in $9,000$ data pages. Notice that this database is created using only a fraction of the original data, which we obtained from Yahoo. We used the remaining data to generate the queries for the experiments. All reported results are averaged over execution of 100 queries.

Our first aim was to study the pruning power of the competitive techniques. Therefore, we fixed the number of index dimensions to 9, but we varied the query selectivities from 0.02 to 0.1.

[3]This data set was obtained from Yahoo ("http://chart.yahoo.com/").

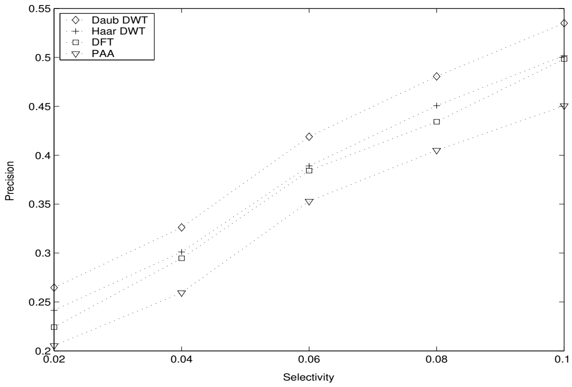

Figure 3: Precision of wavelets

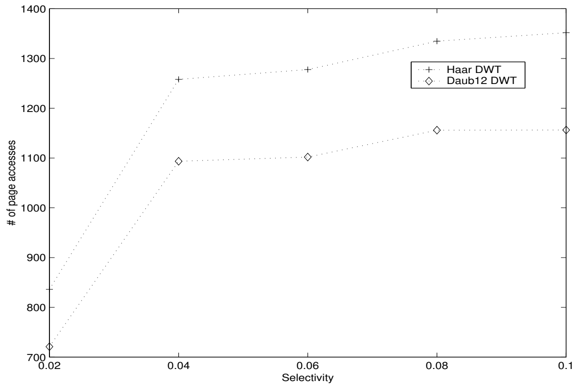

Figure 4: Performance of wavelets

Figure 3 shows the precision over different query selectivities. Note that the results for DFT and Haar are very close as is suggested by Wu *et al* [WAA00].[4] However, the precision of the Daubechies wavelet is significantly better. This confirms our prediction that the precision can be improved using wavelets that model the characteristics of the dataset better. PAA is the Piecewise Aggregation Approximation (also known as segmented means) that has been used in several recent studies [KP00b, KCPM00, YF00]. We did not expect PAA to perform well on this pruning-power test. However, we have included it since it has been shown to have good performance in increasing classification accuracy for weighted Euclidean distance [KCPM00]. Furthermore, it is the only technique suitable for applications that require similarity models based on all $\mathcal{L}_p$ norms [YF00].

To understand the impact on performance of this improvement, we study the relationship between precision and IO (expressed as the number of page accesses). The page access performance over the range of selectivities is shown in Figure 4. For reference, the database has 9,000 pages so even at the highest selectivity of 10%, the page IO is only 13% of the total database size. The improvement of Daubechies 12 over Haar is about 15%.

We performed this experiment with a large class of wavelets from our library. For this data set, which is an example of brown noise ($\beta = 1.99$), we found the Daubechies wavelet with filter length 12 outperformed all other wavelets in the library. This included the Coiflet family and two families of Bathlet wavelets (around 20 wavelets of different filter lengths) among many others. We expect the reason for this performance has to do with the length of the Daubechies filter. Most of the other wavelets in our library had filter lengths less than 10. Intuitively, the length of the filter reflects the length of the patterns that the wavelet is able to capture and use in compressing the data. Given the success of the wavelets with longer filters (the Daubechies 16 performed comparably to the Daubechies 12), we studied in detailed the effects of filter length in a separate experiment (Section 4.4).

[4] Chan and Fu did find that Haar showed improvement over DFT, but we believe this is because they did not use DFT with Rafiei and Mendelzon's symmetry optimization [RM98].

4.3 Dimensionality Study

To study the impact of the index structure on the overall performance, we analyzed the number of page accesses over a range of dimensionalities. For this study, we used the same stock data as in the previous study and the best transformation for this type of data, the Daubechies 12. The query selectivity was 0.01. To understand the influence of the number of dimensions used in indexing, we plot both the precision for different dimensions (Figure 5) and the performance (Figure 6).

