International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.12 (2015), pp.355-368
http://dx.doi.org/10.14257/ijmue.2015.10.12.34

An Efficient Social Search Method Based on Location and User Preferences in Mobile Environments

Byoungyup Lee[1], Minje Ahn[2], Jongtae Lim[2],
Kyungsoo Bok[2] and Jaesoo Yoo[2]

[1]Paichai University
[2]Chungbuk National University
bylee@pcu.ac.kr, {mjahn, jtlim, ksbok, yjs}@chungbuk.ac.kr

Abstract

Due to advances in mobile networks and smartphone popularity in recent years, users can now connect to the internet anytime from anywhere and social searches are becoming popularly. Unlike conventional web searches, the social search provides optimal search results according to user preferences. In this paper, we propose a mobile social search method based on popularities and user preferences. The popularity is calculated by collecting the visiting records of users. The user preferences are generated by the actual visiting information among the search results. We process a skyline query to extract the meaningful information from the candidate objects with multiple features. The proposed method ranks social search results by combining user preferences and popularity with the skyline query processing mechanism. To show the superiority of the proposed method, we compare it with the existing method through performance evaluation.

Keywords: *Social search, Social network, Location based service, Popularity, User preferences*

1. Introduction

The rapid development of wireless communication technology and smart phones makes it possible to access the internet anywhere and anytime. Since most of the mobile devices are equipped with GPS functions, it is easy to obtain the location information of mobile users. Recently, mobile social network services are growing quickly due to the ubiquity of mobile users. The location based mobile social search is very important for the mobile users of mobile social network services. Different from traditional web searches, the mobile social search analyzes user preferences from various social network services and then takes into account them to search the optimal results for users. The social search in mobile environments has important location issues such as how to maintain the recent locations and how to rank search results by using location information. Different from conventional computing environments, the mobile computing environments should view the search results in a mobile phone with a small screen size. Therefore, efficient filtering methods are important for mobile social searches [1- 3].

Recently, various studies have been conducted to provide effective social search services. The popularity and experts rating are used for improving social searches [1]. Popularity is evaluated by analyzing contents published in social network services and historical searching keywords. The related experts are selected based on the obtained keywords. The rating information about the selected experts is used for improving social searches. However, the problem is that the popularity and experts rating cannot represent personal preferences of each mobile user. The existing social search methods based on user preferences were proposed to improve social searches. In order to do this, user behaviors can be obtained by analyzing social data collected from social network services. The related social data within a specific period has to be collected [7-8]. If enough related

ISSN: 1975-0080 IJMUE

social data cannot be collected, the accuracy of social search results will be affected. In [9], the profile information of users is used for social search. In this method, when performing mobile social searches, the profile information of users is considered. The problem is that most of the user profile information is not changed from the first time they are registered in spite that user preferences may change when time passes. Therefore, profile information of users also cannot represent user preferences well.

User location information can be collected and used from social network services. Since the location information is highly related with user privacy, some anonymous data collection techniques [4-5] are used to collect user location data. The location data used in social search services is different from that used in location search services [6]. Location search services need to search precise location data, while social search services utilize location to search social data but do not need to collect precise location data.

In order to resolve the problems of the existing methods, in this paper, we propose a mobile social search method that considers popularity and user preferences. The popularity is calculated by collecting real visiting records of users in the social network services. The user preferences are evaluated by analyzing the social data such as published contents, tags, and email of users in social network services. We process a skyline query to extract the meaningful information from the candidate data sets with multiple features. The proposed method ranks search results by combining user preferences and popularity with the skyline query processing mechanism.

The remainder of this paper is organized as follows. Related work is covered in Section 2. We present the proposed method in Section 3. Experiment evaluations are presented in Section 4. Finally, Section 5 contains our conclusions and future works.

2. Related Work

In order to support social searches, user preferences are important to be obtained. Generally, user preferences are obtained by analyzing collected social data. The implicit social data collection methods obtain social data from user life with the approval of users. Social data is collected to evaluate user preferences, such as emails, messages, location information, and various actions from social network services. In [7], SLANT was proposed to search results by using user information collected from emails and twitter, such as analyzing the keywords and links in email bodies and follower information. The hot search keywords of historical social search results in social network services can be obtained. In SLANT, social data can be collected only when getting approval from users because of privacy problems. In order to obtain user preferences reliably, social data needs to be collected at least one month from the time when getting approval from users.

