The Haar$^+$ Tree: a Refined Synopsis Data Structure[*]

Panagiotis Karras
Department of Computer Science
University of Hong Kong
Pokfulam Road, Hong Kong, China
pkarras@cs.hku.hk

Nikos Mamoulis
Department of Computer Science
University of Hong Kong
Pokfulam Road, Hong Kong, China
nikos@cs.hku.hk

Abstract

We introduce the Haar$^+$ tree: a refined, wavelet-inspired data structure for synopsis construction. The advantages of this structure are twofold: First, it achieves higher synopsis quality at the task of summarizing data sets with sharp discontinuities than state-of-the-art histogram and Haar wavelet techniques. Second, thanks to its search space delimitation capacity, Haar$^+$ synopsis construction operates in time linear *to the size of the data set for* any *monotonic distributive error metric. Through experimentation, we demonstrate the superiority of Haar$^+$ synopses over histogram and Haar wavelet methods in both construction time and achieved quality for representative error metrics.*

1 Introduction

The need to reduce a very large data set into a compact synopsis that captures its basic characteristics arises frequently in various database applications, like, for example, OLAP/DSS systems [35], approximate query answering [1, 30, 22, 3], cost-based query optimization [27] and time-series mining [4]. Over the past years, two principal methods have emerged as recommendable choices for efficient synopsis construction: histograms [19, 21, 32, 31, 22, 30, 10, 23, 11, 15, 17, 12, 16, 34] and Haar wavelets [27, 35, 3, 6, 7, 8, 25, 26, 29, 12, 13, 14]. The main objective of both approaches is to minimize some appropriate error measure [9], given a space budget.

However, previous research has not attempted to examine how state-of-the-art Haar wavelet and histogram-based synopsis construction techniques compare to each other. A comparison is required both in terms of time and space for synopsis construction, and in terms of synopsis quality, depending on the characteristics of the data at hand. Such attempts as were made in this direction [27, 17] carried out uneven comparisons by setting provably optimal methods for the one technique against non-optimal ones for the other. From a qualitative point of view, a histogram-based synopsis is arguably recommendable when summarizing smooth datasets without sharp discontinuities or bursts. On the other hand, wavelet-based techniques are more well-suited for approximating datasets with such discontinuities. A B-term histogram defines only $B + 1$ distinct value intervals, but has total freedom on the bucket boundaries it chooses and the value it assigns to each of them. On the other hand, a B-term wavelet synopsis defines $B + 1$ to $3B + 1$ distinct value intervals, but is constrained on the boundaries and the values assigned to each of them. In particular, a wavelet coefficient contributes its value positively to the former and negatively to the latter of the two halves of the fixed-size interval that it affects; hence the resulting summarization value in a wavelet-defined interval may be sub-optimal.

Still, at the task of minimizing a *distributive* error metric (as opposed to a maximum error metric), such as the average absolute or relative error, both optimal histogram-based [23, 17] and quality-aware wavelet-based schemes [8, 12, 29, 13, 14] run in time super-linear to the size of the input.[1] This state of affairs renders previous techniques inapplicable for the time-efficient summarization of very large data sets with a distributive error quality guarantee and calls for a different approach.

In this paper, we introduce the Haar$^+$ tree: a refined, wavelet-inspired synopsis data structure, which alleviates the aforementioned deficiencies. First, it adds flexibility to the values constructed by a classical Haar wavelet synopsis, while maintaining its compression power over a histogram; therewith it outperforms previous techniques in the approximation quality for hard-to-summarize data sets. Second, it allows for easy delimitation of the search space, resulting to a synopsis construction algorithm for *general* error metrics that operates in time *linear* to the size of the data. We demonstrate the superiority of Haar$^+$ over histogram and Haar wavelet methods in both construction time and achieved quality for representative error metrics.

[*] Work supported by grant 7160/05E from Hong Kong RGC.

[1] The simple wavelet synopsis algorithm for Euclidean error [33, 27] notwithstanding.

2 Background and Related Work

Presently, the principal structures employed for quality-aware synopsis construction are histograms and Haar wavelets. Under both approaches, given an n-size data vector $\mathbf{D}$, the problem is to devise a representation of $\mathbf{D}$ in B space, so that an error measure in the produced approximation $\hat{\mathbf{D}}$ is minimized. A *normalized* Minkowski-norm error metric, generally expressed in its *weighted* version: $\mathcal{L}_p^w(\hat{\mathbf{D}}, \mathbf{D}) = \sum_i \{\frac{(w_i|\hat{d}_i - d_i|)^p}{n}\}^{\frac{1}{p}}$, covers most practically interesting cases of a point-wise error metric; n denotes the size of the data set, $\hat{d}_i$ the reconstructed value for d_i and w_i a related weight; in the case of relative error, this weight is defined as $w_i = \frac{1}{\max\{|d_i|,S\}}$, where $S > 0$ is a sanity bound that prevents small values from unnaturally dominating the error result [8]. The methods we propose in this paper are applicable to all such metrics, and to any metric in a wider class of *monotonic distributive* metrics. For illustration purposes, we use three instances of a normalized Minkowski-norm: the average absolute error $\mathcal{L}_1$, the sum of squared errors $\mathcal{L}_2$, and the maximum absolute error $\mathcal{L}_\infty$.

2.1 Histogram-based Data Reduction

Given a data sequence $\mathbf{D}$, a *histogram*, also called segmentation and partitioning, divides $\mathbf{D}$ into $B \ll n$ successive disjoint intervals $[b_i, e_i]$, $1 \leq i \leq B$ called *buckets* or *segments* and attributes a single value v_i to each of them that approximates all consecutive values therein, d_i, $i \in [b_i, e_i]$. Thus, a single bucket (segment) can be expressed by the triplet $s_i = \{b_i, e_i, v_i\}$. $2B-1$ numbers suffice to represent a B-bucket histogram (since $\forall r, 1 < i \leq B, b_i = e_{i-1} + 1$ and the edges are fixed). For a particular target error metric, the optimal value of v_i is defined as a function of the data values within $[b_i, e_i]$.[2]

Most initial work on histogram construction algorithms in the database literature focused on heuristics that exhibited low errors in some estimation problem, such as the MaxDiff [32] and MHIST [31] heuristics; it did not attempt to detect the optimal bucket boundaries for the given problem [20]. The problem of computing optimal bucket boundaries for an error metric was first addressed within the database community by Jagadish et al. [23]; the histogram construction problem was then also disengaged from the specific application of approximating frequency distributions, and posed as the general problem of defining a *piecewise constant approximation* of a finite data sequence. In fact, this is a special case of the problem of approximating a curve by line segments; hence the dynamic programming algorithm in [23] is a special case of the line-segmentation algorithm introduced by the inventor of dynamic programming [2]. Guha et al. [17] specialized this solution for the case of the maximum-relative-error and average-relative-error metrics (by extension, any weighted maximum-error and average-error metrics). Later, Guha [12] introduced a generic space-efficiency paradigm applicable on all those histogram construction algorithms.

2.2 Haar Wavelet-based Data Reduction

Wavelet analysis is a mathematical technique for hierarchical function decomposition [33]. The simplest wavelet transform, introduced by Haar [18], can be visualized through a complete binary tree, herein called *Haar tree*. This tree holds the coefficients representing the original data in successive layers of detail; the final tree layer holds the original data. The coefficient in the root node contains the overall average value and each other coefficient value c_i contributes the value $+c_i$ to all data values (leaves) in its *left* sub-tree and $-c_i$ to those in its *right* sub-tree. Hence each original data value is reconstructed by adding/subtracting the coefficients in the path towards its position.

