Optimal Workload-based Weighted Wavelet Synopses *

Yossi Matias

School of Computer Science
Tel Aviv University
Tel Aviv 69978, Israel
matias@tau.ac.il

Daniel Urieli†

School of Computer Science
Tel Aviv University
Tel Aviv 69978, Israel
daniel1@post.tau.ac.il

Abstract

In recent years wavelets were shown to be effective data synopses. We are concerned with the problem of finding efficiently wavelet synopses for massive data sets, in situations where information about query workload is available. We present linear time, I/O optimal algorithms for building optimal workload-based wavelet synopses for point queries. The synopses are based on a novel construction of weighted inner-products and use weighted wavelets that are adapted to those products. The synopses are optimal in the sense that the subset of retained coefficients is the best possible for the bases in use with respect to either the mean-squared absolute or relative errors. For the latter, this is the first optimal wavelet synopsis even for the regular, non-workload-based case. Experimental results demonstrate the advantage obtained by the new optimal wavelet synopses, as well as the robustness of the synopses to deviations in the actual query workload.

1 Introduction

In recent years there has been increasing attention to the development and study of data synopses, as effective means for addressing performance issues in massive data sets. Data synopses are concise representations of data sets, that are meant to effectively support approximate queries to the represented data sets [10]. A primary constraint of a data synopsis is its size. The effectiveness of a data synopsis is measured by the accuracy of the answers it provides, as well as by its response time and its construction time. Several different synopses were introduced and studied, including random samples, sketches, and different types of histograms. Recently, wavelet-based synopses were introduced and shown to be a powerful tool for building effective data synopses for various applications, including selectivity estimation for query optimization in DBMS, approximate query processing in OLAP applications and more (see [16, 20, 21, 2, 6, 9, 8], and references therein).

The general idea of wavelet-based approximations is to transform a given data vector of size N into a representation with respect to a wavelet basis (this is called a *wavelet transform*), and approximate it using only $M \ll N$ wavelet basis vectors, by retaining only M coefficients from the linear combination that spans the data vector (*coefficients thresholding*). The linear combination

*Research partly supported by a grant from the Israel Science Foundation.

†Contact author

that uses only M coefficients (and assumes that all other coefficients are zero) defines a new vector that approximates the original vector, using less space. This is called *M-term approximation*, which defines a *wavelet synopsis* of size M.

Wavelet synopses. Wavelets were traditionally used to compress some data set where the purpose is to reconstruct, in a later time, an approximation of the *whole* data using the set of retained coefficients. The situation is a little different when using wavelets for building synopses in database systems [16]: in this case only *portions* of the data are reconstructed each time, in response to user queries, rather than the whole data at once. As a result, portions of the data that are used for answering frequent queries are reconstructed more frequently than portions of the data that correspond to rare queries. Therefore, the approximation error is measured over the *multi-set of actual queries*, rather than over the data itself.

Another aspect of the use of wavelets in database systems is that due to the large data-sizes in databases (giga-, tera- and peta-bytes), the efficiency of building wavelet synopses is of primary importance. Disk I/Os should be minimized as much as possible, and non-linear-time algorithms may be unacceptable.

Optimal wavelet synopses. The main advantage of transforming the data into a representation with respect to a wavelet basis is that for data vectors containing similar values, many wavelet coefficients tend to have very small values. Thus, eliminating such small coefficients introduces only small errors when reconstructing the original data, resulting in a very effective form of lossy data compression.

Generally speaking, we can characterize a wavelet approximation by three attributes: how the approximation error is measured, what wavelet basis is used and how coefficient thresholding is done. Many bases were suggested and used in traditional wavelets literature. Given a basis with respect to which the transform is done, the selection of coefficients that are retained in the wavelet synopsis may have significant impact on the approximation error. The goal is therefore to select a subset of M coefficients that minimizes some approximation-error measure. This subset is called an *optimal wavelet synopsis*, with respect to the chosen error measure.

While there has been a considerable work on wavelet synopses and their applications [16, 20, 21, 2, 6, 9, 8], so far there were only a few optimality results. The first one is a linear-time Parseval-based algorithm, which was used in traditional wavelets literature (e.g [12]), where the error was measured over the *data*. This algorithm minimizes the L_2 norm of the error vector, and equivalently it minimizes the mean-squared-absolute error over all possible point queries. No algorithm that minimizes the mean-squared-relative error over all possible point queries was known. The second one, introduced recently [9], is a polynomial-time ($O(N^2 M \log M)$) algorithm that minimizes the max relative or absolute error over all possible point queries. Another optimality result is a polynomial time dynamic-programming algorithm that obtains an optimal wavelet synopsis over multiple measures [6]. The synopsis is optimal w.r.t. an error metric defined as weighted combination of L_2 norms over the multiple measures (this weighted combination has no relation with the notion of weighted wavelets of this paper).

Workload-based wavelet synopses. In recent years there is increased interest in workload-based synopses – synopses that are adapted to a given query workload, with the assumption that the workload represents (approximately) a probability distribution from which future queries will be taken. Chaudhuri et al [4] argue that identifying an appropriate precomputed sample that avoids large errors on an *arbitrary* query is virtually impossible. To minimize the effects of this problem,

previous studies have proposed using the *workload* to guide the process of selecting samples [1, 3, 7]. By picking a sample that is tuned to the given workload, we can reduce the error over frequent (or otherwise "important") queries in the workload.

In [4], the authors formulate the problem of pre-computing a sample as an *optimization* problem, whose goal is to pick a sample that minimizes the error for the given workload.

Recently, *workload-based wavelet synopses* were proposed [14, 18]. Using an adaptive-greedy algorithm, the query-workload information was used during the thresholding process in order to build a wavelet synopsis that decreases the error w.r.t. to the query workload. While these workload-based wavelet synopses demonstrate significant imporvement with respect to prior synopses, they are not optimal.

In this paper, we address the problem of finding efficiently *optimal* workload-based wavelet synopses.

1.1 Contributions

We introduce efficient algorithms for finding optimal workload-based wavelet synopses using *weighted Haar (WH)* wavelets, for workloads of point queries. Our main contributions are:

- Linear-time, I/O optimal algorithms that find optimal Workload-based Weighted Wavelet (WWW) synopses: [1]
  - An optimal synopsis w.r.t. workload-based mean-squared *absolute*-error (*WB-MSE*).
  - An optimal synopsis w.r.t. workload-based mean-squared *relative*-error (*WB-MRE*).

  Equivalently, the algorithms minimize the *expected* squared, absolute or relative errors over a point query taken from a given distribution.
- The *WB-MRE* algorithm, used with uniform workload, is also the first algorithm that minimizes the mean-squared-relative-error over the *data values*, with respect to a wavelet basis.
- Both WWW synopses are also optimal with respect to *enhanced wavelet synopses*, which allow changing the values of the synopses coefficients to arbitrary values.
- Experimental results show the advantage of our synopses with respect to existing synopses.
- The synopses are robust to deviation from the pre-defined workload, as demonstrated by our experiments.

The above results were obtained using the following novel techniques.

- We define the problem of finding optimal workload-based wavelet synopses in terms of a *weighted norm*, a *weighted-inner-product* and a *weighted-inner-product-space*. This enables linear time I/O optimal algorithms for building optimal workload-based wavelet synopses.

  The approach of using a weighted inner product can also be used to the general case in which each data point is given different priority, representing its significance (an example is shown in Sec. 6). Using these weights, one can find a weighted-wavelet basis, and an optimal weighted wavelet synopsis in linear time, with $O(N/B)$ I/Os.

[1] No relation whatsover to the world-wide-web.

- We introduce the use of *weighted wavelets* for data synopses. Using weighted wavelets [5, 11] enables finding optimal workload-based wavelet synopses efficiently. In contrast, it is not known how to obtain optimal workload-based wavelet synopses with respect to the Haar basis efficiently. If we ignore the efficiency of finding a synopsis, the Haar basis is as good as the weighted Haar basis for approximation.

In wavelets literature (e.g [12]), wavelets are used to approximate a given signal, which is treated as a vector in an inner-product space. Since an inner-product defines an L_2 norm, the approximation error is measured as the L_2 norm of the error vector, which is the difference between the approximated vector and the approximating vector. Many wavelet bases were used for approximation, as different bases are adequate for approximating different collections of data vectors. By using an orthonormal wavelet basis, an optimal coefficient thresholding can be achieved in linear time, based on Parseval's formula. When using non-orthogonal wavelet basis, or measuring the error using other norms (e.g L_∞), it is not known whether an optimal coefficient thresholding can be found efficiently, so usually non-optimal greedy algorithms are used in practice.

A WH basis is a generalization of the standard Haar basis, which is typically used for wavelet synopses due to its simplicity.

There are several attributes by which a wavelet basis is characterized, which affects the quality of the approximations achieved using this basis (for full discussion, see [12]). These attribute are: the set of nested spaces of increasing resolution which the basis spans, the number of vanishing moments of the basis, and its compact support (if exists). Both Haar basis and a WH basis span the same subsets of nested spaces, have one vanishing moment, and a compact support of size 1.

Haar basis is orthonormal for uniform workload of point queries. Hence it is optimal for the MSE error measure. The WH basis is orthonormal with respect to the *weighted* inner-product defined by the problem of finding optimal workload-based wavelet synopses. As a result, an optimal workload-based synopses with respect to WH basis is achieved efficiently, based on Parseval's formula, while for the Haar basis no efficient optimal thresholding algorithm is known, in cases other than uniform workload.

