센서 네트워크를 위한 PCA 기반의 데이터 스트림 감소 기법

A PCA-based Data Stream Reduction Scheme for Sensor Networks

알렉산더 페도시브* 최 영 환** 황 인 준***
Alexander Fedoseev Young-Hwan Choi Eenjun Hwang

요 약

데이터 스트림이란 새로운 개념과 기존의 단순 데이터 사이에 존재하는 개념적 차이를 극복하기 위해서는 많은 연구가 필요하다. 대표적인 예로써 센서 네크워크에서의 데이터 스트림 처리를 들 수 있는 데, 이를 위해서는 대역폭이나 에너지, 메모리와 같은 자원적 한계에서 부터 연속 질의를 포함하는 질의처리의 특수성까지 고려해야 할 대상이 광범위하다. 본 논문에서는 데이터 스트림 처리에서의 물리적 제약사항에 해당하는 한정된 메모리 문제를 해결하기 위해 PCA 기법을 기반으로 하는 데이터 스트림 축소 방안을 제안하다. PCA는 상호 관련된 다수의 변수들을 관련이 없는 적은 수의 변수로 변환해준다. 본 논문에서는 질의 처리 엔진의 협력을 가정하고서 센서 네크워크의 스트림 데이터 처리를 위해 PCA 기법을 적용하며, 다른 센서로부터 얻어진 많은 측정값 사이에 시공간적 관련성을 이용한다. 최종적으로 그러한 데이터 처리를 위한 프레임워크를 제시하고 다양한 실험을 통하여 기법의 성능을 분석한다.

Abstract

The emerging notion of data stream has brought many new challenges to the research communities as a consequence of its conceptual difference with conventional concepts of just data. One typical example is data stream processing in sensor networks. The range of data processing considerations in a sensor network is very wide, from physical resource restrictions such as bandwidth, energy, and memory to the peculiarities of query processing including continuous and specific types of queries.

In this paper, as one of the physical constraints in data stream processing, we consider the problem of limited memory and propose a new scheme for data stream reduction based on the Principal Component Analysis (PCA) technique. PCA can transform a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables. We adapt PCA for the data stream of a sensor network assuming the cooperation of a query engine (or application) with a network base station. Our method exploits the spatio-temporal correlation among multiple measurements from different sensors. Finally, we present a new framework for data processing and describe a number of experiments under this framework. We compare our scheme with the wavelet transform and observe the effect of time stamps on the compression ratio. We report on some of the results.

☞ keyword : 센서 네트워크, 데이터 스트림, 데이터 감쇄, 데이터 근사화, 주성분 분석
sensor network, data stream, data reduction, data approximation, principal component analysis

1. INTRODUCTION

The principle of ubiquitous computing was first elucidated by Mark Weiser [1] in 1991. The main idea of ubiquitous computing is to transparently integrate computations into the environment by making wireless networks consisting of tiny nodes containing several sensors for monitoring the environment such as temperature, humidity, illumination, etc. [2, 3, 4]. There are many emerging applications based on such a sensor network including health monitoring, passenger support

* 준 회 원 : LG 전자
alisher@korea.ac.kr
** 준 회 원 : 고려대학교 전기전자공학과 석박사 통합과정
work48@korea.ac.kr
*** 정 회 원 : 고려대학교 전기전자전파공학부 부교수
ehwang04@korea.ac.kr(교신저자)
[2009/01/06 투고 - 2009/01/21 심사 - 2009/02/13 심사완료]

systems, various financial applications, smart home, and network monitoring [5, 6].

Since its emergence, enabling data stream processing has caused a wide variety of new tasks and challenges related to different fields of modern science. In this paper, we consider a data stream as a distributed and continuous information source. Data streams have a number of differences compared to conventional stored relational data [7]:

1. The data elements in the data stream arrive online, while conventional data is permanently stored.
2. The receiving system has no control over the order in which data elements arrive for processing, while traditional data can be distributed in any suitable manner.
3. Data streams are potentially unbounded in size, which is not acceptable for stored relational data.
4. Once a data stream element has been processed, it is eliminated or archived.

