Hierarchical Synopses with Optimal Error Guarantees

PANAGIOTIS KARRAS
National University of Singapore
and
NIKOS MAMOULIS
University of Hong Kong

Hierarchical synopsis structures offer a viable alternative in terms of efficiency and flexibility in relation to traditional summarization techniques such as histograms. Previous research on such structures has mostly focused on a single model, based on the Haar wavelet decomposition. In previous work, we have introduced a more refined, wavelet-inspired hierarchical index structure for synopsis construction: the Haar$^+$ tree. The chief 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* in the size of the data set for *any* monotonic distributive error metric. Contemporaneous research has introduced another hierarchical synopsis structure, the compact hierarchical histogram (CHH). In this article, we elaborate on both these structures. First, we formally prove that the CHH, in its default binary-hierarchy form, is a simplified variant of a Haar$^+$ tree. We then focus on the summarization problem, with both these hierarchical synopsis structures, in which an error guarantee expressed by a *maximum-error* metric is required. We show that this problem is most efficiently solved through its dual, space-minimization counterpart, which can also achieve *optimal quality*. In this case, there is a benefit to be gained by specializing the algorithm for each structure; hence, our algorithm for optimal-quality maximum-error CHH requires *low polynomial* time; on the other hand, optimal-quality Haar$^+$ synopses for maximum-error metrics are constructed in exponential time; hence, we also develop a low-polynomial-time approximation scheme for the maximum-error Haar$^+$ case. Furthermore, we extend our approach for both general-error and maximum-error Haar$^+$ synopses to arbitrary dimensionality. In our experimental study, (i) we confirm the theoretically expected superiority of Haar$^+$ synopses over Haar wavelet methods in both construction time and achieved quality for representative error metrics; (ii) we demonstrate that Haar$^+$ synopses are also constructed faster than optimal plain histograms, and, moreover, achieve higher synopsis quality with highly discontinuous data sets; such an advantage of a hierarchical synopsis structure over

This work was supported by grant HKU 7155/06E from Hong Kong RGC.
Authors' addresses: P. Karras, National University of Singapore, Computing 1, Law Link, Singapore 117590; email: karras@comp.nus.edu.sg; N. Mamoulis, University of Hong Kong, Pokfulam Road, Hong Kong; email: nikos@cs.hku.hk.

a histogram had been intuitively expressed, but never experimentally verified; and (iii) we show that Haar$^+$ synopsis quality supersedes that of a CHH.

Categories and Subject Descriptors: H.2.4 [**Database Management**]: Systems—*Query processing*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms: Algorithms, Experimentation, Performance

Additional Key Words and Phrases: Summarization, data synopses, approximate query processing

ACM Reference Format:

Karras, P. and Mamoulis, N. 2008. Hierarchical synopses with optimal error guarantees. ACM Trans. Datab. Syst. 33, 3, Article 18 (August 2008), 53 pages. DOI = 10.1145/1386118.1386124 http://doi.acm.org/10.1145/1386118.1386124

1. INTRODUCTION

The need to reduce a very large data set into a compact synopsis that captures its basic characteristics arises frequently. Database applications that sustain interest in this problem include OLAP/DSS systems [Vitter et al. 1998; Vitter and Wang 1999], approximate query answering [Acharya et al. 1999; Poosala et al. 1999; Ioannidis and Poosala 1999; Chakrabarti et al. 2001], cost-based query optimization [Matias et al. 1998, 2000], and time-series mining [Chakrabarti et al. 2002]. Over the past years, two principal methods have emerged as recommendable choices for quality-aware synopsis construction: histogram-based methods [Ioannidis and Poosala 1999; Poosala et al. 1999; Gibbons et al. 2002; Jagadish et al. 1998; Guha et al. 2004], and methods based on a hierarchical index structure; such a structure has traditionally been provided by the Haar wavelet decomposition [Matias et al. 1998; Chakrabarti et al. 2001; Garofalakis and Kumar 2005; Guha and Harb 2008]. The main objective of both approaches is to minimize some appropriate error measure [Gibbons and Matias 1999], given a space budget.

Still, previous research has not attempted to examine how state-of-the-art hierarchical 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 carried out uneven comparisons, by setting provably optimal methods for the one technique against nonoptimal ones for the other [Matias et al. 1998; Guha et al. 2004], or nonoptimal methods for both [Chakrabarti et al. 2002]. Still, Graps [1995] and Guha et al. [2004] have provided some intuitive remarks about such a comparison. From a qualitative point of view, a histogram-based synopsis is arguably recommendable when summarizing smooth data sets without sharp discontinuities or bursts [Guha et al. 2004]. On the other hand, hierarchical techniques such as Haar wavelets are better suited for approximating datasets *with* such discontinuities [Graps 1995]. This disposition is due to the fact that a B-term histogram defines B consecutive distinct value intervals, with no restrictions on their relative locations and sizes; on the other hand, a B-term Haar wavelet synopsis can define B to $3B+1$ distinct consecutive value intervals [Guha et al. 2004]. Nevertheless, the Haar wavelet technique delimits the allowed bucket

boundaries to a predefined set, while the approximation values in these buckets are constrained by their interdependence. A wavelet coefficient contributes its value positively to the former and negatively to the latter of the two halves of the fixed-size interval it affects, hence the resulting summarization value in a wavelet-defined interval may be suboptimal.

Moreover, for the task of minimizing a *distributive* error metric in general (as opposed to a maximum-error metric in particular), such as the average absolute or relative error, both optimal histogram-based [Jagadish et al. 1998; Guha et al. 2004] and quality-aware wavelet-based schemes[1] [Garofalakis and Kumar 2005; Muthukrishnan 2005; Guha 2008; Guha and Harb 2008] run in time super-linear in the size of the input. The recent proposal of an alternative hierarchical summarization structure, the compact hierarchical histogram (CHH) [Reiss et al. 2006], was accompanied by exact solutions for limited versions of the problem and by heuristics for the general, longest-prefix-match CHH problem, but not by an algorithm providing deterministic error guarantees for this general problem. This state of affairs renders previous techniques inapplicable for the time-efficient summarization of very large data sets with a general error guarantee, and calls for a different approach.

In Karras and Mamoulis [2007], we have introduced the Haar$^+$ tree: a refined, wavelet-inspired synopsis data structure which offsets the aforementioned deficiencies. First, it adds flexibility to the classical Haar wavelet synopsis, while maintaining its compression advantage over a histogram; therewith it outperforms previous techniques in approximation quality with hard-to-summarize data sets. Second, it allows for easy delimitation of its search space, resulting in a synopsis construction algorithm for *general* error metrics that operates in time *linear* in the size of the data. Furthermore, Muthukrishnan [2005] and Karras et al. [2007] have shown how offline maximum-error summarization problems, whose importance has been noted by Garofalakis and Gibbons [2004], Garofalakis and Kumar [2004], and Karras and Mamoulis [2005], can be more efficiently solved via their dual, error-bounded counterparts.

In this article we add to our previous work by providing a significant insight, a major algorithmic contribution, a generalization to higher dimensionality, and further experimentation. Our insight consists of the proof that the independently proposed compact hierarchical histogram (CHH) [Reiss et al. 2006] is equivalent to a simplified variant of the Haar$^+$ structure; in consequence, our Haar$^+$ synopses algorithms also provide novel and generalized solutions to the problems studied in Reiss et al. [2006]. With the benefit of hindsight, the Haar$^+$ structure can be seen as a merging of a classical Haar tree with a CHH, superseding both these antecedents in terms of quality. Our algorithmic contribution consists of a focused application of the dual-problem-based methodology to hierarchical synopsis construction problems. We show that this *indirect* methodology can indeed provide not only benefits of efficiency and scalability, but also advantages in the areas of tractability of the problem and the accuracy of the solution, as opposed to *direct* solutions to the space-bounded

[1]The simple algorithm for Euclidean error notwithstanding [Matias et al. 1998].

problem. Hence, we exhibit the full potential of this method. In particular, we provide an algorithm that effectively solves the optimal longest-prefix-match CHH partitioning problem for maximum-error metrics in low polynomial time through this dual-problem approach, in the footsteps of Muthukrishnan [2005] and Karras et al. [2007]. To that end, we also devise a novel solution to this dual, error-bounded CHH partitioning problem, that is, the problem of constructing a minimal-space CHH synopsis that satisfies a given maximum-error bound. This solution guarantees an *optimal* solution to the space-bounded problem with any maximum-error metric as the target of optimization. We apply this approach to the general Haar$^+$ case and observe that the worst-case complexity of the algorithm becomes exponential in that case, due to the higher complexity of the Haar$^+$ structure itself in comparison to the CHH. Hence, for the sake of completeness with regard to the Haar$^+$ model, we also present and analyze an efficient approximation scheme for maximum-error Haar$^+$ synopsis construction, also following the dual-problem approach of Muthukrishnan [2005] and [Karras et al. 2007]. All our solutions provide either optimal or approximate error guarantees, hence gain a definite quality advantage over the heuristics in Reiss et al. [2006]. Lastly, we generalize our Haar$^+$ techniques to higher dimensionality; thereby also show that the benefit of the indirect approach becomes stronger as dimensionality increases.

In our experimental study, we demonstrate the superiority of Haar$^+$ synopses over other hierarchical techniques, including the recently proposed CHH, as well as their competitiveness with optimal histograms, for certain types of data, and we verify that Haar$^+$ synopses are constructed in linear time. To our knowledge, this study is the first face-to-face comparison between any pair of the state-of-the-art techniques for nonEuclidean-error-optimal synopses with plain histograms, Haar wavelets, and CHH; hence it supplements the studies in the following: [Matias et al. 1998; Chakrabarti et al. 2002; Guha et al. 2004; Garofalakis and Kumar 2005; Reiss et al. 2006; Guha and Harb 2008].

2. BACKGROUND AND RELATED WORK

Previous research has established two principal methods for the construction of data approximations with deterministic[2] quality guarantees. The former, *histograms*, is based on the creation of buckets of contiguous values that are approximated by a single representative value. The latter utilizes an appropriate *hierarchical index structure* for concise data representation. Under both approaches, given an n-size data vector $\mathbf{D} = \langle d_0, d_1, \ldots, d_{n-1} \rangle$, the *space-bounded* synopsis problem is to devise an approximate representation $\hat{\mathbf{D}}$ of $\mathbf{D}$ within a space budget B, so that a given error metric in the approximation is minimized. A *normalized* Minkowski-norm error metric, generally expressed in its *weighted* version:

$$\mathcal{L}_p^w(\hat{\mathbf{D}}, \mathbf{D}) = \left(\frac{\sum_i \left(w_i \left| \hat{d_i} - d_i \right| \right)^p}{n} \right)^{\frac{1}{p}}, \tag{1}$$

[2]Approximation schemes that do not provide deterministic guarantees, such as sketches [Gilbert et al. 2003; Cormode et al. 2006], are outside the scope of this work.

covers most practically interesting cases of a pointwise error metric; $\hat{d}_i$ denotes the reconstructed value for d_i and w_i a related weight; in the case of relative-error-based metrics, this weight is $w_i = \frac{1}{\max\{|d_i|,S\}}$, where $S > 0$ is a sanity bound that prevents small values from unnaturally dominating the error result [Garofalakis and Gibbons 2004]. The following definition specifies a broader class of error metrics [Garofalakis and Kumar 2005].

Definition 2.1. 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 *cumulative* 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

$$f_{\mathcal{E}}(\|\hat{\mathbf{D}} - \mathbf{D}\|_{R_i}) = G\big(f_{\mathcal{E}}(\|\hat{\mathbf{D}} - \mathbf{D}\|_{R_j}),\, f_{\mathcal{E}}(\|\hat{\mathbf{D}} - \mathbf{D}\|_{R_k})\big) \tag{2}$$

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

In this article we introduce techniques applicable to any monotonic distributive error metric, as well as specialized ones for the case where G is the max function, that is, for maximum-error metrics (notationally expressed with $p = \infty$ in the Minkowski-norm). For illustration, we use instances of a normalized Minkowski-norm: the average absolute error $\mathcal{L}_1$, the root-mean-squared (Euclidean) error $\mathcal{L}_2$, and the maximum absolute error $\mathcal{L}_\infty$.

We now provide more details on state-of-the-art data reduction methods for the space-bounded synopsis problem under monotonic distributive error metrics.

2.1 Histogram-Based Summarization

A *histogram synopsis* (also called segmentation [Terzi and Tsaparas 2006], partitioning, or piecewise constant approximation [Chakrabarti et al. 2002]) divides $\mathbf{D}$ into $B \ll n$ disjoint intervals $[b_i, e_i]$, $1 \le i \le B$, called *buckets* or *segments*, and attributes a single value v_i to each of them that approximates all consecutive values therein, d_j, $j \in [b_i, e_i]$. In a *dense* histogram these intervals are successive; in a *sparse* histogram there may be void areas between them. 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 dense B-bucket histogram (since $\forall i$, $1 < i \le B$, $b_i = e_{i-1}+1$ and the edges are fixed). For a particular target error metric, the optimal value of v_i is defined as a function[3] of the data values within $[b_i, e_i]$.

Initial work on histogram construction algorithms in the database literature [Ioannidis 1993; Ioannidis and Poosala 1995] focused on heuristics that exhibited low errors in some estimation problem, such as the end-biased

[3]For $\mathcal{L}_1$ it is the median of the values in $[b_i, e_i]$ [Terzi and Tsaparas 2006]; for $\mathcal{L}_2$, their mean [Jagadish et al. 1998]; for $\mathcal{L}_\infty$, the mean of the maximum and minimum value among them; while Gupta et al. [2004] analyzes the respective relative error cases.

[Ioannidis and Poosala 1995], MaxDiff [Poosala et al. 1996], and MHIST [Poosala and Ioannidis 1997] heuristics; such approaches did not attempt to detect the optimal bucket boundaries [Ioannidis 2003]. Jagadish et al. [1998] presented an $O(n^2B)$ dynamic-programming (DP) algorithm that calculates optimal (dense) bucket boundaries for Euclidean ($\mathcal{L}_2$) error. Its basic observation is that the b-optimal histogram for a data vector $\mathbf{D}$ can be recursively derived from the space of $(b-1)$-optimal partitionings of all prefix vectors of $\mathbf{D}$. Hence, the minimal distributive error $E(i, b)$ of a b-bucket histogram of the prefix vector $\langle d_0, d_1, \ldots, d_i\rangle$ is recursively expressed as

$$E(i,b) = \min_{1\leq j<i}\{G\left(E\left(j, b-1\right), \mathcal{E}\left(j+1, i\right)\right)\}, \tag{3}$$

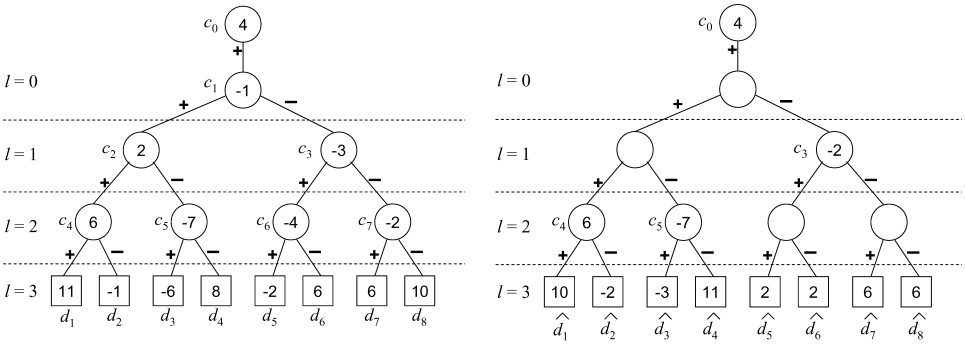
where G is the cumulative error function and $\mathcal{E}(j+1, i)$ the optimal error in a bucket that contains the items $\langle d_{j+1}, \ldots, d_i\rangle$. In fact, this problem is a special case of the problem of approximating a curve by line segments; hence, the DP solution of Jagadish et al. [1998] is a special case of the line-segmentation algorithm of Bellman [1961]. For an arbitrary error metric, this algorithm requires $O(n^3B)$ time. Still, Guha et al. [2004] proposed more efficient specializations for a variety of relative-error-based metrics. Related research paths have supplied histogram methods that efficiently extend the basic idea to multiple dimensions [Poosala and Ioannidis 1997; Bruno et al. 2001; Thaper et al. 2002; Muthukrishnan and Strauss 2003a; Furfaro et al. 2005; Gunopulos et al. 2005], and to workload-based [Muthukrishnan et al. 2005] and range query [Koudas et al. 2000; Gilbert et al. 2001; Guha et al. 2002; Muthukrishnan and Strauss 2003b] optimization.

2.2 Hierarchical Summarization

An alternative stream of research proposes index structures that represent the data in consecutive hierarchical levels of detail. This approach started with the application of the Haar wavelet transform, which has long been used in signal processing [Jawerth and Sweldens 1994]. Recently, Reiss et al. [2006] have proposed a related hierarchical structure for data approximation.

2.2.1 *The Haar Wavelet Hierarchy.* The Haar wavelet hierarchy, based on the transform introduced by Alfréd Haar [Haar 1910], can be visualized through a complete binary tree, the *Haar tree*. This tree holds coefficients representing $\mathbf{D}$ in successive layers of detail; the final tree layer holds the original data. The coefficient in the Haar tree root node contains the overall average value, and each other coefficient value c_i contributes the value $+c_i$ to data values (leaves) in its *left* subtree and $-c_i$ to those in its *right* subtree. Hence, each original data value is reconstructed in terms of the coefficients in its root-to-leaf path.

A *Haar wavelet synopsis* of $\mathbf{D}$ is a vector $\hat{\mathbf{Z}}$ of $B \ll n$ nonzero $\langle i, c_i\rangle$ terms, such that its inverse wavelet transform $\hat{\mathbf{D}} = \mathcal{W}^{-1}(\hat{\mathbf{Z}})$ approximates the data vector $\mathbf{D}$. Figure 1(b) shows a $\{\langle 0, 4\rangle, \langle 3, -2\rangle\ \langle 4, 6\rangle\ \langle 5, -7\rangle\}$ synopsis for the data array of Figure 1(a) with maximum absolute error 4. This is the $\mathcal{L}_\infty$-optimal synopsis with $B = 4$.



(a) Haar tree ($n = 8$). (b) Unrestricted Haar wavelet synopsis.

Fig. 1. A Haar tree and unrestricted synopsis ($n = 8$).

For Euclidean error ($\mathcal{L}_2$), the optimal Haar wavelet synopsis is conveniently computed, consisting of the top-B *normalized* coefficients in the complete Haar wavelet transform of $\mathbf{D}$ [Matias et al. 1998]; the normalized value of a coefficient c is $\frac{|c|}{\sqrt{2^\ell}}$, where ℓ is the level where c resides in the Haar tree. For example, the $\mathcal{L}_2$-optimal synopsis, with $B = 2$, for the data vector in Figure 1(a) is $\{\langle 0, 4\rangle, \langle 5, -7\rangle\}$. The computational convenience of the $\mathcal{L}_2$-synopsis methodology has allowed for its extension to data streams of the cash register model [Gilbert et al. 2003; Cormode et al. 2006], multiple-measure [Deligiannakis et al. 2007] and multidimensional [Jahangiri et al. 2005] data sets, as well as to synopses customized for a given workload [Matias and Urieli 2007], for range-sums [Matias and Urieli 2006], or for both [Mathioudakis et al. 2006; Chen and Nucci 2007; Guha et al. 2008]. Yet this convenience does not extend to *nonEuclidean* metrics. Still, recent studies have strived to construct optimal synopses for such metrics within the Haar framework.

