Efficient Similarity Search Techniques with a Real-Time Approximate Analysis in Streaming Database

Ling Wang[1], Tie Hua Zhou[2], Kyung Ah Kim[3], Eun Jong Cha[3] and Keun Ho Ryu[2*]

[1]*Department of Computer Science and Technology, School of Information Engineering, Northeast Dianli University, Jilin, China*
[2]*Database/Bioinformatics Laboratory, School of Electrical & Computer Engineering, Chungbuk National University, Chungbuk, Korea*
[3]*Department of Biomedical Engineering, Chungbuk National University, Chungbuk, Korea*

[1]*smile2867@dblab.chungbuk.ac.kr,* [2]*{thzhou, khryu}@dblab.chungbuk.ac.kr,*
[3]*{kimka, ejcha}@chungbuk.ac.kr*

Abstract

In many applications such as sensor networks, similarity search is more practical than exact match in stream processing, where both the queries and the data items are always change over time. The volumes of multi-streams could be very large, since new items are continuously appended. The main idea is to build a small size of synopsis instead of keeping original streams by using our proposed techniques, then to provide approximate answers for many different classes of aggregate queries. In this paper, we present D-skyline and T-skyline methods give almost "true" results on approximated analysis for similarity search query in streaming environments.

Keywords: *skyline, similarity search, stream processing*

1. Introduction

There are many real applications require the manipulation of data streams [1-3], *e.g.*, sensor networks, telecommunications, web applications, red ocean prediction. In a stream application, we need mechanisms to support continuous queries over real time data that is continuously updated from the environment. This requirement is significantly different than what happens in a traditional database where data is stored in static tables and then existing data. Furthermore, a stream scenario brings a number of unique query processing challenges due to its vast applicability where the incoming streams go fast, in a very huge size, especially for a continuously query emphasizing on fast responses. Here, "continuous" means it will never been stop to process data and queries what are always refreshed by real time, not only for existing data. One direction is to exploit the sharing computation among the queries and objects considered computation sharing of sliding window aggregates. Another direction is building small space summaries over data streams to provide approximate answers for many classes of aggregate queries. In this paper, we considered both of them to do the analyses for the streaming process. Figure 1 shows our general work flows. Multiple evolving streams as the input of data stream database, a common approach is to use sliding window structures to manage the input data cells and use our proposed skyline techniques to mining the most

* Corresponding author

recent useful datasets, then retain the most representative ones to store in memory and construct a synopsis by timestamp segmentation.

Since multi-streams are usually too large to be stored in main memory, skyline algorithms on similarity search are used in the sense that emerged as an important summarization technique happens in the main memory. Several algorithms [4-7] have been proposed targeting the efficient skyline evaluation on large datasets. These solutions always classified into two categories, depending on whether they assume an index. Intuitively, the index-based schemes are faster than index-independent strategies, since they avoid accessing the entire data collection, yet their applicability is significantly limited by the indexing requirement. They may not to be indexed which the data are dynamically produced in many streaming applications (such as moving sensors, predicted analysis).