1.2 Paper outline

The rest of the paper is structured as follows. In Sec. 2 we describe the basics of wavelet-based synopses. In Sec. 3 we describe the basic ideas we rely on in our development, including the workload-based error metrics and optimal thresholding in orthonormal bases. In Sec. 4 we define the problem of finding optimal workload-based wavelet synopses in terms of weighted inner product, and solve it using an orthonormal basis. In Sec. 5 we describe the optimal algorithm for minimizing *WB-MSE*, which is based on the construction of Sec. 4. In Sec. 6 we extend the algorithm to work for the *WB-MRE*. In Sec. 7 we present experimental results, and in Sec. 8 we draw our conclusions.

2 Wavelets basics

In this section we will start by presenting the Haar wavelets and continue with presenting wavelet based synopses, obtained by thresholding process, described in Sec. 2.2. The error tree structure will be presented next (Sec. 2.3), along with the description of the reconstruction of original data from the wavelet synopses in Sec. 2.4.

Wavelets are a mathematical tool for the hierarchical decomposition of functions in a space-efficient manner. Wavelets represent a function in terms of a coarse overall shape, plus details that

range from coarse to fine. Regardless of whether the function of interest is an image, a curve, or a surface, wavelets offer an elegant technique for representing the various levels of detail of the function in a space-efficient manner.

2.1 One-dimensional Haar wavelets

Haar wavelets are conceptually the simplest wavelet basis functions, and were thus used in previous works of wavelet synopses. They are fastest to compute and easiest to implement. We focus on them for purpose of exposition in this paper. To illustrate how Haar wavelets work, we will start with a simple example borrowed from [16].

Suppose we have one-dimensional "signal" of $N = 8$ data items: $S = [2, 2, 0, 2, 3, 5, 4, 4]$. We will show how the Haar wavelet transform is done over S. We first average the signal values, pairwise, to get a new lower-resolution signal with values $[2, 1, 4, 4]$. That is, the first two values in the original signal (2 and 2) average to 2, and the second two values 0 and 2 average to 1, and so on. We also store the pairwise differences of the original values (divided by 2) as detail coefficients. In the above example, the four detail coefficients are $(2-2)/2 = 0$, $(0-2)/2 = -1$, $(3-5)/2 = -1$, and $(4-4)/2 = 0$. It is easy to see that the original values can be recovered from the averages and differences.

This was one phase of the Haar wavelet transform. By repeating this process recursively on the averages, we get the Haar wavelet transform (Table 1). We define the *wavelet transform* (also called *wavelet decomposition*) of the original eighth-value signal to be the single coefficient representing the overall average of the original signal, followed by the detail coefficients in the order of increasing resolution. Thus, for the one-dimensional Haar basis, the wavelet transform of our signal is given by
$\tilde{S} = [2\frac{3}{4}, -1\frac{1}{4}, \frac{1}{2}, 0, 0, -1, -1, 0]$

Resolution	Averages	Detail Coefficients
8	[2, 2, 0, 2, 3, 5, 4, 4]	
4	[2, 1, 4, 4]	[0,-1,-1, 0]
2	[1.5, 4]	[0.5, 0]
1	[2.75]	-1.25

Table 1: Haar Wavelet Decomposition

The individual entries are called the wavelet coefficients. The wavelet decomposition is very efficient computationally, requiring only *O(N)* CPU time and *O(N/B)* I/Os to compute for a signal of N values, where B is the disk-block size.
No information has been gained or lost by this process. The original signal has eight values, and so does the transform. Given the transform, we can reconstruct the exact signal by recursively adding and subtracting the detail coefficients from the next-lower resolution. In fact we have transformed the signal S into a representation with respect to another basis of R^8: The Haar wavelet basis. A detailed discussion can be found, for example, in [19].

2.2 Thresholding

Given a limited amount of storage for maintaining a wavelet synopsis of a data array A (or equivalently a vector S), we can only retain a certain number $M \ll N$ of the coefficients stored in

the wavelet decomposition of A. The remaining coefficients are implicitly set to 0. The goal of coefficient thresholding is to determine the "best" subset of M coefficients to retain, so that some overall error measure in the approximation is minimized.

One advantage of the wavelet transform is that in many cases a large number of the detail coefficients turn out to be very small in magnitude. Truncating these small coefficients from the representation (i.e., replacing each one by 0) introduces only small errors in the reconstructed signal. We can approximate the original signal effectively by keeping only the most significant coefficients.

For a given input sequence $d_0, ..., d_{N-1}$, we can measure the error of approximation in several ways. Let the i'th data value be d_i. Let q_i be the i'th point query, which it's value is d_i. Let $\hat{d}_i$ be the estimated result of d_i. We use the following error measure for the absolute error over the i'th data value:

$$e_i = e(q_i) = |d_i - \hat{d}_i|$$

Once we have the error measure for representing the errors of individual data values, we would like to measure the norm of the vector of errors $e = (e_0, ..., e_{N-1})$. The standard way is to use the L_2 norm of e divided by $\sqrt{N}$ which is called the *mean squared error*:

$$MSE(e) = \|e\| = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1} e_i^2}$$

We would use the terms *MSE* and L_2 norm interchangeably during our development since they are completely equivalent, to a positive multiplicative constant.

The basic thresholding algorithm, based on Parseval's formula, is as follows: let $\alpha_0, ..., \alpha_{N-1}$ be the wavelet coefficients, and for each α_i let $level(\alpha_i)$ be the resolution level of α_i. The detail coefficients are normalized by dividing each coefficient by $\sqrt{2^{level(a_i)}}$ reflecting the fact that coefficients at the lower resolutions are "less important" than the coefficients at the higher resolutions. This process actually turns the wavelet coefficients into an orthonormal basis coefficients (and is thus called "normalization"). The M largest normalized coefficients are retained. The remaining $N - M$ coefficients are implicitly replaced by zero. This deterministic process *provably* minimizes the L_2 norm of the vector of errors defined above, based on Parseval's formula (see Sec. 3).

2.3 Error tree

The wavelet decomposition procedure followed by any thresholding can be represented by an *error tree* [16].

Fig. 1 presents the error tree for the above example. Each internal node of the error tree is associated with a wavelet coefficient, and each leaf is associated with an original signal value. Internal nodes and leaves are labelled separately by $0, 1, ..., N-1$. For example, the root is an internal node with label 0 and its node value is 2.75 in Fig. 1. For convenience, we shall use "node" and "node value" interchangeably. The construction of the error tree exactly mirrors the wavelet transform procedure. It is a bottom-up process. First, leaves are assigned original signal values from left to right. Then wavelet coefficients are computed, level by level, and assigned to internal nodes.

2.4 Reconstruction of original data

Given an error tree T and an internal node t of T, $t \neq a_0$, we let $leftleaves(t)$ ($rightleaves(t)$) denote the set of leaves (i.e., data) nodes in the subtree rooted at t's left (resp., right) child. Also, given any (internal or leaf) node u, we let $path(u)$ be the set of all (internal) nodes in T that are

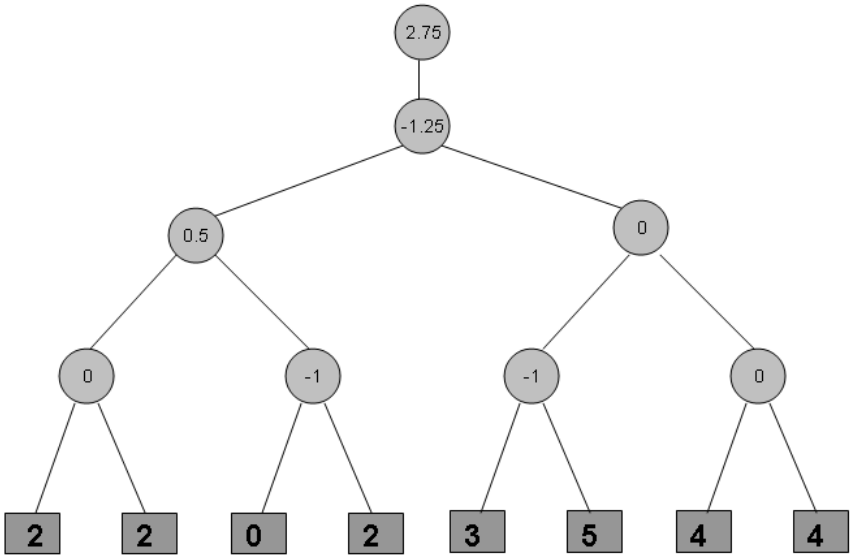

Figure 1: Error tree for $N = 8$

proper ancestors of u (i.e., the nodes on the path from u to the root of T, including the root but not u) with nonzero coefficients.
Finally, for any two leaf nodes d_l and d_k we denote $d(l:h)$ as the range sum $\sum_{i=l}^{k} d_i$
Using the error tree representation T, we can outline the following reconstruction properties of the Haar wavelet decomposition [16]:

2.4.1 Single value.

The reconstruction of any data value d_i depends only on the values of the nodes in $path(d_i)$.

$$d_i = \sum_{\alpha_j \in path(d_i)} \delta_{ij} \cdot \alpha_j$$

where $\delta_{ij} = +1$ if $d_i \in leftleaves(\alpha_j)$ or $j = 0$, and $\delta_{ij} = -1$ otherwise.