2.2.2 *Restricted Haar Wavelet Synopses.* The *space-bounded* Haar wavelet synopsis problem for general error metrics was first suggested by Matias et al. [1998]. Its first systematic treatment was supplied by Garofalakis and Gibbons [2004], highlighting the practical importance of maximum-error metrics in particular. This treatment was based on a probabilistic model, followed by a fast approximation scheme [Deligiannakis et al. 2005]; however, as shown in Guha et al. [2004] and Garofalakis and Kumar [2005], it does not produce results of high quality. Subsequently, Garofalakis and Kumar [2005] suggested a dynamic programming (DP) algorithm that retains the optimal coefficient subset from the Haar wavelet transform of $\mathbf{D}$ for the target metric. Karras and Mamoulis [2005] proposed a streaming-capable greedy counterpart to this solution for maximum-error metrics. Guha [2008] reduced both the time and space complexity of the DP scheme [Garofalakis and Kumar 2005]. Muthukrishnan [2005] suggested that an algorithm solving the dual, *error-bounded* problem[4] can provide a shortcut to the solution of the space-bounded problem. Still, these solutions all tackle the problem in a *restricted* fashion, in which the nonzero

[4]That is, find a minimal-space synopsis achieving an error bound ϵ.

value that a coefficient may be assigned is fixed in advance; it is either the value provided by the Haar wavelet decomposition itself, as in the following: [Garofalakis and Kumar 2005; Karras and Mamoulis 2005; Muthukrishnan 2005; Guha 2008], or that value normalized according to a randomized rounding scheme, as in Garofalakis and Gibbons [2004] and Deligiannakis et al. [2005]. However, such values are not the *optimal* that could be assigned to the selected set of nodes in the hierarchy for a nonEuclidean metric; hence, as Guha and Harb [2008] observed, synopsis quality is confined by this restriction. This observation also holds for the workload-oriented algorithms of the following: [Mathioudakis et al. 2006; Matias and Urieli 2007; Chen and Nucci 2007; Guha et al. 2008], which also use the coefficient values provided by a Haar wavelet transform, resulting in suboptimal solutions for nonEuclidean metrics.

2.2.3 *Unrestricted Haar Wavelet Synopses.* Guha and Harb [2008] discerned that the values of the B nonzero Haar wavelet synopsis terms need not be defined by the dataset's Haar wavelet transform; they can be set *unrestrictedly*, leading to higher quality than the *restricted* model. For example, the synopsis of Figure 1(b) is unrestricted, since the value assigned to c_3 is not derived from the corresponding coefficient in the complete Haar decomposition in Figure 1(a). Guha and Harb [2008] provided a fully polynomial-time approximation scheme (FPAS) for unrestricted space-bounded Haar wavelet synopses under any Minkowski-norm error metric; this solution is a DP algorithm guided by a two-dimensional tabulation per Haar tree node. A node c_i calculates the minimum attainable error $E(i, v, b)$ over *both* every possible incoming value[5] v *and* every possible amount of space b allocated to the subtree rooted at c_i; possible incoming values are discretized by a resolution step δ. For each $E(i, v, b)$ entry, both the δ-optimal assigned value z (also quantized as a multiple of δ) and the δ-optimal distribution of b units of space among the left c_{i_L} and right c_{i_R} branches of c_i are detected. This DP recursion can be summarized by the equation below:

$$E(i,v,b) = \min \left\{ \begin{array}{c} \min\limits_{0 \le b' \le b} \left\{ \max \left\{ \begin{array}{c} E(i_L, v, b'), \\ E(i_R, v, b-b') \end{array} \right\} \right\}, \\ \\ \min\limits_{z, 0 \le b' \le b-1} \left\{ \max \left\{ \begin{array}{c} E(i_L, v+z, b'), \\ E(i_R, v-z, b-1-b') \end{array} \right\} \right\} \end{array} \right\} \tag{4}$$

Computing $E(0, 0, B)$ determines the best B nonzero term positions in the Haar hierarchy to retain for the synopsis and the best values to assign to each of them for a given δ. The ranges of incoming values v and assigned values z to be tested per node are contained by a guessed upper-bound $\mathcal{E}$ for the target minimized error [Guha and Harb 2008]. The cardinality $R = O(\frac{\mathcal{E}}{\delta})$ of the set of examined values enters the complexity expressions.

2.2.4 *Compact Hierarchical Histograms.* The compact hierarchical histogram (CHH) [Reiss et al. 2006] is an alternative hierarchical approximation

[5]The incoming value of a node c_i is the value constructed by the path from the root of the sparse Haar tree up to c_i. For example, the incoming value of node c_7 in the tree of Figure 1(b) is $c_0 - c_3 = 6$.

structure. A CHH defines a binary hierarchy of dyadic intervals, by default identical to that of a Haar tree; likewise, it selects a subset of nodes to represent the approximated data set. However, the representation mechanism is different; a data value is simply approximated by the value of its lowest nonzero ancestor node in the CHH hierarchy. In a *longest-prefix-match* (LPM) CHH, each such node should be assigned the *optimal* value, under the target error metric, for the exact set of data values it approximates [Reiss et al. 2006]; this optimal value is defined as for a histogram bucket (see Section 2.1). Remarkably, a binary CHH is analogous to the hierarchical summarization structure independently proposed by Agarwal et al. [2007].

Reiss et al. [2006] proposed exact solutions for limited versions of the CHH construction problem and heuristics for the LPM CHH problem. The best-performing CHH algorithm is a greedy heuristic that improves upon an optimal *overlapping* partitioning. In such a partitioning, the value assigned to a CHH node is the optimal value for the *whole* data interval under the scope of c_i with the target metric (as in a plain histogram bucket [Jagadish et al. 1998; Guha et al. 2004; Terzi and Tsaparas 2006]), but not for the value set c_i *actually* approximates (which depends on the other nodes with nonzero assigned values in the subtree rooted at c_i). The greedy heuristic first establishes the optimal occupied node positions for an overlapping partitioning with the target error metric, and then uses these positions, but adjusts the values assigned to them so as to be optimal for the data set they actually approximate (i.e., a subset of the whole data interval under the node's scope).

2.3 A Space-Efficiency Technique

The state-of-the-art for both synopsis construction methods features DP algorithms that tabulate over space allocations [Guha et al. 2004; Garofalakis and Kumar 2005; Guha and Harb 2008; Reiss et al. 2006], raising high time and space complexity demands. Guha [2008] identified space as the most significant resource for summarization and provided a paradigm that reduces the space demands of these DP schemes in the offline case. Its main idea is to avoid storing all tabulated results throughout the DP; part of them can be dropped and recomputed later. In histogram construction, the tabulation on $\{i, b\}$ should progress with increasing b, $1 \leq b \leq B$ (i.e., the loop of b is the outer loop). Since the values $E(*, b)$ are fully determined by $E(*, b-1)$, after a b-column has been used to calculate the $(b+1)$-column, it is dropped; hence the space is $O(n)$. The tabulation (Eq. (3)) also detects and stores the single bucket $M(i, b)$ in the optimal b-partitioning of $\langle d_0, d_1, \ldots, d_i \rangle$ that contains the *middle* data item $\lfloor \frac{n}{2} \rfloor$ of $\mathbf{D}$ ($M(i, b)$ exists only when $i \geq \frac{n}{2}$, where n is the size of $\mathbf{D}$). After the optimal error $E(n, B)$ and middle-item bucket $\mathcal{M} = M(n, B)$ for the complete problem have been found, the two $O\left(\frac{n}{2}\right)$ independent subproblems for the intervals on the left and right of $\mathcal{M}$ are re-solved recursively. Hence, the total time for the general-error histogram construction problem [Jagadish et al. 1998] remains $O(\sum_{\ell=1}^{\log n} 2^{\ell}(\frac{n}{2^{\ell}})^2 B) = O\left(n^2 B\right)$, that is, the recomputation cost is amortized.

Guha [2008] applies the same methodology to the restricted Haar wavelet synopsis algorithm of Garofalakis and Kumar [2005]. In this case, the required

tabulation progresses in a bottom-up fashion in the Haar tree; all table entries on a parent node are computed from the tables of its children nodes, which can then be dropped. Accordingly, at most $\log n + 1$ tables need be concurrently stored, covering one path through the Haar tree. After the solution is established at the top level of the tree, the two half-size subproblems in the two subtrees of c_1 are re-solved [Guha 2008]. Restricted Haar wavelet synopsis construction requires time quadratic in n, because each of n Haar tree nodes has to consider $O(2^{\log n}) = O(n)$ possible choices of values in its ancestor-set [Garofalakis and Kumar 2005]. Hence, the recomputation cost is amortized in this case as well. The same technique is applicable to the *unrestricted* Haar wavelet synopsis algorithm of Guha and Harb [2008] (Eq. (4)); likewise, after the arrays $E(i_L, *, *)$ and $E(i_R, *, *)$ have been used to calculate the entries of $E(i, *, *)$, they are dropped, hence at most $\log n + 1$ arrays need be concurrently stored. However, as we show in Section 5.4, in this case the price for space-efficiency is an extra $\log n$ time complexity factor due to recomputation.

3. MOTIVATIONS

In this section we outline the motivations for our research on the questions of quality and efficiency in hierarchical synopsis construction, as well as in the experimental arena.

3.1 Quality

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 [Guha et al. 2004]. Therefore, histograms are not good at approximating sharp discontinuities. In contrast, hierarchical structures are advantaged by their ability to see beyond local interrelations. Besides, a B-term histogram approximates only B value ranges, as opposed to the Haar wavelet which defines B to $3B + 1$ value intervals, and the CHH, which defines B to $2B + 1$ intervals. Observably, the Haar framework can make the most economic use of a space budget, as a retained wavelet coefficient can split a pre-existing interval into four new ranges. This four-way split derives from the fact that a coefficient bears on the two binary intervals that it affects two opposite-signed contributions of equal absolute magnitude. Besides, the *differential* nature of Haar wavelet coefficients is also responsible for the Parseval-based near-linear computation of $\mathcal{L}_2$-optimal synopses [Jawerth and Sweldens 1994]. However, this differential form is also a liability, as it restricts the flexibility of representation. Still, we *do not have to* hold on to this restriction in cases where the computational effectiveness that it allows for does not apply (i.e, on non-$\mathcal{L}_2$ metrics).

3.2 Efficiency

3.2.1 *General Error Metrics.* In addition to the quality consideration above, the state-of-the-art unrestricted Haar wavelet synopsis algorithm [Guha and Harb 2008] presents a complexity defect depending on the p parameter of

the target Minkowski norm (Eq. (1)); its complexity is $O((\frac{\mathcal{E}}{\delta})^2 n^{1+\frac{2}{p}} B)$ in time, that is, cubic in n for the $\mathcal{L}_1$ metric, and $O(\frac{\mathcal{E}}{\delta} n^{\frac{1}{p}} B \log \frac{n}{B} + n)$ in space (Table II, Section 8.1). Chen and Nucci [2007] observed this defect, correctly characterized the complexity as "too high for synopses used for databases," and resorted to studying the workload-based Haar wavelet synopsis problem, for *weighted* Euclidean error in a restricted formulation. Nevertheless, in this workload-based case, the restricted formulation does not provide an error guarantee with respect to the optimal solution. In our opinion, this withdrawal to the restricted model discards what is worth keeping; the unrestricted model of Guha and Harb [2008] treats the workload-aware problem with the same effectiveness as the regular problem, *notwithstanding* its complexity dependence on p. For maximum-error metrics, the solution of Guha and Harb [2008] is already linear in n and achieves higher quality than a restricted solution. In our opinion, we should not abandon the unrestricted model, but treat its complexity defect, that is, render its complexity independent of p. As we will show, expanding the Haar tree structure in a way that allows for higher accuracy of approximation also *simplifies* the synopsis computation process and achieves equally low complexities for all monotonic distributive error metrics.

3.2.2 *Maximum-Error Metrics.* The algorithm of Guha and Harb [2008] is linear in n for any maximum-error metric. Yet, despite the reduction by Guha [2008], it still features a demanding two-dimensional tabulation over space allocations and value assignments. Besides, the amortization achieved with the paradigm of Guha [2008] holds for algorithms of time quadratic or near-quadratic in n (e.g., those reviewed in Section 2.3 and in Muthukrishnan [2005]), but *not* for algorithms of time *linear* or near-linear in n (e.g., the solution of Guha and Harb [2008] for maximum-error metrics and the winner heuristic in Reiss et al. [2006]). Applied on them, the paradigm creates a tradeoff between space- and time-efficiency (Table III, Section 8.1), which also appears with our general-error Haar$^+$ synopsis construction algorithm (in Section 5.4). Muthukrishnan [2005] and Karras et al. [2007] have shown that *space-bounded* hierarchical synopsis problems for maximum-error metrics can be more efficiently solved *via* their *error-bounded* counterparts, in which the goal is to minimize the synopsis space subject to a maximum-error bound ϵ; this *indirect* approach lightens the complexity burdens. Yet the importance of maximum-error metrics [Garofalakis and Gibbons 2004; Garofalakis and Kumar 2005; Karras and Mamoulis 2005] not only calls for a more space-efficient solution, but also for one that would provide optimal error guarantees. As we show, this indirect method *does* allow for *optimal* solutions. This benefit is undermined by an exponential worst-case time complexity for Haar$^+$ synopsis construction; still, applied on the CHH, that is, a simplified version of the Haar$^+$ tree, this method yields synopses with *optimal* maximum-error guarantees in *low polynomial* time.

3.3 Experimentation

To the best of our knowledge, previous research has not attempted a face-to-face comparison between state-of-the-art hierarchical and histogram approximation techniques. In particular, Matias et al. [1998] compared heuristic Haar wavelet

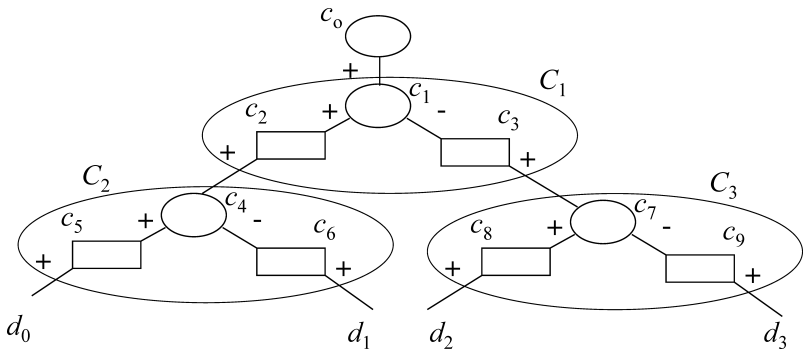


Fig. 2. An one-dimensional Haar$^+$ tree.

synopsis methods (as well as the $\mathcal{L}_2$-optimal method) to the MaxDiff [Poosala et al. 1996] and MHIST [Poosala and Ioannidis 1997] histogram heuristics; the contemporaneously developed optimal-histogram algorithm of Jagadish et al. [1998] did not make it into that study. Chakrabarti et al. [2002] compared the pruning power of the APCA dimensionality-reduction technique to that of simple Haar wavelet synopses consisting of the B highest terms, applied on similarity search among indexed time series. Guha et al. [2004] measured the accuracy of approximation achieved with probabilistic Haar wavelet synopses [Garofalakis and Gibbons 2004] against that of error-optimal histograms for several relative-error based metrics. Unluckily, the contemporaneously developed error-optimal restricted Haar wavelet synopsis algorithm of Garofalakis and Kumar [2004] was not available for that study. Garofalakis and Kumar [2005] showed that the techniques of Garofalakis and Kumar [2004] outperform those of Garofalakis and Gibbons [2004] in terms of accuracy; yet, their relation to optimal histograms was not inspected. Similarly, Guha and Harb [2008] demonstrated that unrestricted Haar wavelet synopses outperform the restricted ones; however, a comparison to histogram methods was not provided. Lastly, Reiss et al. [2006] compared the accuracy achieved with heuristic CHH techniques to the that of $\mathcal{L}_2$-optimal [Jagadish et al. 1998] and end-biased [Ioannidis and Poosala 1995] histograms; other error-optimal histograms [Guha et al. 2004] and Haar wavelet methods [Garofalakis and Kumar 2005; Guha and Harb 2008] were not cross-examined. Hence, an investigation of the relative performance of diverse approximation techniques, examining the intuition expressed in Graps [1995] and Guha et al. [2004], is long due. Section 9 provides a first attempt in that direction.

4. 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 2 depicts a simple one-dimensional Haar$^+$ tree that may be used for summarizing a four-element data set $\{d_0, d_1, d_2, d_3\}$. It contains a single root coefficient node c_0 that contributes its value to all approximated data, followed by a binary tree of coefficient nodes grouped in triads, depicted as C_1,

C_2, and C_3. Triads substitute what are single wavelet coefficients in a classical Haar tree. In each triad, the *head coefficient*, namely c_1, c_4, and c_7, behaves as a classical wavelet coefficient: it contributes its value positively to its left subtree and negatively to its right subtree; the other two, left and right *supplementary coefficients*, namely, c_2 and c_3 in C_1, c_5 and c_6 in C_2, and c_8 and c_9 in C_3, contribute their value positively in the single interval they affect. For example, c_3 contributes its value positively to d_2 and d_3, if such a nonzero 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, and the triad where this parent node resides *parent triad* of C. For example, the parent node of C_2 is c_2 and its parent triad is C_1; in reverse, C_2 is the *left child triad* of C_1 and C_3 is its *right child triad*.

An *optimal* synopsis of space budget B for a given error metric $\mathcal{E}$ places B nonzero coefficient values at any positions in the Haar$^+$ tree so that $\mathcal{E}$ is minimized. For example, for the four-element data set $\{5, 3, 12, 4\}$ a 2-term Haar$^+$ synopsis that minimizes the un-weighted pointwise errors $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_\infty$ consists of the coefficients $\{c_0 = 4, c_8 = 8\}$. The resulting approximation is $\{4, 4, 12, 4\}$ with absolute error values $\{1, 1, 0, 0\}$, hence $\mathcal{L}_1 = 0.5$, $\mathcal{L}_2 = \frac{\sqrt{2}}{2}$ 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 $\mathcal{L}_1 = 2$, $\mathcal{L}_2 = \frac{3\sqrt{2}}{2}$, $\mathcal{L}_\infty = 3$; 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 $\mathcal{L}_1$ and $\mathcal{L}_\infty$ is $\{c_0 = 5.5, c_7 = 4\}$, with $\mathcal{L}_1 = 2$ 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 absolute error values $\{1, 1, 4, 4\}$, hence $\mathcal{L}_2 = \frac{\sqrt{34}}{2}$ and $\mathcal{L}_\infty = 4$. An $\mathcal{L}_1$-optimal 2-bucket histogram is $\{5, 5, 5, 4\}$ with $\mathcal{L}_1 = 2.25$. This simple example demonstrates that both the classical Haar synopsis model and piecewise-constant histogram techniques may not achieve as high accuracy of approximation as the Haar$^+$ structure. We emphasize the following points:

—The classical Haar structure is a special case of the generalized Haar$^+$ structure. Hence, 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 coefficients in three distinct groups, one for each 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, hence $\lceil \log n \rceil$ bits per index, as with the indexes in a classical Haar wavelet synopsis and the bucket boundaries in a histogram.

4.1 Basic Properties

A Haar$^+$ tree is a sparse vector $\mathbf{H}$ of $n = 3 \times 2^d - 2$ 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. We use the notation $a = \mathsf{P}(b)$ to denote that coefficient a resides on the parent node of coefficient b, $a \in \mathsf{Rleaves}(b)$ to denote that data item (leaf) a lies in the right subtree of node b, and $a \in \mathsf{path}(b)$ 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, such that $i = (j + N) \setminus 2$. This data item is constructed as $d_j = \sum_{i \in \mathsf{path}(j)} \delta_{ij} c_i$, where

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

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 $v = \sum_{k \in \mathsf{path}(i)} \delta_{ki} c_k$ 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 nonzero values in a Haar$^+$ tree $\mathbf{H}$.

The following lemma shows that a nonzero head coefficient does not need to inhabit a triad with at least one nonzero supplementary coefficient.

LEMMA 4.1. *A triad C of a Haar$^+$ tree representation $\mathbf{H}$ does not need to contain both a nonzero head coefficient and a nonzero supplementary coefficient.*

PROOF. The case where C contains three nonzero coefficients can be directly reduced to a more sparse variant with only two nonzero supplementary coefficients. $C = [v, p, q, r]$ is equivalent to $C = [v, 0, q + p, r - p]$; a nonzero head coefficient is redundant in this case. The case that C contains exactly two nonzero values being a head and a supplementary coefficient can be treated similarly, with the number of nonzero values unchanged. In effect, a nonzero head coefficient only need be used as a triad's single nonzero term. □

Figure 3 depicts an adjustment of $C = [v, p, q, 0]$ to $C = [v, 0, p+q, -p]$ as in Lemma 4.1.

The following theorem expands on the result of Lemma 4.1.

