AIMS: An Immersidata Management System*

Cyrus Shahabi

Computer Science Department
University of Southern California
Los Angeles, CA 90089-0781
shahabi@usc.edu

Abstract

We introduce a system to address the challenges involved in managing the multidimensional sensor data streams generated within immersive environments. We call this data type, *immersidata*, which is defined as the data acquired from a user's interactions with an immersive environment. Management of immersidata is challenging because they are: 1) multidimensional, 2) spatio-temporal, 3) continuous data streams (CDS), 4) large in size and bandwidth requirements, and 5) noisy.

By focusing on two specific applications, Attention Deficit Hyperactivity Disorder (ADHD) diagnosis and American Sign Language (ASL) recognition, we propose to study the challenges of two main modes of operations on immersidata: off-line and online query and analysis. In addition, we propose complementary approaches for efficient acquisition and storage of immersidata. The core promising idea behind our proposed approaches is a *'database friendly'* utilization of linear algebraic transformations on both data sets and queries to efficiently abstract, aggregate, classify and/or approximate multidimensional data streams.

*This research has been funded in part by NSF grants EEC-9529152 (IMSC ERC) and IIS-0082826, NIH-NLM grant nr. R01-LM07061, NASA/JPL contract nr. 961518, DARPA and USAF under agreement nr. F30602-99-1-0524, and unrestricted cash gifts from Microsoft, NCR, and Okawa Foundation.

Proceedings of the 2003 CIDR Conference

1 Introduction

With *Immersive Environments*, a user is immersed into an augmented or virtual reality environment in order to interact with people, objects, places, and databases. In order to facilitate a natural interaction (beyond keyboard and mouse), the users in typical immersive environments are traced and monitored through various sensory devices such as: tracking devices on their heads, hands, and legs, video cameras and haptic devices. Immersive sensors are the user interfaces of the future; as a research community we should study their generated data or we will miss the boat. To define this data type, in an earlier publication [26], we coined the term *immersidata* as the data acquired from a user's interactions with an immersive environment. Immersidata can be considered as several continuous data streams (CDS) generated by several sensors in an immersive environment.

Management of immersidata becomes crucial as the number of immersive applications grows and as they become more common. Due to specific characteristics of the immersidata, its management requires database expertise combined with signal processing and continuous math flavors. Immersive applications have ample common data management needs, justifying the design of a general-purpose system for management of immersidata. The grand challenge is to design this system in such a way that incorporates decades of experience in dealing with signals, rather than reinventing the wheel.

1.1 Motivation

While researchers in the areas of *graphics* and *human-computer-interfaces* (HCI) have been investigating the interaction aspects of the immersive sensors and trackers, we are not aware of any work in recording and storing immersidata for future query and analysis. We believe an immersidata set is a rich source of information by analyzing which one can learn about users' behaviors in immersive applications. This in turn can serve several important purposes such as conducting human factor studies, enabling more natural interaction methods, improving the performance of the system compo-

nents, customizing the environment towards a user's preferences, and identifying flaws and pitfalls of the environment.

To illustrate, consider the analogous scenario of web usage analysis. With the WWW, keyboard strokes and mouse clicks have been the dominant modes of interaction with webpages since late 1980's. However, it was not until mid 1990's that the database community realized the richness of the large volumes of usage data collected from these clickstreams. We were among the pioneers in the web-usage mining research area (from [31] to [25]) and hence it is natural to extend our research to the analysis of immersidata collected from immersive sensors.

To be specific, we discuss two examples of immersive applications in Sec. 2 with which we have been involved for the last two years. One of these applications focuses on the off-line querying of immersidata, by which we mean that the data are collected beforehand and stored in a database for future query and analysis. We demonstrate that in a very interesting setting, analyzing multidimensional immersidata helps experts for better diagnosis of Attention Deficit Hyperactivity Disorder (ADHD) in children. The second application motivates online querying and analysis of immersidata in order to recognize users' hand and body motions. For this application, we need to analyze aggregate streams of sensor/tracker data in real-time to match the motions to a library of known motions.

Note that the size of immersidata can potentially grow very large as the immersive applications become more and more common. Current immersive interfaces can generate an average of 40 K-Bytes per second to capture body motions of a single user. For example, a CyberGlove, which is a virtual reality glove that captures a single hand's motions, has 28 sensors with 100 Hz sampling rate. Compare this *per user* rate with the rate that a single user can generate clickstreams on the web or that of any other current data stream application, and the exponential difference in data size and rate would become obvious[1].

1.2 Contributions

We propose the design of an immersive sensor data streams management system as depicted in Fig. 1. Towards this end, we focus our research on four major system components: 1) acquisition, 2) storage, 3) off-line query and analysis, and 4) online query and analysis subsystems. In this paper, we use the terms *immersidata*, *sensor data streams* and *multidimensional data* interchangeably to refer to the same type of data, depending on the specific aspect on which we want to emphasize. Consider each component in turn.

[1] Of course this rate is much less than video data rates; however, the challenges in video streaming applications are different than those we have focused on in the area of query and analysis on data streams.

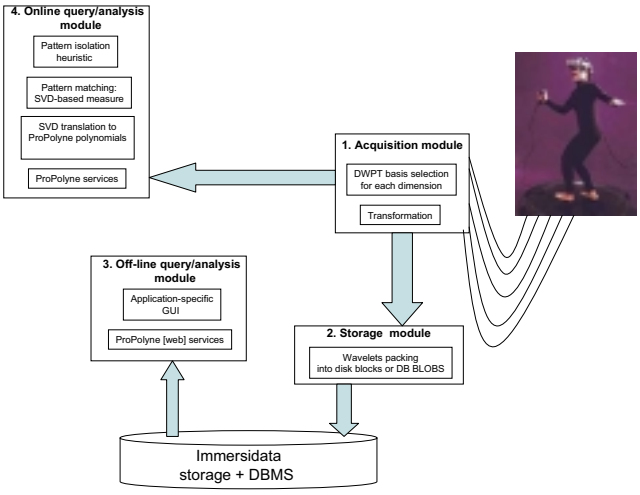


Figure 1: AIMS block diagram

Acquisition: The main challenge in acquisition of immersidata is that similar to any other physical signal, it needs to be cleaned from noise (filtered) and be abstracted for analysis (transformed). Our prior experiences with this data set [27, 29] show that conventional signal processing techniques are sufficiently effective. Therefore, we propose to focus on selecting a signal processing technique that would be *'database friendly'* for our future query and analysis proposes. Our approach is to study a general basis library, Discrete Wavelet Packet Transform (DWPT), to automatically select and apply different transformations on different dimensions.