In order to reduce the period time of data collection, explicit data collection methods evaluate user preferences by collecting user profile data from register information of users. The user preferences are used to improve social searches. In [9], user social data was collected by using explicit data collection methods when users join social network services. The user similarity is computed by taking the keywords of user profile information. Users are assigned to different groups according to their user similarity. The user information of the same group is used to improve social searches. For instance, if user j has profile information such as keywords a, b, c, and d, and user i has profile information such as keywords c and d, the user similarity between user i and user j is 0.5. Like this, the user similarity is computed by using the profile information when a new user joins a social network service. The problem is that user profile information is not changed when time passes but user preferences are changed. Therefore, this method cannot represent user preferences well.

3. Proposed Social Search Method

3.1. System Structure of the Proposed Method

In this Section, we propose an efficient social search method in location based mobile services. The proposed method consists of four steps. First, the candidate location sets are generated by analyzing historical search keywords. The unavailable locations are removed during the first step. Second, the popularity of each location is evaluated by using the access records of users that can be obtained from the existing social network services. Third, the meaningful locations are selected by using the skyline query processing mechanism and the user preferences are evaluated by using the weight values of the selected locations. Finally, the mobile social search results are provided to the users according to ranking values that are computed by combining the popularity and user preferences.

Figure 1 shows the system structure of the proposed method. The collector module is used to gather user information from the social network services. User information, location, search results, and visiting records of users are collected and stored in databases. The location data is stored only when the data is not in the server or update is needed. The unique ID is assigned to the location data when the location is generated for the first time. The related data about the location data are stored in the databases, such as position coordinates, shop name, type, and price. If the visiting positions already exist in the database, only the user visiting information is recorded, which includes user ID, position, and time information. Mobile users use mobile devices to search the requested contents. There are four modules used to process search queries from mobile users. The query processor module is used to evaluate keywords, time, and location from mobile users. The candidate generator module is used to evaluate suitable candidates and popularity values. The skyline module is responsible for assigning weight values according to user preferences. The ranking engine module is used to compute the final social search results according to the popularity generated from the candidate generator module and user preferences generated from the skyline module.

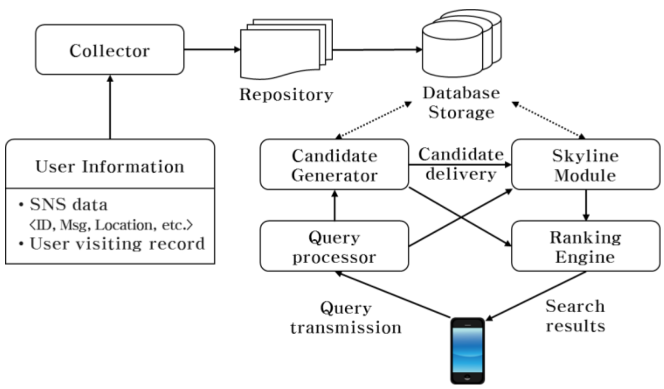


Figure 1. System Structure of the Proposed Method

3.2. Database Schemas

Figure 2 shows the database schema of the proposed method. The collected user social data, location data, and visiting records are stored in the database. Some Tables are used to store the collected data. The location Table is used to store the detailed information of each location, such as the unique ID, name, location coordinates, category, and the specified parameter value. In order to improve efficiency, the location data are managed hierarchically. The user record Table is used to store visiting information of each user. It consists of time, location ID, and feedback. The user Table is used to store the information of user preferences. Each weight value Wi represents user preferences according to different attributes. The weight value Wi is updated by using the existing feedback data of the user record Table. The user preferences are computed by the equation (1), where k is the total number of feedbacks within the most recent T period and di is the value of the ith parameter which can be computed by equation (2). In equation (2), ni is a normalization value aivi of collected data vi by feedback. ai is a parameter to make the normalization value of each attribute within a range 0~1. vi represents the attributes (prices, distances) of locations based on the real visiting locations of users.