2.4.2 Range sum.

An internal node α_j contributes to the range sum $d(l:h)$ only if $\alpha_j \in path(d_l) \cup path(d_k)$.

$$d(l:h) = \sum_{\alpha_j \in path(d_l) \cup path(d_h)} x_j$$

where

$$x_j = \begin{cases} (h-l) \cdot \alpha_j & \text{if } j = 0 \\ (|leftleaves(\alpha_j, l:h)| - |rightleaves(\alpha_j, l:h)|) \cdot \alpha_j & \text{otherwise} \end{cases}$$

and where $leftleaves(\alpha_j, l:h) = leftleaves(\alpha_j) \cap \{d_l, d_{l+1}, ..., d_h\}$ (i.e., the intersection of $leftleaves(\alpha_j)$ with the summation range) and $rightleaves(\alpha_j, l:h)$ is defined similarly.
Thus, a reconstruction of a single data values involves the summation of at most $\log N + 1$ coefficients, and reconstructing a range sum involves the summation of at most $2 \log N + 1$ coefficients, regardless of the width of the range.

3 The basics of our development

3.1 Workload-based error metrics

Let $D = (d_0, ..., d_{N-1})$ be a sequence with $N = 2^j$ values. Denote the set of point queries as $Q = (q_0, ..., q_{N-1})$, where q_i is a query which its answer is d_i. Let a workload $W = (c_0, ..., c_{N-1})$ be a vector of weights that represents the probability distribution from which future point queries are to be generated. Let $(u_0, ..., u_{N-1})$ be a basis of R^N, than $D = \sum_{i=0}^{N} \alpha_i u_i$. We can represent D by a vector of coefficients $(\alpha_0, ..., \alpha_{N-1})$.

Suppose we want to approximate D using a subset of the coefficients $S \subset \{\alpha_0, ..., \alpha_{N-1}\}$ where $|S| = M$. Then, for any subset S we can define a weighted norm WL_2 with respect to S, that provides a measure for the errors expected for queries drawn from the probability distribution represented by W, when using S as a synopsis. S is then referred to as a *workload-based wavelet synopsis*.

Denote $\hat{d}_i$ as an approximation of d_i using S. There are two standard ways to measure the error over the i'th data value (equivalently, *point query*):
The absolute error: $e_a(i) = e_a(q_i) = |d_i - \hat{d}_i|$; and *the relative error*: $e_r(i) = e_r(q_i) = \frac{|d_i - \hat{d}_i|}{max\{|d_i|, s\}}$, where s is a positive bound that prevents small values from dominating the relative error.

While the general (non-workload-based) approach is to reduce the L_2 norm of the vector of errors $(e_1, ..., e_N)$ (where $e_i = e_a(i)$ or $e_i = e_r(i)$), here we would generalize the L_2 norm to reflect the query workload. Given a workload W that consists of all the queries' probabilities $c_1, ..., c_N$ (where c_i is the probability that q_i appears), the *weighted-L_2 norm* of the vector of (absolute or relative) errors $e = (e_1, ..., e_N)$ would be:

$$WL_2(e) = \|e\|_w = \sqrt{\sum_{i=0}^{N-1} c_i \cdot e_i^2}$$

where $0 < c_i \leq 1$, $\sum_{i=0}^{N-1} c_i = 1$. The intuition behind this definition of norm is to give each data value d_i (or equivalently each point query q_i) some weight that represents its significance. In the above case the square of the WL_2 norm is the expected squared error for a point query that is drawn from the given distribution. In other words, to minimize that norm of the error is to *minimize the expected squared error of an answer to a query*.

In general, the weights given to data values need not necessarily represent a probability distribution of point queries, but any other significance measure. For example, in Sec. 6 we use weights to solve the problem of minimizing the mean-squared relative error measured over the *data values* (the non-workload-based case).

Notice that it is a generalization of the MSE norm: by taking equal weights for each query, meaning $c_i = \frac{1}{N}$ for each i and $e_i = e_a(i)$, we get the standard MSE norm. We use the term *workload-based error* for the WL_2 norm of the vector of errors e. When e_i are absolute (resp. relative) errors the workload-based error would be called the *WB-MSE* (resp. *WB-MRE*).

3.2 Optimal thresholding in orthonormal bases

The construction is based on Parseval's formula, and a known theorem that results from it (Thm. 1).

3.2.1 Parseval's formula.

Let V be a vector space, where $v \in V$ is a vector and $\{u_0, ..., u_{N-1}\}$ is an orthonormal basis of V. We can express v as $v = \sum_{i=0}^{N-1} \alpha_i u_i$. Then

$$\|v\|^2 = \sum_{i=0}^{N-1} \alpha_i^2 \tag{1}$$

An M-term approximation is achieved by representing v using a subset of coefficients $S \subset \{\alpha_0, ..., \alpha_{N-1}\}$ where $|S| = M$. The error vector is than $e = \sum_{i \notin S} \alpha_i u_i$. By Parseval's formula, $\|e\|^2 = \sum_{i \notin S} \alpha_i^2$. This proves the following theorem.

Theorem 1 (Parseval-based optimal thresholding) *Let V be a vector space, where $v \in V$ is a vector and $\{u_0, ..., u_{N-1}\}$ is an orthonormal basis of V. We can represent v by $\{\alpha_0, ..., \alpha_{N-1}\}$ where $v = \sum_{i=0}^{N-1} \alpha_i u_i$. Suppose we want to approximate v using a subset $S \subset \{\alpha_0, ..., \alpha_{N-1}\}$ where $|S| = M \ll N$. Picking the M largest coefficients to S minimizes the L_2 norm of the error vector, over all possible subsets of M coefficients.*

Given an inner-product, based on this theorem one can easily find an optimal synopses by choosing the largest M coefficients.

3.3 Optimality over enhanced wavelet synopses

Notice that in the previous section we limited ourselves to picking subsets of coefficients with original values from the linear combination that spans v (as is usually done). In case $\{u_0, ..., u_{N-1}\}$ is a wavelet basis, these are the coefficients that results from the wavelet transform. We next show that the optimal thresholding according to Thm. 1 is optimal even according to an enhanced definition of M-term approximation. We define *enhanced wavelet synopses* as wavelet synopses that allow *arbitrary values* to the retained wavelet coefficients, rather than the original values that resulted from the transform. The set of possible standard synopses is a subset of the set of possible *enhanced* synopses, and therefore an optimal synopsis according to the standard definition is not necessarily optimal according to the enhanced definition.

Theorem 2 *When using an orthonormal basis, choosing the largest M coefficients with original values is an optimal enhanced wavelet synopses.*

Proof: The proof is based on the fact that the basis is orthonormal. It is enough to show that given some synopsis of M coefficients with original values, any change to the values of some subset of coefficients in the synopsis would only make the approximation error larger:
Let $u_1, ..., u_N$ be an orthonormal basis and let $v = \alpha_1 u_1 + ... + \alpha_N u_N$ be the vector we would like to approximate by keeping only M wavelet coefficients. Without loss of generality, suppose we choose the first M coefficients and have the following approximation for v: $\tilde{v} = \sum_{i=1}^{M} \alpha_i u_i$. According to Parseval's formula $\|e\|^2 = \sum_{i=M+1}^{N} \alpha_i^2$ since the basis is orthonormal. Now suppose we would change the values of some subset of j retained coefficients to new values. Let us see that due to the orthonormality of the basis it would only make the error larger. Without loss of generality we

would change the first j coefficients, meaning, we would change $\alpha_1, ..., \alpha_j$ to be $\alpha'_1, ..., \alpha'_j$. In this case the approximation would be $\tilde{v}' = \sum_{i=1}^{j} \alpha'_i u_i + \sum_{i=j+1}^{M} \alpha_i u_i$. The approximation error would be $v - \tilde{v}' = \sum_{i=1}^{j} (\alpha_i - \alpha'_i) u_i + \sum_{i=M+1}^{N} \alpha_i u_i$. It is easy to see that the error of approximation would be: $\|e\|^2 = \langle v - \tilde{v}', v - \tilde{v}' \rangle = \sum_{i=1}^{j} (\alpha_i - \alpha'_i)^2 + \sum_{i=M+1}^{N} \alpha_i^2 > \sum_{i=M+1}^{N} \alpha_i^2$. □

4 The workload-based inner product

In this section, we define the problem of finding an optimal workload-based synopses in terms of a weighted-inner-product space, and solve it relying on this construction. Here we deal with the case where e_i are the absolute errors (the algorithm minimizes the *WB-MSE*). An extension to relative errors (*WB-MRE*) is introduced in Sec. 6
Our development is as follows:

1. Transforming the data vector D into an equivalent representation as a function f in a space of piecewise constant functions over $[0, 1)$. (Sec. 4.1)
2. Defining the *workload-based inner product*. (Sec. 4.2)
3. Using the inner product to define an L_2 norm, showing that the newly defined norm is equivalent to the *weighted* L_2 *norm* (WL_2). (Sec. 4.3)
4. Defining a *weighted Haar basis* which is orthonormal with respect to the new inner product. (Sec. 4.4)

Based on Thm. 1 and Thm. 2 one can easily find an optimal workload-based wavelet synopses with respect to a weighted Haar wavelet basis.

4.1 Transforming the data vector into a piecewise constant function

We assume that our approximated data vector D is of size $N = 2^j$. As in [19], we treat sequences (vectors) of 2^j points as piecewise constant functions defined on the half-open interval $[0, 1)$. In order to do so, we will use the concept of a vector space from linear algebra. A sequence of one point is just a function that is constant over the entire interval $[0, 1)$; we'll let V_0 be the space of all these functions. A sequence of 2 points is a function that has two constant parts over the intervals $[0, \frac{1}{2})$ and $[\frac{1}{2}, 1)$. We'll call the space containing all these functions V_1. If we continue in this manner, the space V_j will include all piecewise constant functions on the interval $[0, 1)$, with the interval divided equally into 2^j different sub-intervals. We can now think of every one-dimensional sequence D of 2^j values as being an element, or vector f, in V_j.