Storage: To store the transformed data, the challenge is in the design of the physical level of the storage system. Since we are storing wavelets, we need to find an optimal method to pack related coefficients together on a disk block to benefit from the locality of reference. Our studies suggest a theoretical upper-bound on the utilization we can expect from a wavelet disk block. Our approach is to study the access patterns of our queries in order to design an allocation strategy that comes as close to the upper-bound as possible.

Off-line Query and Analysis: Since in this case the immersidata are already acquired, transformed and stored, we do not need to worry about the streaming aspect of the data. However, we still need to support efficient queries, perhaps even approximate and/or progressive queries, on this bulky multidimensional data set. Here, we intend to extend our work in supporting progressive and approximate polynomial range-aggregate queries for OnLine Analytical Processing (OLAP) applications. In particular, we want to extend and generalize our ProPolyne technique [24, 23] to support arbitrary queries on immersidata. The encouraging news here is that ProPolyne is originally designed to work on wavelet transformed

multidimensional data. The challenge, however, is that ProPolyne does not yet know how to deal with transformed data where each dimension is transformed through a different basis.

Online Query and Analysis: This component faces the most challenging aspect of immersidata and heavily relies upon the previous three research activities. The main challenge here is as if we need to perform real-time *pattern recognition* on aggregation of several data streams that are incrementally completing! In addition to typical challenges of continuous data streams (CDS) where: 1) queries must be answered based on limited amount of information rather than the entire dataset, and 2) the data can be looked at only once due to the real-time constraints, there are two other challenges as follow. First, queries need to operate on aggregate information from several sources/sensors. Data from each individual sensor do not make much sense for us, rather data from all sensors together form a meaningful point in the hand (or body) motion trajectory. This is harder than the case for general CDS with data coming from distributed sources, where a loose aggregation is needed for the computation. In our case a much tighter aggregation is needed, which essentially renders our problem to be a high dimensional one. Second, a meaningful hand/body motion is formed by a sequence of data samples, rather than individual data samples occurring in a data stream. In addition, a sequence for one hand motion has no fixed length, as different persons may finish a hand motion with different time durations while the data rate of the sensors are fixed. Hence, one needs to separate a series of variable length motions into individual and recognizable actions. Our previous efforts [28, 5] in pattern recognition from this data set focused on using conventional learning techniques such as Bayesian Classifiers, Decision Trees and Neural Nets. However, these techniques are not appropriate for *streaming* data and only work well when the whole data is available. Hence, we propose a new approach that would address aggregation, dimensionality reduction and pattern matching challenges in one shot. We also propose to utilize a concept from information-theory to address the challenge of incrementally completing signals. Another concern is to make these techniques work directly in (wavelet) transformed domain so that we can utilize our acquisition, storage and ProPolyne techniques. In Sec. 3.4.1, we show that this is doable.

2 Motivating Applications

To motivate the main two modes of query and analysis, off-line and online, we focus on two specific applications we have been involved with for the past two years. Consider each application in turn.

a. ADHD subject

b. Virtual classroom

Figure 2: ADHD Immersive Environment

2.1 ADHD Diagnosis: Off-line Query and Analysis

We have been working with a human factor scientist on an interesting application [22]. Within a 3-D immersive environment, *Virtual Classroom*, a group of normal and ADHD-diagnosed (Attention Deficit Hyperactivity Disorder) children are subjects to do particular attention tasks (see Fig. 2a). The application objective is to differentiate between these two groups of subjects by analyzing their interactions with the environment. The environment consists of a typical classroom containing student desks, a teacher's desk, a virtual teacher, a blackboard, a large window looking out onto a playground with buildings, vehicles, and people, and a pair of doorways on each end of the wall opposite the window through which activity occurs (see Fig. 2b).

A typical task consists of alphabetical characters being displayed on the blackboard and having the child press a button when a particular pattern is seen. In one of our designed tasks, the AX task, the subject is instructed to press mouse button as quickly as possible upon detecting an X after an A (a hit) and withhold their response to any other pattern. At the same time, a series of typical classroom distractions are systematically manipulated within the environment (i.e., ambient classroom noise, paper airplane flying around the room, students walking into the room, activity occurring outside the window). During the test, the trackers placed on the head, hands and legs monitor body movements of the child and stream the data continuously. Each tracker data consists of 6 dimensions: X, Y and Z values corresponding to tracker position in the space and H, P and R parameters representing tracker rotation corresponding to the Y, X and Z axis, respectively. Therefore, the data set in general has 8 dimensions: in addition to the above mentioned 6 values, there are the time-stamp and sensor-id attributes.

After collecting immersidata from several tests, psychologists would like to ask a variety of queries over the stored data set. This is why we call this *off-line query and analysis* because the queries happen post-application on the collected immersidata. The queries can be as simple as: "Which distraction was around

when a particular child missed a question?" to most complex queries such as "Automatically distinguish hyperactive kids from normal ones." Actually, in our preliminary experiments, we successfully (with 86% accuracy) distinguished hyperactive kids from normal ones by using a Support Vector Machine (SVM) on the motion speed of different trackers. Alternatively, the set of answers to task questions may be represented as a feature vector per subject that can be classified in order to differentiate between normal and ADHD-diagnosed subjects. Later in Sec. 3.4.1, we argue that this sort of statistical analysis can be performed in the wavelet domain more efficiently. Another type of query is the polynomial range-sum queries (as will be discussed in Sec. 3.3) such as: "What is the average response time during a specific task for each child?" or "Is there a correlation (i.e., covariance) between hits (or misses) and subject's attention period to distractions?"

2.2 ASL Recognition: Online Query and Analysis

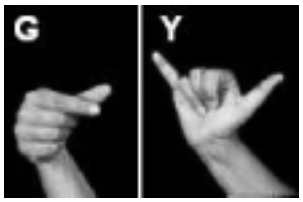
An immersive application can receive input commands through hand and/or body motions as opposed (or in addition to) mouse clicks and keyboard strokes. The main challenge would be to extract in real-time an atomic motion and then recognize the motion by comparing it with a known library of motions, termed *vocabulary*. Due to lack of well-defined vocabulary of motions for an immersive application[2], we focus on recognizing American Sign Language (ASL) signs as examples of well-defined hand motions. In this application, user hand motions are captured via a sensor-equipped virtual reality glove, called CyberGlove (see Fig. 3a).

