基于压缩的传感器数据存储与访问方法 *

冯 涛, 闫 林, 郭 耀+, 陈向群

(高可信软件技术教育部重点实验室, 北京大学信息科学技术学院软件所, 软件工程国家工程研究中心, 北京 100871)

A Compression-Based Storage and Access Method for Sensor Data

FENG Tao, YAN Lin, GUO Yao+, CHEN Xiang-Qun

(Key Laboratory of High Confidence Software Technologies (Ministry of Education)
Institute of Software, School of Electronics Engineering and Computer Science (Peking University)
National Engineering Research Center for Software Engineering, Beijing 100871, China)

+ Corresponding author: Phn: +8610-62753496, Fax: +8610-62751792, E-mail: yaoguo@sei.pku.edu.cn

Abstract: Massive sensor data are taking up more and more storage resources. Although traditional compression-based methods for storage can reduce the total sensor data size, but they normally introduce extra performance overhead on data accesses. Based on the analysis of sensor data characteristics, this paper attempts to develop compression-based method for sensor data storage and accesses. With the help of partition-based compression, we reduce the storage size by eliminating data redundancy and manage to maintain the performance of random data accesses. Based on this method, three different data compression algorithms and corresponding data access algorithms are proposed to eliminate redundancies in sensor data. We implement the proposed methods in a data sensing and publishing platform for a smart lab environment, and evaluate its performance on sensor data collected in a real environment.

Key words: Internet of things, sensor data; data compression; wavelet transform

摘 要: 海量的传感器数据会占用大量的存储资源，传统的数据压缩方法虽然可以降低传感器数据的存储空间，但是会给数据的访问带来额外的性能开销。本文根据传感器数据的特点，讨论了基于压缩的传感器数据存储与访问方法。具体采用了分组压缩的思想，通过消除原始数据中的冗余信息降低数据存储空间，同时保证数据随机访问的性能；本文根据不同的压缩算法，提出了三种不同的存储与访问方法。利用上述的存储与访问方法，实现了一个实验室环境数据采集发布系统，在真实环境中收集的传感器数据上测试了提出的存储与访问方法的有效性。

关键词: 物联网, 传感器数据, 数据压缩, 小波变换

中图法分类号: TP301 **文献标识码**: A

* This work is supported by the National Basic Research Program of China (973) under Grant No. 2011CB302604 (国家重点基础研究发展规划(973)),the Science Fund for Creative Research Groups of China under Grant No. 60821003 (国家创新研究群体), and the National Science Foundation of China under Grant No. 61103026 (国家自然科学基金), National Science and Technology Major Project No. 2011ZX01043-001-002(国家科技重大专项).

1 引言

随着物联网技术的发展，越来越多的传感器被部署在我们的周围，每个时刻都会有海量的新数据产生。国际数据公司（International Data Corporation）的研究表明：下一个十年之后，全世界的数据会增长 50 倍，其中大部分是由嵌入式设备和各种传感器产生的[1]。

海量的传感器数据会占用大量的存储资源，这给物联网应用的开发带来很大的挑战。物联网应用都需要长期地收集和存储传感器数据，大量的存储资源消耗提高了物联网应用的开发的成本。另外，一些资源受限的平台（比如手持设备等），很难满足物联网应用对存储资源的最低需求。如果采用传统的文件压缩方法（比如 zip 等）来减少传感器数据的存储空间，那么每次访问数据都需要解压缩整个文件，性能开销很大。其次，传感器数据是动态增加的，而传统的压缩方法是针对静态数据的，并不适合物联网环境的需求。

通过观察，我们发现，在一个稳定的小环境中，传感器数据会包含大量的冗余信息。这些冗余信息不仅包括连续重复的传感器数据，而且还包括传感器数据的无效精度，以及应用程序并不关心的小范围内的数据波动。有时传感器的硬件精度可能会小于实际采集到的数据的表示精度，比如 YSI 44006 芯片采集到的温度的精确度最好可以达到±0.2 摄氏度，而实际采集到的数据会精确到小数点后一位。

本文根据了物联网中传感器数据的特点，讨论了基于压缩的传感器数据存储与访问方法，在减少数据存储空间的同时，保证数据访问的效率。核心思想是采用分组的压缩方式，把一组数据中的冗余信息通过一定的压缩方法去除，按组存储压缩后的结果。本文具体提出了三种数据存储与访问的算法，分别是：基于变化的存储与访问算法，基于小波的存储与访问算法，以及混合的存储与访问算法。三种算法具有不同的特点，分别适用于不同的场景。