4.2 Defining a workload-based inner product

The first step is to choose an inner product defined on the vector space V_j. Since we want to minimize a *workload based error* (and not the regular L_2 error), we started by defining a new *workload based inner product*. The new inner product is a generalization of the standard inner product. It is a sum of $N = 2^j$ weighted standard products; each of them is defined over an interval of size $\frac{1}{N}$:

$$\langle f, g \rangle = N \cdot \left(\sum_{i=0}^{N-1} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} f(x)\, g(x)\, dx \right) \text{ where } 0 < c_i \leq 1, \sum_{i=0}^{N-1} c_i = 1 \tag{2}$$

Lemma 1 *$\langle f, g\rangle$ is an inner product.*

Proof: Let us check that it satisfies the conditions of an inner product:

- $\langle f, g\rangle : V_j \times V_j \to R$
- Symmetric:

$$\langle f, g\rangle = N \cdot \sum_{i=0}^{N-1} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} f(x)g(x)dx = N \cdot \sum_{i=0}^{N-1} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} g(x)f(x)dx = \langle g, f\rangle$$

- Bilinear:

$$\langle af_1 + bf_2, g\rangle = N \cdot \sum_{i=0}^{N-1} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} (af_1 + bf_2)(x)g(x)dx =$$

$$N \cdot \sum_{i=0}^{N-1} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} af_1(x)g(x)dx + N \cdot \sum_{i=0}^{N-1} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} bf_2(x)g(x)dx =$$

$$aN \cdot \sum_{i=0}^{N-1} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} f_1(x)g(x)dx + bN \cdot \sum_{i=0}^{N-1} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} f_2(x)g(x)dx =$$

$$a\langle f_1, g\rangle + b\langle f_2, g\rangle$$

- and also

$$\langle f, ag_1 + bg_2\rangle = a\langle f, g_1\rangle + b\langle f, g_2\rangle$$

  with a similar proof.

- positive definite:

$$\langle f, f\rangle = N \cdot \sum_{i=0}^{N-1} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} f(x)f(x)dx = N \cdot \sum_{i=0}^{N-1} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} f^2(x)dx \geq 0$$

$$\text{and } \langle f, f\rangle = 0 \text{ iff } f \equiv 0 \text{ since } c_i > 0 \text{ for each } i$$

□

As mentioned before, a coefficient c_i represents the probability (or a weight) for the i'th point query (q_i) to appear. Notice that the answer of which is the ith data value, which is function value at the i'th interval. When all coefficients c_i are equal to $\frac{1}{N}$ (a uniform distribution of queries), we get the standard inner product, and therefore this is a generalization of the standard inner product.

4.3 Defining a norm based on the inner product

Based on that inner product we define an inner-product-based (IPB) norm:

$$\|f\|_{IPB} = \sqrt{\langle f, f \rangle} \tag{3}$$

Lemma 2 *The norm $\|f\|_{IPB}$ measured over the vector of absolute errors is the weighted L_2 norm of this vector, i.e $\|e\|^2_{IPB} = \sum_{i=0}^{N-1} c_i e_i^2 = \|e\|^2_w$.*

Proof: Let $f \in V_j$ be a function and let $f' \in V_j$ be a function that approximates f. let the error function be $e = f - f' \in V_j$. Then the norm of the error function is:

$$\|e\|^2_{IPB} = \langle e, e \rangle = N \cdot \sum_{i=0}^{N-1} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} e(x)\, e(x)\, dx =$$

$$N \cdot \sum_{i=0}^{N-1} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} e^2(x)\, dx = N \cdot \sum_{i=0}^{N-1} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} (f - f')^2(x)\, dx =$$

$$N \cdot \sum_{i=0}^{N-1} \frac{c_i}{N} \left(f\left(\frac{i}{N}\right) - f'\left(\frac{i}{N}\right) \right)^2 = N \frac{1}{N} \sum_{i=0}^{N-1} c_i \left(f\left(\frac{i}{N}\right) - f'\left(\frac{i}{N}\right) \right)^2 = \sum_{i=0}^{N-1} c_i e_i^2$$

where e_i is the error on the i'th function value. This is exactly the square of the previously defined *weighted L_2 norm.* □

Notice that when all coefficients are equal to $\frac{1}{N}$ we get the regular L_2 norm, and therefore this is a generalization of the regular L_2 norm (MSE).
Our purpose is to minimize the *workload based error* which is the WL_2 norm of the vector of errors.

4.4 Defining an orthonormal basis

At this stage we would like to use Thm. 1. The next step would thus be finding an orthonormal (with respect to a workload based inner product) wavelet basis for the space V_j . The basis is a *Weighted Haar Basis.* For each workload-based inner product (defined by a given query workload) there is corresponding orthonormal weighted Haar basis, and our algorithm finds this basis in linear time, given the workload of point queries. We describe the bases here, and see how to find a basis based on a given workload of point queries. We will later use this information in the algorithmic part.

In order to build a weighted Haar basis, we take the Haar basis functions and for the k'th basis function we multiply its positive (resp. negative) part by some x_k (resp. x_k). We would like to choose such x_k and y_k so that we get an orthonormal basis with respect to our inner product. Let us illustrate it by drawing. Instead of using Haar basis functions (Fig. 2), we use functions of the kind illustrated in Fig. 3, where x_k and y_k are not necessarily (and probably not) equal, so our basis looks like the one in (Fig. 4). How do we choose x_k and y_k?

Let u_k be some Haar basis function as described above. Let $[a_{k_0}, a_{k_1})$ be the interval over which the basis function is positive and let $[a_{k_1}, a_{k_2})$ be the interval over which the function is negative. Recall that a_{k_0}, a_{k_1} and a_{k_2} are both multiples of $\frac{1}{N}$ and therefore the interval precisely contains some number of continuous intervals of the form $[\frac{i}{N}, \frac{i+1}{N}]$ (also $a_{k_1} = \frac{a_{k_0}+a_{k_2}}{2}$). Moreover, the size of the interval over which the function is positive (resp. negative) is $\frac{1}{2^i}$ for some $i < j$ (As we remember, $N = 2^j$). Recall that for the i'th interval of size $\frac{1}{N}$, meaning $[\frac{i}{N}, \frac{i+1}{N})$ there is a

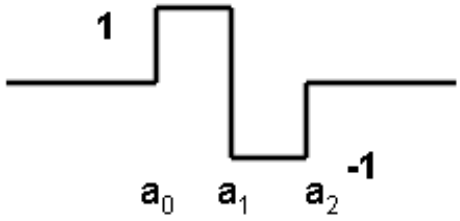

Figure 2: An example for a Haar basis function

corresponding weight coefficient c_i which is the coefficient that is used in the inner product. Notice that each Haar basis function is positive (negative) over some number of (whole) such intervals. We can therefore associate the sum of coefficients of the intervals "under" the positive (negative) part of the function with the positive (negative) part of the function.
Let us denote the sum of weight coefficients (c_i's) corresponding to intervals that are under the positive (resp. negative) as l_k (resp. r_k).

Lemma 3 *Suppose for each Haar basis function v_k we choose x_k and y_k such that*

$$x_k = \sqrt{\frac{r_k}{l_k r_k + l_k^2}} \quad y_k = \sqrt{\frac{l_k}{l_k r_k + r_k^2}}$$

and multiply the positive (resp. negative) part of v_k by x_k (resp. y_k); by doing that we get an orthonormal set of $N = 2^j$ functions, meaning we get an orthonormal basis.

Proof: We first show that when taking x_k and y_k such that $\frac{x_k}{r_k} = \frac{y_k}{l_k}$ the basis is orthogonal. It is enough to show that the inner product of any v_k and a constant function is 0. In order to see why that suffices:
Let u and v be some 2 Haar basis functions and let I_u and I_v be the intervals over which u and v are different from zero, respectively. If there is some point (interval) over which both functions are different from zero, then by the Haar basis definition we get either $I_u \subset I_v$ or $I_v \subset I_u$. Suppose $I_v \subset I_u$ then I_v is contained only in the negative part of I_u or only in the positive part of I_u, again, by the Haar basis definition. Consequently, when multiplying u and v by an inner product, there are two possible results: either there is no point that both functions are different from zero, or the non-zero interval of one function is completely contained in a constant part of the other function. Obviously this goes for our Weighted Haar Basis as well. Now, let us verify that the inner product of some v_k with a constant function $f(x) = m$ is zero:

$$\langle v_k, f \rangle = N \cdot \sum_{i=0}^{N-1} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} v_k(x) f(x)\, dx = N \cdot \sum_{i=0}^{N-1} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} v_k(x)\, m dx =$$

$$mN \cdot \sum_{i=0}^{N-1} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} v_k(x)\, dx =$$

$$mN \cdot \sum_{\{i|v_k(\frac{i}{N})>0\}} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} v_k(x)\, dx + mN \cdot \sum_{\{i|v_k(\frac{i}{N})<0\}} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} v_k(x)\, dx =$$