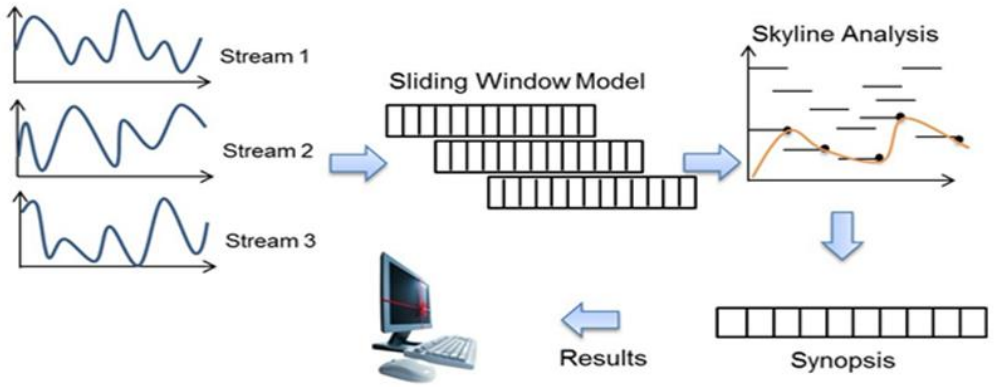

Figure 1. Work Flow

In many stream applications [8-10], similarity search is more practical than exact match in stream processing, where both query and data change over time. The length of multi-streams can be very large, since new values are continuously appended. Therefore, the similarity of multi-streams is expressed by means of the last values of each stream, using a sliding window approach. The naïve approach is to delete the old items by using timestamp techniques, to re-apply the data reduction mining technique to the new items, and finally to store the resulting summarized in the access method for the further final approximated result analysis. This process is very efficiently both in CPU time and numeric items calculated by one-pass processing. Here, we just pay attention to the challenge of real time skyline maintenance on data streams [11-15]. Actually, large database is not simply considered as a stream database because of streaming data is not only containing huge data volumes, but distributed, continuous, rapid, time varying. Therefore, the general techniques may not suit for streams exactly. Accuracy and fast responses require approximated answers are more important in distributed stream processing for the similarity search. So, focus on similarity search in streaming environment, D-skyline and T-skyline methods proposed in this paper to give almost "true" results on approximated analysis for different kinds of users needed. We also perform data reduction through synopsis data structures to batch processing with particular relevance to the data stream computation model over sliding windows. Our experimental evaluation has shown the detailed effect for approximated analysis by using different kinds of skyline methods, then effectiveness and efficiency of our approach.

2. Skyline Operator

In this paper, we present two algorithms for efficiently processing similar search queries. Our proposed two methods named D-skyline and T-skyline which are focus on requirement for the lowest space usage and fast response to the users in a high accuracy guarantee results. D-skyline algorithm organizes the already computed multi-streams into sub-windows such that the candidate similar search tuples can quickly prune if they are dominated by some other streams. For a candidate query, only their distance under the threshold are calculated and stored into sub-windows. In a defined time series, only the frequent times for each stream could be shown in each sub-window, then the most appeared streams as an approximated result of similar search queries for the whole sliding window domain. The other unsigned items will be removed from memory in order to release space for continuous coming streams. T-skyline is similar to D-skyline except one pass on the sub-windows, which need top-k items as a list only when there are needed later in a whole sliding window process. Figure 2 shows the general skyline example which is only focus on one-to-one process. In the next parts, we will show a detailed analysis for our proposed methods, and finally compare with general skyline technique give a detailed experiment.

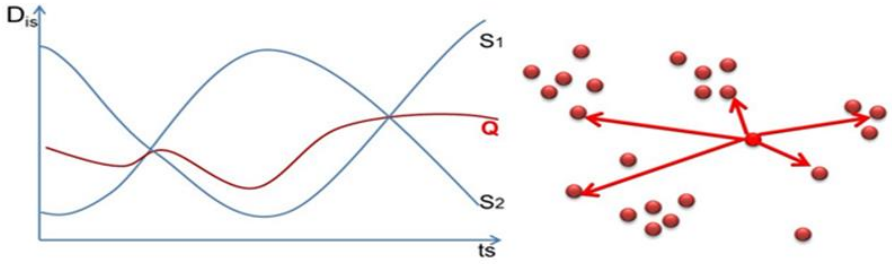

Figure 2. An Example of General Skyline One-to-One Process

2.1. D-Skyline Technique

The first method D-skyline is inspired by the distance-sensitive: under users afforded threshold to select all of multi-streams are compared with object query point to judge them who can be processed in the further synopsis process. Figure 3 is an example to express that a simply descript D-skyline working flows, incompletely. Q means a query stream, and other original streams present in this figure use S1, S2. And the attribute in x-axis shows a timestamp running, and in y-axis are compared distance values. D-skyline only concerned about those points that contained in the area of afforded distance threshold. Generally, skyline always pay attention to solve multi-dimension also say multi-attributes problem. So, our proposed technique is not only a distance mining algorithm to reduce the volumes of original datasets, also give a method to classify the datasets which direct at some different attributes, see the right part of Figure 3. Actually, for these different attributes there are some certain special contacts with each other. And these outliers could not simply think that are noisy data, maybe they are not having a distinguished feature with others.