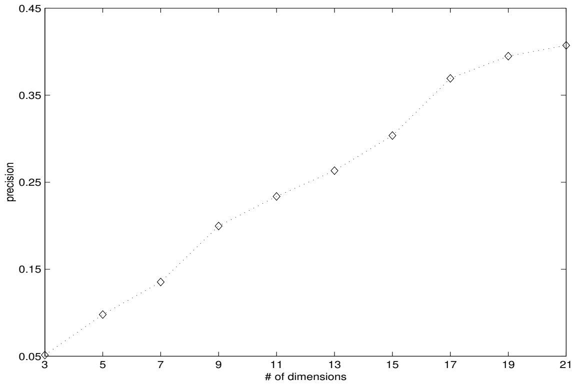

Figure 5: Precision vs. # dimensions

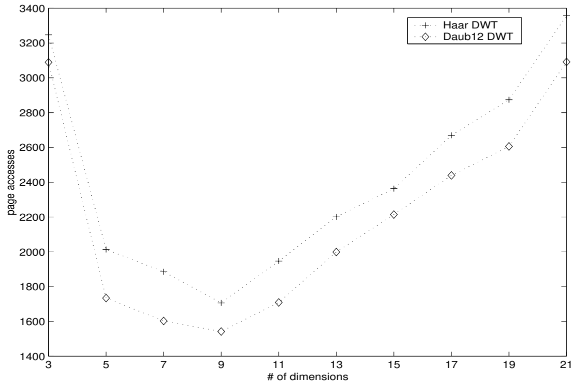

Figure 6: Performance vs. # dimensions

Figure 5 confirms our intuition that more index dimensions improve query precision. Increasing the number of dimensions also improves search performance, up to a point. To explain this, first notice that the page accesses can be broken down into two components.

$$\text{Page accesses} = \text{Index page accesses} + \text{Database page accesses}.$$

An improvement in precision means that the number of database page accesses in the above formula is decreasing. At the same time, the number of index page accesses is steadily growing as the performance of all multidimensional index structures decreases with added dimensions. Before the minimum (in this experiment, 9 dimensions), the growth in the number of index page accesses is less than the reduction in the number of database page accesses, caused by the improved precision. Hence, the search performance is improved. After this minimum, the situation is reversed. This same trade-off will be present with any index structure although the optimal point may be reached at a different number of dimensions.

For the index structure we chose, we found 9 dimensions to be optimal for most wavelets. We also found the same optimum for other data sets that were Brown or midway between Brown and Pink (for example, the River data set described in Section 4.5). Hence, we believe this number is influenced primarily by the index structure.

To confirm that the optimum depends on the index structure, we decided to perform similar experiments using other multi-dimensional indices. Collosi and Nascimento have bench-marked

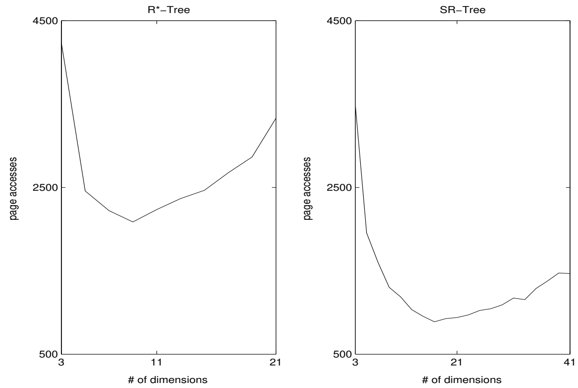

Figure 7: R*-tree vs. SR-tree

a set of promising high-dimensional index structures [Col00]. The SR-tree [KS97] and the M-tree [CPZ97] are among the structures that clearly outperform the other competitive techniques in their experimental set. We ran an experiment to compare the R*-tree and the SR-tree, an implementation of which is publicly available. We fixed the selectivity to 0.01 and we used the Haar wavelet transform for both index structures. For the SR-tree, the optimal number of dimensions shifted to 17. Furthermore, we observe that the SR-tree exhibits much slower deterioration when the number of dimensions grows. However, Figure 7 confirms that the overall behavior of SR-tree is similar to the one observed for the R*-tree.