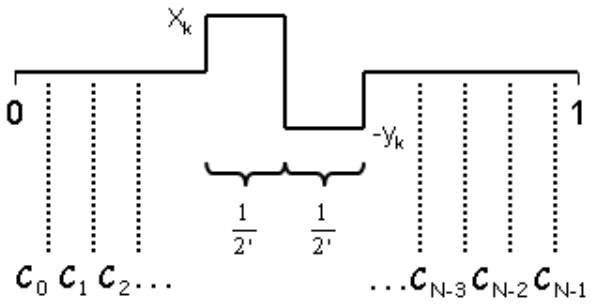

Figure 3: An example for a Weighted Haar Basis function

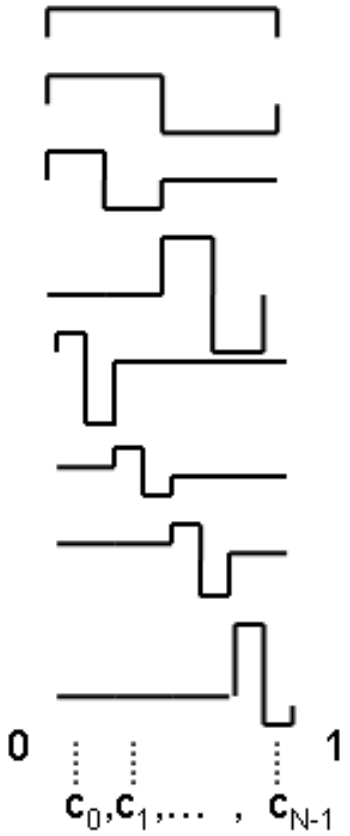

Figure 4: the weighted Haar Basis along with the workload coefficients, each coefficient under its corresponding interval. For each level, the functions of the level are different from zero over intervals of equal size.

$$mN \cdot \sum_{\{i|v_k(\frac{i}{N})>0\}} c_i \frac{x_k}{N} - mN \cdot \sum_{\{i|v_k(\frac{i}{N})<0\}} c_i \frac{y_k}{N} =$$

$$mN \cdot \frac{x_k}{N} \sum_{\{i|v_k(\frac{i}{N})>0\}} c_i - mN \cdot \frac{y_k}{N} \sum_{\{i|v_k(\frac{i}{N})<0\}} c_i = m\left(x_k l_k - y_k r_k\right) = 0$$

Now, in order to get an orthonormal basis all we have to do is to normalize those basis functions.

Let us compute the norm of some v_k whose positive part is set to x_k and its negative part is set to y_k:

$$\langle v_k, v_k \rangle = N \cdot \sum_{i=0}^{N-1} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} v_k^2(x)\,dx =$$

$$N \cdot \sum_{\{i|v_k(\frac{i}{N})>0\}} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} v_k^2(x)\,dx + N \cdot \sum_{\{i|v_k(\frac{i}{N})<0\}} c_i \int_{\frac{i}{N}}^{\frac{i+1}{N}} v_k^2(x)\,dx =$$

$$N \cdot \sum_{\{i|v_k(\frac{i}{N})>0\}} c_i \frac{x_k^2}{N} + N \cdot \sum_{\{i|v_k(\frac{i}{N})<0\}} c_i \frac{y_k^2}{N} =$$

$$N\frac{x_k^2}{N} \sum_{\{i|v_k(\frac{i}{N})>0\}} c_i + N\frac{y_k^2}{N} \sum_{\{i|v_k(\frac{i}{N})<0\}} c_i = x_k^2 l_k + y_k^2 r_k$$

From the orthogonality condition we will take $y_k = \frac{x_k l_k}{r_k}$:

$$x_k^2 l_k + y_k^2 r_k = 1 \Leftrightarrow x_k^2 l_k + \left(\frac{x_k l_k}{r_k}\right)^2 r_k = 1 \Leftrightarrow x_k^2 l_k + \frac{x_k^2 l_k^2}{r_k} = 1 \Leftrightarrow$$

$$x_k^2 \left(l_k + \frac{l_k^2}{r_k^2}\right) = 1 \Leftrightarrow x_k^2 = \frac{1}{l_k + \frac{l_k^2}{r_k^2}} \Leftrightarrow x_k = \sqrt{\frac{1}{l_k + \frac{l_k^2}{r_k^2}}} = \sqrt{\frac{r_k}{l_k r_k + l_k^2}}$$

So we will take:

$$x_k = \sqrt{\frac{r_k}{l_k r_k + l_k^2}} \quad y_k = \sqrt{\frac{l_k}{l_k r_k + r_k^2}}$$

There is a special case which is the computing of the constant basis function (which represents the total weighted average) $v_0(x) = const$. We would like the norm of this function to be 1. We just have to put $x_k = y_k$ in the equation $x_k^2 l_k + y_k^2 r_k = 1$ and get $f(x) = x_k = y_k = \sqrt{\frac{1}{l_k + r_k}} = const$. Again, notice that had all the workload coefficients been equal ($c_i = \frac{1}{N}$) we would get the standard Haar basis used to minimize the standard L_2 norm. □

Again, notice that had all the workload coefficients been equal ($c_i = \frac{1}{N}$) we would get the standard Haar basis used to minimize the standard L_2 norm.

As we have seen, this is an orthonormal basis to our function space. In order to see that it is a wavelet basis, we can notice that for each $k = 1, ..., j$, the first 2^k functions are an orthonormal set belonging to V_k (its dimension is 2^k) and which is therefore a basis of V_k.

5 The algorithm for WWW transform

In this section we describe the algorithmic part. Given a workload of point queries and a data vector to be approximated, we build workload-based wavelet synopses of the data vector using a weighted Haar basis. The algorithm has two parts:

1. Computing efficiently a *Weighted Haar basis*, given a workload of point queries. (Sec. 5.1)
2. Computing efficiently the *Weighted Haar Wavelet Transform* with respect to the chosen basis. (Sec. 5.2)

5.1 Computing efficiently a weighted Haar basis

Note that at this point we already have a method to find an orthonormal basis with respect to a given workload based inner product. Recall that in order to know x_k and y_k for every basis function we need to know the corresponding l_k and r_k. We are going to compute all those partial sums in linear time. Suppose that the basis functions are arranged in an array like in a binary tree representation. The highest resolution functions are at indexes $\frac{N}{2}, ..., N-1$, which are the lowest level of the tree. The next resolution level functions are at indexes $\frac{N}{4}, ..., \frac{N}{2}-1$, and so on, until the constant basis function is in index 0. Notice that for the lowest level (highest resolution) functions (indexes $\frac{N}{2}, ..., N-1$) we already have their l_k's and r_k's. These are exactly the workload coefficients. It can easily be seen in Fig. 4 for the lower four functions. Notice that after computing the accumulated sums for the functions at resolution level i, we have all the information to compute the higher level functions: let u_k be a function at resolution level i and u_{2k}, u_{2k+1} be at level $i+1$, where their supports included in u_k's support (u_k is their ancestor in the binary tree of functions). We can use the following formula for computing l_k and r_k:

$$l_k = l_{2k} + r_{2k} \quad r_k = l_{2k+1} + r_{2k+1}$$

It can be seen in the example of Fig. 4. Thus, we can compute in one pass only the lowest level, and build the upper levels bottom-up (in a way somewhat similar to the Haar wavelet transform). At the end of a phase in the algorithm (a phase would be computing the functions of a specific level) we would keep a temporary array holding all the pairwise sums of all the l_k's and r_k's from that phase and use them for computing the next phase functions. Clearly, the running time is $\frac{N}{2} + \frac{N}{4} + ... + 1 = O(N)$. The number of I/Os is $O(N/B)$ I/Os (where B is the block size of the disk) – since the process is similar to the computation Haar wavelet transform. A pseudo-code of the computation can be found in Fig. 14. The *createFunction*() function takes two sums of weight coefficients corresponding to the function's positive part and to the function's negative part, and build a function whose positive (resp. negative) part's value is x_k (resp. y_k) using the following formulae:

$$x_k = \sqrt{\frac{r_k}{l_k r_k + l_k^2}} \quad y_k = \sqrt{\frac{l_k}{l_k r_k + r_k^2}}$$

5.2 Computing a weighted Haar wavelet transform

Given the basis we would like to efficiently perform the wavelet transform with respect to that basis. Let us look at the case of $N = 2$ (Fig. 5). Suppose we would like to represent the function in Fig. 6. It is easy to compute the following result (denote α_i as the coefficient of f_i):

$$\alpha_0 = \frac{yv_0 + xv_1}{x+y} \qquad \alpha_1 = \frac{v_0 - v_1}{x+y}$$

(by solving 2x2 matrix). Notice that the coefficients are weighted averages and differences, since the transform generalizes the standard Haar transform (by taking $x = y = \sqrt{2^i}$ we get the standard Haar transform). It's easy to reconstruct the original function from the coefficients:

$$v_0 = \alpha_0 + x\alpha_1 \qquad v_1 = \alpha_0 - y\alpha_1$$

This implies a straightforward method to compute the wavelet transform (which is I/O efficient as well) according to the way we compute a regular wavelet transform with respect to the Haar

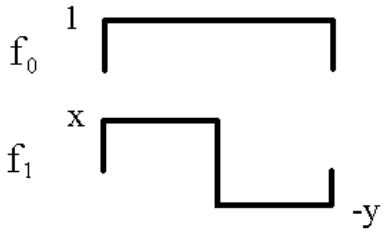

Figure 5: Weighted Haar Transform with two functions

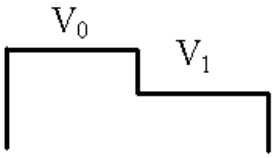

Figure 6: a simple function with 2 values over $[0, 1)$

basis: we go over the data, and compute the weighted differences which are the coefficients of the bottom level functions. We keep the weighted averages, which can be represented *solely* by the rest of the basis functions (the "lower resolution" functions - as in the regular Haar wavelet transform), in another array. We repeat the process over the averages time and time again until we have the overall average, which is added to our array as the coefficient of the constant function ($v_0(x) = const$). While computing the transform, in addition to reading the values of the signal, we need to read the proper basis function that is relevant for the current stage (in order to use the x_k and y_k of the function that is employed in the above formula). This is easy to do, since all the functions are stored in an array F and the index of a function is determined by the iteration number and is identical to the index of the corresponding currently computed coefficient. A pseudo code of the algorithm is can be found in Fig. 15.