User

User_ID	W_1	W_2	...
1402954	0.38	0.62	
1402955	0.71	0.29	

User Record

Time	Location_ID	Feedback
20121009T131523	274	0.43, 0.57
20121009T153010	455	

Location

Location_ID	Name	Location_X	Location_Y	Category	Param_1	...
135	Dunkin	30.234891	-97.7951395	restaurant, seafood	15	
136	BergerKing	30.234890	-97.7951393	restaurant, bakery	20	

Figure 2. Database Schema

$$W_i = \frac{1}{k}\sum_{i=1}^{k} d_i \tag{1}$$

$$d_i = 1 - \frac{n_i}{\sum_{j=1}^{J} n_j} \tag{2}$$

For example, Figure 3 shows 3 feedback data collected from users. Normalized user feedback data are the normalization values of user feedback data. The feature value of each attribute is computed by equation (2). The user preference is computed by equation (1) based on the feature value.

Normalized User Feedback

$\alpha_1 v_1$	$\alpha_2 v_2$
0.68	0.15
0.72	0.30
0.56	0.41

Feature Value

d_1	d_2
0.819	0.181
0.705	0.295
0.577	0.423

User Preference

User 1	
W_1	W_2
0.701	0.299

Figure 3. Feedback Data Collected from Users

3.3. Candidate Result Generation

In the proposed scheme, a visit time is included and transferred with the search information. Therefore, the search efficiency can be increased by applying a method of assigning a high weight to publicly popular locations in the search time slot utilizing a mixed model. The utilization of time information can produce three positive effects. First, the hours of operation of the search location can be utilized in the search process. These can be inferred through the visit records of a corresponding time slot by checking user visit records within a valid time range in the search time. Therefore, locations that a user cannot visit within the expected visit time slot are excluded from the candidate group based on operating hours. Second, effective processing results in response to comprehensive queries can be provided. As shown in Figure 4, the most active check-in records occur between 12:00 and 20:00. The expected value of a comprehensive keyword, such as Restaurant, differs slot by slot when the time is divided into 12:00 to 14:00, 14:00 to 18:00, and after 18:00. It may be a simple location search for regular meals or a location search including Café or Bar depending on the search time. That is, if a query requires a wide range of search results, the number of locations to be searched can be reduced. Third, unnecessary computations can be decreased. The amount of accumulated check-in information collected continuously within a specific period is enormous. Therefore, it would take a great deal of time to sort out the rankings of candidates based on the assignment of user preference and public popularity scores in each module. Therefore, the utilization of time information only exploits check-in information within the valid range so that it can reduce the number of candidates, thereby increasing the search efficiency and computation speed.

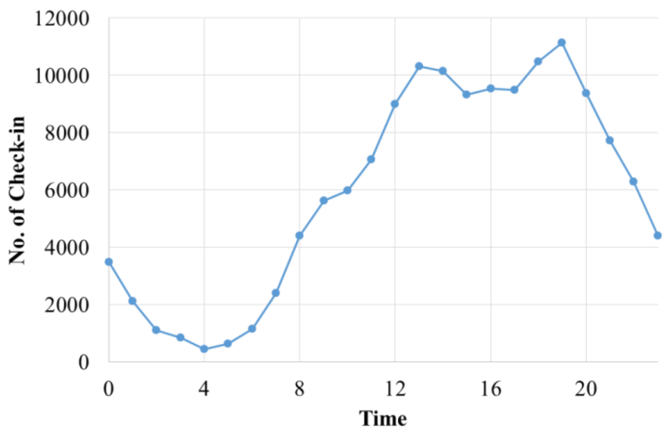


Figure 4. Gowalla User Check-In Record by Time

It is vital to create an appropriate candidate group in order to provide search results required by users. Figure 5 shows a candidate-generation process. First, it extracts location information that includes a core keyword in the location category information in the unit-region Table that corresponds to a user's current location. Locations that have no visit records within the search time are removed from the temporarily generated candidate group based on the time information specified by a user. This is because the businesses in the removed locations are regarded as not operating in the specified time. Through this process, a final candidate list is generated to provide search results. For example, assuming that a user called User34 arrives at a meeting place one and a half hours earlier than the appointment time, which was 16:00, and searches for a suiTable café nearby to wait until the appointment time, he will send a query to the server in the following form: <User34, cafe, 36.002453, 145.356363, 20131102T123124>. The server then identifies his current location through geocoding based on the latitude and longitude sent by the user. Through the location hierarchy structure, the city where the user is currently located is searched, and the locations that have cafés in the sub-category field are searched from the corresponding city Table. Once an initial candidate list is generated via the search keyword, the time information of 14:00 is extracted from the search time "20131102T143124." Locations for which no check-in records are found within the valid time period based on 14:00 are removed from the candidate list. In this way, locations that are currently operating can be checked, and nearby cafés that can be visited are included in the final candidate list.