Actually, compare with the general skyline method that each point needs to calculate with query stream points and store them all in the memory. However, how to get more exactly result on low space usage and fast response are more important for streaming process. Typically, skyline is a summarized dataset which have been processed by some useful aggregates as a preprocessing phase instead of the whole original datasets from incoming streams. Generally, preprocess over stream data is very difficult to implement and costly in

processes. One way to evaluate the query over sliding windows which only control the most recent datasets and use timestamp to determine data expire. By this way, preprocessing can be used and only store the necessary results in synopsis after some partial aggregates is enough. Our D-skyline technique is a distance-sensitive technique to construct an unequal synopsis under a user afforded distance threshold, and then classify all these datasets labeled. Finally base on these labeled datasets in synopsis to find out the most popular ones. The detailed algorithm shows in the following:

```
D-skyline algorithm:
Input: multi-streams
Output: Final_list
θ :threshold
Dis: distance between query object
Stream_list: list of selected streams
Final_list: Top-k streams over returned list
Begin
  For the whole selected sliding window,
      Sort the set of stream_list;
    If meet timestamp*, calculate distance -θ ≤Dis≤ θ;
    Then sign streams into the stream_list;
  Return Final_list;
End.
```

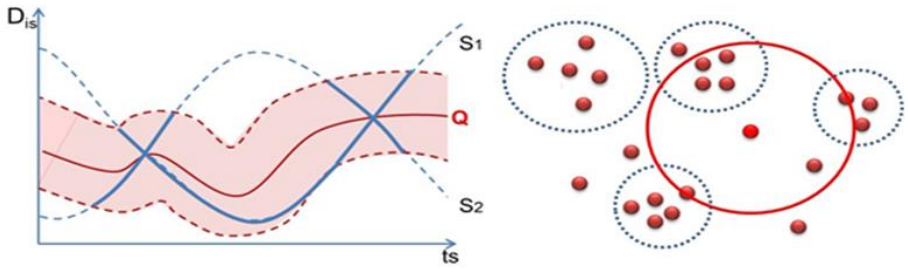

Figure 3. An Example of D-skyline Technique

2.2. T-Skyline Technique

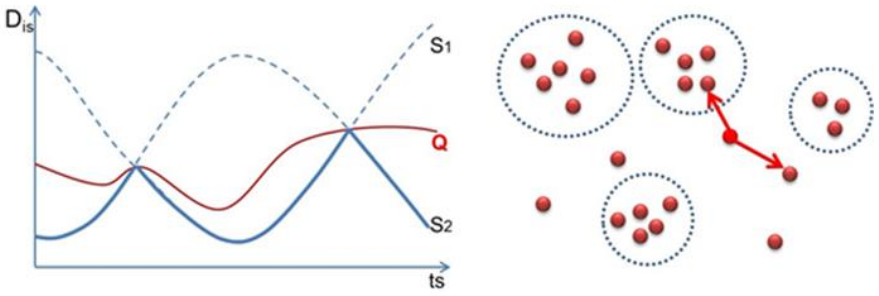

Figure 4. An Example of T-skyline Technique

Our second method T-skyline is inspired by the Top-k combined: it's also base on transformed synopsis that contains several the same size sub-windows of Top-k lists, as a