THEOREM 4.2. *Let $\mathbf{H}$ be an arbitrary Haar$^+$ tree producing the data vector $\mathbf{D}$, in which at least one triad contains more than one nonzero values. 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 nonzero value and $\|\mathbf{H}'\| \leq \|\mathbf{H}\|$.*

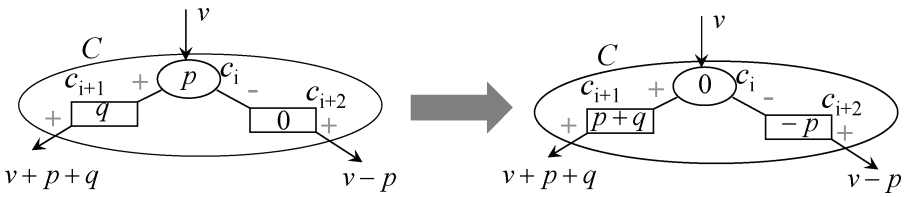


Fig. 3. Adjustment of triad as in Lemma 4.1.

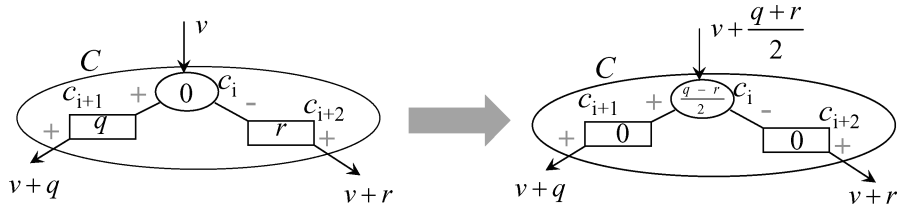


Fig. 4. Basic transformation of triad in Theorem 4.2.

PROOF. By Lemma 4.1, any assignment of more than one nonzero value in a triad C can be reduced to the assignment of two nonzero values, one on each supplementary coefficient; hence C is brought to the state $[v, 0, q, r]$ and produces the contribution vector $[v + q, v + r]$. Yet 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 of more than one nonzero coefficients is reducible to a triad of one nonzero coefficient by changing its *incoming* value from v to $v + \frac{q+r}{2}$, as follows:

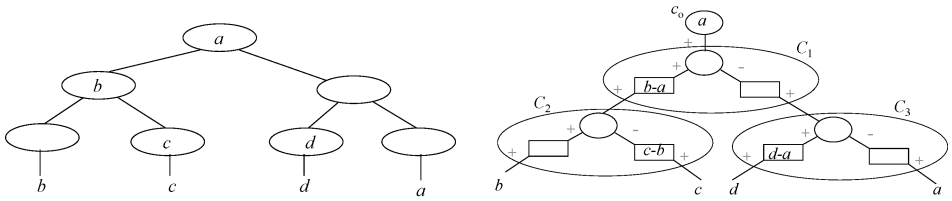
(1) If the parent of C is the root node, then we add the value $\frac{q+r}{2}$ to the root coefficient.
(2) If the parent of C is a triad Q, then we add the value $\frac{q+r}{2}$ to the parent node of C in Q. If this addition results in more than one nonzero value in Q, then we proceed to reduce Q to a triad with one nonzero 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 nonzero 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 nonzero value and $\|\mathbf{H}'\| \leq \|\mathbf{H}\|$. □

Figure 4 depicts the basic transformation of a triad with two nonzero supplementary coefficients q, r to one with only one nonzero head coefficient.

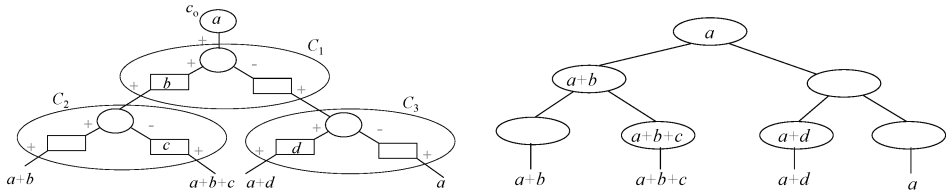
Although three coefficients need never be chosen in the same triad, it is still advisable to maintain the structure in this form. The assignment of two opposite-signed nonzero values such that $c_l = -c_r$ is also reduced to the assignment of a single nonzero value $c_h = c_l$; this characteristic is an advantage of the classic Haar tree that the Haar$^+$ tree maintains. Based on its triad structure, the Haar$^+$ tree allows for refined, high-quality summarization.

The following corollary follows from Theorem 4.2.



(a) CHH (b) Haar$^+$ tree

Fig. 5. A CHH and its equivalent Haar$^+$ tree.



(a) Haar$^+$ tree (b) CHH

Fig. 6. A Haar$^+$ tree and its equivalent CHH.

COROLLARY 4.3. *The optimal B-term Haar$^+$ tree representation* **H** *of a data vector* **D** *that minimizes a given error measure* $\mathcal{E}$ *can be expressed as a Haar$^+$ tree with at most one nonzero value in each triad.*

4.2 Equivalence of a Compact Hierarchical Histogram to a Simplified Haar$^+$ Tree

In this section we show the equivalence of a CHH to a simplified Haar$^+$ tree.

THEOREM 4.4. *Any B-nonzero-term binary CHH [Reiss et al. 2006] can be represented as a Haar$^+$ tree of B nonzero supplementary (or root) coefficients. In reverse, any Haar$^+$ tree of B nonzero terms which are all supplementary (or root) coefficients can be represented as a B-term binary CHH.*

PROOF. Let **C** be a B-term binary CHH. For each nonzero term node c_i, let v_i be the value of its lowest nonzero ancestor node. Then, a Haar$^+$ tree **H** in which the supplementary coefficient corresponding to the position of each nonzero term c_i is assigned the value $c_i - v_i$ produces the same representation as **C**. In reverse, let **H** be a Haar$^+$ tree with exactly B nonzero supplementary coefficients. Let v_i be the *incoming value* to the node of a nonzero term c_i. Then, a CHH **C** in which the node corresponding to the position of each nonzero term c_i is assigned the value $c_i + v_i$ produces the same representation as **H**. □

Figures 5 and 6 depict the transformation from a CHH to a Haar$^+$ tree and vice versa. By Theorem 4.4, the CHH synopsis problem is reduced to the Haar$^+$ synopsis problem. The Haar$^+$ tree can also achieve higher accuracy than a CHH, thanks to its inclusion of classical wavelet (head) coefficients. In addition, the hierarchical structure recently proposed by Agarwal et al. [2007] is also equivalent to a Haar$^+$ tree of only supplementary coefficients. In this case the relationship is straightforward; the nodes in Agarwal et al. [2007] combine their contributions additively, exactly like Haar$^+$ supplementary coefficients.

With the benefit of hindsight, a Haar$^+$ tree can be seen as the merger of a CHH and a Haar tree; it retains the compression advantage of a Haar tree over a histogram, adding more flexibility to it. Furthermore, it achieves more succinct representations than a CHH, as the use of a head coefficient adds two distinct constant-value intervals in its approximation.

In Reiss et al. [2006], CHH techniques are tested on approximating Internet traffic data, in comparison to $\mathcal{L}_2$-optimal [Jagadish et al. 1998] and end-biased [Ioannidis and Poosala 1995] histograms. The authors emphasize that the CHH structure can be useful in a broad range of applications. In fact, as has been expressed intuitively in Graps [1995] and Guha et al. [2004], and we verify in Section 9, hierarchical synopsis structures perform better than histograms when approximating more *discontinuous* data sets; Internet traffic data is a particular instance of such data. Besides, Reiss et al. [2006] proposed an extension of CHH techniques on predefined hierarchies beyond the default binary one; the same extension applies straightforwardly to the Haar$^+$ tree, since, as Reiss et al. [2006] show, any arbitrary hierarchy can be expressed as a binary one, and, in effect, such a binary hierarchy can incorporate Haar$^+$ tree head and supplementary coefficients. Still, *predefined* hierarchies impose an arbitrary constraint. In Karras and Mamoulis [2008] we examine the problem of detecting the most appropriate hierarchical pattern for the problem at hand.

5. HIERARCHICAL SYNOPSES FOR GENERAL DISTRIBUTIVE ERROR METRICS

We now construct a DP approximation scheme for the optimal hierarchical synopsis representation of a data vector $\mathbf{D}$, employing Corollary 4.3. As we discussed, the CHH construction problem can be treated as a special case of the Haar$^+$ synopsis construction problem. The simplification of the structure from a Haar$^+$ tree to a CHH does not yield a complexity advantage, while compromising synopsis quality; hence, our methodology is developed for the general, Haar$^+$ case. The space-bounded problem that we address is defined as follows:

Problem 5.1. Given a data vector $\mathbf{D}$ and a monotonic distributive error metric $\mathcal{E}$, construct a B-nonzero-term Haar$^+$ representation $\mathbf{H}$ of $\mathbf{D}$ that produces an 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 nonzero terms we can keep. Since each triad C_i needs to contain at most one nonzero 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.

5.1 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 C_i and its descendants. We can establish the solution in a bottom-up process, calculating $Q(i, v, b)$ on each triad

Table I. Notation

Symbol	Meaning
D	Summarized data vector
H	Optimized Haar$^+$ representation of **D**
C_i	Triad in **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 coefficient of C_i
z_0	Value assigned to root coefficient of **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 **D**
D_i	Domain of allocated space values b at C_i

for each possible v and b. Let ℓ_i denote the layer of triads in which C_i resides, counting from the bottom; then at most $2^{\ell_i} - 1$ nonzero values can be used by triad C_i and its descendants; hence the domain of b is $D_i = \{0, 1, \ldots, \min\{B, 2^{\ell_i} - 1\}\}$. In order to delimit the domain of v, we quantize it into multiples of a resolution step δ. But we still need to set lower and upper bounds for this domain. Fortunately, the Haar$^+$ structure allows us to do so tightly. In the following discussion, we use the notation in Table I. We start out with the following proposition.

PROPOSITION 5.2. *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 $\hat{d}_l \geq v$.*

PROOF. At C_i there exists a reconstruction path in which v is not increased, as well as one in which it is not decreased, obtained if we choose the appropriate direction in case C_i holds a nonzero head coefficient or the all-null direction otherwise. The same applies at every subsequent layer. Hence, there exist reconstructed values $\hat{d}_k$ and $\hat{d}_l$ such that $v \in [\hat{d}_k, \hat{d}_l]$. □

Proposition 5.2 finds application in the following.

PROPOSITION 5.3. *If C_i has a nonzero 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, $v \notin (m_i, M_i) \Rightarrow z_h = 0$.*

PROOF. Assume that $v \notin (m_i, M_i)$ and $z_h \neq 0$. Then, by Corollary 4.3, 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 subtree of C_i may lead to a good approximation of the values therein by decreasing v. Nevertheless, the incoming value $v + z_h$ to the left subtree does not gain in approximation by increasing v, as v is larger than the maximum value M_i to be approximated, and, by Proposition 5.2, 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 value $z_r = -z_h$ to the right supplementary coefficient of C_i alone. Similar reasoning applies to other cases. Thus the assignment of a nonzero 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 assigned thus after we introduce the following proposition, which follows from Proposition 5.3.

PROPOSITION 5.4. *An incoming value $v < m_i$ ($v > M_i$) at C_i cannot result in better approximation quality, by any monotonic error metric, than a value v' such that $v < v' \leq m_i$ ($v > v' \geq M_i$), with the number of nonzero terms in the subtree of C_i being equal.*

PROOF. Assume $v < m_i$. By Proposition 5.3, the first nonzero coefficient encountered on any subsequent reconstruction path can be a supplementary coefficient without affecting the quality of approximation. Still, 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 nonzero coefficient is not encountered, v' has a default quality advantage over v, since it has a 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 an 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 now delimit the assigned value of a head coefficient with the following theorem.

THEOREM 5.5. *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* **H**. *If a nonzero 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 $[v + z_h, v - z_h]$ to its two subtrees. Without loss of generality, assume that $z_h > 0$ and $v + z_h > M_i$, $v - z_h < m_i$. Then, by Proposition 5.4, the approximation quality on both subtrees can be bettered by decreasing z_h so that at least one of the produced 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, z_h should place at least one of $v + z_h$, $v - z_h$ inside the interval $[m_i, M_i]$:

$$\begin{array}{lll} m_i \leq v + z_h \leq M_i & \vee \quad m_i \leq v - z_h \leq M_i & \Leftrightarrow \\ m_i - v \leq z_h \leq M_i - v & \vee \quad v - M_i \leq z_h \leq v - m_i & \Leftrightarrow \\ z_h \in \left[\min\{v - M_i, m_i - v\}, \max\{M_i - v, v - m_i\}\right] & & \Leftrightarrow \\ |z_h| \leq \max\{M_i - v, v - m_i\} & & \end{array}$$ □

Reasoning analogous to that of Theorem 5.5 leads to the following theorem.

THEOREM 5.6. *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) subtree of triad C_i. If a nonzero 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 nonzero value z_0 is assigned to the root coefficient, then $z_0 \in [m, M]$, where m (M) is the global minimum (maximum) in* **D**.

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

THEOREM 5.7. *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 nonzero supplementary coefficient, or from the root coefficient, the proof follows directly from Theorem 5.6. We examine incoming values derived from an ancestor nonzero head coefficient. Consider a nonzero head coefficient z_h encountered at a triad C_k. Then, by Proposition 5.3, the incoming value v at C_k lies within the interval (m, M). Besides, according to Theorem 5.5, $|z_h| \leq \max\{M - v, v - m\}$. 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 5.7 is that, in the worst case, a nonzero head coefficient covers the difference $M - m$ in one direction and replicates it 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 $|\mathcal{S}| \leq \lfloor \frac{3\Delta}{\delta} \rfloor + 1 = O(\frac{\Delta}{\delta})$.[6] Furthermore, let $\mathcal{S}^v_{i,H} \subset \mathbb{R}$, $\mathcal{S}^v_{i,L} \subset \mathbb{R}$, $\mathcal{S}^v_{i,R} \subset \mathbb{R}$ denote the set of potential values assigned to the head, left and right supplementary coefficient of triad C_i that are multiples of δ, for incoming value v. By Theorems 5.5 and 5.6, the cardinality of these sets is also $O(\frac{\Delta}{\delta})$.

5.2 Deriving the Answer

The derivation of the optimal error result and the respective B-nonzero-term Haar$^+$ tree representation $\mathbf{H}$ of $\mathbf{D}$ does not pose a novel algorithmic problem. As in previous synopsis construction algorithms [Jagadish et al. 1998; Deligiannakis et al. 2007; Garofalakis and Gibbons 2004; Guha et al. 2004; Garofalakis and Kumar 2005; Guha and Harb 2008; Muthukrishnan 2005], a DP solution can be applied. In particular, our algorithm draws from the unrestricted Haar wavelet synopsis construction algorithm of Guha and Harb [2008]. We compute the $Q(i, v, b)$ 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 Guha [2008], and analyze the emerging tradeoff 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 precalculated 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 subtree rooted at C_i; it contains: (i) the δ-optimal value z_h, z_l, or z_r to assign to a coefficient 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

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

$|\mathcal{S}_i| \cdot |D_i|$. A recursive procedure MinError emerges, which computes $E(i, v, b)$ as

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

$$E(i,v,b) = \min \left\{ \begin{array}{l} \min\limits_{z_h \in \mathcal{S}^v_{i,H}, b' \in D_i} \left\{ \begin{array}{l} E(i_l, v + z_h, b') + \\ E(i_r, v - z_h, b - b' - (z_h \neq 0)) \end{array} \right\} \\ \min\limits_{z_l \in \mathcal{S}^v_{i,L}, b' \in D_i} \left\{ \begin{array}{l} E(i_l, v + z_l, b') + \\ E(i_r, v, b - b' - (z_l \neq 0)) \end{array} \right\} \\ \min\limits_{z_r \in \mathcal{S}^v_{i,R}, b' \in D_i} \left\{ \begin{array}{l} E(i_l, v, b') + \\ E(i_r, v + z_r, b - b' - (z_r \neq 0)) \end{array} \right\} \end{array} \right\} \quad (5)$$

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 these is the least achievable error, in the subtree rooted at C_i, among all allowed combinations of a value assigned to the examined coefficient[7] and a distribution of the available space to the branches of that subtree. For the economy of presentation the -1 term, which decreases the space allocated to the right branch in case a nonzero 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 $A[v, b].z_h$, $A[v, b].z_l$, or $A[v, b].z_r$ stores the coefficient 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 Guha [2008], 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 postorder recursion reaches the rightmost 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 triad C_i hold one entry for each possible incoming value in $|\mathcal{S}|$, hence their size is $O(\frac{\Delta}{\delta} \min\{B, 2^{\ell_i} - 1\})$; 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 $O(\frac{\Delta}{\delta} \min\{B, 2^{\ell_i} - 1\})$ time. Hence, the worst-case running time of MinError is $O((\frac{\Delta}{\delta})^2 \sum_{i=1}^{n} \min\{B, 2^{\ell_i} - 1\}^2) = O((\frac{\Delta}{\delta})^2 nB)$. Under the assumption that $\frac{\Delta}{\delta}$ (that is, the largest input value) is polynomially-bounded in n, this algorithm provides a fully-polynomial-time approximation scheme. For the special case of a maximum-error metric, the B factor becomes $\log^2 B$, thanks to the application of binary search for search through space allocations; this method is used in the following: [Garofalakis and Kumar 2005; Guha 2008; Guha and Harb 2008]. Since at most $\log n + 1$ arrays need to be concurrently stored, the space complexity[8] is $O(\frac{\Delta}{\delta} \sum_{\ell=1}^{\log n+1} \min\{B, 2^{\ell_i} - 1\}) = O(\frac{\Delta}{\delta} B \log \frac{n}{B})$.

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

[8]A $1 + \log \frac{n}{B}$ factor is simplified to $\log \frac{n}{B}$ under the assumption that $B < \frac{n}{2}$. In applications where a distinction between *total space* and *working space* complexity is meaningful, as in Garofalakis and

5.3 Approximation Guarantee

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

THEOREM 5.8. *If a data set* $\mathbf{D}$ *of size* n *is optimally summarized in* B *terms by a Haar*$^+$ *representation* $\mathbf{H}^*$ *in* $\mathbb{R}$*, 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, achieving 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 $\mathbf{D}^*$ be the approximation of $\mathbf{D}$ produced by $\mathbf{H}^*$, $\hat{\mathbf{H}}_\delta$ the representation of $\mathbf{D}$ derived after rounding all coefficients in $\mathbf{H}^*$ to the nearest multiple of δ, $\mathcal{E}'_\delta$ 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 $\mathcal{E}_\delta \leq \mathcal{E}'_\delta$. Still, by the triangle inequality, $\mathcal{E}'_\delta \leq \mathcal{E}^* + \mathcal{L}_p(\mathbf{D}^*, \hat{\mathbf{D}})$. Each reconstructed data value is the sum of at most $\min\{B, \log n\}$ terms (at most one per 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-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, $\mathcal{E}_\delta \leq \mathcal{E}^* + \frac{\delta}{2}\min\{B, \log n\}$. □

5.4 Constructing the Synopsis

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

5.4.1 *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 runtime is the sum of the basic runtime for all re-entered subproblems. Setting ℓ as the Haar$^+$ tree layer, this sum becomes $O((\frac{\Delta}{\delta})^2 B \sum_{\ell=0}^{\log n} 2^\ell \frac{n}{2^\ell}) = O((\frac{\Delta}{\delta})^2 nB \log n)$, specialized as $O((\frac{\Delta}{\delta})^2 n \log n \log^2 B)$ for a maximum-error metric. Hence, the price for space-efficiency is an extra $\log n$ time complexity factor due to recomputation. On the other hand, the space becomes $O(\frac{\Delta}{\delta} B \log \frac{n}{B} + n)$, where n stands for the necessary storage of the data set.

5.4.2 *The Time-Efficient Solution.* Alternatively, we may maintain all computed solutions throughout the computation, keeping the time at $O((\frac{\Delta}{\delta})^2 nB)$. 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 $O(\frac{\Delta}{\delta}\min\{B, 2^{\ell_i}\})$. The summation over the second factor gives $\sum_i \min\{B, 2^{\ell_i}\} = \sum_\ell 2^{\log n - \ell} \min\{B, 2^\ell\} = n \log B$. Hence the space complexity in this case is $O(\frac{\Delta}{\delta} n \log B)$.

Kumar [2005], we need only keep three arrays in the main memory at any time, hence the working space complexity is $O(\frac{\Delta}{\delta} B)$.

