

# Theoretical Frameworks for Improving the Computation of Skylines over Uncertain Data

Ma'aruf Mohammed Lawal<sup>1,2\*</sup>, Hamidah Ibrahim<sup>2</sup>, Nor Fazlida Mohd Sani<sup>3</sup>, and Razali Yaakob<sup>4</sup>

<sup>1</sup>Department of Computer Science, Faculty of Physical Sciences Ahmadu Bello University, Zaria.

<sup>2,3,4</sup>Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia  
mmlawal@abu.edu.ng<sup>1\*</sup>, hamidah.ibrahim@upm.edu.my<sup>2</sup>, fazlida@upm.edu.my<sup>3</sup>, razaliy@upm.edu.my<sup>4</sup>

\*Corresponding Author

## Abstract

*Skyline query is preeminent for realizing solutions that support multi-criteria decision making, particularly for modern applications that often capture or generate large volume of high dimensional database. The essence of skyline query is to return a set of objects that is not dominated by any other objects. In this paper, we proposed frameworks namely; SQUiD, CSQUiD, and ASQUiD that deal with the computational problem associated with processing constrained skyline queries over uncertain data with high dimensionality. Although, solutions for processing such queries over uncertain data are available, but they are inefficient as they do not utilize the result of earlier computations. Since multiple constrained skyline queries might span over the same part of the database, this research work attempt to avoid unnecessary skyline computations of these queries by identifying the conditions specified in these queries.*

**Keywords:** *constrained skyline query, uncertain data, Indexing technique.*

## 1. Introduction

The essence of skyline query processing is to return a set of objects that is not dominated by any other objects. Based on user's preference, skyline analysis has become a way of realizing non-trivial and interesting solutions from the underlying database. While conducting literature review on preference evaluation techniques, a number of works have applied the concept of skyline evaluation technique to provide answers to a user's query. These include Branch and Bound Skyline (BBS) (Papadias et al., 2005), Divide-and-Conquer (DC) (Borzsonyi et al., 2001), Block Nested Loop (BNL) (Papadias et al., 2005), Bitmap and Index (Tan et al., 2001), and Nearest Neighbor (NN) (Kossmann et al., 2002). Common to all of these proposed algorithms, is the attempt to indirectly reduce the computational cost of computing the skylines, through the reduction in the number of pairwise comparisons among objects of the underlying database.

Several research works on skyline query processing exist, as most of the proposed algorithms for computing skylines assumed that the values along a dimension of the database are all exact values (Cui et al., (2008), Dellis & Seeger (2007), Gothwal et al., (2018), Gulzar et al., (2017), Lee et al., (2010), Li et al., (2012), Liu et al., (2013), Papadias et al., (2005), Pei et al., (2007), Saad et al., (2014)). However, this assumption is not always true in the real-world. Particularly, with modern database applications that capture uncertain data with high dimensionality.

Uncertain database in our work is referred to as a collection of objects with continuous range values, whose exact values are not known at the point of computing the skylines of a given skyline query. With uncertain database, performing dominance tests among objects without calculating the skyline probabilities is almost impossible. The conventional dominance test which derives skylines by directly comparing the values of objects to identify dominant objects no longer work in this setting, as continuous range value is neither comparable to an exact nor other continuous range values. Similarly, it becomes more complicated when a constrained skyline query is to be processed over uncertain database. Unlike conventional skyline queries, a constrained skyline query returns a set of skylines that strictly satisfies the constraints on specific dimensions that are attached to the query, that is, a constraint on a

dimension is a range specifying the user's interest. Identifying whether an object satisfies these constraints is straightforward if the values of the object are all exact values; thus, those objects whose dimension values are within the range specified in the constraints are selected for skyline computation. However, objects whose dimension values are in a continuous range form that intersect or overlap with the range specified in the constraint needs further examination before the objects can be considered for skyline computation. Although there are works on skyline queries as well as constrained skyline queries over uncertain database, these works which employ a probability-based approach require additional computational time, as they rely heavily on probability calculations. Moreover, processing every single constrained skyline query independently might incur unnecessary repetition of skyline computations as these constrained skyline queries might have similar constraints specified in the queries. Thus, avoiding these unnecessary skyline computations of multiple constrained skyline queries is crucial. The main contributions of this research work are presented as follows:

- i We proposed the Skyline Query on Uncertain Database (*SQUiD*) framework which is the baseline framework of the subsequent frameworks with the main aim to efficient process skyline query over uncertain database without employing either the arithmetic mean calculation or the probability methods.
- ii The second framework that we have proposed named Constrained Skyline Query on Uncertain Database (*CSQUiD*) aim at computing constrained skyline queries over uncertain database with high dimensionality.
- iii Finally, Abating the computation of constrained Skyline Query on Uncertain Database (*ASQUiD*) is introduced mainly to streamline the process of processing multiple constrained skyline queries over uncertain database

The rest of this paper is organized as follows: Related works are presented in Section 2, while the proposed theoretical frameworks are presented in Section 3, and Section 4 concludes this research work.

## 2. Related Works

The introduction of a number of preference evaluation techniques is borne out of the quest to derive solutions for multi-criteria problems, as traditional query is too rigid as it does not give users the privilege to specify their interests. Over the years, considerable attention in databases focused on preference-based queries (Afshani et al., (2011), Aggarwal & Philips, (2011) Alwan et al., (2014), Arslan et al., (2010), Atallah & Qi (2009), Bartolini et al., (2008), Borzsonyi et al., (2001), Chomicki et al., (2005), Gulzar et al., (2019), Han et al., (2013), Jiang et al., (2012), Khalefa et al., (2010), Kossmann et al., (2002), Lee et al., (2010), Li et al., (2012), Liu et al., (2013), Papadias et al., (2005), Pei et al., (2007), Saad et al., (2014), Saad et al., (2016), Saad et al., (2018), Saad et al., (2019), Tan et al., (2001)). This is due to the effectiveness in returning interesting results that provides optimum solution. With databases emerging from processing environments, an extension of the standard Structured Query Language (SQL) with preference evaluation technique is required in order to provide answers to personalized queries. Among the preference evaluation techniques are the skyline, top-k, top-k dominating, k-dominance, and the k-frequency. Each of these techniques is unique in the way preference evaluation has been performed (Atallah & Qi (2009), Bartolini et al., (2008), Borzsonyi et al., (2001), Chomicki et al., (2005), Han et al., (2013), Jiang et al., (2012), Kossmann et al., (2002), Papadias et al., (2005), Saad et al., (2018), Saad et al., (2019), Tan et al., (2001)). These preference evaluation techniques are discussed in detail with more emphasis on skyline query in the work of (Lawal et al., (2018), Lawal et al., (2017)). The most popular among the preference evaluation techniques used in a database are the top-k and skyline queries.

Inspired by the benefits that can be achieved if applications for retrieving non-trivial information are developed, several solutions on skyline computation have been proposed (Afshani et al., (2011), Aggarwal & Philips, (2011), Alwan et al., (2014), Arslan et al., (2010), Atallah & Qi (2009), Bartolini et al., (2008), Borzsonyi et al., (2001), Chomicki et al., (2005), Gothwal et al., (2018), Han et al., (2013), Kossmann et al., (2002), Lee et al., (2010), Pei et al., 2007, Tan et al., (2001)). Quite a number of works adopted the probabilistic approach in processing skyline queries in finding the dominant objects (Jiang et al., (2012), Pei et al., (2007), Saad et al., (2014), Saad et al., (2016), Saad et al., (2018), Saad et al., (2019)). In respective of the underlying database to be processed, basically, there are

two main approaches employed by existing works for processing skyline queries over high dimensional databases, namely: the index and non-index-based approaches.