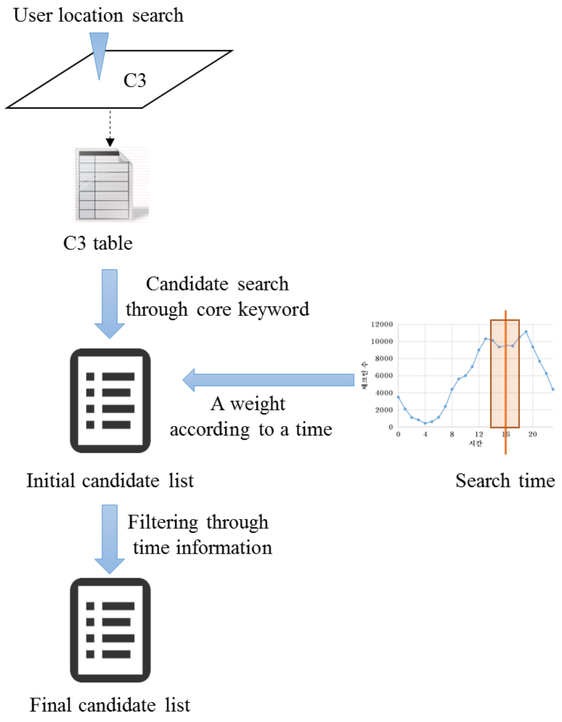


Figure 5. Candidate Group Generation Process

In the proposed scheme, the time information v used to check the operating hours can be calculated via Equation (3). A weight is assigned within a valid range of u_1 and u_2 with respect to the past visit times $t(O_i)$ of each location O_i based on an expected visit time $t(r)$. Here, u_1 and u_2 are values specified by repetitive experiences. If there are no visit records within a valid range, locations are removed from the candidate group. They are taken into consideration to calculate a popularity score $P(O_i)$ through weights by utilizing time information v only when visit records are found.

$$v = \begin{cases} 0, & u_1 < t(O_i) \\ 0.5, & t(r) < t(O_i) \le u_1 \\ 1, & t(r) = t(O_i) \\ 0.5, & u_2 \le t(O_i) < t(r) \\ 0, & t(O_i) < u_2 \end{cases} \qquad (3)$$

3.4. Ranking Algorithm

The proposed scheme performs a four-step processing procedure to return appropriate results to users based on data collected continuously. Once an appropriate candidate group is selected based on the user’s current location, popularity and user preference scores are calculated via the candidate generator and the skyline process, respectively. The ranking engine assigns a final score S_i through Equation (4) by taking the popularity and user preferences. Here, $L(O_i)$ is the location weight, $P(O_i)$ is the popularity, and β is the weight for each in a candidate group O_i. The weight β is determined by the search frequency. The candidate group is rearranged based on the final score, and the final result is returned to the user.

$$S_i = \beta \frac{L(O_i)}{Max(L(O_i))} + (1-\beta)\frac{P(O_i)}{Max(P(O_i))} \tag{4}$$

The Query Processor extracts core keywords and coordinates information from the user search query. In this way, a candidate group is selected based on the user's current location. The information of the selected candidate locations is calculated as a score in which public popularity is considered during the recent period T based on a search period. Equation (5) calculates popularity. Here, C is the number of total visitors of total candid$t(Q)$ate locations during the recent period T, while m is the number of visitors of the corresponding candidate location O_i. The popularity is calculated by assigning a higher weight to a more recent visit record $t_j(O_i)$ of location O_i from the query time. In addition, a higher weight is given to visit records that have times closer to the search time through the time information weight v according to the search time.

$$P(O_i) = \frac{1}{C}\sum_{j=1}^{m} v\left(\frac{1}{t(Q) - t_j(O_i) + 1}\right) \tag{5}$$