A *Haar wavelet synopsis* is a Haar wavelet vector z of B non-zero terms, such that its inverse wavelet transform $\hat{d} = \mathcal{W}^{-1}(z)$ approximates a data vector d, with $B \ll n$. For the Euclidean error metric, the optimal Haar wavelet synopsis consists of the top-B *normalized* coefficients of the original data vector's Haar wavelet transform [33]. This computational convenience has allowed for the extension of the Euclidean error minimization methodology into data streams of the cash register model [5], multiple-measure [6] and multi-dimensional [24] data sets, and workload-aware optimization based on a weighted version of the Haar basis [26]. Unfortunately, this convenience does not extend to non-Euclidean *distributive* (decomposable) error metrics. Still, recent studies have strived to construct algorithms that provide optimal synopses for such metrics within the Haar framework. The first systematic endeavor was made by Garofalakis and Gibbons with a randomized rounding scheme [7]. However, as shown in [17], the quality of this scheme's results is not high in practice. Subsequently, Garofalakis and Kumar [8] developed a dynamic programming (DP) algorithm that detects the optimal B-term subset of a dataset's Haar wavelet transform to retain, for any distributive error metric. A dependable greedy counterpart to this solution for maximum-error metrics, applicable on time-series data streams, was proposed in [25]. Both the time and space complexities of the DP scheme [8] were reduced in [12]. Later, Guha and Harb [13, 14] discerned that the values of the B non-zero wavelet terms need not be obtained from the dataset's Haar wavelet transform; they can be set *unrestrictedly*, leading to higher quality than the *restricted* model. [13, 14] provided a fully polynomial-time approximation scheme (FPAS) for unrestricted Haar wavelet synopses under any Minkowski-norm error metric.

[2] E.g. for $\mathcal{L}_1$ it is the median of the values in $[b_i, e_i]$ [34], for $\mathcal{L}_2$ their mean [23], for $\mathcal{L}_\infty$ the mean of the maximum and minimum value among them, while [17] analyzes the respective relative error cases.

2.3 Motivation

We observe that the quality of approximation achieved by existing techniques is constrained by their nature. In the case of histograms, the primary limitation is that of *locality*; a bucket is supposed to approximate *neighboring* values, which are expected to exhibit small variations. Therefore, histograms are not good at approximating sharp discontinuities. In addition, a B-sized histogram approximates only $B + 1$ value ranges, as opposed to the Haar wavelet which can define up to $3B + 1$ value intervals. Regarding the Haar framework, any retained wavelet coefficient is supposed to bear on the two binary intervals that it affects two opposite-signed contributions of equal absolute magnitude. While this is the very property of the decomposition that allows for a near-linear computation of the Euclidean-error-optimal synopsis, it is not obligatory to maintain this restriction when the computational effectiveness that it allows does not apply any more. As we will show in the next sections, dropping this restriction not only *increases* the accuracy of approximation, but also *simplifies* the synopsis computation process, rendering the time complexity independent of the target error metric.

3 The Haar+ Tree

In this section we introduce the Haar+ tree, an enhanced and more powerful synopsis data structure, by dropping the restrictions of the classical Haar model. Figure 1 depicts a simple one-dimensional Haar+ tree structure that may be used for summarization of a four-element data set $\{d_0, d_1, d_2, d_3\}$. The structure contains a single root coefficient node c_0 that contributes its value to all approximated data values. This root is followed by a binary tree of coefficient nodes grouped in triads, depicted as C_1, C_2 and C_3. Every triad of coefficients substitutes what is a single wavelet coefficient in a classical Haar tree structure. In each such triad, the *head coefficient*, namely coefficients c_1, c_4 and c_7, behaves as a classical wavelet coefficient in reconstruction: it contributes its value positively to its left sub-tree and the same value negatively to its right sub-tree. However, the other two, left and right *supplementary coefficients*, namely c_2 and c_3 in group C_1, c_5 and c_6 in group C_2, and c_8 and c_9 in group C_3, contribute their value positively in the single subinterval that they affect. For example, coefficient c_3 contributes its value positively to data values d_2 and d_3, if such a non-zero value is maintained in a synopsis. The parent of the node where the head coefficient of a triad C resides is called *parent node* of C, while the triad where this parent node resides is called *parent triad* of C. For example, the parent node of triad C_2 is c_2 and its parent triad is C_1.

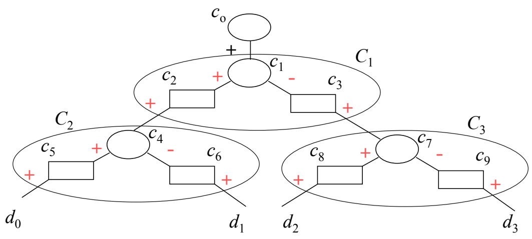


Figure 1. One-Dimensional Haar+ Tree

An *optimal* synopsis of space budget B for a given error metric $\mathcal{E}$ places B non-zero coefficient values at any positions in the Haar+ tree so that $\mathcal{E}$ is minimized. For example, for the four-element data set $\{5, 3, 12, 4\}$ the 2-term Haar+ synopsis that minimizes the sum of absolute errors $\mathcal{L}_1$, the sum of squared errors $\mathcal{L}_2$, and the maximum absolute error $\mathcal{L}_\infty$, consists of the coefficients $\{c_0 = 4, c_8 = 8\}$. The resulting estimated data set is $\{4, 4, 12, 4\}$ with absolute error values $\{1, 1, 0, 0\}$, hence, using non-normalized values, $\mathcal{L}_1 = 2$, $\mathcal{L}_2 = \sqrt{2}$ and $\mathcal{L}_\infty = 1$. The optimal 2-term *restricted* Haar synopsis for all three considered metrics is $\{c_0 = 6, c_7 = 4\}$, producing the errors $\{1, 3, 2, 2\}$ with $\mathcal{L}_1 = 8$, $\mathcal{L}_2 = 3\sqrt{2}$, $\mathcal{L}_\infty = 3$; by default, this is also the $\mathcal{L}_2$-optimal 2-term *unrestricted* Haar synopsis. On the other hand, the optimal 2-term *unrestricted* Haar synopsis for both other considered metrics is $\{c_0 = 5.5, c_7 = 4\}$, with $\mathcal{L}_1 = 8$ and $\mathcal{L}_\infty = 2.5$. Likewise, the $\mathcal{L}_2$- and $\mathcal{L}_\infty$-optimal 2-bucket histogram for the same data set approximates it as $\{4, 4, 8, 8\}$ with $\mathcal{L}_2 = \sqrt{34}$ and $\mathcal{L}_\infty = 4$. An $\mathcal{L}_1$-optimal 2-bucket histogram is $\{5, 5, 5, 4\}$ with $\mathcal{L}_1 = 9$. The observed difference of approximation quality between Haar+ synopses and other techniques can be generalized to any gap using multiplication and to any data set size. This simple example demonstrates that both the classical Haar synopsis model and piecewise-constant histogram techniques may not achieve high accuracy of approximation in relation to the Haar+ structure. This observation illustrates the powerfulness and versatility of the Haar+ tree at producing synopses of high quality under diverse error metrics. We emphasize the following points:

- The classical Haar structure is a special case of the generalized Haar+ structure. It follows that a Haar+ synopsis is always at least as good as the equivalent Haar-wavelet synopsis.