The index-based approach requires a progressive preconstructed data structure that allows and supports the computation of skylines in an efficient manner. The index-based techniques employ the indexing techniques to organize data into Minimum Bounded Rectangles (MBRs) in order to streamline the process of computing skylines (Li et al., (2012), Liu et al., (2013), Papadias et al., (2005)). Index based techniques have been seen to be useful in query optimization, by playing a major role in reducing the computational time required for processing a query. The preconstructed data structure is meant to significantly speed up skyline computation by pruning off objects which do not satisfy the query when computing the skylines (Lawal et al., 2020). Most of the works that adopt the index-based approach typically uses the *R*-based trees by (Guttman, 1984), spatial indexing techniques for supporting the process of computing skylines over a database (Lee et al., (2010), Li et al., (2012), Liu et al., (2013), Papadias et al., (2005)). The non-index-based approach requires preprocessing operations such as sorting and grouping. Unlike the index-based approach which required using an indexing technique to organize a data, with non-index approach, the operations to be executed are performed directly on the database (Borzsonyi et al., 2001).

Among the proposed solutions for processing skyline queries over uncertain data, are that of (Khalefa et al., (2010), Lee et al., (2010), Saad et al., (2014), Saad et al., (2016), Saad et al., (2018), Saad et al., (2019)) which employ a non-index-based approach that was inspired by the work of (Pei et al., 2007). In their proposed solution, a probability calculation method was utilized just like in the works of (Jiang et al., 2012) and (Pei et al., 2007). Even though the works of (Khalefa et al., (2010), Saad et al., (2014), Saad et al., (2016), Saad et al., (2018), Saad et al., (2019)), employ a method that captures the uncertainty associated with continuous range values, the solution seems inefficient especially when processing skyline query over uncertain data with high dimensionality as many probability calculations need to be performed before the skylines are derived. Meanwhile, (Li et al., 2011) employs the median value calculations method to compute the exact value of each continuous range value. However, the method is not realistic as there is no certainty that the exact value for a continuous range value is most likely to be at the central point of the continuous range value. Therefore, it has become imperative to support the design and development of an efficient skyline computation framework for processing skyline query over uncertain data by avoiding extensive probability calculations and not relying on the assumption that the exact value of each continuous range value is at the central point of the continuous range value.

### 3. Proposed Theoretical Frameworks

In dealing with the computational problem associated with processing skyline queries over uncertain database, in this work we proposed three distinct nonprobability index-based frameworks, namely: Skyline Query Processing on Uncertain Database (*SQUiD*), Constrained Skyline Query Processing on Uncertain Database (*CSQUiD*), and Abating Skyline Query Recomputation over Uncertain Database (*ASQUiD*). Common to these indexed based frameworks is the use of *X*-tree structure in organizing the underlying data set. The choice of the *X*-tree is justified based on the analysis conducted in Lawal et al., (2020)

#### 3.1 The *SQUiD* framework

This framework consists of two main modules, namely: Data Pre-processing module and Real-time module as presented in Figure 1. Essentially, the *SQUiD* framework is meant for reducing the number of pairwise comparisons that needs to be performed among objects and for minimizing the searching space which indirectly relate to an improved data access performance. The Data Pre-processing module aims at streamlining the process of skyline computation by employing the *X*-tree indexing technique to organize the uncertain database into MBRs in such a way that objects that are being dominated by other objects within the same MBR can be filtered. Nevertheless, our choice of *X*-tree indexing technique is based on in-depth performance analyses conducted among the existing *R*-based indexing techniques, namely: *R*-tree, *R*\*-tree, and *X*-tree. Meanwhile, the Real-time module aims at transforming the uncertain database (continuous range value) into certain data (exact value) so that the conventional dominance test can be directly employed.