In this paper, we focus mostly on the third item in the list. Since data streams are unbounded, we may need an unlimited amount of memory to provide an appropriate speed of processing. Here, we present a framework for data reduction: query preprocessing that works as collaboration between a network base station and a query processing engine. Our approach is based on the inherent property of real world sensor measurements, namely, a spatio-temporal correlation of data tuples. At the core of the framework lies a well-known dimensionality reduction technique, Principal Component Analysis (PCA). PCA transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components. We establish two different types of data reductions, horizontal and vertical reductions. We propose a technique for selecting one between them for better performance and prove the effectiveness of our scheme through experiment.

2. RELATED WORK

Although we know that data approximation is unavoidable, there still exist applications with an exact query evaluation requirement. This means that lossy approximation is inappropriate. In [8], conjunctive SPJ (Select-Project-Join) queries with an arithmetic comparison over a data stream were considered and an algorithm was presented for determining whether or not a query can be evaluated using a bounded amount of memory.

One of the simplest and natural approximation techniques is Sliding Windows [9]. In this technique, queries are evaluated using only recent data in the time domain or using the number of readings. This technique is well-defined and easily understood which makes the semantics of the approximation clear.

Random Samples [7, 10] are based on the assumption that a small sample captures the essential characteristics of the data set. The actual computation for the random sample over a data stream is relatively easy. In order to reduce error from the variance in data and group-by queries, stratified sampling has been proposed recently as an alternative to uniform sampling. The reservoir sampling algorithm of Vitter [11] makes one pass over the data set and is well suited for the data stream model.

Sketching Techniques use frequency moments which capture the statistics of the distribution of the values in the data stream [12]. Sketching involves

building a summary of a data stream using a small and limited amount of memory making it possible to estimate the answer to certain queries over the data set.

Histograms also find application in data stream reduction. In fact, histograms are commonly used summary structures to concisely capture the distribution of values in a data set. Popular histograms are V-Optimal Histograms [13] and End-Biased Histograms.

Wavelets are often used as a technique to provide a summary representation of the data. Wavelets' coefficients are projections of the given signal (set of data values) onto an orthogonal set of basis vectors. Often Haar wavelets are used in databases for their ease of computation. In [14], the use of wavelets was proposed for general purpose approximate query processing and how to compute joins, aggregations, and selections was entirely demonstrated in the wavelet coefficient domain.

It becomes important to devise techniques for computing wavelets in the streaming model in order to extend this body of work to data streams. There has been recent work in computing the top wavelet coefficients in the data stream model. The technique, described in [15], to approximate the best dyadic interval that most reduces the error, gives rise to an easy greedy algorithm to find the best B-term Haar wavelet representation. This work improves upon a previous result by Gilbert, et al. [16].

Compressing Historical Information [17] is based on the notion of the base signal which is constructed and periodically updated from new data tuples. Here, the main assumption is that the data stream has an inherent real world property, namely, spatial/temporal correlation.

Recently, a new notion of the probabilistic or stochastic stream [18, 19, 20, 21] has appeared. The main difference from the regular data stream is that the probabilistic distribution law of data tuples is assumed to be known or discoverable. Our framework benefits from taking into account this new sort of data streams as we demonstrate later. In [19], an extension of a conventional relational model called Probabilistic Stream Relational Algebra (PSRA) was introduced to model existing deterministic data stream models.

PCA is a useful statistical technique having applications in many fields such as face recognition and digital image compression [22]. In [23], we already presented the motivation and basic algorithm for the PCA-based data stream approximation. In this paper, we describe our extensive experiments to prove its effectiveness. Also, we consider data time stamps to find their influence on the compression ratio in terms of the relative approximation error.

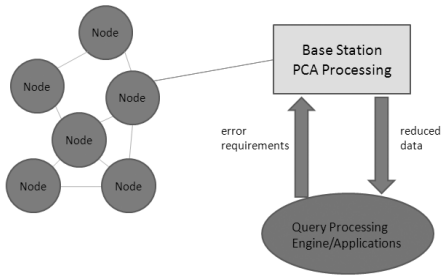
3. FRAMEWORK DESCRIPTION

In this section, we describe some details about our framework for implementing data reduction. Since we assume that we work with a probabilistic data stream and that the probabilistic characteristics of the data stream are constant during a significant period of time, we do not need to send the FeatureVector with every data set from the base station to the query processing engine. We can permanently store this vector at the engine. In order to keep information up-to-date, we can update it periodically. All the assumptions we made for this work can be found in [23]

3.1 ARCHITECTURE

The framework architecture is depicted in Figure 1. In the figure, sensor nodes proactively send information in the direction of the base station. The

base station collects the stream in a sliding window of size m and processes it by applying the PCA algorithm. The query processing engine provides error requirements to the base station and receives back the reduced data.