4.4 Filter Length Study

For this experiment, we studied the effect of the filter length on the precision of the query and the performance of the search. We used the Daubechies family of wavelets for several reasons.

- It includes the longest filter length (20) that we were able to find. Given the success of longer filters in earlier studies, we felt it was important to systematically study these filters.
- It includes a larger set of filter lengths than other families. For example, the Coiflet family is represented by filters of three different lengths (2, 4, 6), for Daubechies family we have filters of eight different lengths (4, 6, 8, 10, 12, 14, 16, 20).[5]
- It performs well for all type of data.

We varied the filter length from 4 to 20 and studied the precision. The results are shown in Figure 8.

The precision of the wavelets increases with increased filter length to a point, then begins to decrease. For the Daubechies family, both the filter lengths 12 and 16 performed comparably, with only a slight variation depending on the data class. Furthermore, these were the best of all the

[5]Note that our library only came with Daubechies 12 and 20, but we were able to add in Daubechies 14 and 16.

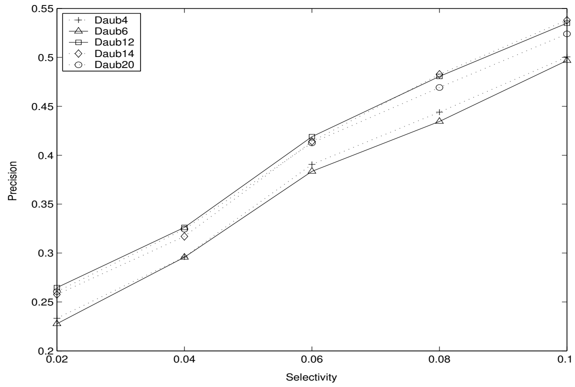

Figure 8: Precision vs. filter length

filter lengths. Note that the filter of length 20 is worse than length 12. We believe the explanation for this is that time-series data exhibits strong patterns of length $12 - 16$ so shorter filters are not able to take advantage of these trends in compression. For longer filters, these trends are obscured by additional data.

To better quantify the impact to a user of this performance difference, we measured the corresponding access times for Daubechies 4 and the Daubechies 12 wavelets. The query precision, thus, the number of data page accesses is the most reliable way to present performance evaluation when comparing dimensionality reduction techniques. However, for this study we used the same transformation library which permitted us to also do a reasonable time comparison. For this experiment we fixed the dimensionality of our R*-tree to 9 and we varied the selectivity of the queries. To measure time, we recorded the wall clock time by using the 'times' system call which is similar to UNIX 'time' utility. We used one of our computational servers to perform this test. To avoid the

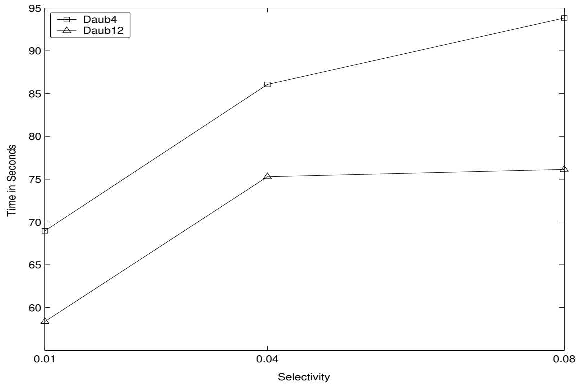

Figure 9: Daubechies 4 vs Daubechies 12

influence of the server workload, we increased substantially the number of tests and we ran the tests for the same selectivity consecutively. We used the "median" method to present the results, i.e., we threw away the highest and the lowest result for each run and we took the average over the rest.

The time used for the false alarm pruning stage is shown in Figure 9. Despite our precautions, we observed some anomalies, that confirmed our assumption that measuring time cannot be considered universal. However, it was notable that Daubechies 12 outperformed the Daubechies 4 by a margin of at least 9 seconds for all of the experiments. For a the .01 selectivity, this corresponds to almost a 20% improvement.