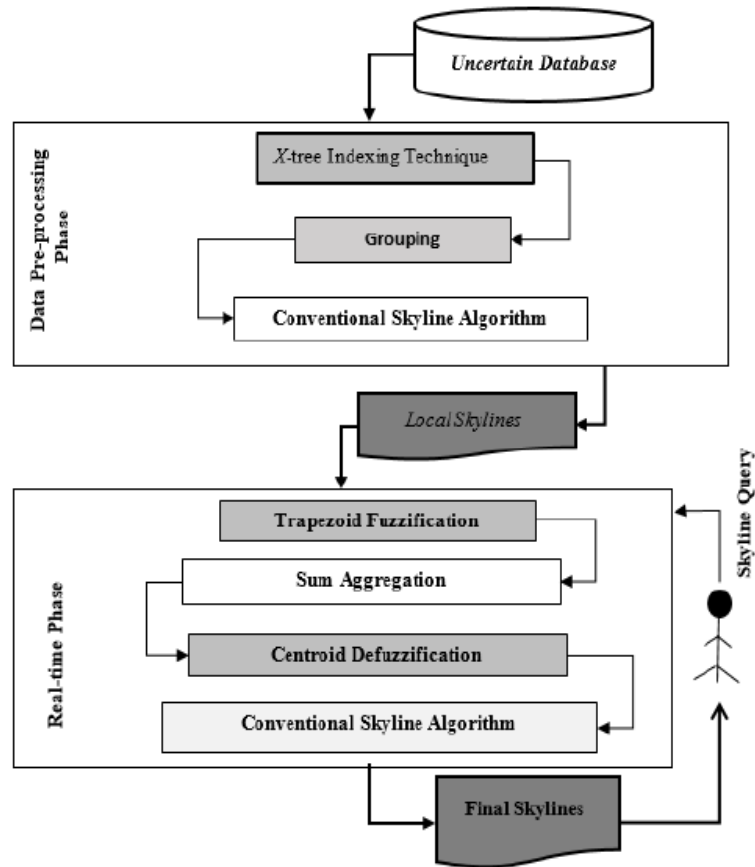


Figure 1: The Proposed *SQUiD* Framework for Computing Skylines over Uncertain Data

To realize *SQUiD* framework, we have identified the following procedures: Grouping method that groups the objects within each MBR based on the values of each dimension; dominance relation that performs the conventional dominance test among objects with comparable values to realize the candidate skylines of each MBR; kNN algorithm that identifies the *k* nearest objects to an object – these objects have similar features that can be used to predict the continuous range values of an object, and fuzzy algorithm that consists of *trapezoid fuzzification*, *sum aggregation*, and *centroid Defuzzification* methods to predict the exact value of a continuous range value based on the features of *k* nearest objects. Eventually, as the values of these objects are now comparable the conventional dominance test can be employed to compute the final skylines.

**Table 1.** An Example for Illustrating the Proposed *SQUiD* Framework for Computing Skylines over Uncertain Data

Algorithm/ Method	Operation	Output
X-Tree Indexing Technique	Use the indexing Algorithm to organize objects {A, B, C, D, E, F, G, H}	R{5} = {D, C, H} R {6} = {B, I} R3 = {G} R4 + {E, A}
Grouping	Grouping of objects into two distinct groups $O_c$ and CS	CS = {R3 = {F}, R4 = {A}, R5 = {C}, R6 = {}} $O_c = \{R3 = \{G, H\}, R4 E, R5 = \{D, h\}, R6 \{B, I\}\}$
Conventional Skyline Algorithm	To return local skyline for each MBR $\{O_c\}$	R3 = {F, G}, R4 = {A, E}, R5 = {C, D, H}, R6 = {B}

Trapezoid Fuzzification	Return fuzzify membership function values for each range value	A = 0.48043478 & 0.48152174 F = 0.49913797 & 0.50086207
Conventional Skyline Algorithm	Apply the skyline Algorithm on those objects derived	B & C