本文的组织结构如下：第二部分介绍了基于压缩的传感器数据的存储与访问方法；第三部分介绍了采用了上述方法的数据采集和发布系统的设计与实现；第四部分对实验结果进行分析，讨论了三种方法的特点和使用场景；第五部分对相关工作进行了概述；第六部分总结本文的工作。

2 基于压缩的传感器数据的存储与访问方法

2.1 基于压缩的存储与访问方法的核心思想

在一个稳定的小环境中，采集的传感器数据包含大量的冗余信息。这些冗余信息包括重复的数据值，无效的数据精度，和应用不关心的小范围内数据波动。我们可以通过去除这些冗余的信息来极大地减少传感器数据的存储空间，这样压缩存储方式由于不需要对原始的数据记录进行重新编码，所以不会给访问数据带来很大的开销。

根据去除冗余信息的思想，我们提出了分组的压缩存储的方法，具体为：

1. 对每个的传感器数据源（一个传感器上可能有多个数据源）都建立一个缓冲区。缓冲区中保存传感器数据的值和采集的时间。在大多数情况下，数据的采集频率是不变的，因此采集时间的间隔相同。为了简化问题，我们可以用缓冲区中的位置，即 0 开始的一系列整数来代替采集时间。当采集频率变化时，我们可以通过自动的插入数据来模拟一个频率不变的采集过程。我们约定缓冲区中的原始数据为 $\overrightarrow{\text{Data}}$， $\overrightarrow{\text{Data}}$ 中的每个元素为形如(index, value)的值对。
2. 根据传感器数据源本身的特点设定分组的大小，以及数据本身的误差范围。分组的大小与该传感器数据源的采集频率和数据源的类型有关。数据本身的误差范围取决于传感器的硬件指标，也可以根据应用的需求来指定。我们约定 N 为分组大小，数据本身的误差范围用 accept_err 来表示。
3. 当缓冲区中的数据大小超过 N 时，那么我们对这 N 个数据进行压缩，去除其中的冗余信息。压缩算法的输入为 $\overrightarrow{\text{Data}}$ 和 accept_err。压缩后的结果会被持久存储，原始的数据从缓冲区中删除。我们约定压缩后的结果表示为$\overrightarrow{\text{Record}}$，$\overrightarrow{\text{Record}}$的大小为 M，其中每个元素为多维的向量，具体的维度以及含义与压缩算法相关。
4. 压缩后的结果被持久存储在文件中。为了加快访问我们在文件头部固定区域内记录了所有$\overrightarrow{\text{Record}}$的

信息，包括该$\overrightarrow{\text{Record}}$的起始的文件偏移和结束的文件偏移。我们约定该固定的区域为 Header，Header 可以保持在内存中。用户根据采集时间来查询某个传感器数据源的值。为了简化问题，我们假设查询的输入为一个 0 开始的整数。

在下面我们会详细描述基于上述思想提出的三种压缩存储方法的具体算法，以及对应的数据访问算法。

2.2 基于变化的存储与访问算法

基于变化的压缩采用了一种增量存储的思想。详细的算法描述如图 1 所示。数据压缩的具体过程为：对于给定的数据集，首先根据可接受误差范围，舍弃不需要的精度。然后遍历所有数据，对于一组连续的相同数据只保存一次。输出结果既包括数据本身，也包括该数据在原始数据集中第一次出现的位置。数据的存储按照第一次出现的位置进行排序。数据的压缩过程的时间代价为 O(N)。由于数据按照位置有序存储，因此我们可以用二分法查找最接近的压缩记录。数据访问过程的时间代价为 O(lgM)。

```
算法：基于变化的压缩
输入：数据Data，误差范围 accept_err
1.  计算 N，使得 N*accept 是大于 1 的整数
2.  for d in Data then
3.      d := d * N
4.  计算Data的 r 进制表示Data_r，其中 r := N*accept
5.  for d_r in Data_r then
6.      d_r = Floor (d_r/r)*r
7.  计算Data_r的十进制表示Data_10
8.  pre := 0
9.  把(0, Data[pre])插入 result_list
10. for d in Data then
11.     if d != Data[pre] then
12.         pre = d 的索引下标
13.         把(pre,d)插入 result_list
14. return result_list
```

```
算法：基于变化压缩的访问
输入：数据的索引 index
1.  读取文件头部信息 Header
2.  begin := 获取 index 所在组的起始位置，通过 Header，index
3.  end := 获取 index 所在组的起始位置，通过 Header，index
4.  mid := binary_search( index, begin, end)
5.  if mid <= index then
6.      读取 mid 处的记录( i, value)
7.  else
8.      读取 mid 的前一条记录( i, value)
9.  return value
```