ASL is a complex visual-spatial language used by vocally or aurally disabled persons in the U.S. and Canada. ASL uses hand gestures, hand movements, or facial expressions to convey meanings such as the English Alphabet, numerals, colors, and so on. An ASL sign is a language unit in ASL, just like a letter or a word with specific meaning in English. According to ASL rules, there are no hand movements involved in most of the Alphabet letter signs (see Fig. 3b); however, hand movements are required for representing ASL words and color signs. For example, color green (or yellow) is conveyed using hand shape of that of letter "G" (or "Y") with the wrist twisting twice.

There are 22 sensors at different positions of the glove to generate the data representing the angle of joints at different parts of a hand (see Fig. 3a). The detailed description of these sensors included with the CyberGlove device are provided in Table 1. In addition to CyberGlove, there is a device called the Polhemus Tracker, which is located on the wrist to measure the

[2] An example of such a visionary application has been illustrated in a recently released movie, *Minority Report*, where the main character, Tom Cruise, interacts with a video-browsing application through hand motions and voice.

a. CyberGlove: Joint angles are measured at positions marked with a circle



b. Some ASL signs

Figure 3: ASL Immersive Environment

hand position (in terms of values for X, Y, and Z coordinates relative to an initial setting) and the hand rotation (in terms of rotation of the palm plane to the X-Y, Y-Z and Z-X planes). In sum, the device software uses the angles to model a human hand, and uses the tracker values to determine the hand motion trajectory; collectively the data from the 28 sensors capture the entirety of a hand motion.

The application takes samples of these data at each sensor clock, which is about 0.01 second. These samples constitute the immersidata in this application which is intrinsically high dimensional. The main query in this application is to recognize signs in particular, or specific hand motions in general. This query is imposed on sensor data streams as they become available and needs to be answered in real-time, hence motivating the on-line query and analysis mode.

3 AIMS System Components

The two applications discussed in Sec. 2, as well as many other immersive applications in training and simulation domains, share common data management requirements that cannot be met by conventional databases[3]. We argue that instead of building a customized system for the data management needs of each immersive application, one can design a general-purpose system providing many of the required functionalities. This is the purpose of AIMS, **A**n **I**mmersidata **M**anagement **S**sytem.

AIMS specifically focuses on the importance of understanding the user behavior in the immersive environments. Consequently, AIMS provides support to collect as much information as possible from the user interactions with the environment. In addition, it provides support for real-time recognition of user's immersive commands as well as query support of the immersidata for future data analysis. In particular, with AIMS, we intend to provide the following common functionalities in support of immersive applications.

1. Acquisition of multiple immersive sensor streams and their appropriate transformation for both

[3] For a description of the shared data-types across these immersive applications see reference [30].

Sensor number	Sensor description
1	thumb roll sensor
2	thumb inner joint
3	thumb outer joint
4	thumb-index abduction
5	index inner joint
6	index middle joint
7	index outer joint
8	middle inner joint
9	middle middle joint
10	middle outer joint
11	middle-index abduction

Sensor number	Sensor description
12	ring inner joint
13	ring middle joint
14	ring outer joint
15	ring-middle abduction
16	pinky inner joint
17	pinky middle joint
18	pinky outer joint
19	pinky-ring abduction
20	palm arch
21	wrist flexion
22	wrist abduction

Table 1: CyberGrasp Sensors

real-time and future query and analysis.

2. Efficient storage of transformed signals in disk blocks or database BLOBs.

3. Progressive and approximate evaluation of polynomial analytical queries on the stored and transformed signals in support of future data mining tasks.

4. Real-time recognition of abstract commands from spatio-temporal aggregation of several transformed sensor streams.

In this section, we discuss the four components or subsystems of AIMS in support of each of the above-mentioned functionalities.

3.1 Acquisition

The first step in managing immersidata is to acquire sensor data streams and store them for the purpose of future query and analysis. A major challenge with data acquisition is the vast amount of noisy data generated in real time by various sensors. The question is how fast one should record a sensor value. A naive approach may record the sensor status as fast as the acquisition system (both hardware and software) can operate. The intuition is that the more samples we collect, the more accurate the acquisition of the occurred event. On the other hand, due to either device limitations or nature of the human motion, the value/status of a sensor might not change as fast as the system is sampling it. Hence, the higher than needed sampling rate will result in more power consumption, storage space and bandwidth requirements for the acquisition task without providing any useful information. Acquiring immersidata becomes even more complicated if we consider that the optimal sampling rate also depends on both the specific sensor being sampled and the immersive session being recorded.

We conducted several experiments [27] with real immersive sensors to understand different factors that impact the sampling rate of immersive sensory data. In our experiments, we used the CyberGlove Software Development Kit (SDK) to write handlers that record sensor data from a virtual reality glove (the same as in Figure 3a) whenever a particular sampling interrupt was called. The rate at which these handlers were called - thus, the maximum rate we could sample - varied as a function of the CPU speed.

To sample and record data asynchronously, we developed a simple multi-threaded double buffering approach. One thread was associated with answering the handler call and copying sensor data into a region of system memory. A second thread worked asynchronously to process and store that data to disk. The CPU was never 100% busy during this process, so we do not believe our recording strategy interfered with the rendering process itself. Furthermore, there is obvious room for optimization here: we could run our experiments on dual-processor machines and we could also adjust the thread priorities for the second thread.

To maintain accuracy, our sampling techniques are based on the Nyquist theorem [19], which states that a signal must be sampled with a rate twice as fast as the maximum frequency in the signal in order to reconstruct it: $r_{nyquist} = 2f_{max}$.

The standard discrete Fourier transform, autocorrelation, and minimum square error techniques were applied to each signal to identify f_{max} within a specified confidence threshold. Our work focuses on determining when to make these calculations. In particular, we developed four alternative sampling techniques: Fixed, Modified Fixed, Grouped and Adaptive Sampling. The first two fix the sampling rate at the largest common denominator across all sensors. Grouped sampling strives to improve on this by clustering similar sensors (in rates) and use a fix rate per cluster. Finally, adaptive sampling considers the immersive session information as well (within a sliding window) and samples according to the level of activity within the session window.