—As suggested by Guha and Harb [2008] for unrestricted Haar synopses, we may append a list of all retained coefficients in the corresponding solution to each entry $A[v, b]$ of a DP array at triad C_i. Again, at most $\log n + 1$ arrays are stored concurrently. The size of a solution maintained with each array entry at layer ℓ_i is at most $\min\{B, 2^{\ell_i}\}$, hence the space for an array at layer ℓ_i is $O(\frac{\Delta}{\delta}(\min\{B, 2^{\ell_i}\})^2)$. The squared factor, summed over all layers, gives $B^2 \log \frac{n}{B}$, hence the space complexity becomes $O(\frac{\Delta}{\delta} B^2 \log \frac{n}{B})$.

The two space complexity expressions are equal when $n \log B = B^2 \log \frac{n}{B} \Leftrightarrow B = \sqrt{n}$. If $B \ll \sqrt{n}$, then it is preferable to append the solutions. On the other hand, if $B \gg \sqrt{n}$, then it is advantageous to maintain all $E(i, *, *)$ arrays *per se* in memory. Values of B both higher and lower than $\sqrt{n}$ are likely to occur in practice, hence the preferable method depends on the application at hand. This time-efficient solution enables the operation of the algorithm in one pass over the data. As Table III in Section 8.1 shows, a similar tradeoff analysis applies to the unrestricted Haar synopsis algorithms of Guha and Harb [2008] (Section 2.2.3) and to the winner greedy heuristic of Reiss et al. [2006] (Section 2.2.4).

6. HIERARCHICAL SYNOPSES FOR MAXIMUM-ERROR METRICS

The problem of minimizing a maximum-error metric, such as $\mathcal{L}_\infty$ and its weighted variants (including maximum *relative* error), has a special practical interest, since such metrics provide intuitive *deterministic error guarantees* for independent approximate values [Garofalakis and Kumar 2004; Karras and Mamoulis 2005; Muthukrishnan 2005]. Moreover, with this problem we can follow a more time- and space-efficient approach; we exploit the solution to the dual, *error-bounded* synopsis problem in order to solve its *space-bounded* counterpart. Such an approach was first suggested by Muthukrishnan [2005] in the context of restricted Haar synopses and applied in other contexts by Karras et al. [2007]; our targeting of error-bounded synopsis problems is akin to the methodology for explanation of change in hierarchical summaries suggested by Agarwal et al. [2007]. The dual-problem approach employed not only delivers a crucial complexity advantage in relation to the FPAS of Section 5 for maximum-error minimization, but also, as we show in this article, allows for *optimal* solutions to space-bounded synopsis problems.

6.1 An Optimal Solution to the Error-Bounded CHH Problem

The computation of an optimal solution was not considered in the case of the space-bounded Haar$^+$ synopsis problem for distributive error metrics (Section 5); in that case, the problem of calculating the optimal value to assign on a coefficient *per se* cannot be easily isolated from the decisions made on other parts of the hierarchy; hence that problem, as are related hierarchical synopsis problems [Guha and Harb 2008; Reiss et al. 2006], is judged to be computationally hard. Hence, Guha and Harb [2008] resorted to an approximation scheme, while Reiss et al. [2006] developed heuristics for CHH construction. Still, error-bounded hierarchical synopsis problems with

a maximum-error bound allow us to attempt an optimal solution. Moreover, contrary to the case of the space-bounded problem for general distributive error metrics (Section 5), for this problem there *is* a significant complexity advantage to be gained by specializing our solution for the special variant of a Haar$^+$ tree, the CHH [Reiss et al. 2006] (see Sections 2.2.4 and 4.2). Hence, in this section we devise an algorithm that detects an optimal solution to the error-bounded CHH problem. As we show, such an optimal solution can be extracted in low polynomial time. The problem under consideration is defined in a *strong* version as follows:

Problem 6.1. Given a data vector $\mathbf{D}$ and an error bound ϵ for a (weighted) maximum-error metric $\mathcal{L}_\infty^w$, construct a CHH $\mathbf{H}$ that produces an approximation $\hat{\mathbf{D}}$ of $\mathbf{D}$, such that $\mathcal{L}_\infty^w(\|\mathbf{D} - \hat{\mathbf{D}}\|) \leq \epsilon$ and the number of occupied nodes B^* in $\mathbf{H}$ is minimized. Furthermore, of all B^*-term CHH representations satisfying ϵ, select one of minimal actual error $\epsilon^* \leq \epsilon$.

We call this version of the problem *strong* due to the *secondary* optimization requirement to choose an error-optimal representation among those that satisfy the given error bound in the minimum space.

When solving the space-bounded problem, we were interested in tabulating error values $E(i, v, b)$ on each triad C_i as a function of incoming value v and allocated space b. The allocated space parameter does not exist for the error-bounded problem, while the maximum-error bound ϵ is a universal parameter that applies on each individual estimated data value d_i; hence, it does not need to enter the recurrence. Thus, we are now interested in the behavior of an $S(i, v)$ function: the minimum space budget needed by a CHH node c_i and its descendants in order to satisfy the $\mathcal{L}_\infty^w$-error bound ϵ with incoming value v at c_i.

For a given i, $S(i, v)$ is defined for every $v \in \mathbb{R}$ and takes values in $\mathbb{N}$. The value range of $S(i, v)$ is delimited as follows.

THEOREM 6.2. *Let $s_i^* \in \mathbb{N}$ be the minimum value of $S(i, v)$ on a CHH node c_i, $v \in \mathbb{R}$. Then, $\forall v$, $S(i, v) \in \{s_i^*,\ s_i^* + 1\}$.*

PROOF. Let $\tilde{v}$ be an incoming value with which the minimum of $S(i, v)$ is obtained: $\forall v, S(i, v) \geq S(i, \tilde{v}) = s_i^*$. For that value c_i is unoccupied; if it were occupied by a nonzero value z^*, then this value z^* itself as incoming value would allow for an equivalent CHH of reduced space. Thus, for any other incoming value v', we may assign the value $\tilde{v}$ itself to c_i so as to produce the same incoming value for its descendants; the rest of the solution is maintained as with incoming value $\tilde{v}$. The assignment to c_i increases the number of nonzero terms in c_i and its descendants by 1. In effect, $\forall v, S(i, v) \in \{s_i^*,\ s_i^* + 1\}$. □

Theorem 6.2 implies that all possible incoming values $v \in \mathbb{R}$ to a node c_i can be grouped in two sets: (i) the set of values with which the optimal, minimum space $S(i, v) = s_i^*$ is achieved, and (ii) the rest, for which $S(i, v) = s_i^* + 1$. Each of these sets can be expressed as a union of intervals of $\mathbb{R}$. We now look at the computation of $S(i, v)$ more closely.

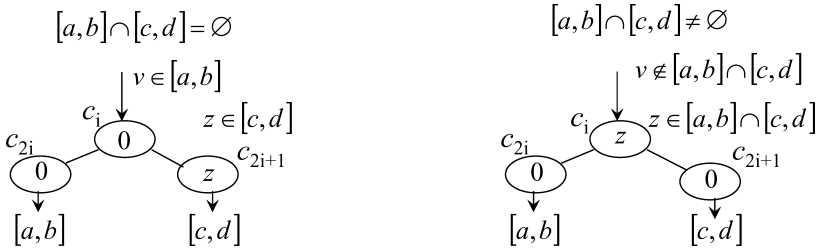


Fig. 7. Two cases in which $S(i, v) = 1$ in a next-to-bottom-level CHH node c_i.

At the bottom CHH level, the values of $S(i, v)$ are directly computed from the affected data. Each data item d_i with associated error weight w_i defines a *tolerance interval* $[d_i - \frac{\epsilon}{w_i}, d_i + \frac{\epsilon}{w_i}]$; approximation values within this interval satisfy the error bound ϵ at d_i. A CHH node at the next-to-bottom level approximates two data values, which define two tolerance intervals, say $[a, b]$ for the left-branch value and $[c, d]$ for the right-branch one. If $[a, b] \cap [c, d] \neq \emptyset$, then the minimum value of $S(i, v)$ is 0, for $v \in [a, b] \cap [c, d]$; in this case, by Theorem 6.2, the worst-case value of $S(i, v)$ is 1, obtained for $v \notin [a, b] \cap [c, d]$, as this value is corrected to optimal by a single nonzero value at c_i. Otherwise, the minimum of $S(i, v)$ is 1, obtained when $v \in [a, b] \cup [c, d]$, since a single nonzero value in the appropriate branch node suffices to satisfy the error bound ϵ; in that case, by Theorem 6.2, the worst-case value of $S(i, v)$ is 2, for $v \notin [a, b] \cup [c, d]$.

Figure 7 presents the two cases in which $S(i, v) = 1$ at a next-to-bottom-level node c_i with incoming value v; the left side of the figure depicts the former case (one of two variants), where a single nonzero coefficient at one of the children of c_i suffices to satisfy the error bound ϵ on both approximated data values at the branches of c_i; the right side shows the latter case, where a single nonzero coefficient assigned to c_i itself suffices for that purpose; in the former case, $[a, b] \cap [c, d] = \emptyset$ and $v \in [a, b] \cup [c, d]$; in the latter case, $[a, b] \cap [c, d] \neq \emptyset$ and $v \notin [a, b] \cap [c, d]$.

Putting it all together, $S(i, v)$ at a next-to-bottom-level CHH node c_i is defined as

$$S(i,v) = \begin{cases} 0, & [a,b] \cap [c,d] \neq \emptyset \ \wedge \ v \in [a,b] \cap [c,d] \\ \hline 1, \quad \vee & [a,b] \cap [c,d] \neq \emptyset \ \wedge \ v \notin [a,b] \cap [c,d] \\ & [a,b] \cap [c,d] = \emptyset \ \wedge \ v \in [a,b] \cup [c,d] \\ \hline 2, & [a,b] \cap [c,d] = \emptyset \ \wedge \ v \notin [a,b] \cup [c,d] \end{cases} \tag{6}$$

According to Eq. (6), in order to represent the full value range of $S(i, v)$ at a next-to-bottom-level CHH node c_i, we only need to store the set $\mathbf{P}_i$ such that $v \in \mathbf{P}_i \Leftrightarrow S(i, v) = s_i^*$, where s_i^* is the minimum value of $S(i, v)$ at c_i. If $[a, b] \cap [c, d] \neq \emptyset$, then $\mathbf{P}_i = [a, b] \cap [c, d]$, otherwise $\mathbf{P}_i = [a, b] \cup [c, d]$, hence $\mathbf{P}_i$ is a union of at most two distinct v-value intervals. For $v \notin \mathbf{P}_i$, it is inferred that $S(i, v) = s_i^* + 1$.

At subsequent CHH levels the computation proceeds recursively. $S(i, v)$ at a node c_i is defined from its values in the children nodes of c_i. Let c_{i_L} be the left child of c_i and c_{i_R} be its right child. The function S, applied on c_{i_L} for any incoming value $v \in \mathbb{R}$, assumes a minimum value $s_{i_L}^* \in \mathbb{N} | \forall v \mathbb{R}, S(i_L, v) \geq s_{i_L}^*$. Similarly, applied on c_{i_R}, S assumes the minimum value $s_{i_R}^* \in \mathbb{N} | \forall v \mathbb{R}, S(i_R, v) \geq s_{i_R}^*$. Then,

according to Theorem 6.2, $S(i_L, v) \in \{s^*_{i_L},\ s^*_{i_L}+1\}$ and $S(i_R, v) \in \{s^*_{i_R},\ s^*_{i_R}+1\}$. We assume that the computation of $S(i_L, v)$ ($S(i_R, v)$) has recursively returned a union of l (m) v-value intervals in which $S(i_L, v)$ ($S(i_R, v)$) achieves its minimum value $s^*_{i_L}$ ($s^*_{i_R}$); the assumption is valid in the next-to-bottom-level case.

Let $\mathbf{L}_i = \bigcup_{j=1}^{l} L_i$ ($\mathbf{R}_i = \bigcup_{j=1}^{m} R_i$) be the union of intervals returned for $S(i_L, v)$ ($S(i_R, v)$), i.e. $v \in \mathbf{L}_i \Leftrightarrow S(i_L, v) = s^*_{i_L}$ ($v \in \mathbf{R}_i \Leftrightarrow S(i_R, v) = s^*_{i_R}$). By analogy to the bottom-level case, if $\mathbf{L}_i \cap \mathbf{R}_i \neq \emptyset$, then the minimum value of $S(i, v)$ is $s^*_i = s^*_{i_L} + s^*_{i_R}$, obtained for $v \in \mathbf{L}_i \cap \mathbf{R}$; such an incoming value is itself an optimal incoming value for both subtrees of c_i. Otherwise, if $\mathbf{L}_i \cap \mathbf{R}_i = \emptyset$, then the minimum value of $S(i, v)$ is $s^*_i = s^*_{i_L} + s^*_{i_R} + 1$, obtained for $v \in \mathbf{L}_i \cup \mathbf{R}_i$; such values of v are optimal for one subtree of c_i and suboptimal (i.e., according to Theorem 6.2, requiring one space unit more than the minimum) for the other. In every case, according to Theorem 6.2, $S(i, v)$ obtains only two values over all the domain v. Hence, if $\mathbf{L}_i \cap \mathbf{R}_i \neq \emptyset$, then $S(i, v) \in \{s^*_{i_L} + s^*_{i_R}, s^*_{i_L} + s^*_{i_R} + 1\}$, otherwise, if $\mathbf{L}_i \cap \mathbf{R}_i = \emptyset$, then $S(i, v) \in \{s^*_{i_L} + s^*_{i_R} + 1, s^*_{i_L} + s^*_{i_R} + 2\}$.

Putting it all together, $S(i, v)$ is expressed as

$$S(i,v) = \begin{cases} s^*_{i_L} + s^*_{i_R}, & \mathbf{L}_i \cap \mathbf{R}_i \neq \emptyset \ \wedge\ v \in \mathbf{L}_i \cap \mathbf{R}_i \\ \hline s^*_{i_L} + s^*_{i_R} + 1, \quad \vee & \mathbf{L}_i \cap \mathbf{R}_i \neq \emptyset \ \wedge\ v \notin \mathbf{L}_i \cap \mathbf{R}_i \\ & \mathbf{L}_i \cap \mathbf{R}_i = \emptyset \ \wedge\ v \in \mathbf{L}_i \cup \mathbf{R}_i \\ \hline s^*_{i_L} + s^*_{i_R} + 2, & \mathbf{L}_i \cap \mathbf{R}_i = \emptyset \ \wedge\ v \notin \mathbf{L}_i \cup \mathbf{R}_i \end{cases} \tag{7}$$

Equation (7) defines $S(i, v)$ recursively throughout a CHH. Again, in order to represent the full value range of $S(i, v)$, we only need to store the set $\mathbf{P}_i$ such that $v \in \mathbf{P}_i \Leftrightarrow S(i, v) = s^*_i$. If $\mathbf{L}_i \cap \mathbf{R}_i \neq \emptyset$, then $\mathbf{P}_i = \mathbf{L}_i \cap \mathbf{R}_i$, otherwise $\mathbf{P}_i = \mathbf{L}_i \cup \mathbf{R}_i$. For $v \notin \mathbf{P}_i$, it is inferred that $S(i, v) = s^*_i + 1$. This representation of the value range of $S(i, v)$ verifies the inductive step of our approach; we assumed that the value ranges of $S(i_L, v)$ and $S(i_R, v)$ were thusly represented at two children nodes, and we have shown that $S(i, v)$ is then thusly represented at the parent node; the assumption holds at the bottom CHH level; hence, this representation is inductively propagated through the CHH in a bottom-up fashion with our recursive scheme.

The set $\mathbf{P}_i$ is appropriately stored as a union of subintervals; each of these subintervals has the form $[m, M]$; in the general case, $[m, M]$ arises from the intersection of *tolerance intervals* of the form $[d_i - \frac{\epsilon}{w_i}, d_i + \frac{\epsilon}{w_i}]$, in which a certain pair of data items d_j, d_k define the limits m, M. For the sake of solving the *strong* version of the problem, each subinterval $[m, M]$ is stored along with appropriate accompanying information. This track-keeping information includes:

(1) The data items d_m, d_M defining the subinterval's limits m, M; in case more than one unequal data item defines the same limit (as a variation in their associated weights may allow for), then the one most distant from this limit, hence of smallest weight, is the critical one (since the error of the data item of larger weight is more rapidly decreased as the approximation value moves away from the limit); in the case of maximum absolute error, d_m, d_M are the minimum and maximum values among the data whose tolerance intervals intersection has produced $[m, M]$ (i.e., the data under the scope of c_i that can be approximated by a value in $[m, M]$).

(2) The optimal incoming/assigned value to c_i, $v^* \in [m, M]$, that is, the value that minimizes the employed maximum-error metric in the data approximation; for maximum absolute error, $v^* = \frac{d_m+d_M}{2}$; in general, for a *weighted* maximum-error metric, $v^* = \frac{w_m d_m + w_M d_M}{w_m+w_M}$, where w_m (w_M) is the weight associated with d_m (d_M); in the case of maximum relative error with a sanity bound S, v^* is calculated according to the case analysis of Guha et al. [2004].

(3) The optimum error e^* achieved by v^*.

The root case $S(0, 0)$ of the CHH recurrence is defined like the regular recursive case (a CHH has no special root coefficient as a Haar$^+$ tree does). Hence, the call of $S(0, 0)$ derives the minimum-space solution. In addition, thanks to the track-keeping of minimum actual errors, it returns the minimum error that can be achieved by a CHH of that minimum space, in a *secondary* optimization. In order to retrieve this optimal CHH, we only need to trace backwards through the choices made at each node after the solution at the top of the CHH has been established; when a nonzero value z has to be assigned to a node c_i, the error-minimizing value v^* is chosen. Furthermore, we may follow the space-efficiency paradigm suggested by Guha [2008]; after the solution is established at the root node, we solve the two half-size problems at the two subtrees of the root and recursively recompute the respective solutions, by the same strategy. This approach stores at most the value-range, assigned-value and error information on only $\log n + 1$ CHH nodes concurrently (one at each level on a root to leaf path, plus one for the last node's sibling).

Complexity Analysis. The space required to store the set (union of intervals) $\mathbf{P}_i$ representing the value range of $S(i, v)$ for a node c_i grows with the CHH level in which c_i resides. In a next-to-bottom-level node c_i, two (2) distinct value intervals need to be stored in the worst case, one for each approximated data value. In the worst case, a parent node c_i at the next level above receives a union of two intervals from both its children nodes c_{i_L} and c_{i_R} as the sets of incoming values $\mathbf{L}_i$, $\mathbf{R}_i$ that achieve the minimum space. Hence, in the worst case, four (4) intervals need to be stored in order to represent the value range of $S(i, v)$ at c_i. Inductively, it follows that a node c_i at level ℓ_i of the CHH, counting from the bottom, requires $O(2^{\ell_i})$ space in order to store the union of intervals $\mathbf{P}_i$ that represents the value range of $S(i, v)$ at c_i. Unions of intervals are kept in sorted order from the bottom level onwards; thus, union and intersection operations are conducted in a merging fashion in linear time. Thus, a node c_i at level ℓ_i requires $O(2^{\ell_i})$ time to compute its sorted union of intervals. At most $\log n + 1$ value range arrays need to be concurrently stored, hence the space complexity is $O(\sum_{\ell=0}^{\log n} 2^\ell) = O(n)$. Similarly, level ℓ contains $\frac{n}{2^{\ell+1}}$ nodes, hence the total time complexity is $O(\sum_{\ell=0}^{\log n} 2^\ell \frac{n}{2^{\ell+1}}) = O(n \log n)$.

In conclusion, our algorithm for the $\mathcal{L}_\infty^w$-bounded longest-prefix-match CHH problem achieves the optimal-space solution, and also secondarily minimizes the *actual* error within that space, in *low polynomial* time.

6.1.1 *Testing Error Optimality.* Even though our algorithm minimizes the actual $\mathcal{L}_\infty^w$-error $\bar{\epsilon}$ within the space $\bar{B}$ required to satisfy the given error bound ϵ, it is still useful, for our purposes, to determine whether the derived $\mathcal{L}_\infty^w$-error

value $\bar{\epsilon}$ is also optimal for other space budgets $B > \bar{B}$, as it may be. The following lemma assists to that end.