The *SQUiD* framework is a two-phase framework. The first phase named the Data Pre-processing phase consists of two methods and utilizes the conventional skyline algorithm to derive the processed MBRs. These methods are (i) *X-tree* indexing technique which is employed for organizing both certain and uncertain data into MBRs, (ii) *Grouping* method that is introduced for eliminating unnecessary pairwise comparisons between object having exact value and object represented with a continuous range value, by grouping objects of each MBR is split into two distinct groups, with one group containing objects with certain data, while the other group containing objects with uncertain data, and (iii) conventional skyline algorithm which is utilized for computing the local skylines over the group with certain data. The Real-time phase incorporates the conventional skyline algorithm and three methods, namely: *Trapezoid Fuzzification*, *Sum Aggregation*, and *Centroid Defuzzification* for predicting the exact value of a continuous range value in order to seamlessly employ the conventional skyline algorithm to compute the final skylines.

### 3.2 The CSQUiD framework

The Proposed framework for processing constrained skyline queries over uncertain data (*CSQUiD*) is presented in Figure 2. Particularly, an illustration for components of the proposed *CSQUiD* framework is typified in Table 2.

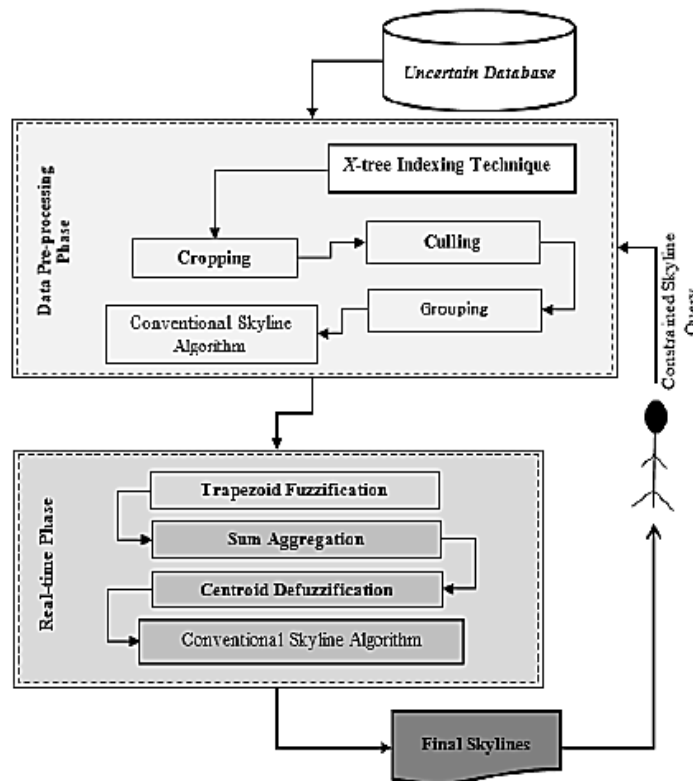


Figure 2: The Proposed *CSQUiD* Framework for Processing Constrained Skyline Query over Uncertain Data

This has become imperative, as identifying whether an object satisfies a constrained skyline query is straightforward when dealing with certain data. Thus, objects whose exact values are within a constrained skyline

query are selected for skyline computation. Likewise, objects with uncertain data need additional evaluation before computing the final skylines to the constrained skyline query. Invariably, this compound and increases the process of evaluating constrained skyline query over uncertain data. Thus, minimizing the computational cost is achieved by pruning off MBRs whose vertex are dominated by the vertex of other MBR. Instead of evaluating objects of each MBR, only objects within the MBR with the vertex are processed for the final skylines. Consequently, the number of pairwise comparisons among objects is significantly reduced.

Primarily, the *CSQUID* framework is proposed for streamlining the process of computing constrained skyline query over uncertain data by evaluating only non-dominated objects within MBR, whose vertex lies within the constrained query. This was achieved by leveraging on an indexing structure to minimize computational cost by pruning off dominated objects within each MBR in order to reduce the number of pairwise comparisons among objects. This framework consists of two phases, namely: Data Pre-processing and Real-time.