We compared the bandwidth requirement of our four proposed sampling technique. We observed that adaptive sampling requires far less bandwidth (and storage) as compared to the other techniques. When compared with a block-based compression technique, e.g., Unix zip software (based on Hoffman coding), adaptive sampling provides superior savings. Later, in a follow-up study [29], we investigated other conventional compression techniques, such as quantization techniques (e.g., Adaptive DPCM). We also combined the above mentioned sampling approaches with ADPCM technique and conducted several experiments to compare the accuracy and efficiency of our different immersidata sampling and compression techniques. The results showed that we only get marginal improvement by combining ADPCM with adaptive sampling.

3.1.1 Multi-Bases Transformation

The main lesson learned from our studies on immersidata *acquisition* [27, 29] is that conventional signal processing techniques are good enough for this data type because after all they behave like any other physical signal. Hence, the choice of the sampling/transformation technique must be determined elsewhere. As we discuss later in Sec. 3.3.1, we need to select a technique that is consistent with our query and analysis modules. In particular, we propose to select a transformation basis per dimension from a general transformation library, Discrete Wavelet Packet Transform (DWPT) [36], to achieve real-time acquisition and sampling of immersidata. There are three important reasons for this decision. First, DWPT is a generalization of wavelet transform that includes wavelet coefficients as well as summary and details of details at different levels. Hence, by recursively applying a summary and a detail filter on both summaries and details, DWPT quickly computes a large amount of information about the space and frequency characteristics of a function at different scales. It is important to note that the Fourier transform, which showed promise in sampling of immersidata in our previous studies, is also a subset of DWPT[4]. Second, the complexity of wavelet transformation for incremental update (append) is low making wavelets the appropriate choice given the continuous data stream nature of immersidata, which is append only. Third, as we illustrate later in Sec. 3.3, we can perform statistical queries and analysis directly on wavelets much more efficiently than on raw data samples. Therefore, storing immersidata as wavelets does not require any extra overhead of reverse transformation at the query time. Note that each dimension requires its own transformation which may be different from others. To illustrate, consider a database of immersive data with schema *(sensor_id, x,y,z, time, sensor_value)*. Suppose a sensor is confined to a limited area possibly a single point in space. Hence, if we project away the *time* and *sensor_value* dimensions, we will have a relatively small result set. Consequently, we may want to use the standard basis (i.e., no transform) on the small relation *(sensor_id, x,y,z)* and use wavelets on the others. In addition, the selected basis per dimension from DWPT must be consistent with those needed by the query engine. Sec. 3.3.1 discusses a hybridization approach that selects the basis per dimension for storage and query, identically.

3.2 Storage

In an earlier study [5], we investigated four different techniques to store immersive sensor data streams in an object-relational database. We conducted several experiments with real data sets to compare the query response times on our four different representations. The results showed that for the type of queries mainly submitted by immersive applications, it is more appropriate to store all the samples from different sensors for a given time frame in one storage unit. In that study, since we were building our representations on top of the relational model, the storage unit was a tuple. However, the main lesson learned from that study was that we were looking at the wrong level of abstraction (i.e., conceptual level). Instead, we should have looked at the physical level and decide what group of samples should be stored together on a *disk block*. Therefore, we propose a disk level storage technique for this multidimensional sensory data.

3.2.1 Disk Level Storage of Immersidata

The main lesson learned from our studies on immersidata *storage* is that new design is required at the physical level to place related immersidata samples into one *disk block*. Given our previous decision on storing immersidata in wavelet domain, we need to devise an optimal technique for grouping related wavelets. Actually, recent work suggests even stronger reasons for us to be interested in efficient storage of wavelet data. By storing a wavelet representation of a relation or data cube instead of a tabular representation, one can provide fast approximate [34], exact, and progressive [24, 23] range aggregate query support.

Generally, in order to speedup the I/O, the granularity of disk access is made large, and each read of the disk brings back a disk block rather than an individual byte or number. Thanks to the principle of locality of reference, we often find that when an application needs to access one datum on a disk block, it is likely to need to access other data on the same block. By designing applications to take advantage of this, we can amortize the cost of disk access over multiple reads, significantly reducing the total I/O cost.

The question we propose to answer is: Is there a principle of locality of reference for wavelet data? Or more precisely, is there a way we can store wavelet data to create such a principle? Our preliminary studies showed that we can, and that for common access patterns we have a much stronger principle. It turns out that for point and range queries, if a wavelet coefficient is retrieved, we are guaranteed that all of its dependent coefficients will also be retrieved. The challenge is that distinct coefficients will have common dependents. In order to make immersive applications that rely on access to wavelet data scalable, we must take full advantage of this unique access pattern.

In particular, we plan to study the access patterns required for processing queries and updates on wavelet data. Our initial study on the space of all possible (non-redundant) allocations of these data to disk blocks suggests the following: For all disk blocks of size B, if a block must be retrieved to answer a query,

[4] The last level of a $(1-d)$ DWPT on a space with 2^j elements corresponds to the DFT on the j-dimensional space $\{0,1\}^j$.

the expected number of needed items on the block is less than $1+lgB$. We use this theoretical upper-bound as our *success metric* and design a technique for allocation of wavelet coefficients to disk blocks that can approach this upper-bound. Our optimal disk allocation technique is based on optimal tiling of the corresponding one dimensional wavelet *error tree*. This one dimensional optimal disk block allocation can subsequently be used to construct optimal allocations for (tensor product) multivariate wavelets. We simply decompose each dimension into optimal virtual blocks, and take the Cartesian products of these virtual blocks to be our actual blocks. Finally, we can define a query dependent importance function on disk blocks (e.g., minimizing worst-case or average error), which would allow us to perform the most valuable I/O's first and deliver approximate results progressively during query evaluation. In other words, this extends our ProPolyne technique discussed in Sec. 3.3 to work with block wavelets.

3.2.2 Related Work

The idea of storing immersidata as wavelet blocks arose out of our efforts to use wavelet-based operator approximation for approximate query answering [24, 23]. While developing these methods, it became clear that efficient disk access would be necessary for any practical system. Most uses of wavelets for databases have taken a different approach, focusing on data approximation instead [34]. Recent work in this area has provided elegant techniques for producing wavelet synopses of data streams [10], and designing synopses to control relative error of point and range queries [7]. All of the data approximation techniques assume that the compressed dataset will fit in main memory or be scanned from disk in its entirety. To our knowledge, efficient disk placement of wavelet data has not been explored before this work.