- The storage of coefficient indexes in a Haar+ synopsis does not impose a storage burden compared to a classical Haar wavelet synopsis or a histogram. A Haar+ triad index corresponds to a classical Haar coefficient index. Hence, a convenient storage scheme is to keep the retained coefficient values in three distinct groups, one for each coefficient type (head, left, and right supplementary), each with its triad index value. A synopsis of n data items requires at most n distinct triad index values in each tree, hence $\lceil \log n \rceil$ bits per index. The same applies to the indexes in a classical Haar wavelet synopsis and the bucket boundaries in a histogram.

3.1 Basic Properties

A Haar$^+$ tree is a sparse vector $\mathbf{H}$ of $N = 3 \times 2^d - 2$ elements (coefficients) $\{c_0, c_1, \ldots, c_{3\times(2^d-1)}\}$, arranged in a tree, that represents a data vector $\mathbf{D}$ of 2^d elements $\{d_0, d_1, \ldots, d_{2^d-1}\}$. The data items reside on the leaf nodes of the tree. In the following, we use the notation $a = \mathsf{P}(b)$ to denote that coefficient a resides on the parent node of coefficient b in the tree, $a \in \mathsf{Rleaves}(b)$ to denote that data item (leaf) a lies in the right sub-tree of node b, and $a \in \mathtt{path}(b)$ to denote that node a lies on the path from the root of the tree to leaf node b. The tree structure is arranged so that:

$$\begin{aligned}
&c_0 = \mathsf{P}(c_1).\\
&i-1 \bmod 3 = 0 \Rightarrow c_i = \mathsf{P}(c_{i+1}) \wedge c_i = \mathsf{P}(c_{i+2}).\\
&i-2 \bmod 3 = 0 \Rightarrow c_i = \mathsf{P}(c_{2i}).\\
&i \bmod 3 = 0 \Rightarrow c_i = \mathsf{P}(c_{2i+1}).
\end{aligned}$$

A data item d_j of the represented data vector $\mathbf{D}$ has a parent node c_i in the tree, such that $i = (j + N) \setminus 2$. This data item is constructed as $d_j = \sum_{i\in\mathtt{path}(j)} \delta_{ij} c_i$, where

$$\delta_{ij} = \begin{cases} -1, (i-1 \bmod 3 = 0) \wedge (d_j \in \mathsf{Rleaves}(c_i)) \\ +1, \qquad\qquad \text{otherwise} \end{cases}$$

We introduce some convenient notation for the discussion that follows. The *state* of a given triad (c_i, c_{i+1}, c_{i+2}) is a four-element vector $[v, a, b, c]$, where $v = \sum_{k\in\mathtt{path}(i)} \delta_{ki} c_k$ is the reconstructed value from the root of the tree up to the node where c_i resides, henceforward called *incoming value* at c_i, and a, b, c are the values at c_i, c_{i+1} and c_{i+2} respectively. We say that this state *produces* the *contribution vector* $[v+a+b, v-a+c]$, meaning that $v + a + b$ is the incoming value at node c_{2i+2} (child of c_{i+1}) and $v - a + c$ is the incoming value at node c_{2i+5} (child of c_{i+2}). $\|\mathbf{H}\|$ denotes the number of non-zero values in a Haar$^+$ tree $\mathbf{H}$. The following theorem shows the redundancy of a dense Haar$^+$ tree representation.

Theorem 1 *Let $\mathbf{H}$ be an arbitrary Haar$^+$ tree that produces the data vector $\mathbf{D}$, in which at least one triad contains more than one non-zero value. Then $\mathbf{D}$ can be represented by an at least equally sparse Haar$^+$ tree $\mathbf{H}'$, such that every triad $C \in \mathbf{H}'$ contains at most one non-zero value and $\|\mathbf{H}'\| \leq \|\mathbf{H}\|$.*

Proof. Consider any triad C in $\mathbf{H}$ that contains more than one non-zero values. Let the state of C be $C = [v, a, b, c]$. This can be equivalently expressed as $C = [v, 0, b{+}a, c{-}a]$, since both these states produce the same contribution vector $[v + b + a, v + c - a]$. Hence, any assignment of more than one non-zero values in a given triad C can be reduced to the assignment of two non-zero values, one on each supplementary coefficient, bringing C to the state $[v, 0, q, r]$, which produces the contribution vector $[v + q, v + r]$. However, this contribution vector is also produced by a triad in the state $[v + \frac{q+r}{2}, \frac{q-r}{2}, 0, 0]$. Hence a triad with two non-zero coefficients in the state $[v, 0, q, r]$ can be reduced to a triad with one non-zero coefficient only by changing its *incoming* value from v to $v + \frac{q+r}{2}$, as follows:

1. If the parent node of C is the root node, then we need to add the value $\frac{q+r}{2}$ to the root coefficient.
2. If the parent triad of C is another triad Q, then we need to add the value $\frac{q+r}{2}$ to the supplementary coefficient in Q which is the parent node of C. If this addition results in more than one non-zero values in Q, then we proceed to reduce Q to a triad with one non-zero value only, as above.

This process leads from any given triad upwards in the tree, hence it terminates in all cases once the root node is reached. Moreover, each step in this process may decrease, but not increase, the amount of non-zero values in the tree as a whole. Hence, it follows that any Haar$^+$ tree $\mathbf{H}$ can be reduced to an at least equally sparse Haar$^+$ tree $\mathbf{H}'$, such that every triad $C \in \mathbf{H}'$ contains at most one non-zero value and $\|\mathbf{H}'\| \leq \|\mathbf{H}\|$. ∎

Figure 2 depicts the basic transformation of a triad with two non-zero supplementary coefficients q, r to one with only one non-zero head coefficient.

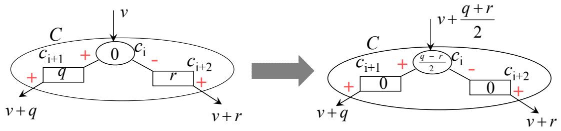


Figure 2. Basic transformation of triad

The following corollary follows from Theorem 1.

Corollary 1 *The optimal B-term Haar$^+$ tree representation $\mathbf{H}$ of a data vector $\mathbf{D}$ that minimizes a given error measure $\mathcal{E}$ can be expressed as a Haar$^+$ tree with at most one non-zero value in each triad.*

Based on Corollary 1, we proceed to construct a dynamic programming approximation scheme for the optimal Haar$^+$ tree representation of a data vector $\mathbf{D}$.

4 Haar$^+$ Synopses for Distributive Error Metrics

Before we proceed, we provide the following definition.