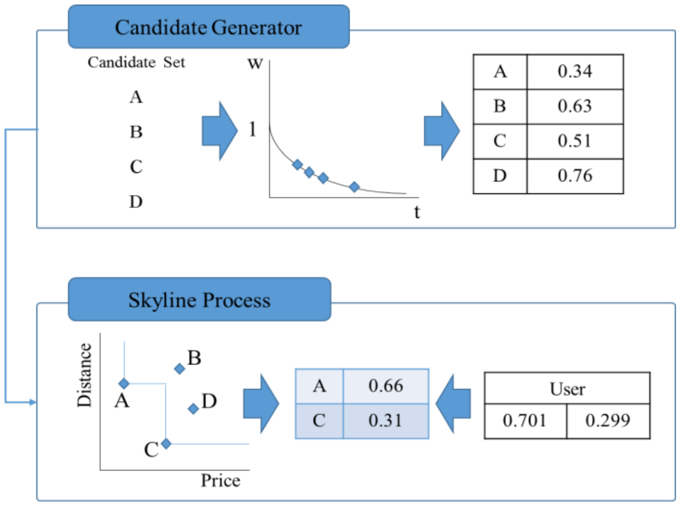
The Skyline Module selects objects that are not dependent on specific attribute values among search target objects that have multiple attributes. Through this process, unnecessary locations can be removed in advance, and priority is given by extracting only meaningful locations to users. Therefore, a location weight is calculated by considering user preference information with respect to objects selected through the skyline. Equation (6) calculates $L(O_i)$. Here, w_k is user preference information, while n_k is a normalized attribute value of locations collected through feedback.

$$L(O_i) = \sum_{k=1}^{n} w_k n_k \tag{6}$$

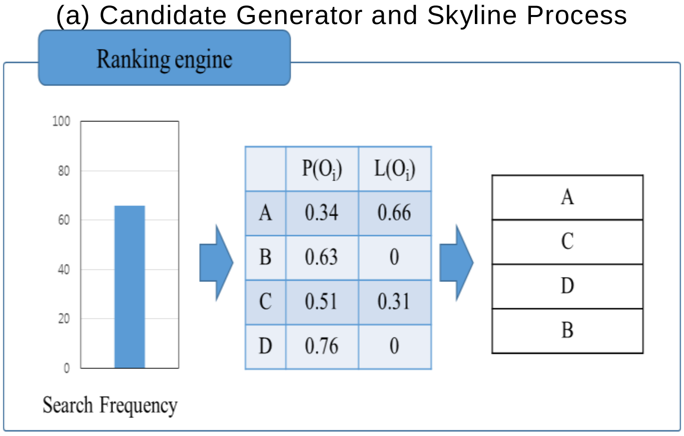
The Ranking Engine sums popularity scores and location weights, thereby assigning rankings and returning the results. Here, a ratio β of popularity score and location weight is reflected through the actual search frequency f_a divided by the search frequency threshold f_t, as shown in Equation (7). The constant δ that is applied to the calculation of β is a value calculated via performance evaluation in various environments. If user preference information exceeds the constant value δ of the equation, search results in which only user preference information is considered (*i.e.*, not general popularity) are provided. Therefore, to adjust this, the actual search frequency f_a should not be accumulated such that it exceeds the search frequency threshold f_t, and the applied proportion should be limited by the constant δ.

$$\beta = \frac{f_a}{f_t}\delta \tag{7}$$

Figure 6 shows a Top-3 processing procedure in the social search. Generally, search results provide all results related to a keyword. As shown in Figure 6(a), nearby information is searched based on the location of a user who requests a search from the Candidate Generator. Here, locations that have no visit records within a corresponding time slot are removed from the candidate list based on the time information transferred by a user. Based on the search keyword and time and location information, A, B, C, and D are extracted. Then, the weights and popularity of each location considering the search time are calculated. The more recent the visit, the higher the weight assigned to calculate public popularity. Similarly, the closer the time of the visit to the time requested by a user, the higher the weight assigned. The Skyline Module selects locations A and C that are not dependent on specific attributes among extracted locations and calculates a location weight by considering user preferences. In the Ranking Engine, popularity $P(O_i)$ and location weight $L(O_i)$ are summed to calculate a final score, as shown in Figure 6(b). The proportions of popularity and location weight vary depending on the user's search frequency. If the user's search frequency shows a 65% utilization rate during a unit period, the proportion of location weight exceeds that of public popularity. Based on this result, a final rank is given for each location, thereby providing A, C, and D to the user. If the user requests additional information using a scroll, locations with rankings lower than B are provided.