preprocess technique on the original data. Figure 4 is a simply example to see how T-skyline works, but not full description. We demonstrate that both of our proposed methods as data reducing mining techniques are efficiency for similar search over multi-streams distributed processing. The important thing is that they give more exactly approximate results as lower as possible on space usage. Compare with just proposed D-skyline method, T-skyline has two more useful features. One thing is that T-skyline may use the lowest space usage among these three methods in a high accuracy. The other one is that T-skyline could give a satisfied exactly answer on a more fast response requirement who only need a large result set not mention the sequence (see Figure 5, these numerics mean some different streams). Stream 3, 8 and 1 are in a same level in the result of T-skyline, that means there are the same values for them. Certainly, if users require for a small set of sequence results, D-skyline should deal and give a better result than T-skyline technique. That's why in this paper we proposed two advanced skyline methods, they are suit for different kinds of user's requirements. In the following evaluation section gives a more detailed explanation.

The detailed algorithm shows in the following:

```
T-skyline Algorithm:
Input: multi-streams
Output: Final_list
θ :threshold
Dis: distance between query object
Stream_list: list of selected stream
T_stream_list: Top-k streams over stream_list
Final_list:Top-k streams over returned list
Begin
  For the whole selected sliding window,
      Sort the set of Stream_list;
    If meet timestamp* calculate the distance -θ≤Dis≤θ;
    Then sign streams into the
         stream_list;
      For sorting stream_list;
     Return T_stream_list;
  Return Final_list;
End.
```

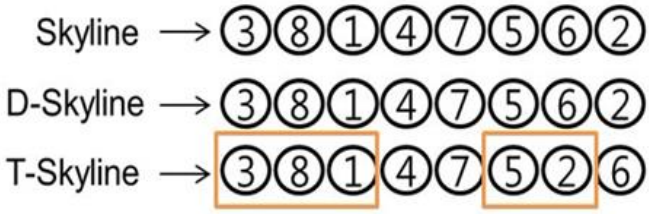

Figure 5. Approximate Results Comparative Analysis

3. Experiment and Evaluation

In our test, we use almost 60000 data from 9 different streams (see Table 1) to evaluate for each method, the detailed discussion looks in the following. The general skyline method is not considered the sliding window, even not sub-window process. But for D-skyline and T-skyline, an important performance measure is the number of candidates that each method reports after the completion of the distance evaluate step, and to discuss the size of sub-

window "w" effect by using 100, 200 and 500 items. The other important performance measure for T-skyline is the exact value of "k" which determines size of list in each sub-window. We also give a detailed evaluation of types retrieve rates for a different value of k is 1, 2 and 5 as shown in Figure 7. By k increasing, labeled different types of streams are also increasing. And by w increasing (see Figure 6), types identified may be not increasing except all of streams will be signed on a very big afford of distance such as 20 as a warning. D-skyline will give a high types retrieval ratio than T-skyline, especially for w increasing as shown in figure 8, but not more than general skyline method of completed types identified.