The Data Pre-processing phase employs an indexing technique to organize and store uncertain data into MBRs, while MBRs that either overlaps or intersect, or lies within a given constrained query are selected using the *Cropping* method. The *Culling* method is utilized for identifying the left coordinates and leftmost coordinate in identifying the non-dominated MBRs whose objects will eventually be the local skylines. Meanwhile, the Grouping method group the objects of the selected MBRs into two distinct groups, and the conventional skyline algorithm computes the local skylines of the selected MBRs. At the Real-time phase, the *Trapezoid Fuzzification*, *Sum Aggregation*, *Centroid defuzzification* methods with conventional skyline algorithm are used to compute the final skylines.

**Table 2.** An Example for Illustrating the Proposed *CSQUID* Framework for Processing Constrained Skyline Queries over Uncertain Data

Algorithm/Method	Operation	Output
X-Tree Indexing Technique	Use the indexing Algorithm to organize objects {A, B, C, D, E, F, G, H}	R3 = {G} R4 = {E, A} R5 = {D, C, H} R6 = {B, I}
Atomization	During atomization of search space, SS each unit space in MBR acquire unit SRD	A = (10 - 12, 8), C = (1 - 5, 6), E = (8, 8), G = (15, 4) H = (12, 1) D = (4, 20) F = (12 - 20, 5) H = (5, 6)
Grouping	Grouping of objects into two distinct groups $O_c$ and CS	CS = {A, C, F} $O_c$ = {B, D, G, H, I}
Centroid Defuzzification	The exact values for each single trapezoid membership function value are computed	C = 73. F = 141.03
Convention Skyline Algorithm		B, C, E, G
X-Tree Indexing Technique	Use the indexing Algorithm to organize objects {A, B, C, E, G}	R3 = {G}, R4 = {E} R5 = {C}, R6 = {B}
Cleaning	Given Constrained query [95 - 130]	{B, E}
Merging	This method is meant to identify and get all objects with a unit space whose SDR lies within the constrained query	E has a SDR (100, 105)(49 - 53)
Conventional Skyline Algorithm		E is returned as the final skyline for [95 130]

### 3.3 The ASQUID framework

The ASQUID framework consists of two phases, namely: Data Pre-processing and Real-time is presented in Figure 3. An example is presented in Table 3 to illustrate the proposed framework.

An index-based materialized view framework to avoid the unnecessary skyline computations of multiple constrained skyline queries. Intuitively, the information of the MBRs as well as their candidate skylines that have been identified in the computations of earlier constrained skyline queries are saved to be utilized in processing subsequent constrained skyline queries which constraints lie within, overlap, or intersect with the space of the saved MBRs. At the Data Preprocessing phase, the whole search space that represents the data is partitioned into a unit space, while the local skylines for each MBR is derived. Subsequently, the derived local skylines are organized into MBRs in order to seamlessly compute the final skylines to a constrained skyline query at the Real-time phase. The Real-time phase consists of the *Merging* and *Glean* methods for identifying the candidate skylines, while the skyline algorithm is employed to compute the final skylines over the candidate skylines derived.