Definition 1 *Consider a data vector $\mathbf{D}$, an approximation thereof $\hat{\mathbf{D}}$, and the function of an error metric $\mathcal{E}$, $f_{\mathcal{E}}$,* *such that $f_{\mathcal{E}}(\|\hat{\mathbf{D}} - \mathbf{D}\|_{R_i})$ denotes the error in the data-value approximation over the range R_i in both $\mathbf{D}$ and $\hat{\mathbf{D}}$. The error metric $\mathcal{E}$ is* distributive *if and only if there exists a two-variable function G, such that the error of any range R_i divided into two disjoint ranges R_j and R_k, $R_i = R_j \cup R_k$ can be expressed as:*

$$f_{\mathcal{E}}(\|\hat{\mathbf{D}} - \mathbf{D}\|_{R_i}) = G\left(f_{\mathcal{E}}(\|\hat{\mathbf{D}} - \mathbf{D}\|_{R_j}), f_{\mathcal{E}}(\|\hat{\mathbf{D}} - \mathbf{D}\|_{R_k})\right)$$

In addition, the error metric $\mathcal{E}$ is monotonic *if and only if the error function $f_{\mathcal{E}}$ is a non-decreasing function of each individual value's absolute error $|\hat{d}_i - d_i|$.*

We now define the error minimization problem:

Problem *Given a data vector $\mathbf{D}$ and a monotonic distributive error metric $\mathcal{E}$, construct a B-non-zero-term Haar$^+$ tree representation $\mathbf{H}$ of $\mathbf{D}$ that produces a approximation $\hat{\mathbf{D}}$ of minimal error $f_{\mathcal{E}}(\|\mathbf{D} - \hat{\mathbf{D}}\|)$.*

In order to solve this problem, we have to determine the optimal positions and values of the B non-zero terms we can keep. Since each triad C_i needs to contain at most one non-zero value, four options are available: either no value is kept, or a value is kept at one of the three positions in the triad. We formalize our solution in the following section.

4.1 Formalizing the Solution

Let $Q(i, v, b)$ express the optimal choice to be made on triad C_i with incoming value v and allocated space b to be used by the triad and its descendants. We can establish the solution in a bottom-up process, by calculating $Q(i, v, b)$ on each triad, for each possible incoming value v and each allocated space b. Let l_i denote the layer of triads in which C_i resides, counting from the bottom; then at most $2^{l_i} - 1$ non-zero values can be used by triad C_i and its descendants; therefore the domain of b is $D_i = \{0, 1, \dots, \min\{B, 2^{l_i} - 1\}\}$. In order to delimit the domain of v, we quantize it into multiples of a resolution step δ. Furthermore, we need to set lower and upper bounds for this domain. For a *maximum error* metric, we can calculate a crude upper bound for the optimal error and contain the domain of v with it, based on the fact that each individual data value cannot exceed it. On the other hand, for general, *monotonic distributive* error metrics, a different approach is called for. Fortunately, the Haar$^+$ structure allows us to delimit the domain of the values we have to search for. In the following discussion, we use the notations described in Table 1. We build the delimitation starting out from the following proposition.

Proposition 1 *For incoming value v at C_i, there exist reconstructed values $\hat{d}_k$ and $\hat{d}_l$ such that $\hat{d}_k \leq v$ and $\hat{d}_l \geq v$.*

Proposition 1 finds application in the following.

Proposition 2 *If C_i has a non-zero head coefficient z_h, then the incoming value v at C_i lies in (m_i, M_i). Symbolically, $z_h \neq 0 \Rightarrow v \in (m_i, M_i)$. In reverse, $v \notin (m_i, M_i) \Rightarrow z_h = 0$.*

symbol	meaning
$\mathbf{D}$	Summarized data vector
$\mathbf{H}$	Optimized Haar$^+$ representation of $\mathbf{D}$
C_i	Triad in $\mathbf{H}$
v	Incoming value to C_i
z_h	Value assigned to head coefficient of C_i
z_l (z_r)	Value assigned to left (right) supplementary coeff. of C_i
z_0	Value assigned to root coefficient of $\mathbf{H}$
m_i (M_i)	Minimum (maximum) data value under scope of C_i
m_l (m_r)	Minimum data value in left (right) sub-tree of C_i
M_l (M_r)	Maximum data value in left (right) sub-tree of C_i
m (M)	Global minimum (maximum) in $\mathbf{D}$

Table 1. Notation Used

Proof. Assume that $v \notin (m_i, M_i)$ and $z_h \neq 0$. Then, according to Corollary 1, both supplementary coefficients are zero, hence C_i produces the contribution vector $[v + z_h, v - z_h]$. Without loss of generality, assume that $v \geq M_i$ and $z_h > 0$. Then the incoming value $v - z_h$ to the right sub-tree of C_i may lead to a good approximation of the values therein by decreasing v. However, the incoming value $v + z_h$ to the left sub-tree of C_i does not gain in approximation quality by increasing v, since v is already larger than the maximum value M_i to be approximated, and, according to Proposition 1, there exists a reconstructed value $\hat{d}_l \geq v$. Hence the error metric $\mathcal{E}$, being monotonic, is not increased by setting $z_h = 0$ and assigning the non-zero value $z_r = -z_h$ to the right supplementary coefficient of C_i alone. Similar reasoning applies to the other cases. Hence the assignment of a non-zero value z_h to the head coefficient of C_i is unnecessary when $v \notin (m_i, M_i)$. ∎

We proceed to delimit the values that may be thus assigned, after we introduce the following proposition which follows from Proposition 2.

Proposition 3 *An incoming value $v < m_i$ ($v > M_i$) at C_i cannot result into better approximation quality, according to any monotonic error metric, than a value v' such that $v < v' \leq m_i$ ($v > v' \geq M_i$), for any synopsis with the same number of non-zero terms in the sub-tree rooted at C_i.*

Proof. Assume $v < m_i$. According to Proposition 2, the first non-zero coefficient encountered on any subsequent reconstruction path can be a supplementary coefficient without affecting the quality of approximation. However, a supplementary coefficient acting on v' can produce the same outcome as when acting on v, rendering the solution equivalent on those paths. On the other hand, in subsequent reconstruction paths where a non-zero coefficient is not encountered, v' has a default quality advantage over v, since it has smaller absolute difference from every data value under the scope of C_i. Hence, for any monotonic error metric, incoming value v' leads to at least as good approximation of all data values under the scope of C_i as v, where $v < v' \leq m_i$. Analogous reasoning applies to the case the $v > M_i$. ∎

We are now ready to delimit the assigned value of a head coefficient with the following theorem.

Theorem 2 *Let m_i be the minimum and M_i the maximum individual data value under the scope of triad C_i and $v \in (m_i, M_i)$ be the incoming value at C_i in $\mathbf{H}$. If a non-zero value z_h is assigned to the head coefficient in C_i, then $|z_h| \leq \max\{M_i - v, v - m_i\}$*

Proof. Since $z_h \neq 0$, C_i advances the contribution vector $[v + z_h, v - z_h]$ towards its two sub-trees. Without loss of generality, assume that $z_h > 0$ and $v + z_h > M_i$, $v - z_h < m_i$. Then, according to Proposition 3, the approximation quality on both sub-trees can be bettered by decreasing z_h so that at least one of the two produced incoming values $v + z_h$, $v - z_h$ reaches the extremum M_i or m_i, respectively. Similar reasoning applies when $z_h < 0$. Hence, under any monotonic error metric, the value of z_h should place at least one of the two produced incoming values $v + z_h$, $v - z_h$ inside the interval $[m_i, M_i]$:

$$\begin{gathered} m_i \leq v + z_h \leq M_i \vee m_i \leq v - z_h \leq M_i \Leftrightarrow \\ m_i - v \leq z_h \leq M_i - v \vee v - M_i \leq z_h \leq v - m_i \Leftrightarrow \\ z_h \in [\min\{v - M_i, m_i - v\}, \max\{M_i - v, v - m_i\}] \Leftrightarrow \\ |z_h| \leq \max\{M_i - v, v - m_i\} \end{gathered}$$

∎

Reasoning analogous to that of Theorem 2 leads to the following theorem.