3.3 Off-line Query and Analysis

Once the multidimensional immersidata are captured, transformed and stored as disk blocks, it is time for providing query and analysis support on this data type. We focus our attention on two modes of query and analysis: off-line and online. In the off-line mode, the data in their entirety are already captured and stored on persistent storage. This mode of operation is useful for post-application analysis of the data, as in the case of the ADHD study (see Sec. 2.1). With this mode, the challenge is how to support efficient statistical queries on this multidimensional and bulky data set.

In the past two years, we have been investigating efficient techniques to support range-sum queries on large multidimensional data sets. This problem is identical to the one studied in the area of OnLine Analytical Processing (OLAP). We have introduced a novel MOLAP (Multidimensional OLAP) technique that can support any polynomial range-sum query (up to a degree specified when the database is populated) using a single set of precomputed aggregates. This extra power comes with little extra cost: the query, update, and storage costs are comparable to the best known MOLAP techniques (see [24]). We achieve this by observing that polynomial range-sums can be translated and evaluated in the wavelet domain. When the wavelet filter is chosen to satisfy an appropriate *moment condition*, most of the query wavelet coefficients vanish making the query evaluation faster. We made this observation practical by introducing the lazy wavelet transform, an algorithm that translates polynomial range-sums to the wavelet domain in polylogarithmic time.

Wavelets are often thought of as a data approximation tool, and have been used this way for approximate range query answering [34, 32, 3, 9]. The efficacy of this approach is highly data dependent; it only works when the data have a concise wavelet approximation. Furthermore the wavelet approximation is difficult to maintain. To avoid these problems, we use wavelets to approximate incoming queries rather than the underlying data[5]. By using our exact polynomial range-sum technique, but using the largest query wavelet coefficients first, we are able to obtain accurate, data-independent query approximations after a small number of I/Os. This approach naturally leads to a progressive algorithm. We brought these ideas together by introducing ProPolyne (Progressive Polynomial Range-Sum Evaluator), a polynomial range-sum evaluation method which

1. Treats all dimensions, including measure dimensions, symmetrically and supports range-sum queries where the measure is *any* polynomial in the data dimensions (not only COUNT, SUM and AVERAGE, but also VARIANCE, COVARIANCE and more). All computations are performed entirely in the wavelet domain.

2. Uses the lazy wavelet transform to achieve query and update cost comparable to the best known exact techniques.

3. By using the most important query wavelet coefficients first, provides excellent approximate results and guaranteed error bounds with very little I/O and computational overhead, reaching low relative error far more quickly than analogous data compression methods.

Our experimental results on several empirical datasets showed that the approximate results produced by ProPolyne are very accurate long before the exact query evaluation is complete. These experiments

[5]Note that the data set is still transformed using wavelet; however, it is not *approximated* since we keep all the coefficients.

also showed that the performance of wavelet based data approximation methods varies wildly with the dataset, while query approximation based ProPolyne delivers consistent, and consistently better, results.

3.3.1 Adapting ProPolyne for Immersidata

While ProPolyne is a good stepping stone, to make it a general and practical tool for supporting arbitrary queries on immersidata, it must be extended in three ways: generalization of the technique, refinement of the technique, and generalization of the applicability.

First, we intend to generalize the mechanism underlying ProPolyne by looking beyond pure wavelets to find another basis which may be more effective on a particular dataset or for a particular query workload. Not only do query evaluation algorithms need to be developed in this setting, but there is also a need for best-basis (or at least good-basis) algorithms that efficiently select an appropriate basis from a library of possibilities. As a first step in this direction we propose to develop a hybrid version of ProPolyne which uses the standard basis in a subset of the dimensions (the standard dimensions) and uses wavelets in all other dimensions. Given this decomposition of the dimensions, relational selection and aggregation operators can be used in the standard dimensions to accumulate the results of ProPolyne queries in the other dimensions. Clearly the best choice of hybridization will perform at least as well as a pure relational algorithm or pure ProPolyne. Our preliminary analysis indicates that for many realistic datasets and query patterns, hybridizations can perform dramatically better. The challenge here is making the correct choice of standard dimensions. We intend to develop one algorithm which efficiently identifies good dimension decompositions as part of the database population process, and a complementary algorithm which selects the most appropriate available basis to use for evaluation of a particular query. As discussed in Sec. 3.1, the basis library used by this hybrid algorithm is a subset of the full wavelet packet basis library. Not only will the techniques developed here be valuable in practice, our understanding of this simplified problem will provide a foundation for future use of the full wavelet packet transform (DWPT). This in turn would allow us to use ProPolyne in coordination with our work on immersidata acquisition and storage (Secs. 3.1 and 3.2).

Second, we plan to refine ProPolyne in several ways. Our experimental results suggest that some information about query workloads can be used to dramatically improve the performance of data approximation version of ProPolyne. If this is the case, then the duality of data and queries leads us to believe that some limited amount of information about the energy distribution of the data can be used to improve the performance of query approximation version of ProPolyne. We will investigate the efficacy of techniques based on this principle, and we want to develop algorithms that exploit this additional information to provide more accurate approximate results quickly without giving up the speed or I/O efficiency of ProPolyne as presented above. ProPolyne can also be improved by the development of dimension reduction techniques such as random projections, improved query iteration algorithms, and forecast error estimation. Also, one benefit of using transforms from Harmonic Analysis is that a great deal of work has already been done to produce refined error estimates for transform-based approximations. We propose to exploit this machinery to provide accurate error estimates and confidence intervals without introducing significant computational overhead.

Finally, we intend to generalize the applicability of the principles underlying ProPolyne. While range aggregate queries are useful, linear algebraic approximation can be used for much more general types of queries. We begin by studying OLAP queries that require the simultaneous evaluation of multiple related range aggregates. These queries are very common and include SQL group-by queries, drill-down queries, or general MDX expressions. The key observation here is that these queries act as linear maps where range queries act as linear functionals. Thus, where we approximate a vector to estimate a range query result, we must approximate a matrix to estimate a general query result. We developed techniques to select bases in which these matrices are very sparse, giving natural query evaluation algorithms with low computational complexity. In [23], we have developed query evaluation algorithms which share I/O maximally and retrieve the most important data first in order to provide fast approximate results. In this setting there are several natural notions of what it should mean for the error to be small: for some applications it is important to minimize the standard deviation (i.e., the standard L^2 norm) of the errors. For other applications it may be more important to ensure that any large differences between results for related ranges are captured early and accurately, making a Sobolev or Besov norm a more appropriate error measure. We formalized these requirements and developed progressive algorithm which attempt to deliver the smallest possible error for a given error measure throughout the computation. The extension of this work will help us to understand the mechanics of matrix approximation for approximate query answering; at the same time it will provide insight into appropriate error measures. Relational Algebra operators also have matrix representations, and once we have a thorough understanding of how matrix approximation works in the simpler setting described above, we will be prepared to develop and analyze fundamentally novel exact, progressive, and approximate evaluation strategies for relational algebra queries.