As we know, the Haar wavelet transform is a linear algorithm. The steps of our algorithm are identical to the steps of the Haar algorithm, with the addition of reading the data at $F[i]$ (the x_k and y_k of the function) during the i'th iteration. Therefore the I/O complexity of that phase remains $O(N/B)$ (B is the disk block size) with $O(N)$ running time.

After having the coefficient of the orthonormal basis we would keep the largest M coefficients, along with their corresponding M functions, and throw the smallest coefficients relying on Thm. 1 We can do it in linear time using the *M-approximate quantile algorithm* [13].

6 Optimal synopsis for mean relative error

We next show a variant of the weighted-wavelets-based algorithm minimizes the weighted L_2 norm of the vector of *relative* errors, weighted by the query workload, using weighted wavelets. We demonstrate another use of giving weights to data values, used to minimize the mean-squared-relative-error measured over the data values.

Recall that in order to minimize the weighted L_2 norm of relative errors, we need to minimize $\sum_{i=1}^{N} c_i \left(\frac{|d_i - \hat{d}_i|}{d_i}\right)^2$ (actually $\sum_{i=1}^{N} c_i \left(\frac{|d_i - \hat{d}_i|}{max\{d_i, s\}}\right)^2$, but the idea is the same). Since $D = d_1, ..., d_N$ is part of the input of the algorithm, it is fixed throughout the algorithm's execution. We can thus

divide each c_i by d_i^2 and get a new vector of weights: $W = \left(\frac{c_1}{d_1^2}, ..., \frac{c_N}{d_N^2}\right)$. Relying on our previous results, and using the new vector of weights we minimize $\sum_{i=1}^{N} \frac{c_i}{d_i^2} \left(|d_i - \hat{d}_i|\right)^2 = \sum_{i=1}^{N} c_i \left(\frac{|d_i - \hat{d}_i|}{d_i}\right)^2$, which is the WL_2 norm of relative errors. Notice that in the case $b_i = \frac{1}{N}$ (the uniform case) the algorithm minimizes the mean-relative-error over all *data values*. As far as we know, this is the first algorithm that minimizes the mean-relative-error over the data values.

7 Experiments

In this section we demonstrate the advantage obtained by our workload-based wavelet synopses. All our experiments were done using the τ-synopses [15] system. For our experimental studies we used both synthetic and real-life data sets. The synthetic data-sets are taken from the TPCH data (www.tpc.org), and the real-life data-sets are taken from the *Forest CoverType* data provided by KDD Data of the University of California (http://kdd.ics.uci.edu). The data-sets are:

1. **TPCH** -
   - TPCH1 - Data attribute 1 from table ORDERS, filtered by attribute O_CUSTKEY, which contains about 150,000 distinct values.
2. **KDD** -
   - KDD2048 - Data attribute Aspect from table CovTypeAgr filtered by Elevation from the KDD data, with a total of 2048 distinct values.

The sets of queries were generated independently by a Zipf distribution generator. We used queries of different skews, distributed by several Zipf parameter values. We took here the zipf parameters 0.2, 0.5 and 0.8, in order to test the behavior of the synopses under different skews, which range from close-to-uniform to highly skewed. The sets of queries contained 3000-5000 queries over each data set.

In Fig. 7 we compared the standard wavelet synopsis from [16] with our WB-MSE wavelet synopsis. The standard synopsis is depicted in solid line. We measured the WB-MSE as a function of synopsis size, measured as the number of coefficients in the synopsis. For each $M = 10, 20, ..., 100$ we built synopses of size M using both methods and compared the WB-MSE error, measured with respect to a given workload of queries. The workload contained 5000 Zipf distributed point queries, with a Zipf parameter of 0.5. The data-set was the TPCH1 data. As the synopsis size increases, the error of the workload-based algorithm becomes much smaller than the error of the standard algorithm. The reason for this is that synopses of sizes 10,...,100 are very small with respect to a data of size 150,000. Since the standard algorithm does not take the query workload into account, the results are more or less the same for all synopses sizes in the experiment. However, the workload-based synopsis adapts itself to the query workload, which is of size 5000. All the data values which are not queried by the workload are given very small "importance weights", so the synopsis actually has to be accurate over less than 5000 values. Thus, there is a sharp decrease in the error of the workload-based algorithm as the synopsis size increases.

In Fig. 8 we used a similar experiment, this time with the KDD2048 data. The standard synopsis is again depicted in solid line. As in the previous experiment, we measured the WB-MSE as a function of synopsis size. For each $M = 20, 40, ..., 200$ we built synopses of size M using both

methods and compared the WB-MSE error, measured with respect to a given workload of queries. The workload contained 5000 Zipf distributed point queries, with a Zipf parameter 0.5. The data was the KDD2048 data, of size 2048. We see that for each synopsis size the error of the standard algorithm is approximately twice the error of the workload-based algorithm. The reason for this is that here the query workload is larger than the data-set, in contrast to the previous experiment. Thus, most of the data is queried by the workload, so the "importance weights" given to data values are more uniform than in the previous experiment. Therefore, the error difference is smaller than in the previous experiment, since the advantage of the workload-based algorithm becomes more significant as the workload gets more skewed. However, since the workload-based synopsis adapts itself to the workload, the error is still better than the standard synopsis, which assumes uniform distribution.

In Fig. 9 we compared the standard wavelet synopsis from [16] and the adaptive-greedy workload-based wavelet synopsis from [14] with our WB-MRE wavelet synopsis. The standard synopsis is depicted in dotted line with "x"s. Since it is hard to distinguish between the other two synopses in this resolution level, we zoom into this figure in Fig. 10. We measured the WB-MRE as a function of synopsis size, measured as the number of coefficients in the synopsis. For each $M = 20, 40, ..., 200$ we built synopses of size M using the three methods and compared the WB-MRE error, measured with respect to a given workload of queries. The workload contained 3000 Zipf distributed point queries, with a Zipf parameter of 0.5. The data-set was the KDD2048 data. Since the standard algorithm does not take into account the query-workload and is not adapted for relative errors, its approximation error is more than 30-40 times larger than the approximation errors of the workload-based algorithms, for each synopsis size.

In Fig. 10 we compare the adaptive-greedy workload-based synopsis from [16] with our WB-MRE synopsis. The adaptive-greedy synopsis is depicted in solid line. We measured the WB-MRE as a function of synopsis size, measured as the number of coefficients in the synopsis. For each $M = 20, 40, ..., 200$ we built synopses of size M using the two methods and compared the WB-MRE error, measured with respect to a given workload of queries. The workload contained 5000 Zipf distributed point queries, with a Zipf parameter of 0.5. The data-set was the KDD2048 data. For each synopsis size, the approximation error of the adaptive-greedy is 10-20 times larger than the error of our WB-MRE algorithm.

In Fig. 11 we depict the WB-MRE as a function of synopsis size, for three given query workloads, distributed with Zipf parameters 0.2, 0.5 and 0.8. The data-set was the KDD2048 data-set, and the workloads consisted 5000 queries. For each of the given three workloads we build synopses of size $M = 50, 100, ..., 500$ and depicted the WB-MRE as a function of synopsis size. It can be seen that many wavelet coefficients can be ignored before the error significantly increases. This is a desired feature for any synopsis. For example, for synopses of size 500 the WB-MRE is smaller than 0.05, and for synopses of size 250 the WB-MRE is smaller than 0.1. It can also be seen that the higher the skew, the more accurate the workload-based synopses. The reason is that when the skew gets higher, the synopsis should be accurate over a smaller number of data values.

In Fig. 12 we compare the standard algorithm from [14] with our WB-MRE algorithm in a different way than before. We compare the ratio between the approximation error of the standard algorithm and the approximation error of the WB-MRE algorithm, for different workload skews. The comparison was done for three different query workloads, distributed with different Zipf parameters. The workloads contained 5000 queries, distributed with Zipf parameters 0.2, 0.5 and 0.8 respectively. The data-set was the KDD2048. For each given workload we measured the error ratio between the two synopses, for each synopses size $M = 50, 100, ...500$. It is clearly seen

that the higher the skew of the workload, the higher the ratio between the approximation errors of the synopses. The reason is than as the workload gets far from uniform, the advantage of the workload-based algorithms naturally becomes more significant over the standard synopsis, which assumes uniform workload.

In Fig. 13 we show the robustness of the workload-based wavelet synopses to deviations from pre-defined workload. The experiment addresses the problem of incorrect future workload estimation. When building our synopsis, we assumed the queries would be distributed as Zipf(0.2). We fixed the synopsis size, and built our synopsis. We then used the synopsis to answer query workloads distributed different than expected, e.g with Zipf parameters 0.3,0.4,... etc. The figure depicts the WB-MRE as a function of the difference between the actual query distribution and our estimated query distribution (estimated as Zipf(0.2)). The skew difference is the difference between the actual Zipf parameter and the estimated Zipf parameter, according to which we assumed the queries are distributed. We show that small errors in the workload estimation introduce only small errors in the quality of the approximation, and that the error grows continuously as the deviation from pre-defined workload increases.

