Continuous Reverse Skyline Query Processing  
over Moving Objects in LBS

Jongtae Lim¹, Yonghun Park¹, Kyoungsoo Bok¹, Jaesoo Yoo¹

¹ Department of Information and Communication Engineering, Chungbuk National University, Korea  
{jtlim, yhpark1119, ksbok, yjs}@chungbuk.ac.kr

Abstract. As the variant of a skyline query, a reverse skyline query has been studied. However, the existing methods for processing a reverse skyline query have the limitation of service domains and require the high costs of computation to provide various location-based services. In this paper, we propose a new reverse skyline query processing method that efficiently processes a query over the moving objects. In addition, the proposed method processes a continuous reverse skyline query efficiently. In order to show the superiority of the proposed method, we compare it with the previous reverse skyline query processing method in various environments. As a result, the proposed method achieves better performance than the existing method.

Keywords: Continuous query, Reverse skyline, Location-based services

1 Introduction

As the variant of skyline [1] query processing, reverse skyline has been studied [2]. The reverse skyline query is a query type returning a set of objects that includes the query object as the result of the skyline query. In recent, various reverse skyline query processing methods have been studied. But, they are based on dynamic skyline [3]. So, we proposed an efficient method for processing reverse skyline queries [3]. This method processes a reverse skyline query using the spatial grid indexing method. This method prunes the objects that do not affect a query using skyline dominant relationship during the reverse skyline query processing. However, this method needs to process the final verification. Because the method proposed in [3] implements the final verification, the method proposed in [3] access more cells in grid index to process the reverse skyline. So, we propose a new reverse skyline query processing method that efficiently processes a query over the moving objects. In addition, the proposed method also processes continuous reverse skyline queries efficiently. The proposed method makes the verification range to grantee the result of reverse skyline query. So, proposed method doesn’t need to implement final verification when the new objects appear or the moving objects are move. Because the proposed method does not need to implement the final verification, the proposed method is an efficient method to process the continuous reverse skyline.
2. The Proposed Query Processing Method

Figure 1 shows the initial step of proposed reverse skyline query processing method. First, the proposed method searches for the same type of object dominating \( q \) without the distance attribute in the order of their proximity to \( q \). Second, if an object dominating \( q \) is found, we draw the bisector between the object and \( q \) such as figure 1(a). All objects located outside the bisector do not contain \( q \) as the skyline result. It is because the objects are always closer to \( o_2 \) than \( q \) and even \( o_2 \) dominates \( q \) for static attributes. \( q \) is dominated by \( o_2 \) on the skyline of both \( c_1 \) and \( c_2 \). If there is the intersection point during processing the reverse skyline query, the proposed method makes the verification range such as figure 1(b). Through the verification range, the proposed method can prune the objects that do not affect a query. These processes are repeated until all objects inside the verification range are checked such as figure 1(c). As a result, \( c_3 \) is result of the reverse skyline query such as figure 1(d).

![Fig. 1. The initial step of the proposed reverse skyline query processing method](image)

When the object \( o \) moves to current location from previous location, the proposed method loads all query list of cell including the previous location and the current location. Next, the proposed method checks the condition of \( q \) included \( RESULT_SET \) at the previous and current location. And then the proposed method checks the condition of \( o \) included reverse skyline range of queries. And the proposed method computes the condition of \( o \) included reverse skyline range of queries using coordinates. Finally, the proposed method compares the condition of \( q \) included \( RESULT_SET \) at the previous location and the condition of \( q \) included \( RESULT_SET \) at the current location. If the condition of \( q \) included \( RESULT_SET \) at the previous location is not same against the condition of \( q \) included \( RESULT_SET \) at the current location, the proposed method updates the result of reverse skyline.
3. Performance Evaluation

We experimentally evaluate the efficiency of the proposed method for the continuous reverse skyline computation. Figure 2 shows the experimental results of continuous reverse skyline by varying the number of moving object from 100 to 1,000. Because the existing method implements the final verification when the new objects appear or the moving objects move, the existing method access more cells to process the continuous reverse skyline query. Therefore, the proposed method achieves about 500% better performance than the existing method to process the continuous reverse skyline when the number of moving object is 1,000.

![Performance comparison according to the number of moving objects](image)

4. Conclusion

In this paper, we proposed a new reverse skyline query processing method that efficiently processes a query over the moving objects. In order to show the superiority of the proposed method, we compared it with the previous reverse skyline query processing method in various performance evaluation environments. As a result, the proposed method achieved better performance than the existing method.

Acknowledgments. This work was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST)(No. 2009-0089128) and by the Korea Institute of Science and Technology Information (K-12-L06-C02-S03)

Reference