Theorem 3 *Let m_l (m_r) be the minimum and M_l (M_r) the maximum individual data value under the scope of the left (right) sub-tree of triad C_i. If a non-zero value z_l (z_r) is assigned to the left (right) supplementary coefficient in C_i, then $z_l \in [m_l - v, M_l - v]$ ($z_r \in [m_r - v, M_r - v]$). Likewise, if a non-zero value z_0 is assigned to the root coefficient, then $z_0 \in [m, M]$, where m (M) is the global minimum (maximum) in $\mathbf{D}$.*

We now proceed to delimit the candidate incoming values for the rest of the triads in terms of these global extrema.

Theorem 4 *The incoming value v to C_i in $\mathbf{H}$ lies within the interval $(m - \Delta, M + \Delta)$, where $\Delta = M - m$.*

Proof. For an incoming value derived from an ancestor non-zero supplementary coefficient, or from the root coefficient, the proof follows directly from Theorem 3. We examine incoming values derived from an ancestor non-zero head coefficient. Consider a non-zero head coefficient z_h encountered at a triad C_k. Then, according to Proposition 2, the incoming value v at C_k lies within the interval (m, M). Besides, according to Theorem 2, $|z_h| \leq \max\{M - v, v - m\}$. Joining the delimitations of v and z_h we get $v \pm z_h \in (2m - M, 2M - m)$. Hence, in both cases, the produced incoming value lies in $(m - \Delta, M + \Delta)$. ∎

The intuition behind Theorem 4 is that, in the worst case, a non-zero head coefficient covers the difference $M - m$ between the two extrema in one direction and replicates this difference in the other. In conclusion, the range of potential incoming values has width 3Δ. Let $\mathcal{S}$ denote the set of such values in $(2m - M, 2M - m)$ that are multiples of the resolution step δ. Then $|\mathcal{S}| \leq \lfloor \frac{3\Delta}{\delta} \rfloor + 1 = O(\frac{\Delta}{\delta})$. [3] Furthermore, let $\mathcal{S}^v_{i,H} \subset \Re$, $\mathcal{S}^v_{i,L} \subset \Re$, $\mathcal{S}^v_{i,R} \subset \Re$ denote the set of potential assigned values at the head, left and right supplementary coefficient of triad C_i that are multiples of δ, for a given incoming value v. Then, according to Theorems 2 and 3, the cardinality of these sets is also $O(\frac{\Delta}{\delta})$.

4.2 Deriving the Answer

The derivation of the optimal error result and the respective B-non-zero-term Haar$^+$ tree representation $\mathbf{H}$ of $\mathbf{D}$ does not pose a novel algorithmic problem. As in previous synopsis construction algorithms [23, 6, 7, 17, 8, 13, 29], a dynamic programming solution can be applied. In particular, our algorithm draws from the unrestricted Haar wavelet synopsis construction algorithm of [13]. We compute the fundamental $Q(i, v, b)$ function with a dynamic programming recursive scheme; however, further elaboration is required at the decision-making process in each triad, due to the multiplicity of options. We also employ the generic space-efficiency paradigm of [12], and analyze the emerging trade-off between time- and space-efficiency.

In a nutshell, the method works in a bottom-up left-to-right scan over the Haar$^+$ tree. At each visited triad C_i it calculates an array A from the pre-calculated arrays L and R of its children triads C_{i_l}, C_{i_r}. The entry $A[v, b]$ corresponds to $Q(i, v, b)$, for the pair of incoming value v *and* space b allocated to the sub-tree rooted at C_i. Such an entry contains: (i) the δ-optimal value z_h, z_l, or z_r to assign at one of the coefficients in C_i, if any; (ii) the amount of space b_L out of b to allocate to the left branch; and (iii) the minimum error $E(i, v, b)$ thus achieved. The size of A is $|\mathcal{S}_i| \cdot |D_i|$. A recursive procedure $\mathsf{MinError}$ emerges, in which the value of $E(i, v, b)$ is computed as:

$$E(0, 0, B) = \min_{z \in \mathcal{S}^0_{0,H}} \{E(1, z, B - (z \neq 0))\}$$

$$E(i, v, b) =$$

$$\min \left\{ \begin{array}{l} \displaystyle\min_{z_h \in \mathcal{S}^v_{i,H}, b' \in D_i} \left\{ \begin{array}{l} E(i_l, v + z_h, b') + \\ E(i_r, v - z_h, b - b' - (z_h \neq 0)) \end{array} \right. \\ \displaystyle\min_{z_l \in \mathcal{S}^v_{i,L}, b' \in D_i} \left\{ \begin{array}{l} E(i_l, v + z_l, b')) + \\ E(i_r, v, b - b' - (z_l \neq 0)) \end{array} \right. \\ \displaystyle\min_{z_r \in \mathcal{S}^v_{i,R}, b' \in D_i} \left\{ \begin{array}{l} E(i_l, v, b')) + \\ E(i_r, v + z_r, b - b' - (z_r \neq 0)) \end{array} \right. \end{array} \right.$$

Addition is used for the sake of simplicity; any distributive error function G can be applied. The latter equation computes the least of three minima, one for each coefficient in C_i. Each of those is the least achievable error, in the subtree rooted at C_i, among all allowed combinations of a value

[3] The inequality $\leq$ accommodates for the variation in the number of integers in a fixed interval.

assigned to the examined coefficient [4] and a distribution of the available space to the branches of that sub-tree. For the economy of presentation the -1 term, which decreases the space assigned to the right branch in case a non-zero value is assigned, is uniformly expressed by the boolean integer $(z_x \neq 0)$. The computed error value is assigned to $A[v,b].e$, while the values of $A[v,b].z_h$, $A[v,b].z_l$, $A[v,b].z_r$ are the coefficient triple (at most one non-zero) that minimizes the expression above. For a last level node, there is no need to scan through the sets of allowed assigned values; the optimal value to assign to each coefficient is directly determined by the incoming value and the data values below.

Following the generic space-efficiency paradigm of [12], for a data set of size n, the maximum number of arrays that need to be concurrently stored is $\log n + 1$: one array per internal triad layer plus the currently used triplet. This maximum is necessitated when the right-bound post-order recursion reaches the right-most triad. Hence an algorithm that derives the minimum error result without constructing the synopsis itself is defined.

Complexity Analysis The result arrays L, R on each node i hold one entry for each possible incoming value in $|\mathcal{S}|$, hence their size is $O(\frac{\Delta}{\delta}\min\{B, 2^{l_i}-1\})$; besides, at each triad C_i and for each $[v,b]$ pair, checking all pairs of an assigned value in $|\mathcal{S}^v_{i,H}|$, $|\mathcal{S}^v_{i,L}|$, or $|\mathcal{S}^v_{i,R}|$ and an amount of space in D_i takes $O(\frac{\Delta}{\delta}\min\{B, 2^{l_i}-1\})$ time. Hence, the worst-case running time of $\mathsf{MinError}$ is $O\left((\frac{\Delta}{\delta})^2 \sum_{i=1}^{n} \min\{B, 2^{l_i}-1\}^2\right) = O\left((\frac{\Delta}{\delta})^2 nB\right)$. For the special case of a maximum-error metric, the B factor becomes $\log^2 B$, thanks to the application of binary search in the procedure that searches through value assignments; this method is used in [8, 12, 13, 14]. Besides, since at most $\log n + 1$ arrays need to be concurrently stored, the space complexity is $O\left(\frac{\Delta}{\delta}\sum_{l=1}^{\log n+1}\min\{B, 2^{l_i}-1\}\right) = O\left(\frac{\Delta}{\delta}B\log\frac{n}{B}\right)$. [5]

For a resolution step δ, the following theorem provides a guarantee of approximation in relation to the optimal solution in $\Re$ for normalized Minkowski-distance error metrics.