4.5 Data Study

White noise (completely random) data, with a spectral exponent $\beta = 0$ is not a subject for our indexing method. On the other hand, brown noise, with spectral exponent $\beta = 2$, has been shown very suitable for indexing using dimensionality reduction techniques. A natural question is how an F-index performs for signals in the gap between white noise and brown noise, or for signals classified as black noise. A related, yet more fundamental, question is whether wavelets are an effective compression technique for different types of time-series data. This question is essential, since there are many real signals exhibiting spectral exponent in the range $\beta \in [1 - \epsilon, 1 + \epsilon]$, also known as *pink noise*. At the same time, many natural phenomenon, for example the level in water resources (like rivers and lakes), are even more skewed. This data has a spectral exponent $\beta > 2$, hence, these signals are classified as black noise.

To explore these questions and to test the limits of application of wavelets to time-series data, we considered a number of different data sets with different noise characteristics.

- **Lakes** - The historical water levels of the following lakes: Erie, Michigan-Huron, Ontario, Superior and Lake St. Clair. Records span the period 1918 - present. The spectral exponent is $\beta = 2.68$.

- **Stocks** - The stock dataset used in the previous two studies. The energy spectrum is $\beta = 1.99$.

- **Pink** - Synthetically generated dataset. We observe an average spectrum of $\beta = 1.01$ for the output of our data generator.

The Lakes dataset was the smallest with only 3,936 sequences. Therefore, we adjusted the size of the other two datasets to be the same. Throughout this experiments the query selectivity was fixed at 0.01, and we used the Daubechies 12 wavelet. Data dimensionality is 128, while we vary the index dimensionality between 16, 32 and 64. All the results are averaged over 100 queries.

In this study, we tried to understand which classes of data can be indexed using an F-index technique. The results are plotted in Figure 10. The vertical axis is the part of the database that is scanned throughout the query answering. Thus, linear scan corresponds to a horizontal line at 1.0. Considering the results, we observe that for pink noise data we need to scan more of the dataset even using 64 dimensions than for brown and black noise data using 16 dimensions. Dimensionality

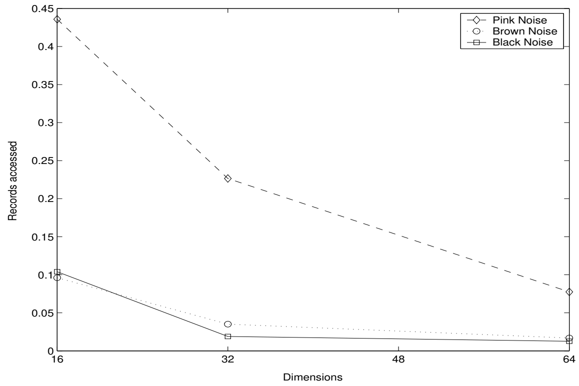

Figure 10: F-index performance for datasets with different characteristics

of 64 is beyond the practical range of almost all multidimensional indexing methods. Furthermore, a common rule of thumb in indexing is that if more than 20% of the data needs to be retrieved using the index, then a linear scan is better [WSB98]. Hence, for pink noise data, an index that performed efficiently for 32 dimensions (at least) would be required. Therefore, we conclude that it is unlikely that "pure" pink noise data has strong enough correlations to enable efficient indexing using widely available indices such as the R*-tree. However, our study of the SR-tree showed that while it does not perform optimally at 32 dimensions, it does exhibit reasonable performance. These studies suggest that a more detailed analysis of the use of sophisticated multi-dimensional indices, like the SR-tree, for indexing pink or almost pink time-series data should be done to see if wavelet transformations of such data is in fact "indexable". Such a study is the subject of our future work.

Our experiment (and conclusion) relied on "pure" synthetically generated pink noise. Hence, we decided to perform an additional experiment to try to determine whether any data less correlated than brown noise could be indexed effectively with this technique. We used the **Rivers** dataset, which we obtained from the Hydro-Climatic Data Network (HCDN).[6] It consists of stream flow records for 1,659 sites throughout the United States and its Territories. Records cumulatively span the period 1874 through 1988. For this dataset we observe, a spectral component of $\beta = 1.4$. Notice that the Lake dataset is excluded in Figure 11, in order to compare datasets of reasonable size. Hence, we are able to use our default settings throughout this experiment (rather than a small sample of the data as was used for Figure 10). The results for the River data set is much closer to Brown noise, than it is to our pure Pink data set. These results motivate using an F-Index (even with a standard R*-tree) as long as the spectral characteristic of the data surpasses 1.5. We expect many data sets to fit into this category.