LEMMA 6.3. *Let* $\mathbf{H}$ *be the* $\bar{B}$*-term CHH synopsis of* $\mathbf{D}$ *for the* $\mathcal{L}_\infty^w$*-error bound* ϵ *returned by our scheme,* $\bar{\epsilon} \leq \epsilon$ *be the minimized actual* $\mathcal{L}_\infty^w$*-error of* $\mathbf{H}$*, and* ϵ^* *be the* minimum $\mathcal{L}_\infty^w$*-error of a CHH synopsis of* $\mathbf{D}$ *in* $B > \bar{B}$ *buckets. Moreover, let* $\tilde{\mathbf{H}}$ *be the* $\tilde{B}$*-term CHH synopsis of* $\mathbf{D}$ *returned by a variant of our scheme in which the condition to be satisfied on each approximated value* d_i *is* $\mathcal{L}_\infty^w(|\hat{d}_i - d_i|) < \bar{\epsilon}$*, that is, error values less than but not equal to* $\bar{\epsilon}$ *are allowed (*$<$ *instead of* $\leq$*), hence the calculated value intervals are open instead of closed. Then* $\bar{\epsilon} = \epsilon^*$ *if and only if* $\tilde{B} > B$*.*

PROOF. By definition, $\bar{\epsilon}$ is the least error that can be achieved in $\bar{B}$ space, while $\tilde{B}$ is the least space required to achieve error *less* than $\bar{\epsilon}$. If it were $\tilde{B} \leq \bar{B}$, then error less than $\bar{\epsilon}$ would be achievable in $\bar{B}$ space; hence, by reduction *ad absurdum*, it must be $\tilde{B} > \bar{B}$. If $\tilde{B} > B$, then any CHH synopsis of $\mathbf{D}$ with $\mathcal{L}_\infty^w$-error less than $\bar{\epsilon}$ requires more than B buckets, hence $\bar{\epsilon}$ is equal to the B-optimal error ϵ^* and $\mathbf{H}$ is also the optimal CHH synopsis in B space (in fact, the optimal synopsis for all space budgets from $\bar{B}$ to $\tilde{B} - 1$). Formally, $\tilde{B} > B \Rightarrow \bar{\epsilon} = \epsilon^*$. In reverse, assume that $\bar{\epsilon}$ is in fact equal to the B-optimal error ϵ^*; then it cannot be $\tilde{B} \leq B$, as then an error value less than the assumed B-optimal would indeed be achievable in B space units, contradicting our assumption; hence $\tilde{B} > B$. Formally, $\bar{\epsilon} = \epsilon^* \Rightarrow \tilde{B} > B$. In conclusion, $\bar{\epsilon} = \epsilon^* \Leftrightarrow \tilde{B} > \bar{B}$. □

In the next section, we show how our solution to the error-bounded problem can provide a beneficial shortcut towards solving the dual, space-bounded problem that we are interested in.

6.2 Solving the Space-Bounded CHH Problem

Our longest-prefix-match CHH construction algorithm for maximum-error metrics invokes the solution to the error-bounded problem; that is, it computes $S(0, 0)$ for a given error bound ϵ by binary search in the domain of ϵ. In contrast to the heuristics of Reiss et al. [2006], this method eschews both a tabulation with respect to space b as well as a tabulation of lowest occupied ancestor nodes. Hence, it gains in terms of both space- and time-efficiency (see Table III, Section 8.1, which follows). Most significantly, it achieves the *optimal* solution to the space-bounded longest-prefix-match CHH problem for any maximum-error metric.

The seed value of the fluctuating error bound ϵ for the target maximum-error metric $\mathcal{L}_\infty^w$ is obtained as the $\mathcal{L}_\infty^w$-error corresponding to a synopsis of the B largest Haar wavelet decomposition coefficients by absolute value, easily computed in $O(n \log B)$ time. For the sake of structure consistency, we could use the nonoverlapping CHH partitioning heuristic of Reiss et al. [2006]; still, its $O(nB \log B)$ time complexity may supersede the $O(n \log n)$ time complexity of our algorithm for the error-bounded problem itself, and hence undermine the overall complexity of our approach. On the other hand, the simple Haar

Algorithm IndirectCHH(B)
Input: space bound B, n-data vector $[d_0, \ldots, d_{n-1}]$
Output: $\mathcal{L}_\infty^w$-error optimal B-sized Haar$^+$ synopsis
1. $\epsilon_u = \mathcal{L}_\infty^w$-error of B-largest-term synopsis;
2. $e_{low} = 0$; $e_{high} = \epsilon_u$;
3. **while** (not finished)
4. $e_{mid} = (e_{high} + e_{low})/2$;
5. $\bar{B} = S(0, 0)$ for error $\leq e_{mid}$ producing synopsis $\mathbf{H}$;
6. $\bar{\epsilon}$ = actual $\mathcal{L}_\infty^w$-error of $\mathbf{H}$; /* $\bar{\epsilon} \leq \epsilon$ */
7. **if** ($\bar{B} < B$)
8. $\tilde{B} = S(0, 0)$ for error $< \bar{\epsilon}$;
9. **if** ($\tilde{B} > B$)
10. finished := 1; /* optimal result found */
11. **else** $e_{high} = \bar{\epsilon}$;
12. **else if** ($\bar{B} > B$) $e_{low} = e_{mid}$
13. **else** finished := 1; /* $\bar{B} = B$ */
14. **return** $\mathbf{H}$;

Fig. 8. Indirect CHH synopsis construction.

wavelet heuristic provides a *reasonable* seed for the error bound in practice without a time complexity overhead. This seed is *not* provably an upper bound for the error of the optimal CHH for the target maximum-error metric. Still, given the poor performance of this heuristic for maximum-error metrics (sufficiently documented in Garofalakis and Gibbons [2004], Garofalakis and Kumar [2005] and Karras and Mamoulis [2005]), it is quite larger in practice; should an exceptional case occur, the binary search accommodates it by doubling the tested error bound value. After the seed error bound is calculated, the solution to the *strong* error-bounded problem is repetitively invoked by binary search on the error bound ϵ. This solution minimizes the error within the optimal space; hence, the binary search is bound to yield the optimal error when it converges to the space budget B. However, a space budget *less* than B may also achieve the B-optimal error. Therefore, in order to ascertain the convergence of the search, our procedure performs an *optimality test*, as defined in Section 6.1.1, for each examined error bound ϵ that requires *less than* B space with actual minimized error $\bar{\epsilon}$ within that space; that is, it computes the value of $S(0, 0)$, for which the condition to be satisfied is not $\mathcal{L}_\infty^w(\|\mathbf{D} - \hat{\mathbf{D}}\|) \leq \bar{\epsilon}$, but $\mathcal{L}_\infty^w(\|\mathbf{D} - \hat{\mathbf{D}}\|) < \bar{\epsilon}$. If this variant requires *more* than B space, then the search can safely terminate; otherwise, it proceeds. Hence, the search terminates when the guessed error bound reaches a value that requires a synopsis of $\bar{B} \leq B$ space with actual error $\bar{\epsilon}$, while the optimality test indicates than any error bound $\epsilon < \bar{\epsilon}$ requires $\tilde{B} > B$ space. Moreover, during the binary search iterations, whenever the tested error bound ϵ is decreased, the actual error $\bar{\epsilon}$ derived for the previous bound is taken into account for determining the new bound. Figure 8 shows a pseudocode for this IndirectCHH algorithm.

Complexity Analysis. The binary search process IndirectCHH adds an $O(\log \frac{\mathcal{E}}{r})$ factor to the time complexity, where $\mathcal{E}$ is the seed of the search, bounded by the

largest input value, and $r > 0$, the precision by which the given machine represents real numbers. After the binary search has converged, a space-efficient synopsis construction process is called, analogous to that of Section 5.4.1; it reenters the problem with the final error bound in the two branches of c_1 and recomputes the respective solutions recursively thereafter. Given that the basic time complexity of the error-bounded solution is $O(n \log n)$, this space-efficient synopsis construction process with subproblem reentry takes $O(\sum_{\ell=0}^{\log n} 2^\ell \ell \frac{n}{2^\ell}) = O(n \log^2 n)$ time. In effect, the total time complexity for computing a B-term longest-prefix-match CHH that minimizes $\mathcal{L}_\infty^w$ is $O(n \log n(\log \frac{\mathcal{E}}{r} + \log n))$. The former log term in the parenthesis expresses the cost of the binary search; the latter expresses the cost of constructing the final B-term CHH in a space-efficient manner after the optimal error value $\frac{\mathcal{E}}{r}$ has been established. This complexity absorbs the $O(n \log B)$ term for determining the seed of the search. Under the assumption that the $\log \frac{\mathcal{E}}{r}$ factor (i.e., the largest input value) does not grow with n, this runtime is lower than the $O(nB \log n \log B)$ runtime of the winner greedy heuristic in Reiss et al. [2006]. Besides, the space requirement of $O(n)$ is lower than the $O(B \log^2 n + n)$ space of that heuristic. Table III (Section 8.1) presents the respective complexities of the more demanding k-holes heuristic of Reiss et al. [2006] as well. In conclusion, IndirectCHH achieves a low-polynomial-time *optimal* solution to the space-bounded longest-prefix-match CHH partitioning problem for maximum-error metrics; hence, the difficulty of choosing an optimal longest-prefix-match partitioning function, correctly identified in Reiss et al. [2006], can indeed be overcome in the case of maximum-error metrics.

6.3 The Question of Convergence

We now study the question of the convergence of our algorithm to the optimal error result in more detail. This question is most interesting for error functions whose values may require higher precision than that by which the given data themselves are expressed. For a maximum-error function such as the maximum absolute error $\mathcal{L}_\infty$, the optimal error result has itself no more precision than that by which the input data themselves are expressed; it can in fact be expressed as the semi-difference of two such data values. Still, this is not the case for other maximum-error functions; for example, an optimal maximum relative error result may be a recurring number (such as $\frac{2}{3} = 0.\bar{6}_{10} = 0.\bar{1}\bar{0}_2$, a recurring number in both decimal and binary). Hence, a question of whether our indirect approach will actually converge to the optimal error result emerges.

We first emphasize that, even if our indirect approach had to calculate the error result based on binary division operations themselves, this requirement would *not* render the error result less exact than that of a (hypothetical) exact algorithm that would compute it directly. That is, such a hypothetical algorithm would itself compute the optimal error with a precision of r, as allowed by the given machine. Our binary-search-based method would also converge to the optimal result with as much precision as the given machine allows. Hence the indirect approach affords the *same* precision as an exact solution on the same finite-precision machine.

Still, as we discussed, the convergence of our indirect approach to the optimal error value is *not* expected to be achieved through the binary division operations themselves, but even more robustly, thanks to the secondary optimization, which calculates the *actual* minimum error value $\bar{\epsilon}$ of a minimum-space synopsis for an error bound ϵ. In effect, the binary search terminates as soon as it reaches a tested error bound ϵ that requires the same amount of space $\bar{B} \leq B$ as the actual optimal error ϵ^* within space B; we use this formulation in order to allow for the possibility that the B-optimal error is also achieved in less than B space units. In other words, the binary search does not have to converge to the exact error value ϵ^* itself, but to *any* value within the appropriate interval.

Formally, let $f : \mathbb{R}_+ \rightarrow \mathbb{N}$ be the function that returns the minimum space $B = f(\epsilon) \in \mathbb{N}$ needed to satisfy the error bound $\epsilon \in \mathbb{R}_+$, according to a given maximum-error metric, in a given summarization problem. Then f is a non-increasing, piecewise-constant function of ϵ, mapping half-closed intervals of the form $[\epsilon^B_{min}, \epsilon^{<B}_{min})$ to natural numbers B, such that ϵ^B_{min} is the minimum error that can be achieved in B space units and $\epsilon^{<B}_{min}$ the minimum error achievable in *less* than B space units; equivalently, B is the minimum space needed to satisfy an error bound ϵ *if and only if* $\epsilon \in [\epsilon^B_{min}, \epsilon^{<B}_{min})$. We call the interval $[\epsilon^B_{min}, \epsilon^{<B}_{min})$ the *error interval* of B. The following theorem proves the convergence of our algorithm *without regard* to the precision of the machine it runs on.

THEOREM 6.4. *Let $B^* \leq B$ be the minimum space in which the same minimum error ϵ^* as in the given space budget B can be achieved in a given summarization problem. Then* IndirectCHH *converges to the optimal error result for that problem in $O(\log \frac{\mathcal{E}}{r_{B^*}})$ iterations, where $\mathcal{E}$ is the seed error value of the binary search and $r_{B^*} = \epsilon^{<B^*}_{min} - \epsilon^{B^*}_{min}$ is the size of the error interval of B^*.*

PROOF. The binary search need only reach any value of the error bound ϵ in the error interval of B^*. As soon as such a value of ϵ is reached, the actual minimum error ϵ^* in space B^* is calculated, and the optimality test is positive, since, by definition, error bounds less than ϵ^* cannot be satisfied within the given space budget B; hence the algorithm terminates. In effect, the binary search does not need proceed to precision higher than that defined by the size r_{B^*} of the error interval of B^*; *ergo*, IndirectCHH converges to the optimal error result in $O(\log \frac{\mathcal{E}}{r_{B^*}})$ iterations. □

Figure 9 depicts a graphic representation of the intervals defined by function f and the state of affairs before the algorithm's termination.

In conclusion, IndirectCHH converges to the optimal error result with as much precision as the given machine allows, and would still robustly converge to it even on an *ideal* machine that allowed for infinite decimal-point (or binary-point) precision.

6.4 An Approximate Solution to the Error-Bounded Haar$^+$ Problem

The methodology of Section 6.1 can be applied to the error-bounded synopsis problem in the general Haar$^+$ case as well. However, due to the expectations raised by head coefficients, the number of distinct value intervals that needs to

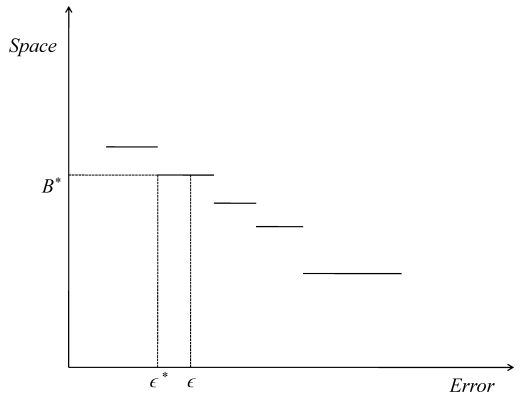


Fig. 9. Function f mapping error intervals to space budgets, and terminating the state of affairs.

be stored per triad grows super-exponentially with the Haar$^+$ tree level, *ergo* exponentially with the data set size. Hence, the worst-case complexity of this solution, presented in the Electronic Appendix that is accessible in the ACM Digital Library, is $O(3^n)$. The exponentially increasing number of intervals this algorithm stores may be scaled down by overlaps among such intervals in practice; still the scalability of this algorithm cannot be guaranteed. Therefore, in this section we present a fully polynomial-time approximation scheme (FPAS) for the error-bounded Haar$^+$ synopsis problem, along the lines of the approximation methodology applied in Section 5 for its space-bounded counterpart. This solution allows for more efficient synopsis construction for maximum-error metrics than the general-error algorithm of Section 5. We start out by defining a *strong* version of the error-bounded problem:

Problem 6.5. Given a data vector $\mathbf{D}$ and an error bound ϵ for a (weighted) maximum-error metric $\mathcal{L}_\infty^w$, construct a Haar$^+$ representation $\mathbf{H}$ of $\mathbf{D}$ that produces an approximation $\hat{\mathbf{D}}$, such that $\mathcal{L}_\infty^w(\|\mathbf{D} - \hat{\mathbf{D}}\|) \leq \epsilon$ and the number of occupied nodes B^* in $\mathbf{H}$ is minimized. Furthermore, of all B^*-nonzero-term Haar$^+$ representations satisfying ϵ, select the one with the minimal actual error $\epsilon^* \leq \epsilon$.

This problem can be solved by a dynamic-programming recurrence analogous to the one introduced for the general space-bounded problem in Section 5.2.

In order to construct our solution, we need to systematically explore the space of possible retained coefficients and values assigned to them. In order to delimit the computational cost, we again quantize the (real-valued) domains of possible incoming values v and, for each v, possible values assigned to the head and left/right supplementary coefficients, z_h, z_l, z_r (see Table I), into multiples of a small resolution step δ. Based on this quantization, our algorithm considers all possible values of v and z_h, z_l, z_r, for a triad C_i. In a bottom-up process, it determines the optimal value to assign to C_i for v. The next section outlines some lemmas that establish upper and lower bounds for these domains.

6.4.1 *Delimiting the Value Domains*

LEMMA 6.6. *Let m_i be the minimum and M_i the maximum individual data values under the scope of triad C_i and $v \in \mathcal{S}_i$ be a possible incoming value at C_i for which the maximum error bound ϵ is satisfied, and $\bar{\epsilon} = \frac{\epsilon}{\min_{j \in I}\{|w_j|\}}$, where I is the interval under the scope of C_i; then $v \in [m_i - \bar{\epsilon}, M_i + \bar{\epsilon}]$.*

PROOF. A root-to-leaf path in the Haar$^+$ tree reconstructs a data value it leads to, while the value v is reconstructed from the root up to triad C_i. According to Theorem 4.2, at most one coefficient per triad needs be occupied. We will show that there exists a path from C_i to a leaf in which v is not increased (decreased). We analyze the cases as follows:

(1) A triad with a nonzero head coefficient, hence zero-valued supplementary coefficients, decreases its incoming value in one direction and increases it in another. Hence there exists a direction in which the incoming value is not increased (decreased).
(2) A triad with a zero head coefficient has at least one zero supplementary coefficient and leaves the incoming value unchanged in the corresponding direction, hence there exists a direction in which the incoming value is not increased (decreased).

Hence, there exist reconstructed values $\hat{d}_j$, $\hat{d}_k$, such that $\hat{d}_k \leq v \leq \hat{d}_j$. However, all reconstructed values should lie within the range defined by the extrema under the scope of C_i, extended by the error tolerance $\bar{\epsilon}$, that is, $\hat{d}_j \leq M_i + \bar{\epsilon}$ and $\hat{d}_k \geq m_i - \bar{\epsilon}$, thus $v \in [m_i - \bar{\epsilon}, M_i + \bar{\epsilon}]$. □

Lemma 6.6 implies that the set $\mathcal{S}_i \subset \mathbb{R}$ of potential incoming values at triad C_i consists of the multiples of δ in the interval $[m_i - \bar{\epsilon}, M_i + \bar{\epsilon}]$; thus, $|\mathcal{S}_i| \leq \lfloor \frac{M_i - m_i + 2\bar{\epsilon}}{\delta} \rfloor + 1 = O(\frac{\Delta}{\delta})$, where $\Delta = M - m$ (see Table I). We now demarcate the values assigned to the head coefficient.

LEMMA 6.7. *Let $v \in \mathcal{S}_i \subset \mathbb{R}$ be a possible incoming value at C_i and $z_h \in \mathcal{S}^v_{i,H} \subset \mathbb{R}$ be a value that can be assigned to the head coefficient of C_i for incoming value v, satisfying the individual-data error bound ϵ; then $|z_h| \leq \min\{M_i + \bar{\epsilon} - v, v - (m_i - \bar{\epsilon})\}$.*

Lemma 6.7 implies that the finite set of possible assigned values we have to examine for the head coefficient at C_i is $\mathcal{S}^v_{i,H}$, where $|\mathcal{S}^v_{i,H}| = O(\frac{\Delta}{\delta})$. The set $\mathcal{S}^v_{i,L}$ ($\mathcal{S}^v_{i,R}$) of possible values assigned to the left (right) supplementary coefficients of triad C_i can be delimited in a similar fashion. We devise our dynamic programming solution based on this delimitation of the search space.

6.4.2 *Deriving the Answer.*

Let $S(i, v)$ be the minimum space that should be allocated to triad C_i and its descendants in the Haar$^+$ tree in order for the given error bound ϵ to be satisfied with incoming value v at C_i. As in Section 5, the solution is derived with a bottom-up dynamic programming recursive scheme. However, now the tabulation is simpler; no distributions of allocated space need to be examined, we only tabulate over bucket values. This convenience renders both the time and, most significantly, the space complexity of

this algorithm lower than the one of Section 5.2. The solution is established after $S(0,0)$ is computed. Again, at each visited triad C_i the algorithm calculates an array A from the precalculated arrays L and R of its children triads C_{i_L}, C_{i_R}. An entry $A[v]$ now contains: (i) the δ-optimal value z to assign to one of the coefficients in C_i (possibly none); (ii) the minimum space required; and (iii) the actual minimized error thus obtained (needed to solve the strong version of the problem). A MinSpace procedure computes $S(i,v)$ recursively:

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

$$S(i,v) = \min \left\{ \begin{array}{l} \min\limits_{z_h \in \mathcal{S}^v_{i,H}} \{S(i_l, v + z_h) + S(i_r, v - z_h) + (z_h \neq 0)\} \\ \min\limits_{z_l \in \mathcal{S}^v_{i,L}} \{S(i_l, v + z_l) + S(i_r, v) + (z_l \neq 0)\} \\ \min\limits_{z_r \in \mathcal{S}^v_{i,R}} \{S(i_l, v) + S(i_r, v + z_r) + (z_r \neq 0)\} \end{array} \right\} \quad (8)$$

The above recurrence follows the same pattern as that for the tabulation of error in Section 5.2, and also observes the redundancy property proved in Theorem 4.2. It differs in the absence of a b parameter and in the inclusion of the boolean terms denoting the space occupied by nonzero coefficients. Moreover, additional care is taken in the recursion in order to select the error-optimal among all solutions minimizing the space requirement; that is, out of all solutions (that is, value assignments) that achieve the optimal space result, the one (or one of those) that also minimizes the error achieved thereby is selected as the optimal assignment; hence the strong version of the problem is solved. A recursive procedure that derives the δ-optimal space result, and the secondary δ-optimal error within that space, without constructing the synopsis itself is defined.