Table 1. Original Data Error Rate

S1	S2	S3	S4	S5	S6	S7	S8
0.3%	42.63%	0.27%	0.3%	0.54%	0.38%	0.29%	0.27%

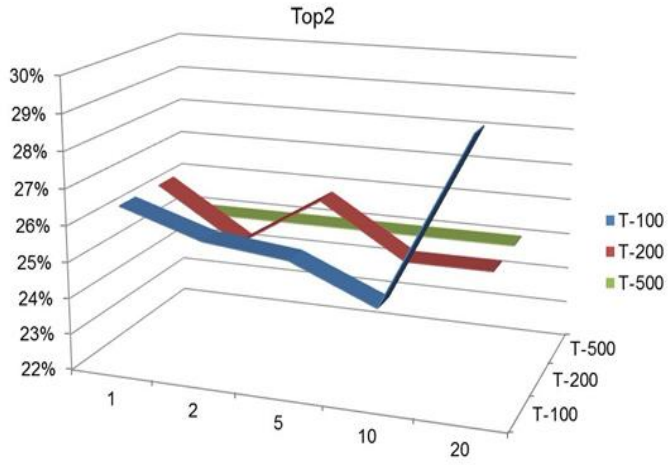

Figure 6. Types Retrieve Rates of T-100, 200 and 500 of T-Skyline Technique

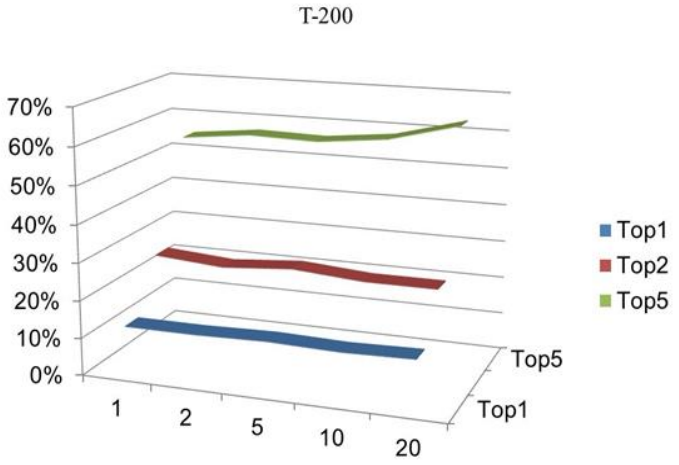

Figure 7. Types Retrieve Rates of Top1, 2 and 5 of T-Skyline Technique

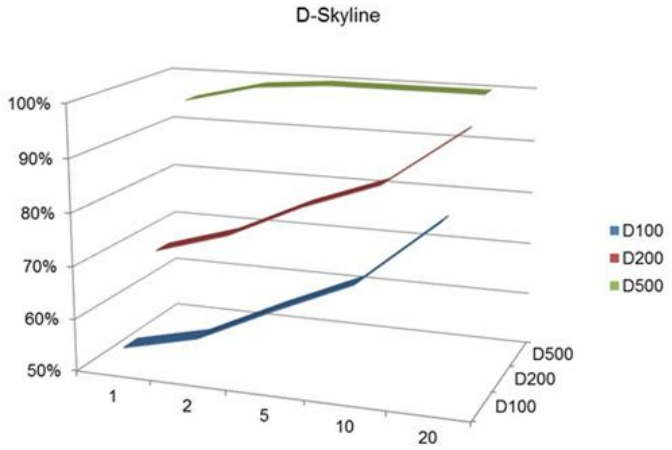

Figure 8. Types Retrieve Rates of D-Skyline Technique

More types identified may be giving a high actuary result, but it also needs more complex computation in the further process. Actually, for a given query, the results of these three methods are the same exactly in our test, although for an approximated result of D-skyline and T-skyline methods. The other more important thing is space usage and the general trend of space usage for D-skyline technique is that by "w" increasing it requires less space (see Figure 9). This law as it happens on T-skyline technique looks the same (see Figure 10). But for a special character "k", it may need more space when "k" increasing. Compare with general skyline shows in Figure 11, general skyline needs the most highest space usage 40% than the other two methods which only less than 1%. In order to reduce the high space usage for stream processing and fast response for continuous queries, D-skyline and T-skyline give more exactly results than traditional method in our test.