3.3.2 Related Work

Recently wavelets have emerged as a powerful tool for approximate answering of aggregate [9, 34, 35, 18, 37] and relational algebra [3] queries. Streaming algorithms for approximate population of a wavelet database are also available [10], making wavelet coefficients a powerful approximate data storage format. Most of the wavelet query evaluation work has focused on using wavelets to compress the underlying data, reducing the size of the problem. A notable exception is [9] which proposes a method to approximate the function that maps ranges to the corresponding range-sum, simultaneously approximating all SUM queries for a given measure. This method is the closest in spirit to the techniques we present; besides supporting a different class of queries, our technique differs by approximating individual queries at the time of submission, rather than approximating all queries at the time of database population.

3.4 On-Line Query and Analysis

The second mode of query and analysis, online mode, strives to support queries and analysis on sensor data streams as they become available. This mode of operation is useful for online recognition of user behavior by matching the behavior to a library of known behaviors (e.g., American Sign Language).

Thus far, we have ignored the continuous data stream (CDS) aspect of immersidata. We assumed that the arrived data are processed off-line and prepared as multidimensional data set for future queries and analysis. Instead, in the online mode, we must recognize a specific behavior by real-time analysis of immersidata as it becomes available, e.g., recognizing an ASL sign from a user's hand motion. We view this problem as *real-time pattern isolation and recognition over immersive sensor data streams.*

In order to recognize a sequence of patterns over the aggregation of several sensor data streams, one needs to address two problems with interdependent solutions (chicken-and-egg problem). To illustrate, suppose as a result of an immersive interaction a sequence of p_1, p_2, ..., p_m patterns have been generated. One problem is to isolate each pattern within the sequence, i.e., identify when (say) p_1 ends and p_2 starts. The other problem is to actually recognize p_1 as a known pattern. The interdependency is that in order to isolate p_1, it should be recognized as a known pattern. However, p_1 must first be isolated in order to be compared with a known set of patterns (termed *vocabulary*) to be recognized!

We first focused on isolated patterns and studied a similarity measure, weighted-sum Singular Value Decomposition (SVD), to compare an input pattern to the members of a known vocabulary. Our weighted-sum SVD addresses several challenges collectively. First, it works directly on an aggregation of several sensor streams (represented as a matrix). Second, it performs dimension reduction due to its capability to linearly transform a given dataset into rotations with optimal set of magnitudes. Finally, it functions as a similarity measure by comparing corresponding eigenvectors weighted by their respective eigenvalues.

To address the isolation problem, we periodically compared sensor streams with each member of the vocabulary using the weighted-SVD measure. Subsequently, we maintained the *accumulated* similarity values. Finally, we developed a heuristic, which in real-time investigates the accumulated values and simultaneously recognizes and isolates the input patterns. The intuition comes from the information theory where the continuously arriving data in a stream forms a process of accumulation in information about the pattern sequence that is currently present in the stream. On the other hand, the stream carries negative information about all the other absent patterns.

3.4.1 Porting Online Pattern Recognition on Top of ProPolyne

We need to expand on these initial studies in several ways. First and foremost, we need to conduct more experiments within our different application domains and utilizing alternative sensor combinations to ensure our preliminary observations hold true in the general case. Second, our techniques work directly on raw sensor data and not the wavelets. However, we are optimistic that our techniques can be re-structured into polynomials that are ProPolyne friendly. We have been encouraged by a study by Shao [33], which points out that all second order statistical aggregation functions (including hypothesis testing, principle component analysis or SVD, and ANOVA) can be derived from SUM queries of second order polynomials in the measure attributes. Higher order statistics can similarly be reduced to sums of higher order polynomials. The power of these observations leads us to believe that ProPolyne's class of polynomial range-sum aggregates can be used directly to compute our SVD-based similarity function on wavelets. Moreover, our pattern isolation heuristic that only depends on the SVD results, is independent from the sensor data set itself. Third, we would like to explore techniques for computing SVD incrementally, i.e., computation of SVD utilizing results that have already been computed in the earlier steps thus reducing the overall computation cost considerably. Finally, we intend to evaluate the effectiveness of other similarity metrics beyond SVD. We believe that our information-theory based heuristic can be evolved into a metric to measure the effectiveness of different similarity measures.

3.4.2 Related Work

Query over Continuous Data Streams (CDS) has been stimulating increasing interests in the database com-

munity lately. Most current research efforts are either on Database Management System (DBMS) support, such as Stream [2], Fjords [17], and NiagaraCQ [4], or on query processing and data mining issues, such as [8, 12, 13, 14].

Data sequences have been used in many applications, such as stock prices, biomedical measurements, weather data, DNA sequences, and sensor data from robotics. New emerging applications, such as data mining and information retrieval by content, require the capability of finding similar patterns, i.e., similarity query. Similarity query on persistent datasets has received a lot of attentions ([1, 21, 11, 15, 16, 20]), however to the best of our knowledge, there are no prior studies on pattern recognition/isolation over CDS.

Trivially, the performance of a similarity query is determined largely by the chosen distance metric. The most straightforward approach for measuring the similarity between two sequences is to use a Minkowski measure such as the Euclidean distance. Euclidean distance metric is not suitable for our problem due to the effect of *"dimensionality curse"* and the requirement of identical length for the two sequences under investigation. Other approaches include DFT (discrete Fourier transform) [1] and DWT (discrete wavelet transform) [21], which are based on linear transformations and effectively treat a sequence with length ℓ as a point in ℓ-D space, and rotate the axes. This is exactly what singular value decomposition (SVD) does, but SVD does this in an optimal way (in terms of L_2-norm) for the given dataset; the reason is that effectively SVD maximizes the variance along the first few rotations [16] thus gives the optimal decomposition of the dataset by way of rotations. Furthermore, the nature of our data requires a 2-D transformation in case of DFT or DWT; however, since our datasets are not correlated on the sensor dimension at any given time, we do not expect DFT or DWT to perform well. These motivate us to use an SVD based approach.