(Fig. 1) Framework architecture

3.2 PCA DATA REDUCTION

In our PCA-based general algorithm for data reduction, the input parameters are required for correct data processing: S is the Data Stream, ε is the error provided by the query processing engine, and m is the sliding window size which represents the maximum number of n-attribute tuples that can be processed at the base station during one cycle.

In the algorithm, the first four lines deal with two special cases: (i) when the application agrees to lose all the data (ε equals one), the base station sends nothing to the query processing engine, and (ii) when the application requires no data loss (ε equals zero), the base station returns the complete data set. The lines 5-10 are for computing the FeatureVector. The loop is responsible for the data reduction restricted error correspondence. The function ERROR in line 15 calculates the error between the OriginalData and RestoredData. We discuss this calculation in Section 3.4.

Algorithm PCA Data Reduction

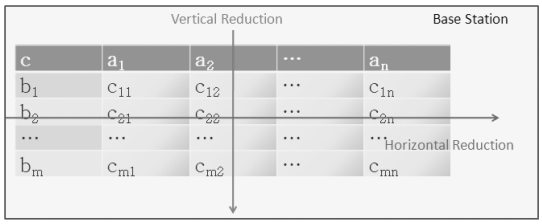
input: S, ε, *m*

```
1:  if (ε == 1) then
2:      return
3:  if ( ε == 0) then
4:      return S
5:  OriginalData = S, Result = OriginalData
6:  compute Mean matrix for OriginalData
7:  DataAdjust = OriginalData - Mean
8:  compute covariance matrix cov for DataAdjust
9:  calculate eigenvectors and eigenvalues matrices
10: obtain FeatureVector by sorting eigenvectors
11:  for i = m down to 1
12:         eliminate column # i from FeatureVector
13:         calculate FinalData
14:         calculate RestoredData
15:         error = ERROR(OriginalData, RestoredData)
16:         if (error < ε ) then
17:                 Result = FinalData
18:                 continue
19:         else return Result
```

3.3 TYPES OF DATA REDUCTION

In this section we consider a base station with window size m × n (m tuples with n attributes) as in Figure 2. We offer two different types of data stream reductions; vertical reduction and horizontal reduction. Later, we give a heuristic for their proper selection.

Horizontal reduction is the data stream compression eliminating complete data tuples. This reduction means elimination of rows (horizontal dimension) from the base station data set. Similarly, we can work with the columns in the same way if we imagine the table rotated 90 degrees and implement a vertical reduction. Hence, vertical reduction is complete elimination of attributes, or columns (vertical dimension) from the base station data set.



(Fig. 2) Types of Data Reduction

In order to find out what kind of reduction is more beneficial in any particular case, we propose a simple heuristic: First we try to estimate how much data we can delete using the vertical and the horizontal reductions and based on this estimation, we can choose a more profitable one. As we have shown, we can eliminate the eigenvectors with the smallest eigenvalues.

Suppose that μ is the upper bound of the eigenvalues we can delete from the FeatureVector. v and h are the number of eigenvalues with a value lower than μ in the vertical and horizontal dimensions, respectively. We can use vertical reduction if v*m < h*n. Otherwise, we can use horizontal reduction. The value of μ is supposed to be chosen experimentally.

3.4 ERROR EVALUATION METRICS

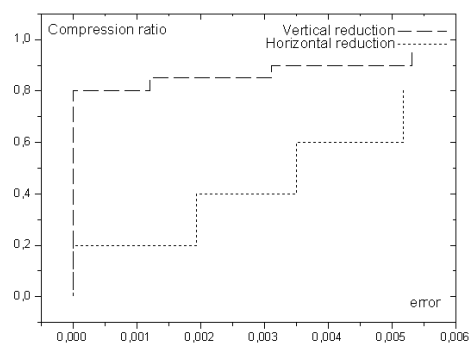
Error evaluation is one of the most important parts of our framework because it has effects on the proper data reduction and thus correct query processing. As possible variants, we can consider any existing error metrics such as sum squared error, maximum error of the approximation, and relative entropy because our method is not oriented toward a specific error metric. Line 15 in the algorithm shows that the procedure ERROR can be chosen independently from other parts of the algorithm.