图 1：基于变化的压缩和访问算法

2.3 基于小波的存储与访问算法

小波变换是一种节省空间的变换，变换后的结果与原始数据的大小一致。如果原始数据变化比较缓慢，那么经过小波变换后的结果中会包含很多的不需要存储的 0 值。本文选取了 Haar 小波变化方法。这种小波变化实现简单，重构任意一个原始数据的代价为 O(lg(N))。Haar 小波变化的过滤函数为：$h = [1/2, 1/2]$ $g = [-1/2, 1/2]$。具体的算法描述如图 2 所示。

数据存储的具体过程为：对于给定的数据集，进行 Haar 小波变化得到一组系数。按照绝对值的大小从小到大遍历所有系数，如果该系数对结果的影响在可接受的误差范围内，那么把该系数置为 0。输出所有非 0 的系数，结果的形式为（系数的原始位置，系数的值）。数据的存储按照系数的原始位置排序。数据压缩过程的时间代价为 O(N)。访问数据首先要计算出重构该数据所需的系数，然后用二分查找的方式在压缩结果中查询相应的系数，如果找不到则为 0。数据访问过程的时间代价为 O(lg(N) * lg(M))。

```
算法：基于小波的压缩
输入：数据Data，误差范围 accept_err
1.  根据Data计算小波变换后的结果Coeff
2.  对Coeff按照每个元素的绝对值从小到大排序，得到SortCoeff
3.  err := { 0 }
4.  for j in [ 0, SortCoeff.length-1] then
5.      coeff := SortCoeff[j]
6.      isremove := true;
7.      for i in [0, Data.length-1] then
8.          if coeff 会影响 Data[i] then
9.              if err[i]+coeff^2 > accept_err^2 then
10.                 Isremove = false;
11.                 break
12.             else err[i]+=coeff*coeff
13.     if isremove = false then
14.         把(j , coeff)插入 result_list
15. return result_list
```

```
算法：基于小波压缩的访问
输入：数据的索引 index
1.  读取文件头部信息 Header
2.  begin := 获取 index 所在组的起始位置，通过 Header，index
3.  end := 获取 index 所在组的起始位置，通过 Header，index
4.  计算重构数据所需的系数的位置 IndexCoeff
5.  for i in IndexCoeff then
6.      binary_search( i,begin,end)
7.      如果没有找到则值为 0
8.  重构 index 处的数据 data
9.  return data
```

图 2：基于小波的压缩和访问算法

2.4 混合的存储与访问算法

小波的压缩算法的访问的时间代价偏高，主要的原因是重构一个数据需要 O(lg(N))个系数，而基于变化的压缩方法只需要查找一条压缩记录即可。实际上，O(lg(N))个系数中可能包含大量的 0，这些 0 值的系数是不需要查找的。

我们定义一个数据的重构链为重构该数据所需的所有非 0 的系数按照下标从小到大的顺序链接起来形成的链表。由于有 0 值的系数存在，数据的重构链的长度一般情况下会小于 O(lg(N))。根据 Haar 小波变换的性

质可知，这些重构链形成一个树，这样我们可以采用父指针的方式来存储所有的数据的重构链。如果一个数据 d 的重构链的最后一个系数是 c，那么我们说 d 是 c 的覆盖数据。我们依次求出所有非 0 系数的覆盖数据集合。显然，这些非 0 系数的覆盖数据集合是不想交的，且并集恰好为原始数据集合 $\overrightarrow{Data}$ 。如果一个系数的覆盖数据集合是不连续的，那么我们通过增加该系数的副本来保证所有系数的覆盖数据集合是连续的。