Theorem 5 *Consider a data set $\mathbf{D}$ of size n optimally summarized in B terms by a Haar⁺ representation $\mathbf{H}^*$ in $\Re$ and by the representation $\mathbf{H}_\delta$ in the domain of multiples of δ, with the normalized Minkowski-distance $\mathcal{L}_p$ error as target, deriving the error values $\mathcal{E}^*$ and $\mathcal{E}_\delta$, respectively. Then $\mathcal{E}_\delta \leq \mathcal{E}^* + \frac{\delta}{2}\min\{B, \log n\}$.*

Proof. Let $\mathbf{D}^*$ denote the approximation of $\mathbf{D}$ produced by $\mathbf{H}^*$; let $\hat{\mathbf{H}}_\delta$ be the representation of $\mathbf{D}$ derived after rounding all coefficients in $\mathbf{H}^*$ to the nearest multiple of δ, $\mathcal{E}'_\delta$ be its $\mathcal{L}_p$ error and $\hat{\mathbf{D}}$ the approximation it produces. Since $\mathbf{H}_\delta$ is the $\mathcal{L}_p$-optimal δ-step representation, it follows that $\mathcal{E}_\delta \leq \mathcal{E}'_\delta$. However, according to the triangle inequality, $\mathcal{E}'_\delta \leq \mathcal{E}^* + \mathcal{L}_p(\mathbf{D}^*, \hat{\mathbf{D}})$. Each approximated data value is the sum of at most $\min\{B, \log n\}$ coefficients (at most one per Haar⁺ triad layer) and each coefficient in $\hat{\mathbf{H}}_\delta$ has been rounded from its value in $\mathbf{H}^*$ by at most $\frac{\delta}{2}$, hence $\mathcal{L}_\infty(\mathbf{D}^*, \hat{\mathbf{D}}) \leq \frac{\delta}{2}\min\{B, \log n\}$. From the definition of the normalized Minkowski-distance norm it follows that $\mathcal{L}_p(\mathbf{D}^*, \hat{\mathbf{D}}) \leq \mathcal{L}_\infty(\mathbf{D}^*, \hat{\mathbf{D}})$. Putting it all together, we get $\mathcal{E}_\delta \leq \mathcal{E}^* + \frac{\delta}{2}\min\{B, \log n\}$. ∎

4.3 Constructing the Synopsis

The construction of the actual synopsis after that optimal error result has been established presents us with a time-space trade-off. We present both variants.

4.3.1 The Space-Efficient Solution

After we have determined the solution at the topmost level we can call a process that reenters the problem in the two branches of C_1 and recomputes the respective solutions for its descendants, recursively. Then the total running time is the sum of the basic running time for all re-entered subproblems. Setting l as the Haar tree level, this sum becomes $O\left((\frac{\Delta}{\delta})^2 B \sum_{l=0}^{\log n} 2^l \frac{n}{2^l}\right) = O\left((\frac{\Delta}{\delta})^2 nB\log n\right)$. On the other hand, the space becomes $O\left(\frac{\Delta}{\delta}B\log\frac{n}{B} + n\right)$, where n stands for the necessary storage of the data set.

4.3.2 The Time-Efficient Solution

Alternatively, we may maintain all computed solutions throughout the computation, keeping the time at $O\left((\frac{\Delta}{\delta})^2 nB\right)$. As far as the space is concerned, we can follow two different approaches:

- We may keep all DP arrays in memory. The size of the array at triad layer l_i is $O\left(\frac{\Delta}{\delta}\min\{B, 2^{l_i}\}\right)$. The second factor sums up to $\sum_i \min\{B, 2^{l_i}\} = \sum_l 2^{\log n - l}\min\{B, 2^l\} = n\log B$. Hence the space complexity in this case is $O\left(\frac{\Delta}{\delta}n\log B\right)$.

- As suggested in [14] for the Haar synopsis problem, we may attach a list with all retained coefficient values in the corresponding solution to each entry $A[v]$ of a DP array at triad C_i. Again, at most $\log n + 1$ arrays are concurrently stored. An array at level l_i maintains lists of size at most $\min\{B, 2^{l_i}\}$ attached on its entries, hence requires $O\left(\frac{\Delta}{\delta}(\min\{B, 2^{l_i}\})^2\right)$ space. The squared factor, summed over all levels, gives $B^2\log\frac{n}{B}$, hence the space complexity in this case becomes $O\left(\frac{\Delta}{\delta}B^2\log\frac{n}{B}\right)$.

The two space complexity expressions are equalized when $n\log B = B^2\log(n/B) \Leftrightarrow B = \sqrt{n}$. For $B < \sqrt{n}$,

[4] The head coefficient is examined only when $v \in (m_i, M_i)$.

[5] A $1 + \log\frac{n}{B}$ factor is simplified to $\log\frac{n}{B}$ under the assumption that $B < \frac{n}{2}$. Besides, in applications where a distinction between *total space* and *working space* complexity is meaningful, as in [8], we need only keep three arrays in the main memory at any time, hence the working space complexity is $O\left(\frac{\Delta}{\delta}B\right)$.

one should opt for annexing the solutions. On the other hand, for $B > \sqrt{n}$, storing the DP arrays per se is favorable. The latter case is likely to apply in practice, especially for large values of n. This time-efficient solution enables the operation of the algorithm in one pass over the data.

4.4 Theoretical Comparison to Other Synopsis Construction Algorithms

The time complexity of the histogram and wavelet synopsis construction algorithms reviewed in Section 2 remains in all cases super-linear to the size of the data set, for generic *distributive* error metrics (although linear or near-linear time versions exist for Euclidean and maximum-error metrics). Table 2 summarizes this complexity terrain, under the $\mathcal{L}_1$ metric, for the state-of-the art, in terms of quality, histogram and Haar wavelet techniques, as well as for the Haar+ method that we introduce; n is the data set size, B the space bound, δ the resolution step, M the maximum absolute value in the data set, $\mathcal{E}$ an upper bound for the optimal $\mathcal{L}_1$ error, and Δ the difference of the minimum from the maximum value in the data set. The fractions $\frac{nM}{\delta}$, $\frac{n\mathcal{E}}{\delta}$ and $\frac{\Delta}{\delta}$ express the cardinality of the examined set of incoming or assigned values for the unrestricted Haar wavelet and Haar+ models, respectively. This set is bounded by an upper bound for the non-normalized $\mathcal{L}_1$ error in [14]; this error measure grows with the size of the summarized data set n, since it is equal to $n\mathcal{E}$, where $\mathcal{E}$ is the $O(1)$ *normalized* $\mathcal{L}_1$ error. As seen in the table, while previous quality-aware techniques are burdened by time complexity at least quadratic to n for this metric, the Haar+ structure allows for synopsis construction in time linear to n. This Haar+ complexity holds for any monotonic distributive error metric. Moreover, Haar+ achieves by theory at least as high (in practice higher) synopsis quality as Haar wavelet techniques, and outperforms histogram techniques too at the task of approximating data sets with sharp discontinuities. We now demonstrate these advantages in practice.