At the time of this writing, we became aware of another related paper [6]. The problem studied in [6] is similar to ours in that both are trying to match the pattern currently in the data stream to a known set of time series (in our case, a set of predefined hand motions) and that patterns are of varying length. However, there are several differences between [6] and our work. The dataset in [6] is one dimensional, while our dataset is high-dimensional (28 D), which makes the problem more challenging due to the required tight aggregation and the impact of the 'dimensionality curse'. Therefore, our choice of weighted SVD for similarity measure is justified and of course different from the choice of Euclidean distance in [6]. Moreover, we deal with the real-time detection and separation of sequences, which has not been addressed in the past to the best of our knowledge. In [6], computation is always performed up to the current time and then the results are reported per each computation, in which case some of the results may not be very meaningful. Another novel aspect of our work is that we work on aggregated sensor streams. Finally, our application domain and datasets are unique.

4 Implementation Details

We have already started on the development of the storage and off-line query components of AIMS as an integrated system. However, currently we only have stand-alone codes for the acquisition and online querying subsystems. Our plan is to integrate those two subsystems into AIMS to realize the architecture depicted in Fig. 1.

Currently, AIMS has been developed as a 3-tier architecture that provides exact, approximate and progressive range-aggregate query supports (e.g., average, count, covariance) on multidimensional data sets. The lowest tier is the NCR Teradata[6] DBMS; the middle-tier is ProPolyne modules implemented as web services using the Microsoft .NET framework; and the top tier consists of a couple of web-accessible GUI's written in $C\#$. Currently, the system is operational for querying atmospheric multidimensional data sets provided to us by NASA/JPL[7] (see Figure 4). The data are transformed to wavelets as they become available and get packed as blocks. Currently, these blocks are stored as BLOBs (using Teradata's BYTE data type) within Teradata. However, we plan to store them as disk blocks on raw disk and instead only store their location IDs in Teradata.

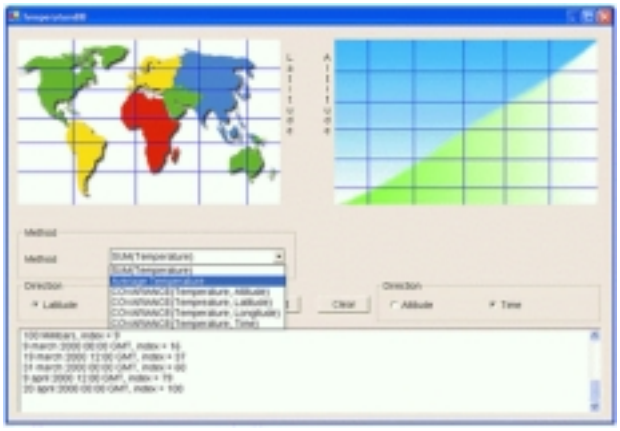
As a long term plan, we intend to integrate the ProPolyne modules into the server tier in order to bring the code closer to data for optimization purposes. To achieve this, we intend to work with NCR to modify the query optimizer of the Teradata DBMS so that given certain queries, it can invoke ProPolyne operators to support progressive and/or approximate evaluation of the queries. Trivially, the system must be extended to allow the creation of ProPolyne data cubes (similar to adding and dropping index structures) on a given data set.

5 Conclusions

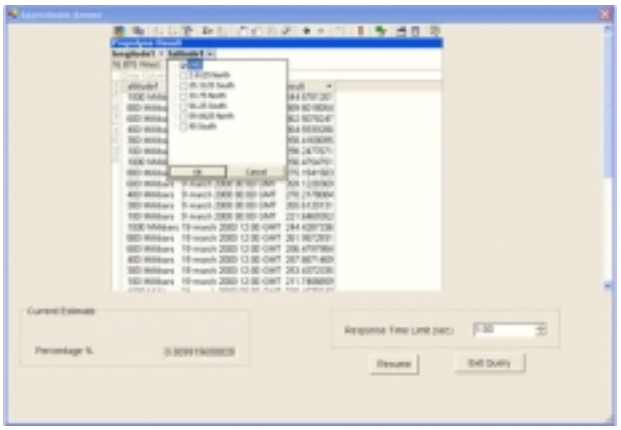
The contribution of this paper has been twofold. First, it introduced a new application domain, immersive applications, and its data set, immersidata. It discussed the database challenges involved in managing immersidata and it explained how some of the techniques proposed within more typical database research areas (e.g., OLAP and multidimensional data mining) can

[6] We have alternative implementations using Informix ORDBMS. The choice of the database server for this specific prototype was mainly due to the agreements with our sponsor.

[7] See a demonstration of this system at http://infolab.usc.edu/NCRwebsite/.



a. Query screen with four dimensional gird cells



b. Result screen as a pivot table

Figure 4: An implementation of approximate and progressive range-sum queries over atmospheric data

be utilized immediately to address some of the challenges in this new area. On the other hand, it discussed why some of the current research contributions (e.g., in the area of data streams) do not immediately apply to the new problems and they need to go through modifications and extensions.

Second, this paper proposed the design of an innovative data systems architecture, AIMS, and provided a detailed report on successes and mistakes relevant to each of the AIMS subsystems. In addition, it provided a brief survey of the research pertaining to each of the subsystems. This paper makes a case that the subsystems are compatible enough that their integration would result in a general-purpose system that would address the common data management needs of a variety of immersive applications.

6 Acknowledgement

The author would like to acknowledge his students, Rolfe Schmidt and Donghui Yan for contributing to some of the research presented in this paper. The author would also like to thank the anonymous referees for their valuable suggestions.

References

[1] R. Agrawal, C. Faloutsos, and A. N. Swami. Efficient Similarity Search In Sequence Databases. In D. Lomet, editor, *Proceedings of the 4th International Conference of Foundations of Data Organization and Algorithms (FODO)*, pages 69–84, Chicago, Illinois, 1993. Springer Verlag.

[2] S. Babu and J. Widom. Continuous queries over data streams. *ACM SIGMOD Record*, 30(3), 2001.

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

[4] J. Chen, D. J. DeWitt, F. Tian, and Y. Wang. Niagaracq: A scalable continuous query system for internet databases. In W. Chen, J. F. Naughton, and P. A. Bernstein, editors, *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA*, volume 29, pages 379–390. ACM, 2000.