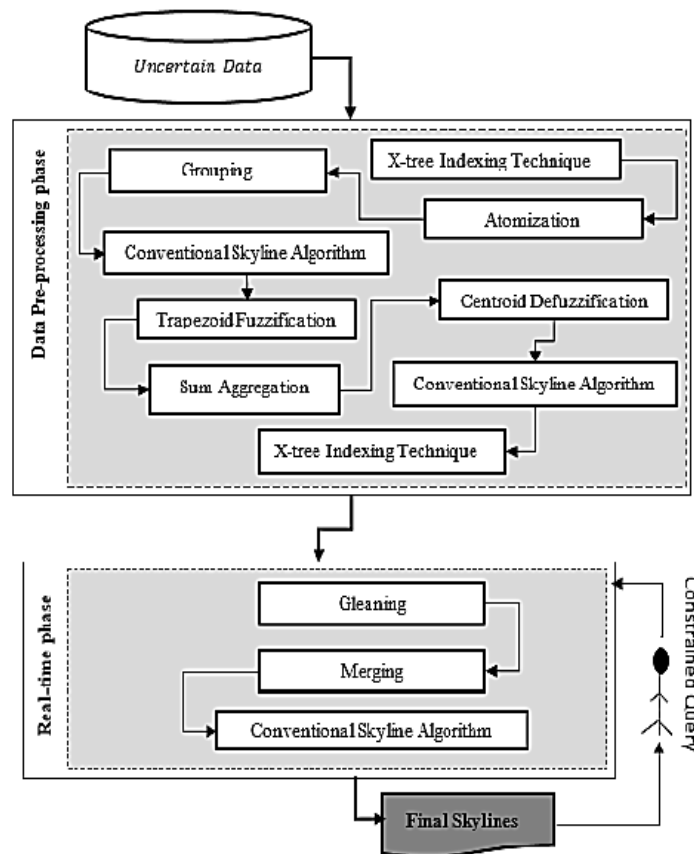


Figure 3: The Proposed ASQUID Framework for Processing Constrained Skyline Query over Uncertain Data

Basically, this is meant to speed up the Central Processing Unit (CPU) processing time in processing a constrained skyline query as it eliminates the need to always process skylines over similar set of objects each time a constrained skyline query is to be evaluated. Computing constrained skyline queries without utilizing the result of the previous computations of skyline queries is unwise as some of the queries might cover the same part of the database which can be easily identified based on the conditions specified in the queries.

**Table 3.** An Example for Illustrating the Proposed *ASQUiD* Framework for Abating Recomputation of constrained Skyline Queries over Uncertain Data

Algorithm/Method	Operation	Output
X-Tree Indexing Technique	Use the indexing Algorithm to organize objects {A, B, C, D, E, F, G & H}	R3 = {G} R6 = {B, I} R5 = {D, C, H} R4 = {E, A}
Atomization	During atomization of search space, SS each unit space in MBR acquire unit SRD	A = (10 - 12, 8) C = (1 - 5, 6), E = (8, 8) G = (15, 4) A = (12, 1) D = (4, 20) F = (12 - 20, 5) H = (5, 6)
Grouping	Grouping of objects into two distinct groups Oc and CS	CS = {A, C, F} Oc = {B, D, G, H, I}
Centroid Defuzzification	The exact values for each single trapezoid membership function value is computed	C = 73. F = 141.03
Conventional Skyline Algorithm		B, C, E, G
X-Tree Indexing Technique	Use the indexing Algorithm to organize objects {A, B, C, E, G}	R {3} = {G} R {4} = {E} R {5} = {C} R {6} = {B}
Cleaning	Given Constrained query [95 - 130]	B, E
Merging	This method is meant to identify and get all objects with a unit space whose SDR lies within the constrained query	E has a SDR (100, 105)(49 - 53)
Conventional Skyline Algorithm		E is returned as the final skyline for [95 130]

#### 4. Conclusion

Today, we are in the era of making multi-criteria decisions based on analysis of available data collected from autonomous databases that usually contain uncertain data. Without doubt, streamlining the process of processing skylines in providing answers to user-specified queries is inevitable. In our work, we proposed *SQUiD*, *CSQUiD* and *ASQUiD* frameworks that combines an index-based technique with a prediction method to reduce the computational time for computing skylines and processing skylines over uncertain data with high dimensionality.

#### Funding

This work was supported by the Ministry of Higher Education Malaysia under the Fundamental Research Grant Scheme (FRGS/1/2020/ICT03/UPM/01/1) and the Universiti Putra Malaysia. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies. We thank the anonymous reviewers for their comments.