4. EXPERIMENTS

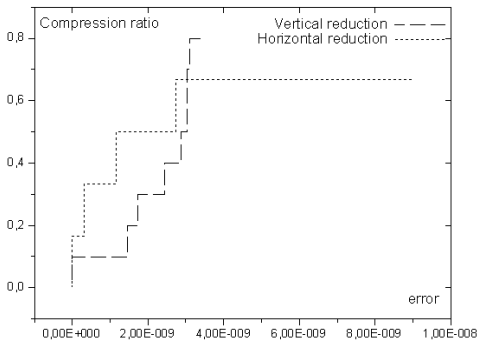
In this section, we describe our implementation of the proposed algorithm and several experiments with some real data sets. We have used Weather Data that includes air temperature, dew point temperature, atmosphere pressure, wind speed, and altitude measurements for a station at the University of Washington for the year 2007 [24]. As a second data set, we have used Stock Data that consists of information on trades that were performed daily in the end of April of the year 2007 [25].

4.1 WEATHER / STOCK DATA EXPERIMENT

Figure 3 shows an illustration of the dependency between the compression ratio and the relative approximation error in the cases of horizontal (dotted line) and vertical (dashed line) reductions for the weather data. The results are stepwise functions since we eliminated entire tuples (or attributes for vertical reduction). The value of the steps depends on the sliding window size m (or number of attributes n). As you can see, in this particular case, the vertical reduction is more beneficial than the horizontal reduction over the entire error range.



(Fig. 3) Weather tuples approximation



(Fig. 4) Stock tuples approximation

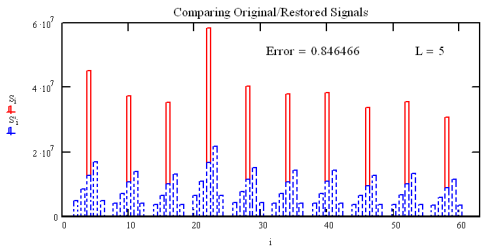
Figure 4 shows the dependency between the compression ratio and the relative approximation error in the cases of horizontal (dotted line) and vertical (dashed line) reductions for the stock data. The Stock Data window size was twice as small as the Weather Data, while the number of attributes was bigger by one unit. In the figure, you may notice that in contrast to the Weather Data, there is an error range where it is more profitable to use horizontal reduction than vertical reduction. This example clearly illustrates a situation where we can select an appropriate reduction technique based on an acceptable error value and a posteriori knowledge.

4.2 WAVELET EXPERIMENT

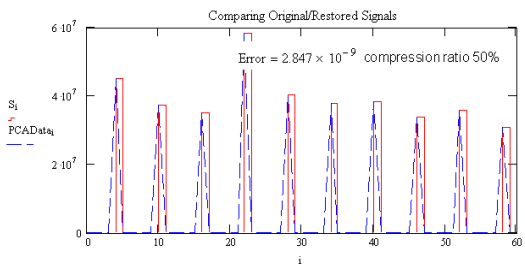
The purpose of this experiment is to compare approximation properties of the PCA and wavelets. We use the same weather and stock market data as in the previous experiment. Also, we use the Daubechies wavelet since it is the most commonly used and the approximation error as error metric.

Figure 5 shows the original and restored stock data after a wavelet approximation. Here, the solid line indicates the original data; the dashed line indicates the restored data. In the figure, the number of used levels of wavelets is five. This is the maximally correct approximation provided by these sorts of wavelets. Obviously, the approximation is not good even with the maximal number of wavelet coefficients.

The PCA approximation, corresponding to the maximally correct approximation provided by the wavelets (with the same compression value of 50%), is indicated in Figure 6. As in Figure 5, the solid line indicates the original data and the dashed line indicates the restored data.