Complexity Analysis. The array A computed by MinSpace on a triad C_i holds $|\mathcal{S}_i|$ entries, one for each possible incoming value, hence its size is $O(\frac{\Delta}{\delta})$; and at each triad C_i and for each $v \in \mathcal{S}_i$, the loop through all possible assigned values needs $O(\frac{\Delta}{\delta})$ time. In conclusion, the time complexity of MinSpace is $O((\frac{\Delta}{\delta})^2 n)$. And, since at most $\log n + 1$ arrays need to be concurrently stored, the space complexity is $O(\frac{\Delta}{\delta} \log n + n)$, where n stands for the storage of the data.

6.5 Solving the Space-Bounded Haar$^+$ Problem

Our Haar$^+$ synopsis algorithm for maximum-error metrics invokes the MinSpace module by binary search in the domain of error. This method avoids a tabulation with respect to space b, hence it pays in terms of both space- and time-efficiency. As in Section 6.2, the seed value of the fluctuating error bound ϵ for the target maximum-error metric $\mathcal{L}^w_\infty$ is obtained as the $\mathcal{L}^w_\infty$-error corresponding to the synopsis of B largest Haar decomposition coefficients by absolute value, easily computed in $O(n \log B)$ time. Thereafter, the MinSpace procedure is repeatedly invoked with binary search on the error bound value ϵ. Again, our solution to the *strong* error-bounded problem minimizes the error within the δ-optimal space; hence, this binary search procedure is bound to yield the δ-optimal error when it converges to the space budget B. However, a space budget *less* than B may also achieve the B-optimal error. Therefore, in order to ensure the convergence of

the search, our procedure also performs an *optimality test*, analogous to that of Section 6.1.1, for each examined error bound ϵ that requires *less* than B space; if this test is positive, then the search can safely terminate; otherwise, it proceeds. The search terminates when the guessed error bound reaches a value that either requires a synopsis of exactly B space, or requires a synopsis of $\bar{B} < B$ space and actual error $\bar{\epsilon}$, while the optimality test indicates than any error bound $\epsilon < \bar{\epsilon}$ requires $\tilde{B} > B$ space. When the tested bound ϵ is decreased, the minimum error derived for the previous bound is taken into account for determining the new bound. As per the convergence of the algorithm, the discussion of Section 6.3 applies. This IndirectHaar$^+$ algorithm is analogous to the IndirectCHH algorithm of Section 6.2 (Figure 8).

Complexity Analysis. As in Section 6.2, the binary search process IndirectHaar$^+$ adds an $O(\log \frac{\mathcal{E}}{r})$ factor to the time complexity, where $\mathcal{E}$ is the seed of the search, bounded by the largest input value, and $r > 0$ the resolution with which the given machine represents real numbers. After the binary search has converged, a space-efficient synopsis construction process is called, analogous to that of Section 5.4.1; it reenters the problem with the final error bound in the two branches of C_1 and recomputes the respective solutions recursively thereafter. Setting ℓ as the Haar$^+$ tree level, the construction time becomes $O((\frac{\Delta}{\delta})^2 \sum_{\ell=0}^{\log n} 2^\ell \frac{n}{2^\ell}) = O((\frac{\Delta}{\delta})^2 n \log n)$. In effect, the total time complexity for computing a B-term Haar$^+$ synopsis that minimizes $\mathcal{L}_\infty^w$ is $O((\frac{\Delta}{\delta})^2 n(\log \frac{\mathcal{E}}{r} + \log n))$. The former log term expresses the cost of the binary search, while the latter one expresses the cost of constructing the final B-term synopsis in a space-efficient manner after the optimal error value ϵ^* has been established. This complexity absorbs the $O(n \log B)$ term for determining the seed of the search. Under the assumption that the $\log \frac{\mathcal{E}}{r}$ factor (i.e., the largest input value) does not grow with n, this runtime is decisively lower than the $O((\frac{\Delta}{\delta})^2 n \log n \log^2 B)$ runtime of the general-case space-efficient (Direct) algorithm applied on a maximum-error metric (Section 5.4.1). Moreover, unless[9] $n \gg B^{\log B}$, it is lower than its $O((\frac{\Delta}{\delta})^2 n \log^2 B)$ *basic* runtime too. The space requirement of $O(\frac{\Delta}{\delta} \log n + n)$ is lower than the $O(\frac{\Delta}{\delta} B \log \frac{n}{B} + n)$ space of the space-efficient Direct algorithm. In conclusion, IndirectHaar$^+$ has better asymptotic behavior than its Direct counterpart in both time and space, while its complexities are independent of the B parameter.

7. MULTIDIMENSIONAL EXTENSION

The efficient handling of multidimensional data is a major challenge for data summarization algorithms. Moreover, hierarchical summarization techniques have an innate advantage over histograms in this area; that is, the construction of an optimal histogram of *arbitrary* nonoverlapping rectangular buckets in more than one dimension is an NP-hard problem [Muthukrishnan et al. 1999]. Algorithms constructing multidimensional histograms are either heuristics [Muralikrishna and DeWitt 1988; Poosala and Ioannidis 1997; Aboulnaga and Chaudhuri 1999; Bruno et al. 2001; Deshpande et al. 2001; Thaper et al.

[9]The constraint is verified for reasonable $\frac{B}{n}$ ratios; e.g. for $B = 16$, $B^{\log B} = 65536$.

2002; Srivastava et al. 2006] or provide solutions with approximation guarantees for limited forms of the problem, in which the arbitrariness of bucket sizes and positions in constrained [Khanna et al. 1997; Muthukrishnan and Suel 2005; Furfaro et al. 2005]. On the other hand, hierarchical structures and their accompanying algorithms are more conveniently extended to multiple dimensions, as they are based on a fixed, nonarbitrary hierarchy. Thus, past hierarchical summarization techniques [Chakrabarti et al. 2001; Muthukrishnan and Strauss 2003b; Garofalakis and Gibbons 2004; Garofalakis and Kumar 2005; Guha and Harb 2008; Reiss et al. 2006] have been extended to the multidimensional case. In this section, we propose a multidimensional extension of the Haar$^+$ tree.

7.1 The Multidimensional Haar Transform

The one-dimensional definition of the Haar wavelet transform has been extended to multidimensional data arrays by two distinct methods, namely the *standard* multidimensional Haar wavelet transform, used in Vitter and Wang [1999], and the *nonstandard* one, used in Chakrabarti et al. [2001]. The standard method is based on iterative applications of the one-dimensional transform; the Haar wavelet coefficients extracted by the application of the decomposition along the rows of one dimension in one step are themselves treated as data and decomposed along another dimension in its next step [Vitter and Wang 1999]. In contrast, the nonstandard method operates directly in the multidimensional domain and constructs a decomposition in one pass [Chakrabarti et al. 2001; Garofalakis and Gibbons 2004]. Moreover, thanks to its regularity, the nonstandard Haar transform allows for concise representation of sign information [Chakrabarti et al. 2001]. Thus, past Haar wavelet-based synopsis construction schemes have opted for the nonstandard method [Garofalakis and Gibbons 2004; Garofalakis and Kumar 2005; Guha and Harb 2008]. A further argument in favor of the nonstandard method is that it preserves the rationale for the use of the Haar wavelet transform in the first place; namely, that a large number of the coefficients (that is, differences) extracted from neighboring (average) values, are likely to be small in magnitude, hence incur small error when truncated [Matias et al. 1998]. This state of affairs is preserved in the nonstandard method, which operates on data values and their averages in successive layers of detail; in contrast, the standard method operates on the data *differences* themselves from its second step onwards.

The basic step of the nonstandard Haar transform in the two-dimensional case is shown in Figure 10. In this case, a two-dimensional array of four values $\{a, b, c, d\}$ is decomposed to their average $V = \frac{a+b+c+d}{4}$ and *three* decomposition coefficients, $A = \frac{a+b-c-d}{4}$, $B = \frac{a-b-c+d}{4}$, $C = \frac{a-b+c-d}{4}$. The original data can be reconstructed in terms of the average and these coefficients by addition and subtraction: $a = V + A + B + C$, $b = V + A - B - C$, $c = V - A - B + C$, $d = V - A + B - C$. The figure depicts, along arrows, the set of the three operations that have be performed between the *incoming value* V to a decomposition node and the three *stored coefficients* in that node, A, B, C respectively, in order to reconstruct the original data values.

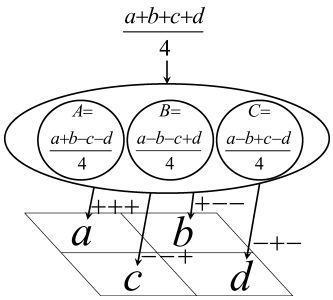


Fig. 10. Two-dimensional nonstandard Haar wavelet decomposition.

The full nonstandard Haar decomposition of a (two-dimensional) $2^m \times 2^m$ data array A is computed by recursively applying the basic decomposition step throughout the array at successive levels of resolution. The overall average results derived from quadruplets of values at one level are collected in the same quadrant of the transform array; the process is recursively reapplied on this quadrant in the next resolution level, where the collected averages play the role of data values. The same process is extended to higher dimensionality. One way of conceptualizing the d-dimensional nonstandard Haar transform is to think of a 2^d hyper-box being shifted across the data array, performing averaging and differencing, and distributing the results to appropriate hyper-quadrants in the transform array [Chakrabarti et al. 2001]. In the next section, we proceed to develop a multidimensional definition of the Haar$^+$ tree, inspired by the nonstandard multidimensional Haar transform.

7.2 The Multidimensional Haar$^+$ Tree

Figure 11 depicts a two-dimensional Haar$^+$ tree that can be used to summarize a 16-element 4×4 two-dimensional data set. A Haar$^+$ tree node now has four (in general, 2^d for d dimensions) children nodes and contains *three* (in general, $2^d - 1$) head coefficients a, b, c, as well as *four* (in general, 2^d) supplementary coefficients q, r, s, t. These are combined by addition and subtraction in order to create the *four* (in general, 2^d) outgoing values of that node, one towards each child node. Each child node summarizes a different region of the data array, called its *support region*. A node's head coefficients play the role of regular Haar wavelet transform coefficients, and all share the same support region; each of the four supplementary coefficients is an additional term on one of the four outgoing values, and shares the same support region as the main coefficients of its child node.

In the two-dimensional case, the *state* of a given node is an eight-element (in general, 2^{d+1}-element) vector $[v, a, b, c, q, r, s, t]$ containing the incoming value v to that node and the coefficient values a, b, c, q, r, s, t in it. A state of a node C *creates* the four-element (2^d-element) *outgoing vector* $[v+q+a+b+c, v+r+a-b-c, v+s-a-b+c, v+t-a+b-c]$. Then the redundancy theorem about Haar$^+$-based data representations can be extended to the two-dimensional case as follows.

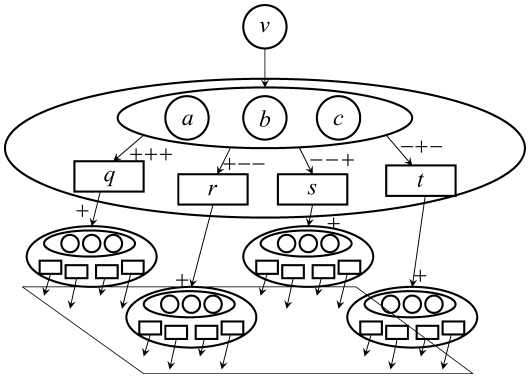


Fig. 11. A two-dimensional Haar$^+$ tree.

THEOREM 7.1. *Any two-dimensional Haar$^+$ tree* $\mathbf{H}$*, in which at least one node contains more than three nonzero coefficients, is equivalent to a Haar$^+$ tree* $\mathbf{H}'$ *such that every node* $C \in \mathbf{H}'$ *contains at most three nonzero coefficients, and* $\|\mathbf{H}'\| \leq \|\mathbf{H}\|$.

PROOF. Let C_i be a node in $\mathbf{H}$ that contains more than three nonzero coefficients and finds itself in the state $[v, a, b, c, k, l, m, n]$. This state is *equivalent* to the state $[v, 1, 1, 1, k{+}a{+}b{+}c, l{+}a{-}b{-}c, m{-}a{-}b{+}c, n{-}a{+}b{-}c]$, as both create the same outgoing vector $[v{+}k{+}a{+}b{+}c, v{+}l{+}a{-}b{-}c, v{+}m{-}a{-}b{+}c, v{+}n{-}a{+}b{-}c]$. Hence, an assignment of more than three nonzero values in a node C_i is reducible to the assignment of exactly four nonzero values, one on each supplementary coefficient. In effect, C_i assumes the state $[v, 0, 0, 0, q, r, s, t]$, where $q \neq 0, r \neq 0, s \neq 0, t \neq 0$, creating the outgoing vector $[v{+}q, v{+}r, v{+}s, v{+}t]$. However, the same outgoing vector is created by a node in the state $[v + \frac{q+r+s+t}{4}, \frac{q+r-s-t}{4}, \frac{q-r-s+t}{4}, \frac{q-r+s-t}{4}, 0, 0, 0, 0]$. In conclusion, a node C_i in the state $[v, 0, 0, 0, q, r, s, t]$, where $q \neq 0, r \neq 0, s \neq 0, t \neq 0$, can be reduced to a node with exactly three nonzero (head) coefficients by modifying its *incoming* value from v to $v + \frac{q+r+s+t}{4}$. Such a modification can be carried out by adding $\frac{q+r+s+t}{4}$ to the value z of the coefficient at the parent node of C_i. If this addition brings about more than three nonzero coefficients in C_j, we reduce C_j to three nonzero values likewise. The process leads from any given node upwards, terminating at the root. Each step may decrease, but not increase, the amount of nonzero coefficients in the tree. In effect, any two-dimensional Haar$^+$ tree $\mathbf{H}$ is reducible to a Haar$^+$ tree $\mathbf{H}'$ such that every node $C \in \mathbf{H}'$ contains at most three nonzero coefficients, and $\|\mathbf{H}'\| \leq \|\mathbf{H}\|$. □

Theorem 7.1 leads to the generalized form of Corollary 4.3:

COROLLARY 7.2. *A B-term d-dimensional Haar$^+$ tree* $\mathbf{H}$ *that approximates a data array* $\mathbf{D}$ *while minimizing an error metric* $\mathcal{E}$ *does not need to contain more than* $2^d - 1$ *nonzero coefficients per node.*

A d-dimensional Haar$^+$ tree representing a data array of $n = m^d$ values has height $\log_{2^d} m^d = \log m$ and contains $O((\frac{m}{2})^d)$ nodes. We now extend our value delimitation to the multidimensional case.

7.3 Multidimensional Value Delimitation

Lemma 5.2 is straightforwardly extended to the multidimensional case, thanks to Theorem 7.1. The outgoing values of a node cannot all be greater than, nor all less than, the incoming value v to that node. Either the incoming value will equal at least one of the outgoing values, or it will be decreased in at least one outgoing value and increased in at least another one. Furthermore, we *postulate* the multidimensional version of Proposition 5.3; that is, if the incoming value v at a node C_i is $v \notin (m_i, M_i)$, then all main coefficients in C_i are set to zero. We introduce this postulate in order to contain the value search space, and hence the complexity of our synopsis construction algorithm. In fact, we can also follow an approach with which we can strictly prove the exact multidimensional equivalent of Proposition 5.3; that is, nonzero head coefficients are unnecessary when $v \notin (m_i, M_i)$. Such an approach would require the incorporation of further supplementary coefficients in the structure—one for every group of child nodes that a head coefficient affects with the same sign; thus, in the two-dimensional case, this approach would require six more supplementary coefficients. Such an overloading of the structure would incur extra computational cost for the sake of marginal approximation benefits. Thus we have chosen to maintain the simplicity the structure. The following theorem constrains the incoming values to all nodes in $\mathbf{H}$ in terms of the global extrema m and M.

THEOREM 7.3. *The incoming value v to a triad C_i in $\mathbf{H}$ satisfies the inequality* $2^d m - (2^d - 1)M < v < 2^d M - (2^d - 1)m$.

PROOF. Let v be an incoming value to a node C_i, coming from an ancestor triplet C_k of C_i, such that the incoming value v' to C_k itself is $v' \in (m, M)$. In the worst case, each of the $2^d - 1$ head coefficients in C_k will assume a value of magnitude $|v' - M|$ ($|v' - m|$), with appropriate signs, in order to produce $2^d - 1$ outgoing values of the extreme value M (m), respectively. For each of these $2^d - 1$ outgoing values, an even number of coefficients cancel each other out, hence the odd one out affects v' in each case. Let v be the *remaining* 2^dth outgoing value, that is, the single one in which all coefficients contribute with the same sign, hence $v = 2^d v' - (2^d - 1)M$ ($2^d v' - (2^d - 1)m$). Still, $v' \in (m, M)$, hence $v \in (2^d m - (2^d - 1)M, 2^d M - (2^d - 1)m)$. □

Theorem 7.3 is the general form of Theorem 5.7. In effect, the difference of the largest from the smallest possible incoming value is $(2^{d+1} - 1)\Delta$, where $\Delta = M - m$. Hence, in the d-dimensional case, $|\mathcal{S}| \leq (2^{d+1} - 1)\lfloor \frac{\Delta}{\delta} \rfloor + 1 = O(2^d \frac{\Delta}{\delta})$. On the other hand, the cardinality of the sets containing the potential assigned values at the head and supplementary coefficients of a triad C_i that are multiples of δ is $O(\frac{\Delta}{\delta})$.

7.4 Multidimensional Algorithms

The one-dimensional algorithm of Section 5.2 can be extended to the multidimensional case. At each node C_i, it needs to consider $O(2^d \frac{\Delta}{\delta})$ incoming values; for each of those, it has to check $O((\frac{\Delta}{\delta})^{2^d - 1})$ combinations of value assignments on $2^d - 1$ main coefficients using the 2^d arrays returned from its children. For

each tabulated value of available space b at node C_i with incoming value v, the algorithm needs to determine the *optimal distribution* of these b space units among the $2^d - 1$ main coefficients on C_i, its 2^d supplementary coefficients, and its 2^d children nodes. We can treat each supplementary coefficient as a member of the subtree at its child node. Hence, for each combination of values assigned to the main coefficients in C_i and amount of allocated space b at a child C_k of C_i rooted on supplementary coefficient c_e, we have to examine two cases: either b space is given to C_k and its subtree with incoming value v, or $b - 1$ space is given to C_k, with a nonzero value z assigned to c_e and modifying v accordingly. The δ-optimal value of z does *not* need to be separately computed for each v. Instead, it is computed only once for each b; thereafter, it is simply adjusted according to the given v, so as to produce the required best incoming value to C_k for the given value of b. The search for the optimal distribution of space b can be efficiently performed by ordering the children of C_i in a binary tree of $2^d - 1$ subnodes and executing binary search on them, as in Garofalakis and Gibbons [2004], Garofalakis and Kumar [2005], and Guha and Harb [2008]. This process takes $O(\log \min\{B, 2^{d\ell_i}\})$ time per entry per subnode per combined value assignment, where ℓ_i is the Haar$^+$ tree layer of node C_i. Hence, the solution takes $O(2^{2d}(\frac{\Delta}{\delta})^{2^d} nB)$ time; only arrays of children nodes in a single root-to-bottom path need to be concurrently stored, hence, as there are at most 2^d children per node, the space is $O(\frac{2^{2d}}{d}\frac{\Delta}{\delta} B \log \frac{n}{B})$.