(a) Candidate Generator and Skyline Process



(B) Ranking Engine

Figure 6. Social Search Processing Procedure

4. Performance Evaluation

4.1. Experiment Environment

In order to show the superiority of the proposed method, we compare it with the existing method [1] through performance evaluation in the same experimental environment. In [1], the objective popularity and experts rating are used for a social search. However, user personal preferences are not considered during a social search. By the experimental results, we can see the objectivity of [1] and identify the advantage of the proposed method which uses user preferences by using the average value of each attribute of the results in [1]. We conducted our experiments on a desktop PC running on Windows 7 professional. The PC has an Intel core i5-3570K 3.20GHz CPU and 8GB memory. All of the experiments were coded in Java 7.0, and MySQL databases were used to store the experimental data. The characteristics of experimental check-in data are shown in Table 1, which is collected by using Gowalla APIs in Stanford University. The price information is not included in the original data set and the random price data is added. Gowalla is a mobile application used to support location based social services. In the experiment, the value of β in equation (3) is set to 0.7, which is selected by performing several experiments in the environment shown in Table 1.

Table 1. Gowalla Check-In Data

Attribute	Value
No. of total users	196,591
No. of total check-ins	6,442,890
Collection duration	Feb. 2009–Oct. 2010 (20 months)
Registered locations	24,533

The initial values used in the experiment are shown in Table 2. User preference is assigned by using price and distance. In order to show the flexibility of the proposed method, the experiments are performed under various types (type1 - type8) of user preferences. The search location is 10011 14th St New York in USA, and the search radius is within the New York City.

As shown in Table 2, the same user preference and various search frequencies are assigned to type 1, 3 and type 2, 4. Conversely, the same search frequency and various user preferences are used from type 5 to type 8. Neutral user preference and extreme user preference are assigned for type 5 and type 6, that is to say, users' preference focuses on a certain aspect, such as distance or price. For type 7 and type 8, the user preference is not slanted. According to the search location, 707 candidate locations are collected around it. In the experiment, the top-20 results are evaluated from the collected 707 candidate locations by performing the existing method and the proposed method.

Table 2. Performance Evaluation Setup Values

Attribute / Type	Price	Distance	Search frequency
Type 1	0.3	0.7	0.8
Type 2	0.7	0.3	0.8
Type 3	0.3	0.7	0.2
Type 4	0.7	0.3	0.2
Type 5	0	1	0.6
Type 6	1	0	0.6
Type 7	0.5	0.5	0.6

4.2. Evaluation Results

Figure 4 shows the top-20 results of a mobile social search. By comparing types 1, 3 and types 2, 4, we can see that the user preference is consistent with search frequency. A high search frequency implies that the user preference focuses on low prices or short distances. Conversely, a low search frequency implies high prices or long distances, which is consistent with general user patterns. Since neutral user preferences and extreme user preferences are assigned for type 5 and type 6, the differences of the price and distance of search results are distinct. Conversely, since the user preference of type 7 and type 8 is not slanted, the differences of the price and distance of search results are negligible. Since the existing method did not consider user preferences, the price and distance of search results are high.