基于上面描述，我们提出了结合基于变化的压缩和小波变换的一种混合的存储和访问算法，如图 3 所示。数据的压缩过程首先进行基于小波的压缩，然后在这个基础上计算系数重构链中的父节点，以及覆盖范围，得到这样的四元组（覆盖数据的最小下标，重构链的父节点的相对偏移，系数的原始下标，系数的值）。因为分组大小 N 通常不会很大，所以可以比较少的位来表示覆盖数据的最小下标和重构链的父节点偏移，相对于基于小波的压缩算法，额外使用的存储空间是可以接受的。存储数据的时候按照覆盖数据的最小下标从小到大排序。

```
算法：混合方法的压缩
输入：数据Data，误差范围 accept_err
1.  利用基于小波的压缩算法得到Coeff，其中部分系数已经被置为 0
2.  Parent[0] := Parent[1] = 0
3.  for i in [ 1 , (N-1)/2 ] then
4.    //计算父节点
5.    if Coeff[i] = 0 then
6.      Coeff[2*i+1] = Coeff[2*i] = Parent[i]
7.    else Coeff[2*i+1] = Coeff[2*i] = i
8.  for i in [ 0 , N-1 ] then
9.    计算 i 的覆盖系数的下标 j
10.   j = (i+N)/2
11.   if Coeff[j] = 0 then
12.     j = Parent[j]
13.   如果 j 的覆盖数据集合为空，或者和 i 相邻，那么把 i 插入 j 的覆盖集合
14.   如果加入 i 后 j 的覆盖数据集合不连续，那么增加 j 的副本 k，把 i 插入 j 的覆盖集合。
15. return 所有非 0 系数的四元组
```

```
算法：混合压缩方法的访问
输入：数据的索引 index
1.  读取文件头部信息 Header
2.  begin := 获取 index 所在组的起始位置，通过 Header，index
3.  end := 获取 index 所在组的起始位置，通过 Header，index
4.  在 begin 和 end 的范围内，二分查找最接近的记录 record
5.  读取 record 的内容，保存 record. index 和 record.coeff
6.  while record.parent != record.index then
7.    读取 record 的内容，保存 record.index 和 record.coeff
8.  重构 index 所指的原始数据 data
9.  return data
```

图 3：混合的压缩和访问算法

数据的访问过程类似于基于变化的算法。首先利用二分法查找到最接近查询索引的四元组记录，即在所有覆盖数据的最小下标小于查询索引的记录中满足覆盖数据的最小下标最大。然后根据父节点查找到所有计算需要的系数。最坏情况下重构链的长度为 O(lg(N))，数据访问过程的时间代价为 $O(\lg(N)+\lg(M))$ 。

3 原型系统的设计与实现

基于上文提出的基于压缩的存储与访问方法，我们设计和实现了一个传感器数据采集和发布系统，结构如图 4 所示。该系统分为设备层，数据层，服务层。设备层包括设备注册，设备健康监控，设备通讯三个模块，负责维护设备的注册信息，监控设备的电量剩余情况，以及与传感器节点通讯。数据层包括数据缓存，数据访问，数据压缩，和数据持久存储模块。数据缓存模块在内存中为每个数据源都维护一个缓冲区；数据压缩模块负责按组对传感器数据进行压缩，把结果传输给数据持久存储模块；数据访问模块负责对采集到的数据进行检索，根据用户的请求同时在数据缓存和数据持久存储模块中查找对应的数据；数据持久存储模块主要负责传感器数据文件的读写，保证读写的一致性，以及缓存常用的数据，加快访问。服务层主要包括 Web Service API，用户注册和认证和授权服务。

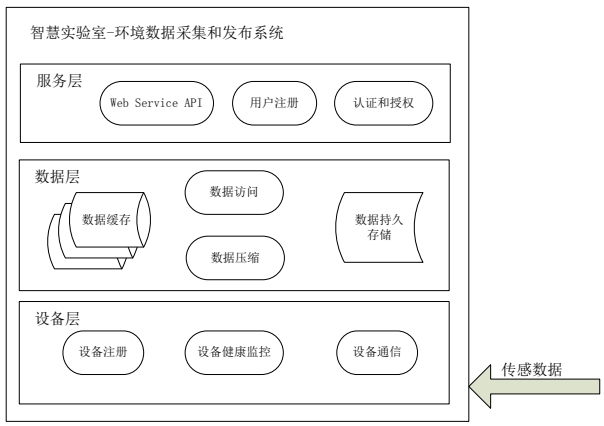

图 4：智慧实验室—环境数据采集和发布系统

4 实验与结果分析

本文利用我们实现的环境数据采集和发布系统对实验室环境进行了长期的监控，主要的传感器包括 IRIS 无线传感器节点，每个实验室成员佩戴的 RFID 以及对应的 RFID 读写器。采集的传感器数据包括：温度，光照，以及 RFID 的信息。传感器采用的是 CrossBow 公司生产的 IRIS 无线传感器节点，使用 MDA100 系列传感器板。具体的传感器数据类型，采集周期，自身的误差范围如表 1 所示。