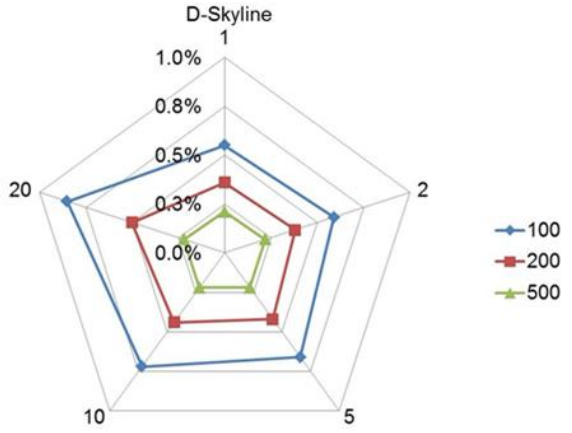

Figure 9. Space Usage of D-Skyline Technique

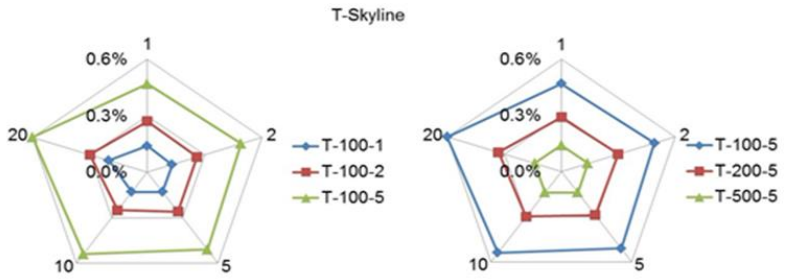

Figure 10. Space Usage of T-Skyline Technique

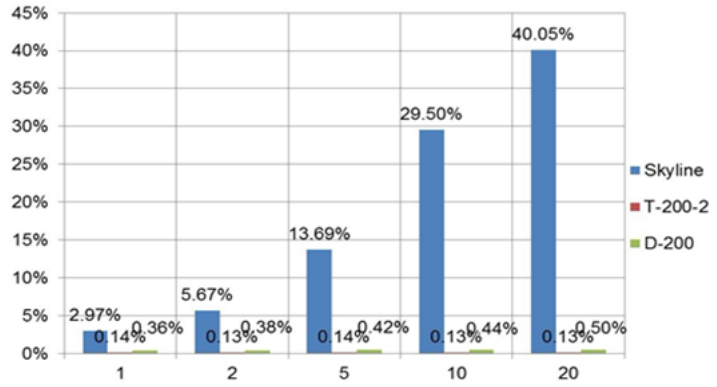

Figure 11. Space Usage of General Skyline, D-skyline and T-skyline Techniques

4. Summary

In this paper, focus on similarity search in streaming environment, our proposed D-skyline and T-skyline methods give almost "true" results on approximated analysis for different kinds of users' needs. We also perform data reduction techniques to activate synopsis data structures in order to batch process a very huge stream datasets concern particular relevance to the data stream computation model. The traditional skyline is not suit for streaming data process, our proposed D-skyline and T-skyline is more excellent to adapt to this kind of data characters. In the test, T-skyline exhibits a more improvement on space usage than the others, but an exact result without sequence. D-skyline is also a space usage improvement method to give a high accurate result in sequence for a special need. Our experimental evaluation has shown the detailed effect for approximated analysis by using different kinds of skyline methods, then effectiveness and efficiency of our approach.

Acknowledgements

This work was supported by the Korea Institute of Energy Research (KIER) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012-0000478).