8 Conclusions

In this paper we introduce the use of weighted wavelets for building optimal workload-based wavelet synopses. We present two time-optimal and I/O-optimal algorithms for workload-based wavelet synopses, which minimize the WB-MSE and and the WB-MRE error measures, with respect to any given query workload. The advantage of optimal workload-based wavelet synopses, as well as their robustness, were demonstrated by experimentations.

Recently, and independently of our work, Muthukrishnan [17] presented an optimal workload-based wavelet synopsis with respect to the standard *Haar* basis. The algorithm for building the optimal synopsis is based on dynamic programming and takes $O(N^2M/\log M)$ time. As noted above, standard Haar basis is not orthonormal w.r.t. the workload-based error metric, and an optimal synopsis w.r.t. this basis is not necessarily also an optimal enhanced wavelet synopsis. Obtaining optimal enhanced wavelet synopses for the standard Haar wavelets may be an interesting open problem. Also, as quadratic time is too costly for massive data sets, it may be interesting to obtain a time efficient algorithm for such synopses. As far as approximation error is concerned, although in general optimal synopses w.r.t. the standard Haar and a weighted Haar bases are incomparable, both bases have the same characteristics. It would be interesting to compare the actual approximation errors of the two synopses for various data sets. This may indeed be the subject of a future work.

Acknowledgments: We thank Leon Portman for helpful discussions and for his assistance in setting up the experiments on the τ-synopses system. We also thank Prof. Nira Dyn for helpful discussions regarding the wavelets theory.

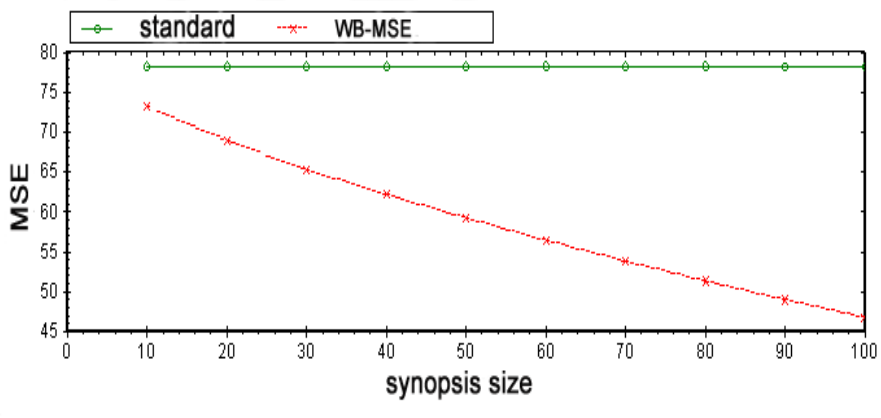

Figure 7: Comparing the WB-MSE of the standard and the workload-based synopses, for different synopses sizes. Data: TPCH1, Workload: 5000 queries distributed as Zipf(0.5).

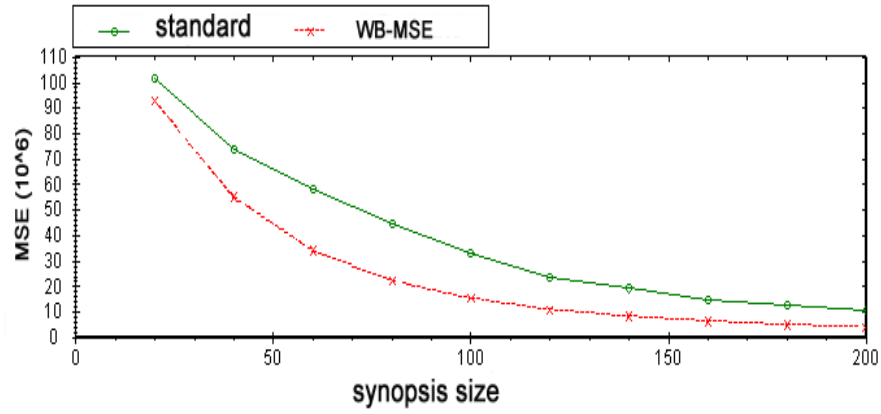

Figure 8: Comparing the WB-MSE of the standard and the workload-based synopses, for different synopses sizes. Data: KDD2048, Workload: 5000 queries distributed as Zipf(0.5).

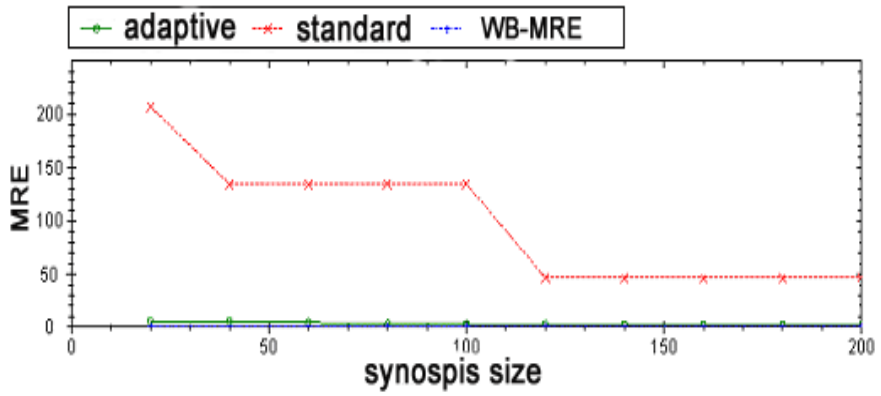

Figure 9: Comparing the WB-MRE of the standard synopsis, the workload-based adaptive-greed synopsis and our WB-MRE synopsis, for different synopses sizes. Data: KDD2048, Workload: 5000 queries distributed as Zipf(0.5). Since the adaptive and the WB-MRE are indistiguishable in this scale, an elaboration of that zone is in Fig. 10

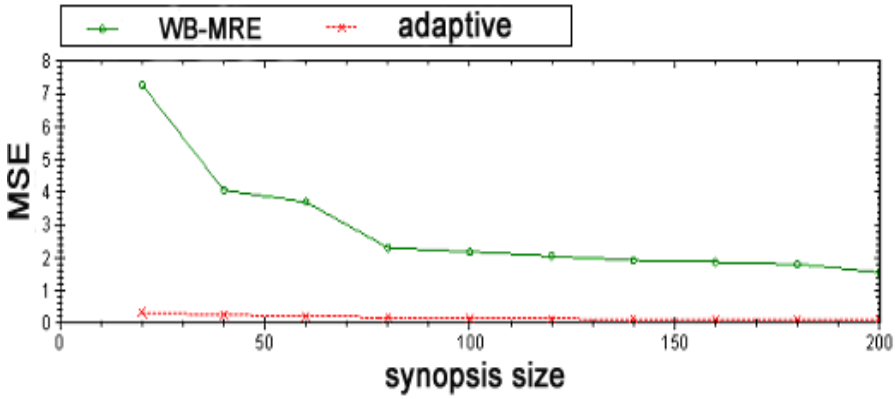

Figure 10: Comparing the WB-MRE of the workload-based adaptive-greed synopsis and our WB-MRE synopsis, for different synopses sizes. Data: KDD2048, Workload: 5000 queries distributed as Zipf(0.5).

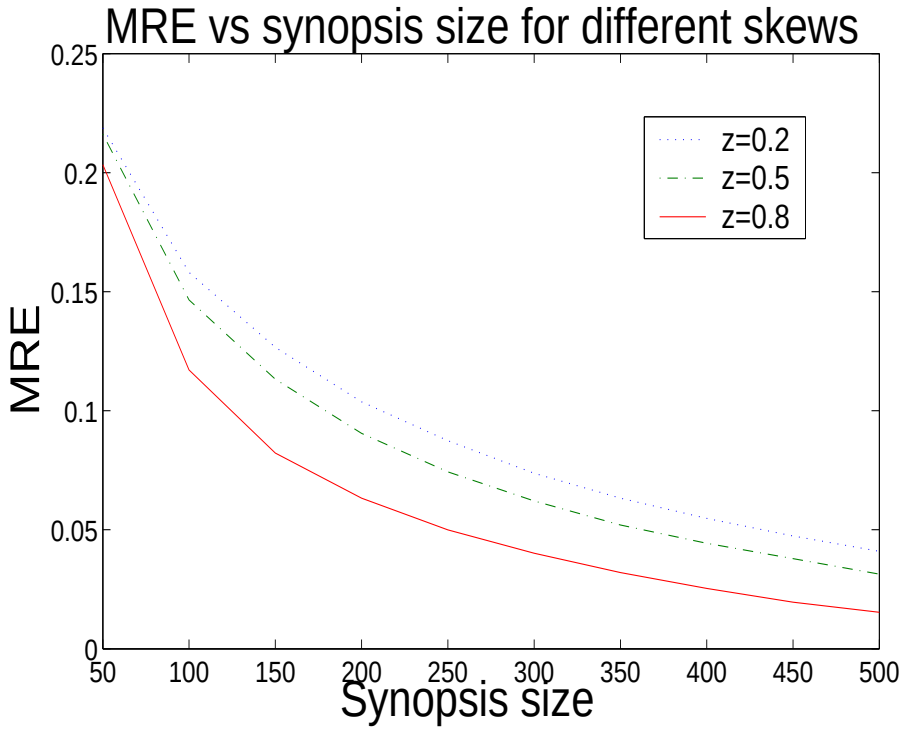

Figure 11: Relative error for different skews. Data: KDD2048, Workload: 5000 queries distributed as Zipf(0.2), Zipf(0.5), Zipf(0.8).