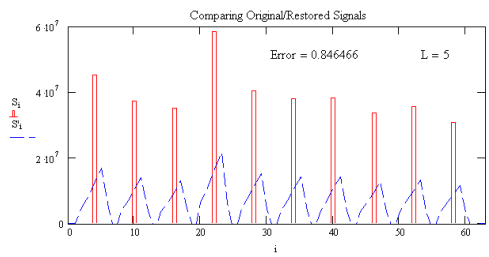


(Fig. 5) Wavelet approximation (1, stock data)



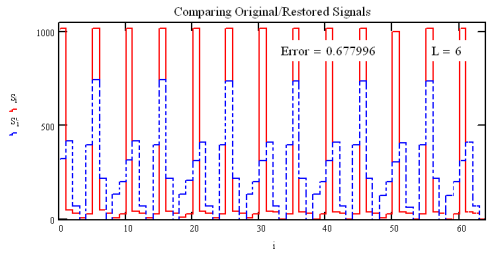
(Fig. 6) PCA approximation (stock data)

The PCA approximation is so much better that the plots overlap each other if they are drawn in the same style. This is why we draw the PCA data by using lines rather than in a step-function style. For comparison, the same plot, but after wavelet approximation, is depicted below in Figure 7.



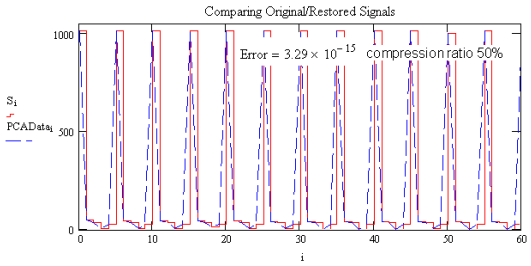
(Fig. 7) Wavelet approximation (2, stock data)

Figure 8 shows the original and restored weather data after wavelet approximation. Here, the solid line indicates the original data and the dashed line indicates the restored data. In the figure, the number of used levels is six. This is the maximally correct approximation provided by this sort of wavelets.



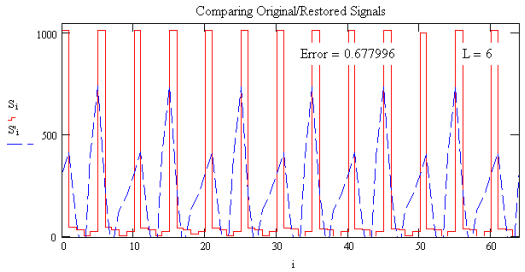
(Fig. 8) Wavelet approximation (1, weather data)

The PCA approximation corresponding to the maximally correct approximation provided by wavelets (with the same compression value 50%) is shown in Figure 9.



(Fig. 9) PCA approximation (weather data)

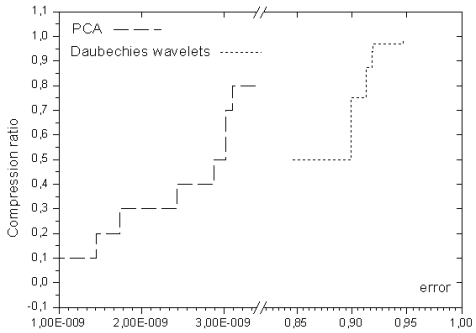
As in the case of stock data, a PCA approximation is also much better. Again, the plots overlap each other if they are drawn in the same style. For comparison, the same plot, but after a wavelet approximation, is depicted below in Figure 10.



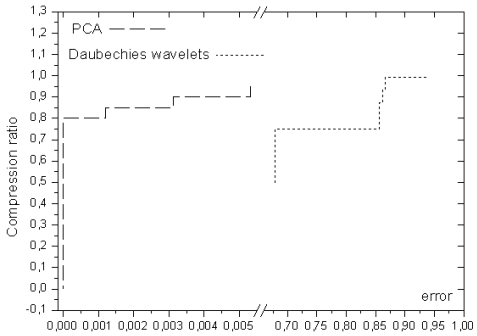
(Fig. 10) Wavelet approximation (2, weather data)

As you can see, error values are extremely different and much better for the PCA. Note that the compression ratios for the both cases are 50%. The compression ratios of the wavelet and the PCA approximation for stock and weather data are shown in Figures 11 and 12, respectively.

From this, we conclude: (i) the minimal compression ratio provided by Daubechies wavelets is limited to 50% while PCA potentially provides all possible values from the entire range. PCA depends only on the sliding window and data stream parameters such as the possible number of tuples allowed for processing and the number of attributes. (ii) PCA is much better than Daubechies wavelets in respect to compression ratio and relative approximation error. Although wavelet approximation generates a bigger error than PCA, it provides a compression ratio close to 100%.