Technique	Time Complexity ($\mathcal{L}_1$)	Reference
Histograms	$O(n^2(B + \log n))$	[23, 17]
Restricted Haar	$O(n^2 \log B)$	[8, 12]
Unrestricted Haar	$O((\frac{M}{\delta})^2 n^3 B)$	[13]
Unrestricted Haar	$O((\frac{\mathcal{E}}{\delta})^2 n^3 B)$	[14]
Haar+	$O((\frac{\Delta}{\delta})^2 nB)$	**This work**

Table 2. Summary of results for quality-aware synopsis construction ($\mathcal{L}_1$ metric)

5 Experimental Comparison of Synopsis Data Structures

In this section we present our experimental results pertaining to the runtime for, and the approximation quality achieved with, Haar+ synopses. We compare the results to those achieved with alternative synopsis construction techniques. Specifically, we have performed comparison with the following algorithms:

- **HIST** The optimal histogram construction algorithm of [23, 17]. This algorithm provides an upper bound to the quality of any approximate histogram construction technique [11, 16, 34].

- **R-Haar** The optimal restricted Haar wavelet synopsis algorithm of [8, 12].

- **U-Haar** The approximation scheme for unrestricted Haar wavelet synopses of [13, 14]. Our implementation of this scheme follows the model of [14], where the examined values are bounded by an upper bound for the final non-normalized Minkowski-norm error. It first calculates, in $O(n \log B)$ time, the targeted non-normalized error metric value $\hat{\mathcal{E}}$ for the synopsis consisting of the B largest Haar terms of $\mathbf{D}$ by absolute value; then it employs it for bounding the search space. In terms of growth, $\hat{\mathcal{E}} = n^{\frac{1}{p}}\mathcal{E} = O(n^{\frac{1}{p}})$, where $\mathcal{E}$ is the $O(1)$ *normalized* Minkowski-norm error.

- **Haar+** The Haar+ synopsis algorithm presented in Section 4.

All algorithms were implemented using the g++ 3.4.3 compiler, while the experiments were run on a 4 CPU Opteron 2.2GHz machine with 4GB of main memory running Solaris.

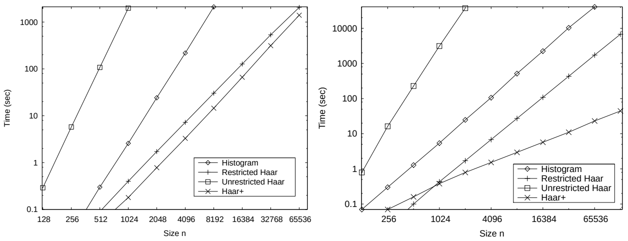
Description of Data We have focused on two real-life data sets with hard to approximate bursts and discontinuities for our quality assessment, as well as an easier to approximate larger real life data set for our runtime assessment. The first data set (FR), discussed in [28], is a sequence of the mean monthly flows for the Fraser River at Hope, B.C.[6] The flows present periodic autoregression features, while they average at 2709 (standard deviation: 2123) and feature discontinuities (min value: 482, max value: 10800). The second data set (FC) is extracted from a relation of 581,012 tuples describing the forest cover type for 30 x 30 meter cells, obtained from US Forest Service. FC contains the frequencies of the distinct values of attribute `aspect` in the relation. The frequencies average at 1613 (standard deviation: 730) and feature spikes of large values (min value: 499, max value: 6308). The third dataset (TM) is a sequence of 178,080 sea surface temperature measures extracted from drifting buoys positioned throughout the equatorial Pacific. The average value in TM is 26.75 and the set has a small standard deviation (1.91). Both FC and TM were downloaded from the UCI KDD Archive [7].

[6] Available at http://lib.stat.cmu.edu/datasets/fraser-river

[7] Available at http://kdd.ics.uci.edu/

5.1 Running Time

In this experiment we evaluate the run-time performance of the four synopsis construction algorithms at the task of minimizing a distributive error metric. In particular, we tried all algorithms on different-sized prefixes of the TM data set with the computationally challenging average absolute error $\mathcal{L}_1$ (equivalently, the sum of absolute errors) as the target metric. In order to facilitate our measurements we opted for a large constant resolution value $\delta = 1$ with both the U-Haar and Haar$^+$ algorithms. For all four algorithms, we measured the time required to derive the error result in two different settings: One in which the space budget B grows along with the data set size n, such that $B = \frac{n}{64}$, and one in which B remains at the constant value $B = 32$ while n grows. Figure 3 plots the results for both settings on logarithmic axes. As expected, the Haar$^+$ algorithm presents the most affordable runtime growth of all. The advantage is particularly striking at the constant B setting, where it is the only algorithm that behaves linearly. When B grows with n it exhibits quadratic behavior, paralleled by R-Haar, whom it exceeds in synopsis quality. On the other hand, the growth of HIST is cubic when B grows with n and quadratic for constant B. Finally, the nature of U-Haar is that of a fourth-power growth when B grows with n and cubic for constant B. It is so because its runtime depends quadratically to the search space bound, which grows linearly with n.

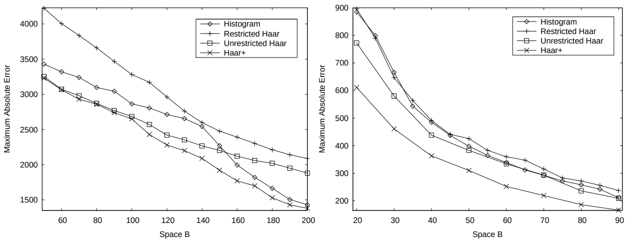


(a) $B = \frac{n}{64}$ (b) $B = 32$

Figure 3. Runtime comparison: $\mathcal{L}_1$

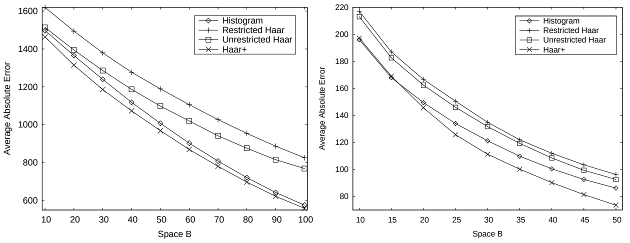
5.2 Synopsis Quality

We present quality results for two representative error metrics at the opposite ends of the Minkowski spectrum: the maximum absolute error $\mathcal{L}_\infty$ and the average absolute error $\mathcal{L}_1$, on the the FR and FC data sets. Results with other metrics, such as $\mathcal{L}_2$, were similar. Figure 4 shows the error results for the $\mathcal{L}_\infty$ metric, in which the resolution value has been set at $\delta = 50$ for the FR and $\delta = 10$ for the FC data set with both U-Haar and Haar$^+$. The Haar$^+$ technique achieves the highest quality for both data sets. U-Haar and HIST outbalance each other for the second best quality, while R-Haar does not achieve high accuracy due to its restricted nature. In order to render the histogram-based summarization directly comparable to the binary-interval-based Haar wavelet-derived techniques in our experiments with the $\mathcal{L}_1$ metric, we have used a 512-value prefix of the FR data set and a 256-value prefix of the FC data set. Figure 5 shows the error results, in which, again, the resolution value has been set at $\delta = 50$ for the FR synopses and $\delta = 10$ for the FC data set with both U-Haar and Haar$^+$. U-Haar is outperformed by HIST with both datasets, while it needed inconveniently long time, in particular for summarizing the larger FR data set, due to its high time complexity for this error metric. However, the Haar$^+$ technique outperforms HIST for this error metric too, achieving the highest quality for both data sets also in this experiment. Finally, R-Haar does not achieve high quality with this error metric either.