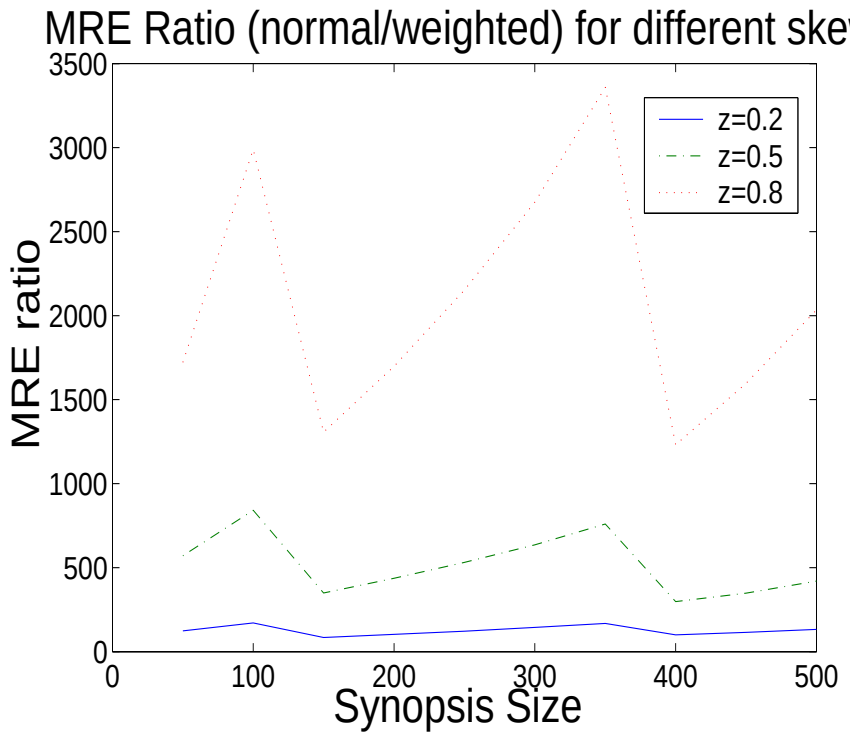

Figure 12: Comparing the ratios between the errors of the standard synopsis and the WB-MRE synopsis, for different workloads. For each z=0.2,0.5,0.8 we used a workload distributed as Zipf(z), and showed the error ratio for different synopses sizes.

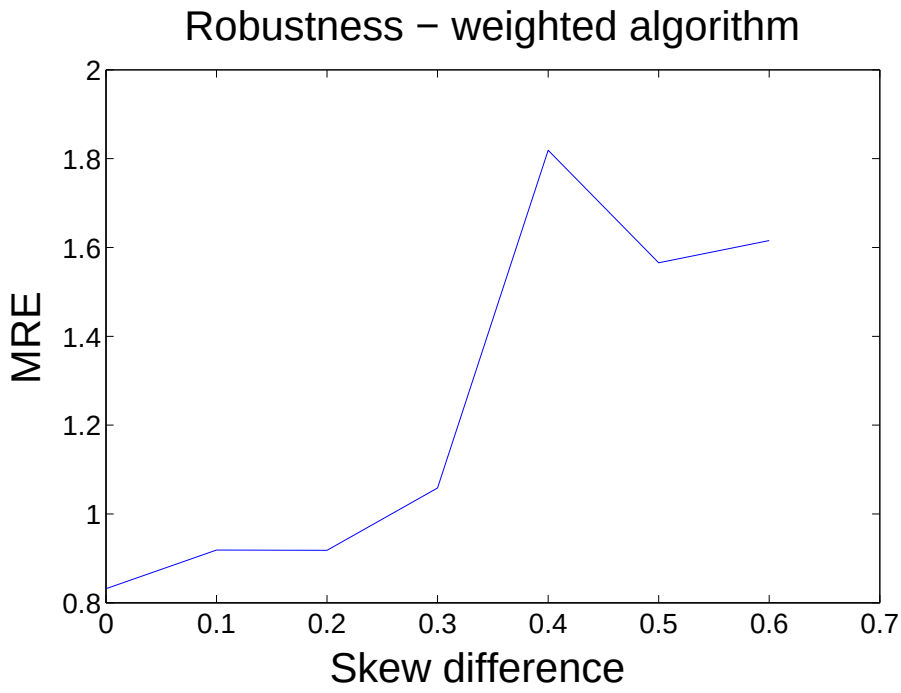

Figure 13: Robustness of the workload based synopses to workload deviations. A synopsis that assumes workload distributed as Zipf(0.2) was built, and used to answer workloads distributed differently. The plot shows the approximation error as a function of the workload estimation error. Data: KDD2048, Workloads: 5000 queries distributed Zipf(0.3),Zipf(0.4),...

```
input: an array W of weight coefficients
output: an array F of basis functions

temp.length = N/2
for i = 0 to N/2 - 1
  F[N/2 + i] = createFunction(W[2i], W[2i+1])
  temp[i] = W[2i] + W[2i+1]
while temp.length > 1
    temp.length /= 2
    offset = temp.length
    for i = 0 to temp.length/2
      F[offset + i] =
      createFunction(temp[2i], temp[2i + 1])
      temp[i] = temp[2i] + temp [2i + 1]
F[0] = createConstFunction (1/temp[0])
```

Figure 14: Construction of a WH Basis

```
input: an array D of data values, an array F of basis functions
output: an array Res of wavelet coefficients
for i = 0 to N/2 - 1
  Res[N/2 + i] = (D[2i] - D[2i + 1])/(F[i].pos + F[i].neg)
  temp[i] = D[2i] * F[i].neg + D[2i + 1] * F[i].pos/(F[i].pos + F[i].neg)
while temp.length > 1
  offset = temp.length/2
  for i = 0 to temp.length/2
    Res[offset + i] = temp[2i] - temp[2i + 1] /
      (F[i].pos + F[i].neg)
    temp[i] = (temp[2i] * F[i].neg + temp[2i + 1] * F[i].pos) /
      (F[i].pos + F[i].neg)
Res[0] = temp[0]/F[0].constValue
```

Figure 15: The wavelet transform

References

[1] A. Aboulnaga and S. Chaudhuri. Self-tuning histograms: Building histograms without looking at data. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, pages 181–192, 1999.

[2] K. Chakrabarti, M. Garofalakis, R. Rastogi, and K. Shim. Approximate query processing using wavelets. In *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, 2000*, pages 111–122.

[3] S. Chaudhuri, G. Das, M. Datar, R. Motwani, , and V. R. Narasayya. Overcoming limitations of sampling for aggregation queries. In *ICDE*, pages 534–542, 2001.

[4] S. Chaudhuri, G. Das, and V. Narasayya. A robust, optimization-based approach for approximate answering of aggregate queries. In *Proceedings of the 2001 ACM SIGMOD international conference on on Management of data*, 2001.

[5] R. R. Coifman, P. W. Jones, , and S. Semmes. Two elementary proofs of the l2 boundedness of cauchy integrals on lipschitz curves. *J. Amer. Math. Soc.*, 2(3):553–564, 1989.

[6] A. Deligiannakis and N. Roussopoulos. Extended wavelets for multiple measures. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 229–240.

[7] V. Ganti, M.-L. Lee, and R. Ramakrishnan. Icicles: Self-tuning samples for approximate query answering. *The VLDB Journal*, pages 176–187, 2000.

[8] M. Garofalakis and P. B. Gibbons. Wavelet synopses with error guarantees. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, 2002.

[9] M. Garofalakis and A. Kumar. Deterministic wavelet thresholding for maximum-error metrics. In *Proceedings of the 2004 ACM SIGMOD international conference on on Management of data*, pages 166–176.

[10] P. B. Gibbons and Y. Matias. Synopsis data structures for massive data sets. In *DIMACS: Series in Discrete Mathematics and Theoretical Computer Science: Special Issue on External Memory Algorithms and Visualization, A*, 1999.

[11] M. Girardi and W. Sweldens. A new class of unbalanced Haar wavelets that form an unconditional basis for L_p on general measure spaces. *J. Fourier Anal. Appl.*, 3(4), 1997.

[12] S. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, 2nd edition, 1999.

[13] G. S. Manku, S. R., and B. G. Lindsay. Approximate medians and other quantiles in one pass and with limited memory. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 426–435, New York, 1998.

[14] Y. Matias and L. Portman. Workload-based wavelet synopses. Technical report, Department of Computer Science,Tel Aviv University, 2003.

[15] Y. Matias and L. Portman. τ-synopses: a system for run-time management of remote synopses. In *International conference on Extending Database Technology (EDBT), Software Demo, 865-867 & ICDE'04, Software Demo*, March 2004.

[16] Y. Matias, J. S. Vitter, and M. Wang. Wavelet-based histograms for selectivity estimation. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 448–459, Seattle, WA, June 1998.

[17] S. Muthukrishnan. Workload-optimal wavelet synopsis. Technical report, May 2004.

[18] L. Portman. Workload-based wavelet synopses. M.sc. thesis, Tel Aviv University, 2003.

[19] E. J. Stollnitz, T. D. Derose, and D. H. Salesin. *Wavelets for Computer Graphics*. Morgan Kaufmann, 1996.

[20] J. S. Vitter and M. Wang. Approximate computation of multidimensional aggregates of sparse data using wavelets. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, pages 193–204, Phildelphia, June 1999.

[21] J. S. Vitter, M. Wang, and B. Iyer. Data cube approximation and histograms via wavelets. In *Proceedings of Seventh International Conference on Information and Knowledge Management*, pages 96–104, Washington D.C., November 1998.