If the target error metric is a maximum error metric, we can do better. As in Section 6.4, we employ the algorithm that solves the complementary, error-bounded problem. We thus gain two advantages. First, we eschew the tabulation of space. Second, the cardinality of the set of possible incoming values is only $|\mathcal{S}| = O(\frac{\Delta}{\delta})$; this is due to the fact that, according to Lemma 5.2, no incoming value can become too distant from the extrema of the data set without producing a reconstructed value violating the maximum-error bound the algorithm operates on. The key operation is now a tabulation only for allowed incoming values at each node C_i. The algorithm determines the δ-optimal assigned value for all $2^d - 1$ head coefficients residing on node C_i for each entry in this tabulation. We have to consider $O(\frac{\Delta}{\delta})$ incoming values at node C_i, and, for each of those, $O((\frac{\Delta}{\delta})^{2^d-1})$ combinations of value assignments on the $2^d - 1$ head coefficients in C_i, scanning through the 2^d arrays returned from the children nodes. Since there are $O((\frac{m}{2})^d)$ nodes in the tree, the basic runtime becomes $O(2^d(\frac{\Delta}{\delta})^{2^d}(\frac{m}{2})^d) = O((\frac{\Delta}{\delta})^{2^d} n)$, and the space is $O(2^d \frac{\Delta}{\delta} \log m) = O(\frac{2^d}{d}\frac{\Delta}{\delta} \log n)$. The tradeoff between time- and space-efficiency is treated as in the one-dimensional case. Using this algorithm in a binary-search iteration, we can solve the space-bounded problem in $O((\frac{\Delta}{\delta})^{2^d} n(\log \frac{\mathcal{E}}{r} + \log n))$ time and $O(\frac{2^d}{d}\frac{\Delta}{\delta} \log n)$ space.

8. THEORETICAL COMPARISON OF SYNOPSIS CONSTRUCTION TECHNIQUES

We now examine how the synopsis construction algorithms we have introduced and the structures they employ relate to other summarization techniques.

Table II. Summary of Results for One-Dimensional Synopsis Construction: $\mathcal{L}_1$ metric.

Reference	Time	Space	Synopsis Model
[Jagadish et al. 1998]	$O(n^3B)$	$O(nB)$	Histogram
[Guha et al. 2004; Guha 2008]	$O(n^2(B+\log n))$	$O(n)$	Histogram
[Garofalakis and Kumar 2004]	$O(n^2B^2)$	$O(n^2B)$	Restricted Haar
[Guha 2008]	$O(n^2 \log B)$	$O(n)$	Restricted Haar
[Guha and Harb 2008]	$O\left(\left(\frac{\mathcal{E}}{\delta}\right)^2 n^3B\right)$	$O\left(\frac{\mathcal{E}}{\delta} nB \log \frac{n}{B} + n\right)$	Unrestricted Haar
[Reiss et al. 2006]	$O(n^{k+1}B^2 \log n)$	$O(n^k B \log^2 n)$	CHH (k-holes)
[Reiss et al. 2006]	$O(nB^2 \log n)$	$O(nB \log n)$	CHH (Greedy) (time efficient)
[Reiss et al. 2006]	$O(nB^2 \log^2 n)$	$O(B \log^2 n + n)$	CHH (Greedy) (space efficient)
This work	$O\left(\left(\frac{\Delta}{\delta}\right)^2 nB\right)$	$O\left(\frac{\Delta}{\delta} B \log \frac{n}{B} + n\right)$	Haar$^+$

8.1 Complexity Comparison

The time complexity of the state-of-the art, in terms of quality, space-bounded histogram, Haar wavelet synopsis, and CHH construction algorithms reviewed in Section 2 remains in all cases super-linear in 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 II summarizes this complexity terrain, under the demanding $\mathcal{L}_1$ metric, and contrasts it to the methods we have introduced; Table III provides a general comparative overview of complexity results for maximum-error metrics (e.g., $\mathcal{L}_\infty$); n is the data set size, B the space bound, q a probability quantization parameter, δ the resolution step, $\mathcal{E}$ an upper bound for the target *normalized* Minkowski-norm error, Δ the difference of the minimum from the maximum value in the data set, and r the machine's resolution. The fractions with denominator δ express the cardinality of the examined set of incoming or assigned values for their respective models. In Guha and Harb [2008], this set is bounded by an upper bound $\bar{\mathcal{E}}$ for the *nonnormalized* $\mathcal{L}_1$ error, that is, the *sum* of absolute errors; yet this aggregate error measure grows at least linearly in the size of the summarized data set n, since $\bar{\mathcal{E}} = n\mathcal{E}$, where $\mathcal{E}$ is the $\Omega(1)$ *normalized* $\mathcal{L}_1$ error, that is, the *average* absolute error. Arguably, the Δ parameter, depending on the *single* maximum value in the data set, is rather comparable to the *average* absolute error $\mathcal{E}$ than to the *sum* of absolute errors $\bar{\mathcal{E}}$. Thus, the question of the dependence of Δ on the data set size n is comparable to the question of the dependence of $\mathcal{E}$ on it; the dependence of the sum of absolute errors $\bar{\mathcal{E}}$ on n is a much clearer matter, as the sum $\bar{\mathcal{E}}$ grows with every additional data value. Hence, for the sake of clarity and comparability, we use the *normalized* error $\mathcal{E}$ in our complexity expressions. Space complexity expressions for Guha and Harb [2008] and Reiss et al. [2006] also take into account their use of the space-efficiency technique of Guha [2008] and the resultant tradeoff, where it applies.

8.2 Structural Genealogy

Figure 12 depicts a genealogy of structures and techniques for synopsis construction. An arrow in the figure denotes that the destination structure *contains*

Table III. Summary of Results for One-Dimensional Synopsis Construction: maximum-error metrics.

Reference	Time	Space	Synopsis Model
[Jagadish et al. 1998]	$O(n^3B)$	$O(nB)$	Histogram
[Guha et al. 2004]	$O(nB\log^2 n)$	$O(nB)$	Histogram (time efficient)
[Guha et al. 2004; Guha 2008]	$O(nB\log^3 n)$	$O(n)$	Histogram (space efficient)
[Garofalakis and Gibbons 2004]	$O(n\mathrm{q}^2B\log(\mathrm{q}B))$	$O(n+\mathrm{q}B\log^2 n)$	Probabilistic Restricted Haar
[Garofalakis and Kumar 2004]	$O(n^2B\log B)$	$O(n^2B)$	Optimal Restricted Haar
[Karras and Mamoulis 2005]	$O(n\log^3 n)$	$O(n\log n)$	Greedy Restricted Haar
[Guha 2008]	$O(n^2)$	$O(n)$	Optimal Restricted Haar
[Muthukrishnan 2005]	$O\left(n^2\frac{\log\frac{\varepsilon}{\Gamma}}{\log n}\right)$	$O(n)$	Optimal Restricted Haar
[Guha and Harb 2008]	$O\left(\left(\frac{\varepsilon}{\delta}\right)^2 n\log n\log^2 B\right)$	$O\left(\frac{\varepsilon}{\delta}B\log\frac{n}{B}+n\right)$	Unrestricted Haar (space efficient)
[Guha and Harb 2008]	$O\left(\left(\frac{\varepsilon}{\delta}\right)^2 n\log^2 B\right)$	$O\left(\frac{\varepsilon}{\delta}\min\left\{\begin{matrix}B^2\log\frac{n}{B},\\ n\log B\end{matrix}\right\}\right)$	Unrestricted Haar (time efficient)
[Reiss et al. 2006]	$O(n^{k+1}B\log B\log n)$	$O(n^kB\log^2 n)$	CHH (k-holes)
[Reiss et al. 2006]	$O(nB\log n\log B)$	$O(nB\log n)$	CHH (time efficient)
[Reiss et al. 2006]	$O(nB\log^2 n\log B)$	$O(B\log^2 n+n)$	CHH (space efficient)
This work	$O\left(\left(\frac{\Delta}{\delta}\right)^2 n\left(\log\frac{\varepsilon}{r}+\log n\right)\right)$	$O\left(\frac{\Delta}{\delta}\log n+n\right)$	Haar$^+$
This work	$O(n\log n(\log\frac{\varepsilon}{r}+\log n))$	$O(n)$	CHH (optimal)

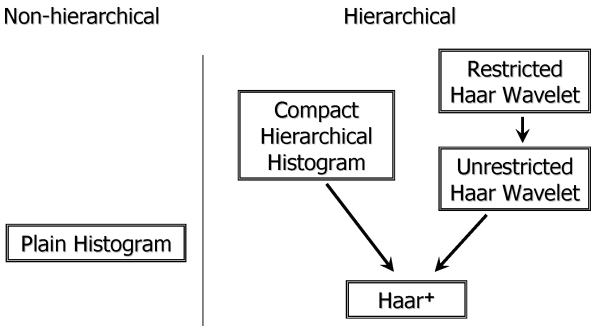


Fig. 12. Genealogy of synopsis structures.

the structure of origin: any representation that can be achieved with the latter can also be achieved with the former. Hence, a restricted Haar wavelet synopsis [Garofalakis and Kumar 2005] is a special case of an unrestricted one [Guha and Harb 2008]; in its turn, a unrestricted Haar wavelet synopsis is a special case of a Haar$^+$ representation. Besides a CHH [Reiss et al. 2006] is a special case of a Haar$^+$ representation as well. The plain histogram [Jagadish et al. 1998; Guha et al. 2004] is unrelated to these hierarchical structures. We infer that the approximation quality achieved with Haar$^+$ is bound to be at least as good as that achieved with a Haar wavelet or a CHH, subject to a sufficiently small value of the resolution step δ, that is, the granularity of examined bucket values. In the next section we verify this relationship experimentally. Greatest experimental interest resides in the quality comparison of Haar$^+$ synopses to histograms, since these two structures are independent.

8.3 Methodological Evolution

Apart from the distinctions defined by the genealogy of structures, synopsis techniques are distinguished by their key methodology. The models for plain histograms [Jagadish et al. 1998; Guha et al. 2004; Guha 2008], restricted Haar wavelets [Garofalakis and Kumar 2005; Muthukrishnan 2005; Guha 2008], and CHH [Reiss et al. 2006] calculate the values to use in the synopsis *per se*. As we mentioned in Section 6.2, Reiss et al. [2006] correctly observed that this calculation for an optimal LPM CHH is computationally hard in the general-error case, due to the interdependence between nodes in the hierarchy, and resorted to heuristic CHH construction techniques for that problem. Besides, both hierarchical synopsis models employing the exact-value-calculation approach [Garofalakis and Kumar 2005; Reiss et al. 2006] are based on a dynamic-programming bookkeeping of choices made at ancestor nodes; the restricted Haar wavelet synopsis algorithm in Garofalakis and Kumar [2005] tabulates a node's all possible subsets of occupied (nonzero) ancestors; likewise, the longest-prefix-match CHH heuristics in Reiss et al. [2006] tabulate the choice of a node's lowest occupied ancestor. As a node in the hierarchy has $O(\log n)$ ancestors, this bookkeeping raises a quadratic time complexity factor in the algorithm of Garofalakis and Kumar [2005] and an $n \log n$ factor in the winning heuristic of Reiss et al. [2006].

In contrast, the unrestricted Haar [Guha and Harb 2008] (Section 2.2.3) and Haar$^+$ methods (Section 5) avoid both these computational obstacles by examining (and tabulating) a quantized set of possible incoming and assigned node values. Thus, they eschew the need for bookkeeping choices made on ancestor nodes; they provably approximate the optimal solution by a small margin of error; and they also achieve a *storage* advantage, as no exact values need be stored, but only integer factors of the chosen resolution δ. Such an approximation scheme with tabulation of incoming values could be applied for general-error longest-prefix-match CHH computation as well, with increased quality and approximation guarantees. The result would be tantamount to a Haar$^+$ algorithm, bar the head coefficients.

9. 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 a comparison of the following algorithms:

—**HIST** The optimal histogram construction algorithm of Jagadish et al. [1998], and Guha et al. [2004]. This algorithm provides an upper bound to the quality of any approximate histogram construction technique [Ioannidis and Poosala 1995; Poosala et al. 1996; Poosala and Ioannidis 1997; Chakrabarti et al. 2002; Gibbons et al. 2002; Gilbert et al. 2002; Guha et al. 2006; Terzi and Tsaparas 2006; Buragohain et al. 2007].

—**R-Haar** The optimal restricted Haar wavelet synopsis algorithm of Garofalakis and Kumar [2005] and Guha [2008].

—**U-Haar** The approximation scheme for unrestricted Haar synopses of Guha and Harb [2008], in which 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 $\bar{\mathcal{E}}$ for the synopsis consisting of the B largest Haar terms of $\mathbf{D}$ by absolute value; it then employs it for bounding the search space. In terms of growth, $\bar{\mathcal{E}} = n^{\frac{1}{p}}\mathcal{E} = \Omega(n^{\frac{1}{p}})$, where $\mathcal{E}$ is the $\Omega(1)$ *normalized* Minkowski-norm error.

—**CHH** The winner greedy heuristic for a compact hierarchical histogram [Reiss et al. 2006]. As explained in Section 2.2.4, this heuristic initially computes an overlapping partitioning in which the value assigned value to a node is optimal for the data interval under the node's scope with the target metric. We have observed that a quality improvement can occur with several metrics if the *median* values in those intervals (which are actually $\mathcal{L}_1$-optimal [Terzi and Tsaparas 2006]) are used instead. This observation is intuitive: the use of medians guides the algorithm more robustly towards the occupation of good *positions*; values do not matter in this stage; only the selection of node positions counts. The $\mathcal{L}_\infty$-optimal mean of extremes and $\mathcal{L}_2$-optimal mean value are unnaturally affected by outlier values in a data interval, which should not be grouped in the same bucket at all. We call this version of the algorithm *Enhanced CHH* to distinguish it from the regular version of Reiss et al. [2006]. We include these CHH algorithms in our experimental study for the sake of completeness; in fact, as we have discussed, the Haar$^+$ tree is bound to outperform them for sufficiently small resolution δ. To our knowledge, this is the first experimental comparison of CHH techniques with *optimal* plain histograms for non-$\mathcal{L}_2$ error metrics [Guha et al. 2004] and other hierarchical synopsis techniques; hence it supplements the study in Reiss et al. [2006].

—**Haar$^+$** The Haar$^+$ synopsis algorithms presented in Sections 5 and 6.4.

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

Description of Data In order to assess the quality achieved with diverse summarization techniques in several real-world environments, we have used two real-life data sets with hard to approximate bursts and discontinuities, as well as another real-world data set with continuity features; we have also employed a larger real-life data set for our runtime assessment. The first data set (FR), discussed in McLeod [1994], is a sequence of the mean monthly flows for the Fraser River at Hope, B.C.[10] The flows present periodic autoregression features, while they average at 2709 (standard deviation: 2123) and feature discontinuities (min value: 482, max value: 10800). We have used a 512-value prefix of the FR data set. 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:

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

730) and feature spikes of large values (min value: 499, max value: 6308). We have used a 256-value prefix of the FC data set. The third data set (DJIA) is the Dow-Jones Industrial Average (DJIA) data set available at StatLib[11] that contains closing values of the Dow-Jones Industrial Average index from 1900 to 1993. Negative values were removed. We used a 512-value subset of closing values from April 14th, 1948 to February 8th, 1950. The closing values average at 182 (standard deviation: 8.73) and exhibit both continuities and hierarchical patterns (min value: 161.6, max value: 205.03). Our last 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). FC and TM were downloaded from the UCI KDD Archive.[12]

9.1 Running Time

In this experiment we evaluate the runtime performance of the four synopsis construction algorithms with approximation guarantees 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 13 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, which 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 in n.

9.2 Synopsis Quality with Nonsmooth Data

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 FR and FC data sets. Results with other metrics, such as $\mathcal{L}_2$, were similar. In order to render the histogram-based summarization directly comparable to the binary-interval-based CHH and Haar wavelet-derived techniques, we have used a 512-value prefix of the FR data set and a 256-value prefix of the FC data set.

[11]Available at http://lib.stat.cmu.edu/datasets/djdc0093
[12]Available at http://kdd.ics.uci.edu/

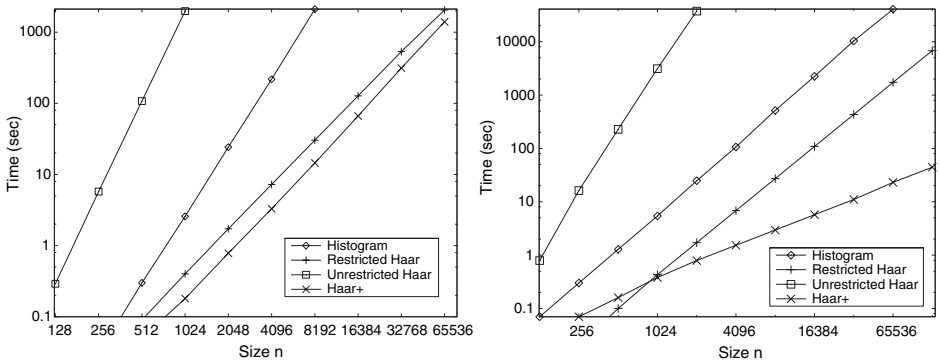


Fig. 13. Runtime comparison: $\mathcal{L}_1$ metric.

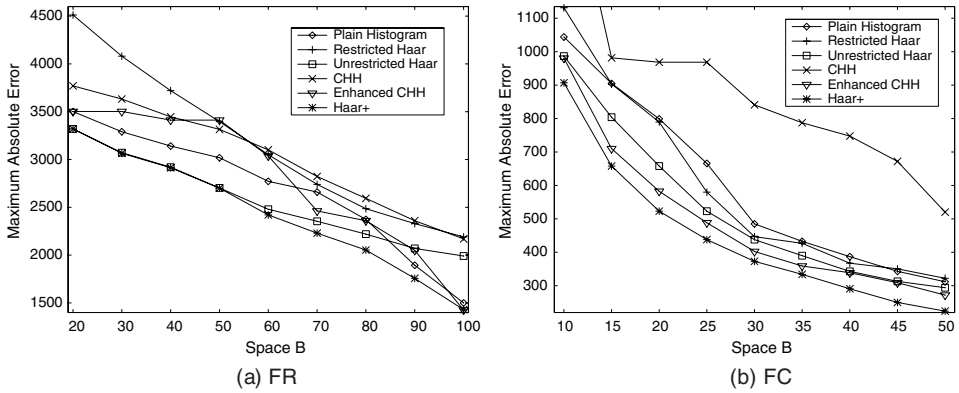


Fig. 14. Quality comparison: $\mathcal{L}_\infty$ metric.

9.2.1 *Maximum Absolute Error.* In this experiment we evaluated the accuracy achieved with the maximum absolute error metric $\mathcal{L}_\infty$ on the the FR and FC data sets. Figure 14 shows the results; the resolution value has been set at $\delta = 50$ for the FR and $\delta = 10$ for the FC data set with both the U-Haar and Haar$^+$ techniques. Haar$^+$ achieves the highest quality for both data sets. The performance of the other methods varies with the data set; U-Haar does well with FR for small space budgets, but not as well with FC; HIST outperforms R-Haar with FR but not with FC. Our Enhanced CHH algorithm could outperform the regular CHH; with FC, it outperforms HIST too; however, its performance is not as stable with FR.

9.2.2 *Average Error.* In our next experiment we evaluated the accuracy achieved with the average error metric $\mathcal{L}_1$ on the the FR and FC data sets. Figure 15 shows the results; 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 data sets, while it needed an inconveniently long time, in particular for summarizing the larger FR data

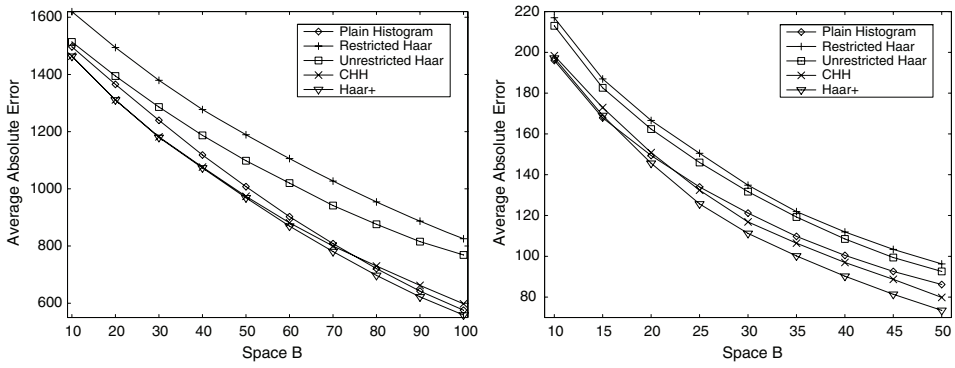


Fig. 15. Quality comparison: $\mathcal{L}_1$ metric.

set, due to its high time complexity for this error metric. CHH fared better with this metric, but was also outperformed by HIST. In this experiment there is no Enhanced CHH technique, as the regular one uses the median values in an interval by default. Still, with the FR data set, the accuracy achieved with the CHH technique deteriorates as B grows, eventually becoming lower than both Haar$^+$ and HIST. However, the Haar$^+$ technique outperforms HIST for this error metric too, achieving the highest quality for both data sets in this experiment also. R-Haar does not achieve high quality with this error metric either.