Previous work on similarity search using F-Indices has speculated that standard transformations (like the DFT) concentrate the energy of the data into few enough dimensions that standard indices (like the R*-tree) can be used [AFS93]. However, these studies focus exclusively on Brown noise

[6] ftp://ftprvares.er.usgs.gov/hcdn92/

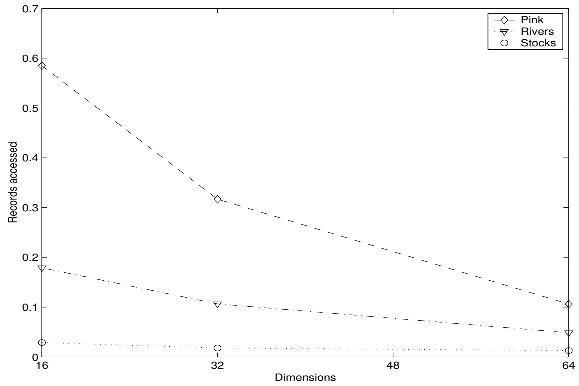

Figure 11: F-index performance for datasets with different characteristics

data. Our results go beyond this to show that less correlated "tan" data can be indexed effectively particularly if we exploit more sophisticated index structures that perform well at higher dimensions.

5 Conclusions

Inspired by the success of wavelets in data compression, we have proposed using full-featured wavelet transforms for similarity search over time-series data. We have first considered the problem of whether general wavelets can in fact be used for this application. We showed that a large class of wavelets, in fact all wavelets with a stable reconstruction (the class of bi-orthonormal wavelets), can in fact be used for similarity search. We then presented a study of how different wavelets perform in this application. We experimentally determined that wavelets with a relatively long (12-16) filter length have the highest precision for this application. We found this to be true for a number of real and synthetic time-series data sets suggesting that these filter lengths best model the important trends in this type of data. In particular, we presented wavelets that outperform the DFT and Haar wavelet for this application.

In addition to our contributions to understanding and optimizing similarity search over time-series data, our work considers the more general question of whether wavelets can be used as part of an effective management or analysis technique for different classes of time-series data. We have shown that some of the more robust wavelets from image compression are indeed effective for not only Brown noise data (a well-studied class that includes most stock datasets), but also for significantly less correlated data. Our results show that data with a spectral component of 1.5 can be efficiently search using off-the-shelf multidimensional indices. They also suggest that even less correlated data may be efficiently search with some of the newly emerging indices designed for much higher dimensions.

We are currently extending our studies using wavelets with some of these more sophisticated, higher dimensional index structures including the M-tree [CPZ97] and the SR-tree [KS97]. We plan to expand our experiments to other indexing structures given the availability of robust implementations of them. Intrigued by the success of wavelets with long (12-16) filter lengths, we also plan to expand our wavelet library to include additional families with similarly long filters. Our long term plans are to investigate the use of similarity search in time-series prediction.

Acknowledgments

We would like to thank Ken Sevcik for his helpful comments and suggestions on earlier drafts of this paper.

References

[ABMD92] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using the wavelet transform. In *IEEE Trans. On Image Processing*, volume 1, pages 205–220, April 1992.

[AFS93] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. Efficient similarity search in sequence databases. In *Foundations of Data Organization and Algorithms, 4th International Conference, FODO'93, Proceedings. Lecture Notes in Computer Science*, volume 730. Springer, 1993.

[BKK96] Stefan Berchtold, Daniel A. Keim, and Hans-Peter Kriegel. The X-tree: An index structure for high-dimensional data. In *VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases*, pages 28–39, Mumbai (Bombay), India, September 1996. Morgan Kaufmann.

[BKSS90] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The R*-tree: An efficient and robust access method for points and rectangles. In *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data*, pages 322–331, Atlantic City, NJ, USA, May 1990. ACM Press.

[Bri96] C. Brislawn. Classification of nonexpansive symmetric extension transforms for multirate filter banks. In *Appl. Comput. Harmon. Anal.*, pages 337–357, 1996.

[CF99] Kin-Pong Chan and Ada Wai-Chee Fu. Efficient time series matching by wavelets. In *Proceedings of the 15th International Conference on Data Engineering*, pages 126–133, Sydney, Austrialia, March 1999. IEEE Computer Society.