(Fig. 11) Compression ratios of PCA and Daubechies wavelets (stock data)

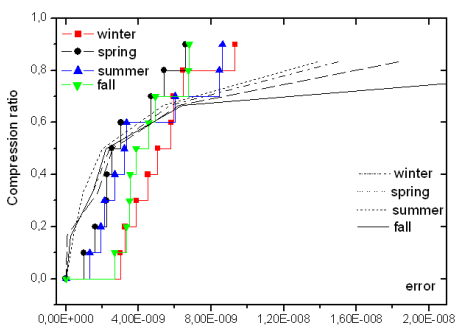


(Fig. 12) Compression ratios of PCA and Daubechies wavelets (weather data)

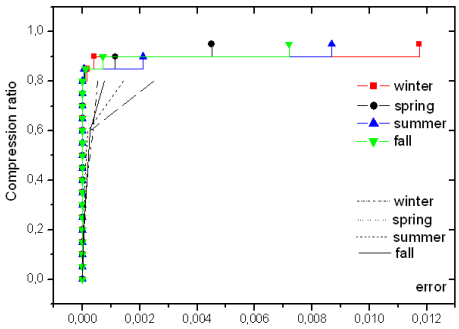
4.3 A TIME STAMP EXPERIMENT

In this experiment, we find how data time stamps influence the compression ratio in terms of the relative approximation error. Basically, we will discover how our method handles a time correlation. We use same weather and stock market data with different time stamps and the relative approximation error

For stock data we use four different data sets created the first months of winter, spring, summer, and fall 2007. We assume the first month of winter, 2007 is January. You may find horizontal and vertical reductions for the stock data in Figure 13.



(Fig. 13) Horizontal and vertical reductions (stock)



(Fig. 14) Horizontal and vertical reductions (weather)

In the figure, the lines without symbols represent vertical reduction, and the lines with symbols represent horizontal reduction. As in our first experiment for each of four time stamps, there is an error range where it is more profitable to use vertical reduction than horizontal reduction. However, even though vertical reduction for the spring period is more preferable among the entire vertical reductions of the experiment, its usability range is the shortest. At the same time, even though horizontal reduction for the winter period is worst, it has the longest usability range for the vertical reduction.

For the weather data, we use four different data sets created the first months of winter, spring, summer, and fall 2007 as we use for the stock data. In Figure 14, the lines without symbols represent vertical reduction and the lines with symbols represent horizontal reduction. For all the entire error range, horizontal reduction operates better than vertical reduction.

From the result, we conclude: (i) Stock data has a pretty strong dependency between the compression ratio and the time stamp while in the case of weather data this dependence is not really sensible. (ii) The compression ratio for horizontal reduction is better for all of the error range in the case of weather data but this is not true for stock data. This difference

proves that the proper selection between two compression techniques is important for every data set and depends on the data set's internal properties. (iii) The degree of usability of one or another reduction technique can depend on the time stamp.

5. CONCLUSION

In this paper, we presented a new PCA-based data stream reduction scheme for sensor networks assuming cooperation between a base station and a query processing engine. We established two different data reduction methods, horizontal and vertical reductions, and proposed a technique for selecting one between them. We tested our scheme on two existing types of real data sets. The simulation result demonstrated that our reduction algorithm achieved a high data compression ratio with a small relative approximation error. This proved that our proposed framework is appropriate for actual sensor network data processing and applicable for real data sets.

REFERENCES

[1] M. Weiser, "The Computer for the 21st Century", Scientific Am.,; reprinted in IEEE Pervasive Computing, pp. 19-25, 2002.

[2] D. Sahs, and A. Mukherjee, "Pervasive Computing: A Paradigm for the 21st Century", IEEE Computer Society, March 2003.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, "A Survey on Sensor Networks," IEEE Communication Magazine, August 2002.

[4] F.L. Lewis, "Wireless Sensor Networks," Smart Environment: Technologies, Protocols, and Applications, New York, 2004.

[5] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer, "EasyLiving: Technologies for Intelligent Environments", HUC 2000, LNCS 1927, pp. 12-29, 2000.

[6] J. Cho, and E. Hwang, "An Exhibition Reminiscent System for Ubiquitous Environ-ment," Proc. of Int'l Conf. on Computer and Information Technology, Sept. 2006.