9.3 Synopsis Quality with Smooth Data

Now we turn our attention to the DJIA data set, which does not present as sharp discontinuities as those we have examined heretofore. In order to enhance the readability of our figures, we do not present the results for R-Haar and U-Haar; the superiority of the Haar$^+$ tree over them has already been decisively demonstrated in both theory and practice.

9.3.1 *Maximum Absolute Error.* We first assess the quality of approximation with the maximum absolute error metric $\mathcal{L}_\infty$, shown in Figure 16(a); the resolution value was set at $\delta = 0.5$ for the Haar$^+$ approximation scheme. Interestingly, none of the hierarchical techniques can match the optimal plain histogram for this error metric with this data set. The performance of both CHH techniques in relation to Haar$^+$ is the expected one, while Enhanced CHH presents a slight advantage over the regular version of the algorithm.

9.3.2 *Average Error.* Figure 16(b) shows the results with the average error metric $\mathcal{L}_1$; resolution values were again set at $\delta = 0.5$. The disadvantage of the examined hierarchical techniques in relation to the optimal plain histogram is repeated with this error metric. Haar$^+$ outperforms CHH, but none of them can reach the quality of the optimal histogram.

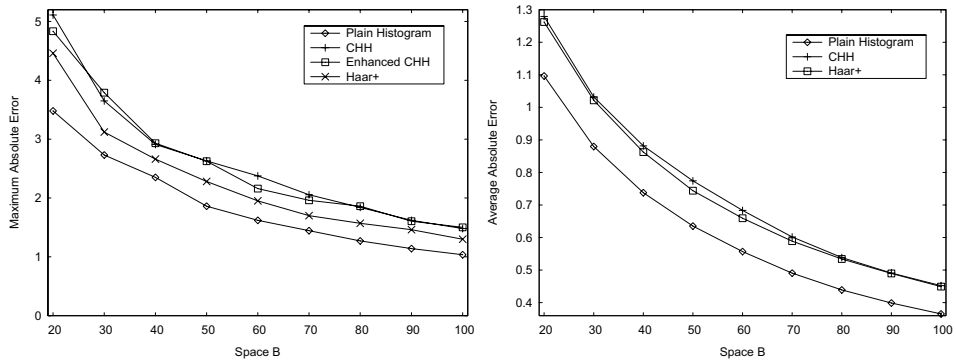


Fig. 16. Quality comparison: $\mathcal{L}_\infty$ and $\mathcal{L}_1$ metrics, DJIA.

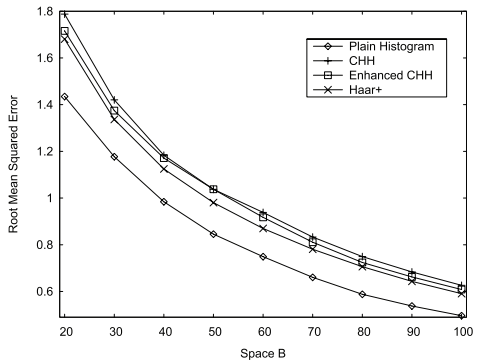


Fig. 17. Quality comparison: $\mathcal{L}_2$ metric, DJIA.

9.3.3 *RMS Error.* In our last experiment we assess the performance on the DJIA data set with the root-mean-squared (Euclidean) error $\mathcal{L}_2$. Figure 17 shows the results; the granularity of values with Haar$^+$ was again set at $\delta = 0.5$. The plain histogram fares best, as in the other cases with this data set. The quality difference is more accentuated in this case, due to the nature of the error metric. The Haar$^+$ technique does better than the CHH heuristics, but cannot reach the histogram quality in this experiment either. Our Enhanced CHH algorithm exhibits a slight quality increase in relation to the regular CHH technique.

10. DISCUSSION

Our results on the superior quality of Haar$^+$ synopses in relation to classical Haar and CHH techniques were expected. A Haar$^+$ synopsis is always at least as good as the equivalent Haar and CHH synopsis. Nevertheless, we have demonstrated that Haar$^+$ *can* also achieve higher accuracy than an optimal histogram. On the other hand, we have witnessed that histograms can do better than hierarchical synopses at approximating those tested data sets that do

not feature sharp discontinuities. These results verify what had hitherto been expressed intuitively [Graps 1995; Guha et al. 2004]. In fact, the intuition of Graps [1995] that wavelet-based techniques are well-suited for approximating sharp discontinuities is consistently verified in relation to a histogram only by the Haar$^+$ technique; previous wavelet-based and CHH schemes do not exhibit the same advantage in relation to an optimal histogram. Moreover, as we saw in the previous section, histograms tend to have an advantage 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 are interesting in themselves, since, as we discussed in Section 3.3, a comparison between the *provably optimal* quality achieved with each of these two synopsis paradigms was missing in previous research.

11. CONCLUSIONS

In this article we have elaborated on hierarchical synopsis structures. 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 shown that this structure supersedes previous hierarchical summarization techniques, such as classical Haar wavelet synopses and the recently proposed compact hierarchical histogram (CHH). In the first, to our knowledge, face-to-face comparison between state-of-the-art hierarchical and (nonEuclidean) histogram summarization techniques, we have demonstrated that Haar$^+$ synopses of data sets with sharp discontinuities can achieve higher quality than optimal histograms under representative error metrics. Furthermore, thanks to the capacity of the Haar$^+$ structure to delimit the search space, Haar$^+$ synopses are constructed in time linear in 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 in the size of the input. And Haar$^+$ synopsis construction can be performed in one pass. In addition, we devised a specialized method for the case that a maximum-error guarantee is required, based on the solution to the dual, error-bounded synopsis problem. We have shown that this methodology effectively solves the longest-prefix-match CHH problem in low polynomial time; consequently, we have shown that this problem is not as computationally hard as previously thought. Moreover, we developed a highly efficient approximation scheme for maximum-error Haar$^+$ synopses by the same methodology. In conclusion, our solutions provide a mostly recommendable option for the high quality and time-efficient summarization of very large discontinuous data sets with any distributive or maximum target error metric. Our techniques have been shown to work well for absolute-error-based error functions. However, a major open problem in the area, also noted by Garofalakis and Gibbons [2004] and Garofalakis and Kumar [2005], is the design of summarization models well-suited for the task of minimizing relative-error-based metrics in data approximation. In the future, we intend to turn our attention to this problem.

ACKNOWLEDGMENTS

We would like to thank the anonymous referees for their insightful suggestions.

REFERENCES

ABOULNAGA, A. AND CHAUDHURI, S. 1999. Self-tuning histograms: Building histograms without looking at data. In *Proceedings of the ACM International Conference on Management of Data* (*SlGMOD'99*). ACM, New York.

ACHARYA, S., GIBBONS, P. B., POOSALA, V., AND RAMASWAMY, S. 1999. Join synopses for approximate query answering. In *Proceedings of the ACM International Conference on Management of Data* (*SIGMOD'99*). ACM, New York.

AGARWAL, D., BARMAN, D., GUNOPULOS, D., YOUNG, N. E., KORN, F., AND SRIVASTAVA, D. 2007. Efficient and effective explanation of change in hierarchical summaries. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining* (*KDD'07*). ACM, New York.

BELLMAN, R. 1961. On the approximation of curves by line segments using dynamic programming. *Comm. ACM 4,* 6, 284.

BRUNO, N., CHAUDHURI, S., AND GRAVANO, L. 2001. STHoles: A multidimensional workload-aware histogram. In *Proceedings of the ACM International Conference on Management of Data* (*SIGMOD'01*). ACM, New York.

BURAGOHAIN, C., SHRIVASTAVA, N., AND SURI, S. 2007. Space efficient streaming algorithms for the maximum error histogram. In *Proceedings of the IEEE International Conference on Data Engineering* (*ICDE'07*).

CHAKRABARTI, K., GAROFALAKIS, M., RASTOGI, R., AND SHIM, K. 2001. Approximate query processing using wavelets. *VLDB J. 10,* 2-3, 199–223.

CHAKRABARTI, K., KEOGH, E., MEHROTRA, S., AND PAZZANI, M. 2002. Locally adaptive dimensionality reduction for indexing large time series databases. *ACM Trans. Datab. Syst. 27,* 2, 188–228.

CHEN, S. AND NUCCI, A. 2007. Dynamic nonuniform data approximation in databases with Haar wavelet. *J. Computers 2,* 8, 64–76.

CORMODE, G., GAROFALAKIS, M., AND SACHARIDIS, D. 2006. Fast approximate wavelet tracking on streams. In *Proceedings of the 10th International Conference on Extending Database Technology.* Springer, Berlin, Germany.

DELIGIANNAKIS, A., GAROFALAKIS, M., AND ROUSSOPOULOS, N. 2005. A fast approximation scheme for probabilistic wavelet synopses. In *Proceedings of the International Conference on Scientific and Statistical Database Management* (*SSDBM'05*).

DELIGIANNAKIS, A., GAROFALAKIS, M., AND ROUSSOPOULOS, N. 2007. Extended wavelets for multiple measures. *ACM Trans. Datab. Syst. 32,* 1.

DESHPANDE, A., GAROFALAKIS, M., AND RASTOGI, R. 2001. Independence is good: Dependency-based histogram synopses for high-dimensional data. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD'01)*. ACM, New York.

FURFARO, F., MAZZEO, G. M., SACCÀ, D., AND SIRANGELO, C. 2005. Hierarchical binary histograms for summarizing multidimensional data. In *Proceedings of the ACM Symposium on Applied Computing*. ACM, New York.

GAROFALAKIS, M. AND GIBBONS, P. B. 2004. Probabilistic wavelet synopses. *ACM Trans. Database Systems 29*, 1 (March), 43–90.

GAROFALAKIS, M. AND KUMAR, A. 2004. Deterministic wavelet thresholding for maximum-error metrics. In *Proceedings of the 23rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. ACM, New York.

GAROFALAKIS, M. AND KUMAR, A. 2005. Wavelet synopses for general error metrics. *ACM Trans. Datab. Syst. 30*, 4, 888–928.

GIBBONS, P. B. AND MATIAS, Y. 1999. Synopsis data structures for massive data sets. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM, New York.

GIBBONS, P. B., MATIAS, Y., AND POOSALA, V. 2002. Fast incremental maintenance of approximate histograms. *ACM Trans. Datab. Syst. 27*, 3, 261–298.

GILBERT, A. C., GUHA, S., INDYK, P., KOTIDIS, Y., MUTHUKRISHNAN, S., AND STRAUSS, M. J. 2002. Fast, small-space algorithms for approximate histogram maintenance. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*. ACM, New York.

GILBERT, A. C., KOTIDIS, Y., MUTHUKRISHNAN, S., AND STRAUSS, M. 2001. Optimal and approximate computation of summary statistics for range aggregates. In *Proceedings of the 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Sysfems*. ACM, New York.

GILBERT, A. C., KOTIDIS, Y., MUTHUKRISHNAN, S., AND STRAUSS, M. J. 2003. One-pass wavelet decompositions of data streams. *IEEE Trans. Knowl. Data Engin. 15*, 3, 541–554.

GRAPS, A. 1995. An introduction to wavelets. *IEEE Computat. Sci. Engin. 2,* 2, 50–61.

GUHA, S. 2008. On the space-time of optimal, approximate and streaming algorithms for synopsis construction problems. *VLDB J.* To appear.

GUHA, S. AND HARB, B. 2008. Approximation algorithms for wavelet transform coding of data streams. *IEEE Trans. Inform. Theory 54,* 2, 811–830.

GUHA, S., KOUDAS, N., AND SHIM, K. 2006. Approximation and streaming algorithms for histogram construction problems. *ACM Trans. Datab. Syst. 37*, 1, 396–438.

GUHA, S., KOUDAS, N., AND SRIVASTAVA, D. 2002. Fast algorithms for hierarchical range histogram construction. In *Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. ACM, New York.

GUHA, S., PARK, H., AND SHIM, K. 2008. Wavelet synopsis for hierarchical range queries with workloads. *VLDB J.* To appear.

GUHA, S., SHIM, K., AND WOO, J. 2004. REHIST: Relative error histogram construction algorithms. In *Proceedings of the 13th International Conference on Very Large Data Bases (VLDB'04)*.

GUNOPULOS, D., KOLLIOS, G., TSOTRAS, V. J., AND DOMENICONI, C. 2005. Selectivity estimators for multidimensional range queries over real attributes. *VLDB J. 14*, 2, 137–154.

HAAR, A. 1910. Zur theorie der orthogonalen functionsysteme. *Mathematische Annalen 69*, 331–371.

IOANNIDIS, Y. E. 1993. Universality of serial histograms. In *Proceedings of the 19th International Conference on Very Large Data Bases (VLDB'93)*.

IOANNIDIS, Y. E. 2003. Approximations in database systems. In *Proceedings of the International Conference on Database Theory (ICDT'03)*.

IOANNIDIS, Y. E. AND POOSALA, V. 1995. Balancing histogram optimality and practicality for query result size estimation. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD'95)*. ACM, New York.

IOANNIDIS, Y. E. AND POOSALA, V. 1999. Histogram-based approximation of set-valued query-answers. In *Proceedings of the International Conference on Very Large Data Bases (VLDB'99)*.

JAGADISH, H. V., KOUDAS, N., MUTHUKRISHNAN, S., POOSALA, V., SEVCIK, K. C., AND SUEL, T. 1998. Optimal histograms with quality guarantees. In *Proceedings of the International Conference on Very Large Data Bases (VLDB'98)*.

JAHANGIRI, M., SACHARIDIS, D., AND SHAHABI, C. 2005. SHIFT-SPLIT: I/O efficient maintenance of wavelet-transformed multidimensional data. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD'05)*. ACM, New York.

JAWERTH, B. AND SWELDENS, W. 1994. An overview of wavelet based multiresolution analyses. *SIAM Rev. 36,* 3, 377–412.

KARRAS, P. AND MAMOULIS, N. 2005. One-pass wavelet synopses for maximum-error metrics. In *Proceedings of the International Conference on Very Large Data Bases (VLDB'05)*.

KARRAS, P. AND MAMOULIS, N. 2007. The Haar$^+$ tree: A refined synopsis data structure. In *Proceedings of the IEEE 23rd International Conference on Data Engineering (ICDE'07)*.

KARRAS, P. AND MAMOULIS, N. 2008. Lattice histograms: A resilient synopsis structure. In *Proceedings of the IEEE 24th International Conference on Data Engineering (ICDE'08)*.

KARRAS, P., SACHARIDIS, D., AND MAMOULIS, N. 2007. Exploiting duality in summarization with deterministic guarantees. In *Proceedings of the 13th ACM International Conference on Knowledge Discovery and Data Mining (KDD'07)*. ACM, New York.

KHANNA, S., MUTHUKRISHNAN, S., AND SKIENA, S. 1997. Efficient array partitioning. In *Proceedings of the 24th International Colloquium on Automata, Languages and Programming (ICALP'97)*.

KOUDAS, N., MUTHUKRISHNAN, S., AND SRIVASTAVA, D. 2000. Optimal histograms for hierarchical range queries. In *Proceedings of the 19th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems.* ACM, New York.

MATHIOUDAKIS, M., SACHARIDIS, D., AND SELLIS, T. 2006. A study on workload-aware wavelet synopses for point and range-sum queries. In *Proceedings of the 9th ACM International Workshop on Data Warehousing and OLAP.* ACM, New York.

MATIAS, Y. AND URIELI, D. 2006. Inner-product based wavelet synopses for range-sum queries. In *Proceedings of the European Symposium on Algorithms Conference (ESA'06)*. Springer, Berlin, Germany.

MATIAS, Y. AND URIELI, D. 2007. Optimal workload-based weighted wavelet synopses. *Theor. Comput. Sci. 371,* 3, 227–246.

MATIAS, Y., VITTER, J. S., AND WANG, M. 1998. Wavelet-based histograms for selectivity estimation. In *Proceedings of the ACM International Conference on Management Data (SIGMOD'98)*. ACM, New York.

MATIAS, Y., VITTER, J. S., AND WANG, M. 2000. Dynamic maintenance of wavelet-based histograms. In *Proceedings of the International Conference on Very Large Data Bases (VLDB'00)*.

MCLEOD, A. 1994. Diagnostic checking of periodic autoregression models with application. *J. Time Series Anal. 15,* 2, 221–233.

MURALIKRISHNA, M. AND DEWITT, D. J. 1988. Equi-depth histograms for estimating selectivity factors for multidimensional queries. In *Proceedings of the ACM International Conference on Management Data (SIGMOD'98)*. ACM, New York.

MUTHUKRISHNAN, S. 2005. Subquadratic algorithms for workload-aware Haar wavelet synopses. In *Proceedings of Foundation of Software Technology and Theoretical Computer Science (FSTTCS'05)*. Springer, Berlin, Germany.

MUTHUKRISHNAN, S., POOSALA, V., AND SUEL, T. 1999. On rectangular partitionings in two dimensions: Algorithms, complexity, and applications. In *Proceedings of the 7th International Conference on Database Theory (ICDT'99)*.

MUTHUKRISHNAN, S. AND STRAUSS, M. 2003a. Maintenance of multidimensional histograms. In *Proceedings of Foundation of Software Technology and Theoretical Computer Science (FSTTCS'03)*. Springer, Berlin, Germany.

MUTHUKRISHNAN, S. AND STRAUSS, M. 2003b. Rangesum histograms. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'03)*. ACM, New York.

MUTHUKRISHNAN, S., STRAUSS, M., AND ZHENG, X. 2005. Workload-optimal histograms on streams. In *Proceedings of the European Symposium on Algorithms Conference (ESA'05)*. Springer, Berlin, Germany, 734–745.

MUTHUKRISHNAN, S. AND SUEL, T. 2005. Approximation algorithms for array partitioning problems. *J. Algor. 54,* 1, 85–104.

POOSALA, V., GANTI, V., AND IOANNIDIS, Y. E. 1999. Approximate query answering using histograms. *IEEE Data Eng. Bull. 22,* 4, 5–14.

POOSALA, V. AND IOANNIDIS, Y. E. 1997. Selectivity estimation without the attribute value independence assumption. In *Proceedings of the International Conference on Very Large Data Bases (VLDB'97)*.

POOSALA, V., IOANNIDIS, Y. E., HAAS, P. J., AND SHEKITA, E. J. 1996. Improved histograms for selectivity estimation of range predicates. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD'96)*. ACM, New York.

REISS, F., GAROFALAKIS, M., AND HELLERSTEIN, J. M. 2006. Compact histograms for hierarchical identifiers. In *Proceedings of the International Conference on Very Large Data Bases (VLDB'06)*.

SRIVASTAVA, U., HAAS, P. J., MARKL, V., KUTSCH, M., AND TRAN, T. M. 2006. In *Proceedings of the 22nd IEEE International Conference on Data Engineering (ICDE'06)*.

TERZI, E. AND TSAPARAS, P. 2006. Efficient algorithms for sequence segmentation. In *Proceedings of the 6th SIAM International Conference on Data Mining* (*SDM*).

THAPER, N., GUHA, S., INDYK, P., AND KOUDAS, N. 2002. Dynamic multidimensional histograms. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD'2)*. ACM, New York.

VITTER, J. S. AND WANG, M. 1999. Approximate computation of multidimensional aggregates of sparse data using wavelets. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD'99)*. ACM, New York.

VITTER, J. S., WANG, M., AND IYER, B. 1998. Data cube approximation and histograms via wavelets. In *Proceedings of the Conference on Information and Knowledge Management (CIKM'98)*.

Received June 2007; revised February 2008; accepted May 2008