[5] J. Eisenstein, S. Ghandeharizadeh, C. Shahabi, G. Shanbhag, , and R. Zimmermann. Alternative representations and abstractions for moving sensors databases. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM'01), Atlanta, Georgia, November 5-10, 2001*. ACM, 2001.

[6] L. Gao and X. S. Wang. Continually evaluating similarity-based pattern queries on a streaming time series. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*. ACM, 2002.

[7] M. N. Garofalakis and P. B. Gibbons. Wavelet synopses with error guarantees. In *SIGMOD 2002*. ACM Press, 2002.

[8] J. Gehrke, F. Korn, and D. Srivastava. On computing correlated aggregates over continuous data streams. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*. ACM, 2001.

[9] A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss. Optimal and approximate computation of summary statistics for range aggregates. In *PODS 2001, Proceedings of the Twenteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 228–237, 2001.

[10] A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In *VLDB 2001*, 2001.

[11] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK*, pages 518–529. Morgan Kaufmann, 1999.

[12] S. Guha, N. Mishra, R. Motwani, and L. OCallaghan. Clustering data streams. In *Proc. of the 2000 Annual Symp. On Foundations of Computer Science*, 2000.

[13] M. R. Henzinger, P. Raghavan, and S. Rajagopalan. Computing on data streams. Technical report tr-1998-011, Compaq Systems Research Center, Palo Alto, California, 1998.

[14] G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 97–106. ACM Press, 2001.

[15] H. V. Jagadish, A. O. Mendelzon, and T. Milo. Similarity-based queries. In *Proceedings of the Fourteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 22-25, 1995, San Jose, California*, pages 36–45. ACM Press, 1995.

[16] F. Korn, H. V. Jagadish, and C. Faloutsos. Efficiently supporting ad hoc queries in large datasets of time sequences. In J. Peckham, editor, *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA*, pages 289–300. ACM Press, 1997.

[17] S. Madden and M. J. Franklin. Fjording the stream: An architecture for queries over streaming sensor data. In *ICDE*, 2002.

[18] Y. Matias, J. S. Vitter, and M. Wang. Dynamic maintenance of wavelet-based histograms. In *VLDB 2000, Proc. of 26th Int'l Conf. on Very Large Data Bases*, pages 101–110. Morgan Kaufmann, 2000.

[19] H. Nyquist. Certain factors affecting telegraph speed. *Bell System Technical Journal*, page 324, April 1924.

[20] S. Park, W. Chu, J. Yoon, and C. Hsu. Similarity searches for time-warped subsequences in sequence databases. In *ICDE*, 2000.

[21] K. pong Chan and A. W.-C. Fu. Efficient time series matching by wavelets. In *Proceedings of the 15th International Conference on Data Engineering, 23-26 March 1999, Sydney, Austrialia*, pages 126–133. IEEE Computer Society, 1999.

[22] A. A. Rizzo and et al. Virtual environments for the assessment of attention and memory processes: The virtual classroom and office. In *Proceedings of The 4th International Conference on Disability, Virtual Reality and Associated Technology, Veszprm, Hungary*, 2002.

[23] R. R. Schmidt and C. Shahabi. How to evaluate multiple range-sum queries progressively. In *Proceedings of the Twenty-first ACM SIGMOD Symposium on Principles of Database Systems, June 3-5, Madison, Wisconsin, USA (PODS'02)*. ACM, 2002.

[24] R. R. Schmidt and C. Shahabi. Propolyne: A fast wavelet-based technique for progressive evaluation of polynomial range-sum queries. In *Conference on Extending Database Technology (EDBT'02)*, Lecture Notes in Computer Science. Springer, 2002.

[25] C. Shahabi and F. Banaei-Kashani. Efficient and anonymous web usage mining for web personalization. *To appear in INFORMS Journal on Computing, Special Issue on Data Mining*, 2002.

[26] C. Shahabi and et al. Immersidata Management: Challenges in Management of Data Generated within an Immersive Environment. In *Proceedings of Multimedia Information Systems (MIS)*, October 1999.

[27] C. Shahabi and et al. Alternative Techniques for the Efficient Acquisition of Haptic Data. In *In the Proceedings of ACM SIGMETRICS/Performance 2001*, June 2001.

[28] C. Shahabi, L. Kaghazian, S. Mehta, A. Ghoting, G. Shanbhag, and M. McLaughlin. Analysis of Haptic Data for Sign Language Recognition. In *To Appear in the International Conference on Universal Access in Human-Computer Interaction*, August 2001.

[29] C. Shahabi, A. Ortega, and M. Kolahdouzan. A comparison of different haptic compression techniques. In *IEEE International Conference on Multimedia and Expo (ICME'02)*, August 2002.

[30] C. Shahabi, M. Sharifzadeh, and A. A. Rizzo. Modeling data of immersive environments. In *Proceedings of the ACM International Workshop on Immersive Telepresence (ITP 2002)*. ACM, 2002.

[31] C. Shahabi, A. Zarkesh, J. Adibi, and V. Shah. Knowledge Discovery from Users Web-Page Navigation. In *Proceedings of the IEEE RIDE97 Workshop*, April 1997.

[32] J. Shanmugasundaram, U. Fayyad, and P. Bradley. Compressed data cubes for OLAP aggregate query approximation on continuous dimensions. In *Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 1999.

[33] S.-C. Shao. Multivariate and multidimensional olap. In H.-J. Schek, F. Saltor, I. Ramos, and G. Alonso, editors, *Advances in Database Technology - EDBT'98, 6th International Conference on Extending Database Technology, Valencia, Spain, March 23-27, 1998, Proceedings*, volume 1377 of *Lecture Notes in Computer Science*, pages 120–134. Springer, 1998.

[34] J. S. Vitter and M. Wang. Approximate computation of multidimensional aggregates of sparse data using wavelets. In *SIGMOD 1999*, pages 193–204. ACM Press, 1999.

[35] J. S. Vitter, M. Wang, and B. R. Iyer. Data cube approximation and histograms via wavelets. In *CIKM 1998, Proceedings of the 7th International Conference on Information and Knowledge Management*, pages 96–104. ACM, 1998.

[36] M. V. Wickerhauser. *Adapted Wavelet Analysis from Theory to Software*. A K Peters Ltd, 1994.

[37] Y.-L. Wu, D. Agrawal, and A. E. Abbadi. Using wavelet decomposition to support progressive and approximate range-sum queries over data cubes. In *CIKM 2000, Proceedings of the 9th Interntationall Conference on Information and Knowledge Management*, pages 414–421. ACM, 2000.