[7] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," PODS, 2002.

[8] A. Arasu, B. Badcock, S. Babu, J. McAlisher, and J. Windom, "Characterizing mem-ory requirements for queries over continuous data streams", Proc. Of ACM Symp. on Principles of Database Systems, June 2002.

[9] M. Datar, et. al, "Maintaining stream statistics over sliding windows," Proc. of Annual ACM-SIAM Symp. on Discrete Algorithms, 2002.

[10] S. Chaudhuri, R. Motwani, and V. Narasayya, "On random sampling over joins," Proc. of ACM SIGMOD, June 1999.

[11] J. Vitter, "External memory algorithm and datastructures", In J. Abello, editor, External Memory Algorithms, Dimacs, 1999.

[12] P. Indyk, "Stable distributions, pseudorandom generators, embeddings and data stream computation," Proc. of Annual IEEE Symp. on Foundations of Computer Science, 2000.

[13] A. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. Strauss, "Fast, small-space algorithms for approximate histogram maintenance", Proc. of the Annual ACM Symp. on Theory of Computing, 2002.

[14] K. Chakrabarti, M. N. Garofalakis, R. Rastogi, and K. Shim, "Approximate query processing using wavelets," Proc. of Conf. on VLDB, Sept. 2000.

[15] A. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. Strauss, "Fast, small-space

algorithms for approximate histogram maintenance," Proc. of the Annual ACM Symp. on Theory of Computing, 2002.

[16] A. Gilbert, et al, "Surfing wavelets on streams: One-pass summaries for approximate aggregate queries", Proc. of Conf. on VLDB, 2001.

[17] A. Deligiannakis, Y. Kotidis, and N. Roussopoulos, "Compressing historical infor-mation in sensor networks," ACM SIGMOD, 2004.

[18] J. Xie, J. Yang, and Y. Chen, "On Joining and Caching Stochastic Streams," ACM SIGMOD June, Baltimore, MD, 2005.

[19] H. Liu, S. Hwang, and J. Srivastava, "Probabilistic Stream Relational Algebra: A Data Model for Sensor Data Streams", TR, July 12, 2004.

[20] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong, "Approximate Data Collec-tion in Sensor Networks Using Probabilistic Models," ICDE 2006.

[21] A. Deshpande, C. Guestrin, S. R. Madden, "Using Probabilistic Models for Data Management in Acquisitioned Environments," Proc. of the 2005 CIDR Conference.

[22] R. Vidal, Y. Ma, and S. Sastry, "Generalized Principal Component Analysis," Proc. of Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 621-628, 2003.

[23] A. Fedoseev and E. Hwang, "Data Stream Approximation Using Principal Component Analysis for Sensor Network", Int'l Conference on Convergence Information Technology, Nov. 2007.

[24] http://www.k12.atmos.washington.edu/k12/gr-ayskies

[25] http://finance.yahoo.com/q/hp?s=GE

◐ 저 자 소 개 ◑

Alexander Fedoseev

2004년 Tomsk State University, Russia 졸업(학사)
2008년 고려대학교 대학원 전기전자공학과(석사)
2008년 ~ 현재 LG 전자
관심분야 : 센서 네트워크, 데이터 스트림 쿼리, 분산 컴파일 시스템
E-mail : alisher@korea.ac.kr

최 영 환(Young-Hwan Choi)

2007년 고려대학교 전자공학과 졸업(학사)
2007년 ~현재 고려대학교 대학원 전기전자공학과 석·박사 통합 과정
관심분야 : Ubiquitous, Bio-Informatics System, U-Healthcare
E-mail : work48@korea.ac.kr

황 인 준(Eenjun Hwang)

1988년 서울대학교 컴퓨터공학과 졸업(학사)
1990년 서울대학교 대학원 컴퓨터공학과 졸업(석사)
1998년 미국 메릴랜드 주립대학교 대학원 전산학과 졸업(박사)
2004년 ~현재 고려대학교 전기전자전파공학부 부교수
관심분야 : 데이터베이스, 멀티미디어 인덱싱 및 검색, 오디오/이미지 특징 추출 및 표현, 대용량 데이터 처리 및 관리, 데이터 분류 및 추천.
E-mail : ehwang04@korea.ac.kr