(a) FR (b) FC

Figure 4. Quality comparison: $\mathcal{L}_\infty$



(a) FR (b) FC

Figure 5. Quality comparison: $\mathcal{L}_1$

6 Discussion

Our results on the superior quality of Haar$^+$ synopses in relation to classical Haar techniques were expected. A Haar$^+$ synopsis is always at least as good as the equivalent Haar synopsis. Still, we have demonstrated that Haar$^+$ can achieve higher accuracy than an optimal histogram too. On the other hand, we have witnessed that histograms can do better than wavelet-derived schemes at approximating data sets *without* sharp discontinuities. Moreover, as we saw in the previous section, they are advantaged in relation to the unrestricted Haar method when the target error metric is the *average* error, which introduces a smoothing factor as well. These findings have interest of their own, as a comparison between the *provably optimal* quality achieved with each of these two models was missing in previous research.[8]

[8] Such comparisons had been made only before optimal-quality methods were available for both models [27, 17].

7 Conclusions

In this paper we have introduced the Haar$^+$ tree: a novel, refined synopsis data structure, inspired from Haar wavelet techniques, eschewing their deficiencies and enhancing on their advantages. We have demonstrated that Haar$^+$ synopses of data sets with sharp discontinuities achieve higher quality than optimal histogram and Haar wavelet schemes under representative error metrics. Moreover, thanks to the capacity of the Haar$^+$ structure to delimit the search space, Haar$^+$ synopses are constructed in time linear to the size of the data for *any* monotonic distributive error metric. To the best of our knowledge, this is the first synopsis construction technique that can achieve higher quality than an optimal histogram for an additive error metric in time linear to the size of the input. Besides, Haar$^+$ synopsis construction can be performed in one pass. In conclusion, the Haar$^+$ structure provides a mostly recommendable option for the high quality and time-efficient summarization of very large discontinuous data sets with any distributive target error metric. In the future, we plan to examine how the Haar$^+$ technique can be extended to multi-dimensional data.

Acknowledgements

We thank the anonymous reviewers for their helpful remarks.

References

[1] S. Acharya, P. B. Gibbons, V. Poosala, and S. Ramaswamy. Join synopses for approximate query answering. In *SIGMOD*, 1999.

[2] R. Bellman. On the approximation of curves by line segments using dynamic programming. *Communications of the ACM*, 4(6):284, 1961.

[3] K. Chakrabarti, M. Garofalakis, R. Rastogi, and K. Shim. Approximate query processing using wavelets. *VLDB Journal (also VLDB 2000)*, 10(2-3):199–223, 2001.

[4] K. Chakrabarti, E. Keogh, S. Mehrotra, and M. Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. *TODS (also SIGMOD 2001)*, 27(2):188–228, 2002.

[5] G. Cormode, M. Garofalakis, and D. Sacharidis. Fast approximate wavelet tracking on streams. In *EDBT*, 2006.

[6] A. Deligiannakis and N. Roussopoulos. Extended wavelets for multiple measures. In *SIGMOD*, 2003.

[7] M. Garofalakis and P. B. Gibbons. Probabilistic wavelet synopses. *TODS (also SIGMOD 2002)*, 29(1):43–90, 2004.

[8] M. Garofalakis and A. Kumar. Wavelet synopses for general error metrics. *TODS (also PODS 2004)*, 30(4), 2005.

[9] P. B. Gibbons and Y. Matias. Synopsis data structures for massive data sets. In *SODA*, 1999.

[10] P. B. Gibbons, Y. Matias, and V. Poosala. Fast incremental maintenance of approximate histograms. *TODS (also VLDB 1997)*, 27(3):261–298, 2002.

[11] A. C. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss. Fast, small-space algorithms for approximate histogram maintenance. In *ACM STOC*, 2002.

[12] S. Guha. Space efficiency in synopsis construction algorithms. In *VLDB*, 2005.

[13] S. Guha and B. Harb. Wavelet synopsis for data streams: minimizing non-euclidean error. In *SIGKDD*, 2005.

[14] S. Guha and B. Harb. Approximation algorithms for wavelet transform coding of data streams. In *SODA*, 2006.

[15] S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss. Histogramming data streams with fast per-item processing. In *ICALP*, 2002.

[16] S. Guha, N. Koudas, and K. Shim. Approximation and streaming algorithms for histogram construction problems. *TODS (also ACM STOC 2001)*, 31(1), 2006.

[17] S. Guha, K. Shim, and J. Woo. REHIST: Relative error histogram construction algorithms. In *VLDB*, 2004.

[18] A. Haar. Zur theorie der orthogonalen functionsysteme. *Math. Annal.*, 69:331–371, 1910.

[19] Y. E. Ioannidis. Universality of serial histograms. In *VLDB*, 1993.

[20] Y. E. Ioannidis. The history of histograms (abridged). In *VLDB*, 2003.

[21] Y. E. Ioannidis and V. Poosala. Balancing histogram optimality and practicality for query result size estimation. In *SIGMOD*, 1995.

[22] Y. E. Ioannidis and V. Poosala. Histogram-based approximation of set-valued query-answers. In *VLDB*, 1999.

[23] H. V. Jagadish, N. Koudas, S. Muthukrishnan, V. Poosala, K. C. Sevcik, and T. Suel. Optimal histograms with quality guarantees. In *VLDB*, 1998.

[24] M. Jahangiri, D. Sacharidis, and C. Shahabi. SHIFT-SPLIT: I/O efficient maintenance of wavelet-transformed multidimensional data. In *SIGMOD*, 2005.

[25] P. Karras and N. Mamoulis. One-pass wavelet synopses for maximum-error metrics. In *VLDB*, 2005.

[26] Y. Matias and D. Urieli. Optimal workload-based weighted wavelet synopses. In *ICDT*, 2005.

[27] Y. Matias, J. S. Vitter, and M. Wang. Wavelet-based histograms for selectivity estimation. In *SIGMOD*, 1998.

[28] A. McLeod. Diagnostic checking of periodic autoregression models with application. *Journal of Time Series Analysis*, 15(2):221–233, 1994.

[29] S. Muthukrishnan. Subquadratic algorithms for workload-aware haar wavelet synopses. In *FSTTCS*, 2005.

[30] V. Poosala, V. Ganti, and Y. E. Ioannidis. Approximate query answering using histograms. *IEEE Data Eng. Bull.*, 22(4):5–14, 1999.

[31] V. Poosala and Y. E. Ioannidis. Selectivity estimation without the attribute value independence assumption. In *VLDB*, 1997.

[32] V. Poosala, Y. E. Ioannidis, P. J. Haas, and E. J. Shekita. Improved histograms for selectivity estimation of range predicates. In *SIGMOD*, 1996.

[33] E. J. Stollnitz, T. D. Derose, and D. H. Salesin. *Wavelets for computer graphics: theory and applications*. Morgan Kaufmann, 1996.

[34] E. Terzi and P. Tsaparas. Efficient algorithms for sequence segmentation. In *SIAM SDM*, 2006.

[35] J. S. Vitter and M. Wang. Approximate computation of multidimensional aggregates of sparse data using wavelets. In *SIGMOD*, 1999.