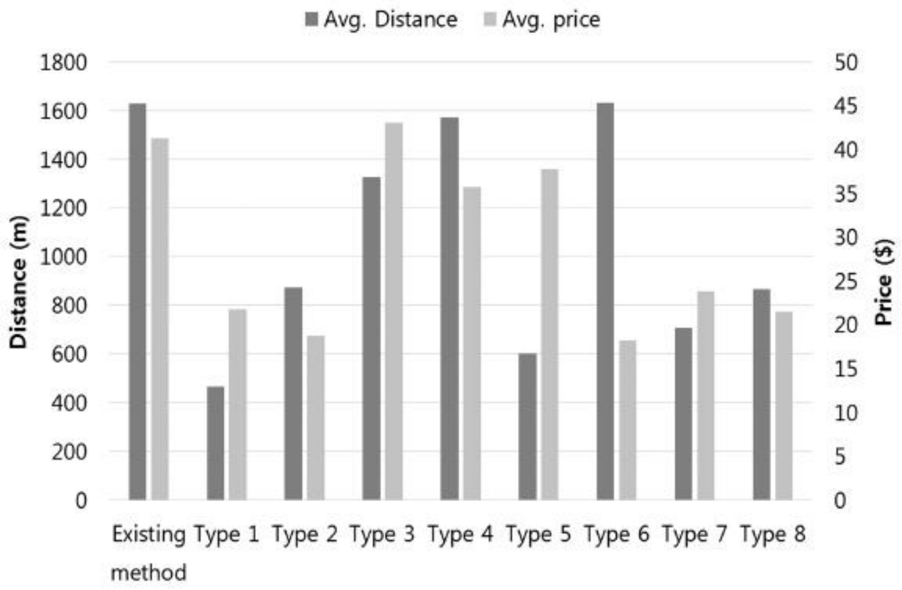


Figure 7. Average Ranking According to Various Types

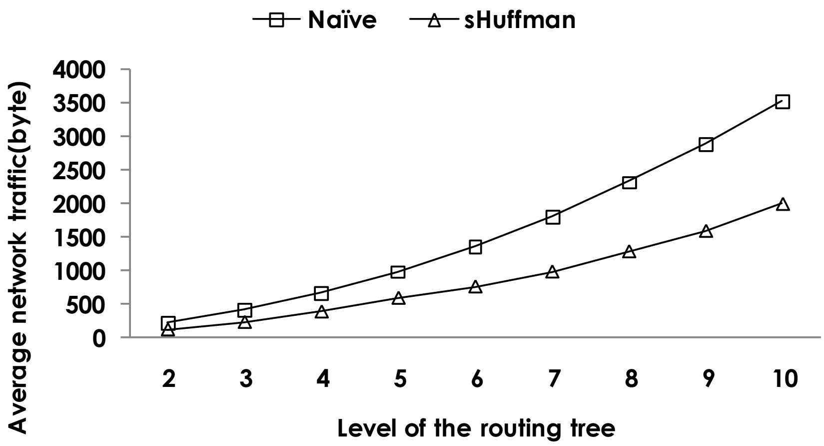


Figure 8. Average Network Traffic

4.3. Network Lifetime

Figure 9 shows the network life with or without conventional compression algorithms. FM (Flooding Method) is an aggregation algorithm based on TAG. To evaluate - the existing compression algorithms, [5] is applied. In the result, the network lifetime of our proposed algorithm is prolonged by about 30%. In case of our proposed algorithm with the existing compression algorithm, it shows additional energy-efficiency.

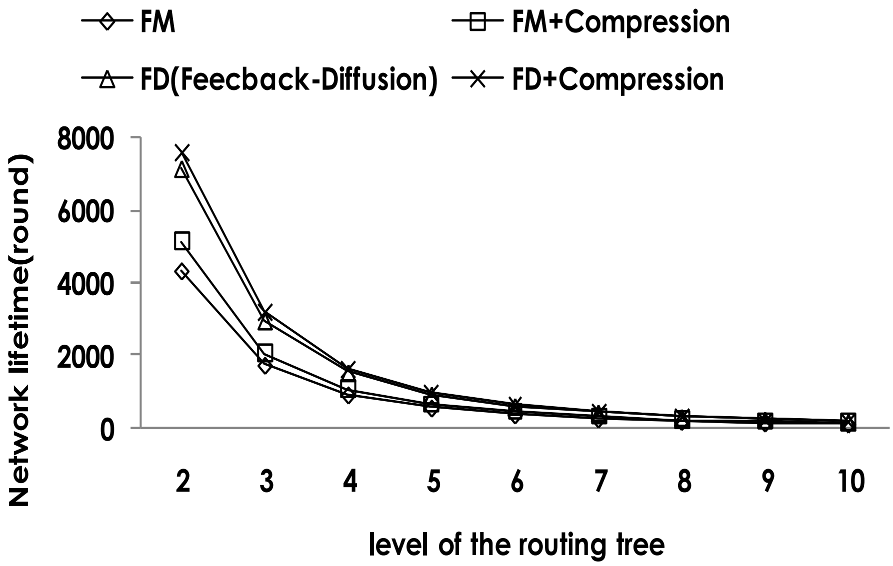


Figure 9. Network Lifetime

5. Conclusions

In this paper, we have proposed a feedback diffusion algorithm based on the variant of Huffman coding, called sHuffman. While existing algorithms just exploited local distribution, our proposed scheme compresses sensor readings efficiently from global distribution. In order to show the superiority of our approach, we compared it with the existing aggregation algorithms in terms of the lifetime of the sensor network. As a result, our experimental results have shown that the whole network lifetime was prolonged by about 30% and we can improve energy efficiency by utilizing existing compression algorithms in parallel. Also, we confirmed that it is possible to efficiently collect data by utilizing an error of the sensor network and an error bound of the sensor data. In the future, we will apply our algorithm to the real sensor network applications.