表 1：传感器数据的信息

传感器数据类型	采集周期（单位秒）	自身的误差	数据范围
温度	60	0.2 摄氏度	-40 到 +60 摄氏度
光照	60	20	0 到 900
RFID 信息	30	0	0,1

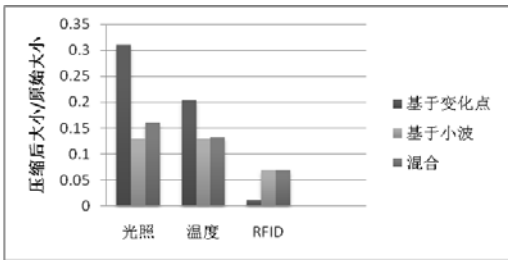

图 5：针对不同数据源的压缩效果统计

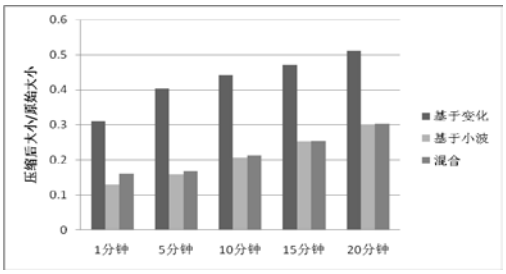

图 6：光照数据在不同频率下的压缩比

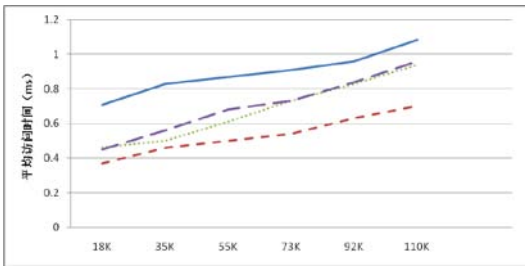

图 7：访问时间统计 (从上到下依次是基于小波的压缩，原始数据，混合压缩，基于变化的压缩)

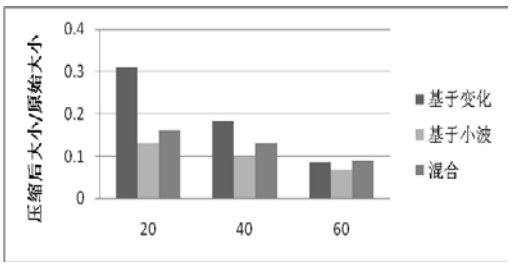

图 8：光照数据在不同误差范围下的压缩

图 5 给出了三种方法在不同数据源上的压缩比（压缩数据大小/原始数据大小）统计，原始数据大小为 1.76MB。可以看到 RFID 数据的压缩效果最好，都达到 18X 以上（压缩后数据的大小分别为 19.53KB，123.44KB，123.44KB）。由于 RFID 数据的变化频率较低，因此采用基于变化的压缩方法效果最为理想（压缩比可以达到 80X）。对于光照和温度数据来说，由于数据变化次数较多，所以基于变化的压缩效果不是太好，而基于小波的算法和混合算法都达到 7-8X 的压缩比（光照数据压缩后的大小依次为 558.59KB，232.42KB，289.84KB；温度数据压缩后的大小依次为 365.23KB，232.42KB，238.28KB），效果较为理想。

图 6 给出了光照数据在不同的采集频率下压缩比的统计。可以看到，随着采集频率的下降，数据的压缩效果都产生了不同程度的下降，这是由于更加密集的数据采集过程会得到冗余度更高的信息，因此可以通过压缩产生更好的效果。另外，混合的压缩算法对于数据采集频率的变化产生的波动较小，因此更加适合于较难以压缩的采集频率较低的数据。