[CGRS00] Kaushik Chakrabarti, Minos N. Garofalakis, Rajeev Rastogi, and Kyuseok Shim. Approximate query processing using wavelets. In *The VLDB Journal*, pages 111–122, 2000.

[Chu92] C.K. Chui. *An Introduction to Wavelets.* NewYork: Academic Press, 1992.

[CM99] Kaushik Chakrabarti and Sharad Mehrotra. The Hybrid Tree: An index structure for high dimensional feature spaces. In *Proceedings of the 15th International Conference on Data Engineering*, pages 440–447, Sydney, Austrialia, March 1999. IEEE Computer Society.

[Col00] M.A. Collosi, N.G. Nascimento. Benchmarking access structures for high-dimensional multimedia data. In *Proc. of the IEEE Intl. Conf. on Multimedia and Expo (ICME'2000)*, pages 1215–1218, New York, USA, July 2000.

[CPZ97] Paolo Ciaccia, Marco Patella, and Pavel Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases*, pages 426–435, Athens, Greece, August 1997. Morgan Kaufmann.

[Dau88] I. Daubechies. Orthonormal bases of compactly supported wavelets. In *Communications on Pure and Applied Mathematics*, volume 61, pages 909–996, 1988.

[Dau92] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.

[Dau93] I. Daubechies. Orthonormal bases of compactly supported wavelets II. Variations on a theme. In *SIAM Journal on Mathematical Analysis*, volume 24, pages 499–519, 1993.

[FJ97] M. Frigo and S. Johnson. The fastest Fourier transform in the West. Technical report, Massachusetts Institute of Technology, 1997. http://www.fftw.org/.

[Fou95] Alain Fournier. Wavelets and their applications in computer graphics. http://www.cs.ubc.ca/nest/imager/contributions/bobl/wvlt/top.html, 1995.

[FRM94] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. Fast subsequence matching in time-series databases. In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, pages 419–429, Minneapolis, Minnesota, USA, May 1994. ACM Press.

[Fuk90] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[JPE00] JPEG 2000 part I final committee draft version 1.0. 2000.

[JWW97] O. Firschein J. Wang, G. Wiederhold and S. Wei. Wavelet-based image indexing techniques with partial sketch retrieval capability. In *International Conference on the Advances in Digital Libraries, Library of Congress*, Washington, D. C., USA, May 1997.

[JWW98] O. Firschein J. Wang, G. Wiederhold and S. Wei. Content-based image indexing and searching using Daubechies' wavelets. In *International Journal of Digital Libraries(IJODL)*, volume 1(4), pages 311–328. Springer-Verlag, 1998.

[KAS98] Kothuri Venkata Ravi Kanth, Divyakant Agrawal, and Ambuj K. Singh. Dimensionality reduction for similarity searching in dynamic databases. In *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data*, pages 166–176, Seattle, Washington, USA, June 1998. ACM Press.

[KCPM00] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzanin, and Sharad Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. Technical report, Department of Information and Computer Science University of California, Irvine, 2000. http://www-db.ics.uci.edu/pages/publications/2000/tr-db-00-08.html.

[Kei01] Daniel A. Keim. Wavelets and their applications in databases. http://www.informatik.uni-konstanz.de/ keim/TutorialNotes/, 2001.

[KP00a] E. J. Keogh and M. J Pazzani. Scaling up dynamic time warping for datamining applications. In *Proc. of the Int'l Conf. on Knowledge Discovery & Data Mining*, pages 285–289, 2000.

[KP00b] E. J. Keogh and M. J Pazzani. A simple dimensionality reduction technique for fast similarity search in large time series databases. In *Pacific-Asia Conf. Knowledge Discovery and Data Mining, Current Issues and New Applications, PAKDD*, pages 122–133, 2000.

[KS97] Norio Katayama and Shin'ichi Satoh. The SR-tree: An index structure for high-dimensional nearest neighbor queries. In *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data*, pages 369–380, Tucson, Arizona, USA, May 1997. ACM Press.

[LYC96] Chung-Sheng Li, Philip S. Yu, and Vittorio Castelli. Hierarchyscan: A hierarchical similarity search algorithm for databases of long sequences. In *Proceedings of the Twelfth International Conference on Data Engineering*, pages 546–553, New Orleans, Louisiana, USA, February - March 1996. IEEE Computer Society.