Acknowledgments

This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2015-H8501-15-1013) supervised by the IITP(Institute for Information & communication Technology Promotion), by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(No. 2014R1A1A2055778) and results of a study on the “Leades INdustry-university Cooperation” Project, supported by the Ministry of Education, Science & Technology (MEST).

References

[1] D. Estrin, L. Girod, G. Pottie and M. Srivastava, "Instrumenting the World with Wireless Sensor Networks", Proceedings of International Conference Acoustics, Speech, and Signal Processing, **(2001)**.

[2] G. J. Pottie and W. J. Kaiser, "Wireless Integrated Network Sensors", Proceedings of Comm. ACM, **(2000)**, pp. 51-58.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramanism and E. Cayirci, "A Survey on Sensor Networks", Proceedings of IEEE Communications Magazine, **(2002)**.

[4] M. H. Yeo, M. S. Lee, S. J. Lee and J. S. Yoo, "Data Correlation-Based Clustering Algorithm in Wireless Sensor Networks", Proceedings of The 2008 International Symposium on Computer Science and its Applications, **(2008)**.

[5] M. Sharaf, J. Beaver, A. Labrinidis and P. Chryanthis, "Tina: A scheme for temporal coherency-aware in-network aggregation", Proceedings of the 2003 ACM Workshop on Data Engineering for Wireless and mobile Access, **(2003)**.

[6] X. Meng, L. Li, T. Nandagopal and S. Lu, "Event contour: An efficient and robust mechanism for tasks in sensor networks", Proceedings of Technical report, **(2004)**.

[7] S. Pattem, B. Krishnamachari and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks", Proceedings of International Conference on Information Processing in Sensor Net-works, **(2004)**.

[8] D. Petrovic, R. Shah, K. Ramchandran and J. Rabaey, "Data funneling: Routing with aggregation and compression for wireless sensor networks", Proceedings of the 2003 IEEE Sensor Network Protocols and Applications, **(2003)**.

[9] "Live from Earth and Mars (LEM) Project", http://www-k12.atmos.washington.edu/k12/grayskies/, **(2006)**.

[10] W. Heinzelman, "Application-Specific Protocol Architectures for Wireless Networks", PhD dissertation, Massachusetts Inst. Of Technology, **(2000)**.

Authors

Byoungyup Lee, he is a professor in the department of electronic commerce at Paichai University, Daejeon, South Korea. He received the M.S. and the PhD degrees in Computer Science from Korea Advanced Institute of Science and Technology, Daejeon, South Korea. His research interests are XML, the database system, intelligence information system, and the electronic commerce.

Minje Ahn, he is a student on department of Information and Communication Engineering from Chungbuk National University, Cheongju, South Korea. He received the M.S. degree in the Department of Information and Communication Engineering from Chungbuk National University, Cheongju, South Korea. His research interests are the database system and the mobile social search.

Jongtae Lim, he is a PhD researcher on department of Information and Communication Engineering from Chungbuk National University, Cheongju, South Korea. He reveived the M.S. and the Ph.D degree in the Department of Information and Communication Engineering from Chungbuk National University, Cheongju, South Korea. His research interests are Query Processing, Moving Object Database, Spatial Database, Location-based Services (LBS), P2P Networks, Big Data.

Kyoungsoo Bok, he is an associate professor of department of Information and Communication Engineering at Chungbuk National University, Korea. He received the M.S. and the Ph.D degree in Department of Information and Communication Engineering at Chungbuk National University, Korea. His main research interests are Database Systems, Data Storage Systems, Moving Object Database, RFID, Sensor Networks, Mobile P2P Networks, Social Network Services, Big Data.

Jaesoo Yoo, he is a professor in the department of Information and Communication Engineering at Chungbuk National University, Cheongju, South Korea. He reveived the B.S. degree in Computer Engineering from Chunbuk National University, Chunju, South Korea. And he received the M.S. and the Ph.D degrees in Computer Science from Korea Advanced Institute of Science and Technology, Daejeon, South Korea. His research interests are the database system and the multimedia database.