我们针对不同大小的原始数据进行了访问时间的实验，结果如图 7 所示。显然，数据的访问代价和原始数据的大小有关，随着原始数据的增长，访问时间会增大。在三种方法的访问时间对比中，我们发现基于变化的压缩算法的访问时间最优，而基于小波的方法代价最高，混合的方法访问时间代价处于二者之间。总体的访问时间开销都在 1ms 以内，应该不会对普通的物联网应用产生影响。

另外，在很多应用场景中，对于传感器数据的精度没有很高的要求，因此如果适当地提高传感器数据允许的误差范围可以进一步降低存储开销。图 8 给出了光照数据在不同误差范围下的压缩效果对比。

综上所述，在传感器数据变化比较缓慢（比如温度），或传感器的采集结果为有限的离散集合（比如 RFID）的情况下，我们采用基于变化的方法可以得到很高的压缩比，同时访问时间最低；而对于那些变化幅度很大的传感器数据，比如光照等，适合采用基于小波或混合的方法，这样可以保证数据存储开销很小。与小波方法对比，混合方法能够在面向不同数据采样频率时保持比较稳定的压缩效果，同时在数据访问时间开销上也比小波算法要表现更好一些。

5 相关工作

在无线传感网络领域，很多研究工作尝试利用有损压缩的方法来减少网络流量，进而减少传感节点的能耗开销。比如 Deligiannakis 等人[2]提出了 Self Based Regression（SBR）压缩算法来压缩传感节点上的历史数据；Lin 等人[3]在 SBR 算法的基础上进行了改进，提高了压缩效率和数据的精度，提出了 Adaptive Linear Vector Quantization（ALVQ）算法；Schoellhammer 等人[4]假设小环境内，传感器数据在一个小的时间窗口内是线性变化的，进而提出了一种时间压缩算法。但是，这些压缩方法都很大地损失了数据的精度。

在联机分析处理领域，很多工作都采用了小波变化的技术来提高响应请求的效率。比如 Vitter 等人[5]利用小波变化的结果来加快计算 Data Cube 的部分和的操作；C.Shahabi 和 R.Schmidt 等人提出了一种迭代的小波变换方法[6]。本文在面向传感器的数据收集中借鉴了这些算法的思想。

6 总结与未来的工作

针对目前物联网环境中对海量传感器数据进行存储和访问的挑战，本文提出了一种高效的传感器数据存储和访问方法，利用分组的压缩方法，把传感器数据中的冗余信息通过一定的压缩方法去除，在减少数据存储空间的同时，保证数据访问的效率。本文具体提出了三种去除冗余信息的压缩算法，分别为基于变化的压缩算法，基于小波的压缩算法，混合的压缩算法。本文利用分组压缩存储方法，实现了一个环境数据采集和发布系统，测试了存储方法的有效性，并讨论了三种压缩方法的特点和适用场景。

在后续的工作中，我们将进一步完善这种传感器数据存储方法，进一步提供压缩比和访问效率；同时设计和实现一个高效的传感器数据存储的中间件，可以透明地完成传感器数据的存储，适用于 PC、嵌入式设备、移动终端等多个平台。此外，我们还将尝试把我们的存储方法应用到图像、视频等传感信息上。

参考文献:

[1] World's data will grow by 50X in next decade, IDC study predicts. **Available**: http://www.computerworld.com/s/article/9217988/.

[2] Deligiannakis A., Kotidis Y and Roussopoulos N., Compressing Historical Information in Sensor Networks. In Proceedings of the ACM SIGMOD International Conference on Management of Data, 2004

[3] Lin S., Kalogeraki V., Gunopulos D. and Lonardi S., Online Information Compression in Sensor Networks. In Proceedings ofIEEE International Conference on Communications, 2006.

[4] Schoellhammer T., Osterweil E., Greenstein B., Wimbrow M. and Estrin D., Lightweight Temporal Compression of Microclimate Datasets. In Proceedings of the 29th Annual IEEE Conference on Local Computer Networks, 2004.

[5] Vitter J.S., Wang M., Iyer B. Data Cube Approximation and Histograms via Wavelets. In: Proc. Seventh International Conference on Information and Knowledge Management, pp. 96–104, Bethesda, Md., November 1998.

[6] C.Shahabi, M. Jahangiri and F.Banaei-Kashani, ProDA: An end-to-end wavelet-based OLAP system for massive datasets. *Computer 41*, 4 (Apr. 2008), 69–77.