References

[1] L. Golab and M. T. Özsu, J. ACM SIGMOD Record, vol. 32, no. 5, **(2003)**.
[2] I. Botan, R. Derakhshan, N. Dindar, L. Haas, R. J. Miller and N. Tatbul, J. VLDB Endowment, vol. 3, no. 232, **(2010)**.
[3] B. Chandramouli, J. Goldstein and D. Maier, J. VLDB Endowment, vol. 3, no. 220, **(2010)**.
[4] K. C. K. Lee, B. H. Zheng, H. J. Li and W. C. Lee, Editors, "Approaching the Skyline in Z-Order", Proceedings of the 33rd International Conference on Very Large Data Bases, Vienna, Austria, **(2007)** September 23-27.
[5] X. B. Wu, Y. F. Tao, R. C. W. Wong, L. Ding and J. X. Yu, Editors, "Finding the Influence Set through Skylines", Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, Saint Petersburg, Russia, **(2009)** March 24-26.
[6] S. M. Zhang, N. Mamoulis and D. W. Cheung, Editors, "Scalable Skyline Computation Using Object-based Space Partitioning", Proceedings of the 35th SIGMOD International Conference on Management of data, Providence, USA, **(2009)** June 29-July 2.
[7] S. M. Zhang, N. Mamoulis, D. W. Cheung and B. Kao, J. VLDB Endowment vol. 3, no. 1255, **(2010)**.
[8] R. V. Nehme, E. A. Rundensteiner and E. Bertino, J. VLDB Endowment, vol. 2, no. 73, **(2009)**.
[9] T. H. N. Vu, N. K. Park, Y. K. Lee, Y. M. Lee, J. Y. Lee and K. H. Ryu, "Systems and Software", vol. 83, pp. 1930, **(2010)**.
[10] Y. K. Lee, J. P. Shin, K. D. Kim and K. H. Ryu, "Innovative Computing, Information and Control", vol. 7, no. 2945, **(2011)**.
[11] X. M. Lin, Y. D. Yuan, W. Wang and H. J. Lu, Editors, "Stabbing the Sky: Efficient Skyline Computation over Sliding Windows", Proceedings of the 21st International Conference on Data Engineering, Tokyo, Japan, **(2005)** April 5-8.
[12] M. Kontaki, A. N. Papadopoulos and Y. Manolopoulos, "Data & Knowledge Engineering", vol. 63, no. 478, **(2007)**.
[13] N. Sarkas, G. Das, N. Koudas and A. K. H. Tung, Editors, "Categorical skylines for streaming data", Proceedings of the ACM SIGMOD International Conference on Management of data, Vancouver, Canada, **(2008)** June 10-12.
[14] D. Fuhry, R. M. Jin and D. H. Zhang, Editors, "Efficient Skyline Computation in Metric Space", Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, Saint Petersburg, Russia, **(2009)** March 24-26.
[15] A. D. Sarma, A. Lall, D. Nanongkai and J. Xu, J. VLDB Endowment, vol. 2, no. 85, **(2009)**.

Authors

Ling Wang is currently an associate professor in the Department of Computer Science and Technology, Northeast Dianli University. She received the Ph.D degree in Computer Science from Chungbuk National University, Korea, in 2013.

Her research interests are mainly in the areas of multimedia database, image processing, data mining, data stream processing, sensor data processing, and spatial-temporal database.

Tie Hua Zhou is a Ph.D. student in Database/Bioinformatics Laboratory of Chungbuk University. He received the M.S. degree from Chungbuk University, Korea, in 2010.

His research interests mainly include multimedia image processing, data mining, and spatial-temporal database.

Kyung Ah Kim is currently an associate professor in the Department of Biomedical Engineering, Chungbuk National University. She received the Master of Science and Master of Engineering degrees from Physics department, Medical and Biological Engineering department, Chungbuk National University, in 1993 and 2001, respectively.

Her research interests mainly in biomedical and measurement, bio-signal analysis, medical informatics and cardio pulmonary tool.

Eun Jong Cha is currently a professor in the Department of Biomedical Engineering, Chungbuk National University. He received the Ph.D degree in Biomedical Engineering from University of Southern California, USA, in 1987.

He worked as a research associate in the Department of Biomedical Engineering, Southern California University (1987 – 1988).

His research interests mainly in biomedical and measurement, bio-signal analysis, medical informatics and cardio pulmonary tool.

Keun Ho Ryu is currently a professor in the Department of Electrical & Computer Engineering, Chungbuk University, as well as a Leader of Database/Bioinformatics Laboratory. He also served a Director of Research Institute for Computer and Communication. He received the Ph.D degree from Yonsei University, Korea, in 1988.

Prof. Ryu worked not only at University of Arizona as Post-doc and research scientist but also at Electronics & Telecommunications Research Institute, Korea. He has served on numerous program committees including AINA, ICWE, WAIM, APWeb, WISE, and FITAT, and so on. He is a member of the IEEE since 1982 and member of the ACM since 1983.

His research interests are mainly in the fields of temporal, spatial, and spatiotemporal databases and their related area included temporal GIS, ubiquitous computing and stream data processing, active database, data mining, database security, knowledge base information retrieval, and biomedical and bioinformatics.