[Mey98] Y. Meyer. Principe d'incertitude, bases hilbertiennes et algébres d'opérateurs. In *Séminaire Bourbaki, 1985/1986, No.662*, 1998.

[MVW00] Yossi Matias, Jeffrey Scott Vitter, and Min Wang. Dynamic maintenance of wavelet-based histograms. pages 101–110, 2000.

[RM97] Davood Rafiei and Alberto Mendelzon. Similarity-based queries for time series data. In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, Tucson, Arizona, USA, May 1997.

[RM98] Davood Rafiei and Alberto Mendelzon. Efficient retrieval of similar time sequences using DFT. In *Proceedings of the International Conference on Foundations of Data Organizations and Algorithms - FODO 98*, Kobe, Japan, November 1998.

[Sch91] Manfred Schroeder. *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise.* W.H. and Company, 1991.

[SCZ00] Gholamhosein Sheikholeslami, Surojit Chatterjee, and Aidong Zhang. Wavecluster: A wavelet based clustering approach for spatial data in very large databases. *VLDB Journal*, 8(3-4):289–304, 2000.

[SS99a] Z. R. Struzik and A. P. J. M. Siebes. The Haar wavelet transform in the time series similarity paradigm. In *Principles of Data Mining and Knowledge Discovery*, pages 12–22. Springer-Verlag, Berlin, Germany, 1999.

[SS99b] Z. R. Struzik and A. P. J. M. Siebes. Measuring Time Series' Similarity through Large Singular Features Revealed with Wavelet Transformation. In *Proceedings of the International Workshop on Database and Expert Systems Application*, pages 162–166, Florence, Italy, September 1999. IEEE Computer Society Press.

[Str81] J.O. Strömberg. A modified Franklin system and higher order spline systems on R^2 as unconditional bases for Hardy spaces. In *Proc. Conf. in Honor of Antoni Zygmund, New York: Wadsworth Mathematics Series.*, volume II, pages 475–493, 1981.

[TlG88] A. Tabatabai and D. le Gall. Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques. In *International Conference on Acoustics, Speech and Signal Processing*, pages 761–765, New York, USA, 1988.

[VK95] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding.* Prentice Hall, 1995.

[VW99] Jeffrey Scott Vitter and Min Wang. Approximate computation of multidimensional aggregates of sparse data using wavelets. In *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data*, pages 193–204, Philadephia, Pennsylvania, USA, June 1999. ACM Press.

[WAA+96] Daniel Wu, Divyakant Agrawal, Amr El Abbadi, Ambuj K. Singh, and Terence R. Smith. Efficient retrieval for browsing large image databases. In *CIKM '96, Proceedings of the Fifth International Conference on Information and Knowledge Management*, pages 11–18, Rockville, Maryland, USA, November 1996. ACM.

[WAA00] Yi-Leh Wu, Divyakant Agrawal, and Amr El Abbadi. A comparison of DFT and DWT based similarity search in time-series databases. Technical report, Department of Computer Science University of California, Santa Barbara, 2000.

[Wan98] J. Wang. Wavelet-based progressive transmission and security filtering for medical image distribution. In *Advances in Biomedical Image Databases, S. Wong (Ed.), Kluwer, to appear*, 1998.

[WJ96] David A. White and Ramesh Jain. Similarity indexing with the SS-tree. In *Proceedings of the Twelfth International Conference on Data Engineering*, pages 516–523, New Orleans, Louisiana, USA, February - March 1996. IEEE Computer Society.

[WSB98] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB'98, Proceedings of 24rd International Conference on Very Large Data Bases*, pages 194–205, New York City, New York, USA, August 1998. Morgan Kaufmann.

[YF00] Byoung-Kee Yi and Christos Faloutsos. Fast time sequence indexing for arbitrary L_p norms. In *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases*, pages 385–394, Cairo, Egypt, September 2000. Morgan Kaufmann.

[YJF98] Byoung-Kee Yi, H. V. Jagadish, and Christos Faloutsos. Efficient retrieval of similar time sequences under time warping. In *Proceedings of the Fourteenth International Conference on Data Engineering*, pages 201–208, Orlando, Florida, USA, February 1998. IEEE Computer Society.