#### Acknowledgment

The authors would like to thank the Ministry of Higher Education Malaysia and the Universiti Putra Malaysia for funding this research work.



## References

- Afshani, P., Agarwal, P. K., Arge, L., Larsen, K. G., and Phillips, J. M. (2011). (Approximate) uncertain skylines. In *Proceedings of the 14th International Conference on Database Theory*, pp. 186–196. ACM.
- Aggarwal, C. C. and Philip, S. Y. (2009). A survey of uncertain data algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering*, 21(5):609–623.
- Alwan, A. A., Ibrahim, H., and Udzir, N. I. (2014). A framework for identifying skylines over incomplete data. In *2014 3rd International Conference on Advanced Computer Science Applications and Technologies*, pp. 79–84. IEEE.
- Arslan, S., Saçan, A., Acar, E., Toroslu, I. H., and Yazici, A. (2010). Comparison of multidimensional data access methods for feature-based image retrieval. In *2010 20th International Conference on Pattern Recognition*, pp. 3260–3263. IEEE.
- Atallah, M. J. and Qi, Y. (2009). Computing all skyline probabilities for uncertain data. In *Proceedings of the twenty-eighth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 279–287. ACM.
- Bartolini, I., Ciaccia, P., and Patella, M. (2008). Efficient sort-based skyline evaluation. *ACM Transactions on Database Systems (TODS)*, 33(4):31.
- Borzsonyi, S., Kossmann, D., and Stocker, K. (2001). The skyline operator. In *Data Engineering, 2001. Proceedings. 17th International Conference on*, pp. 421–430. IEEE.
- Chomicki, J., Godfrey, P., Gryz, J., and Liang, D. (2005). Skyline with presorting: Theory and optimizations. In *Intelligent Information Processing and Web Mining*, pp. 595–604. Springer.
- Cui, B., Lu, H., Xu, Q., Chen, L., Dai, Y., and Zhou, Y. (2008). Parallel distributed processing of constrained skyline queries by filtering. In *2008 IEEE 24th International Conference on Data Engineering*, pp. 546–555. IEEE.
- Dellis, E. and Seeger, B. (2007). Efficient computation of reverse skyline queries. In *Proceedings of the 33rd International Conference on Very large data bases*, pp. 291–302. VLDB Endowment.
- Gothwal, H., Choudhary, J., and Singh, D. (2018). The survey on skyline query processing for data-specific applications. In *Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIOTCT)*, pp. 26 – 27.
- Gulzar, Y., Alwan, A. A., Salleh, N., and Shaikhli, I. (2017). Skyline query processing for incomplete data in cloud environment. In *Proceedings of the 6th International Conference on Computing & Informatics*, pp. 567–576.
- Gulzar, Y., Alwan, A. A., Abdullah, R. M., Xin, Q., and Swidan, M. B. (2019). Scsa: Evaluating skyline queries in incomplete data. *Applied Intelligence*, 49(5):1636–1657.
- Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching, volume 14. ACM.
- Han, X., Li, J., Yang, D., and Wang, J. (2013). Efficient skyline computation on big data. *IEEE Transactions on Knowledge and Data Engineering*, 25(11):2521 – 2535.
- Jiang, B., Pei, J., Lin, X., and Yuan, Y. (2012). Probabilistic skylines on uncertain data: model and bounding-pruning-refining methods. *Journal of Intelligent Information Systems*, 38(1):1–39. 191
- Khalefa, M. E., Mokbel, M. F., and Levandoski, J. J. (2010). Skyline query processing for uncertain data. In *Proceedings of the 19th ACM International Conference on Information and knowledge management*, pp. 1293–1296. ACM.
- Kossmann, D., Ramsak, F., and Rost, S. (2002). Shooting stars in the sky: An online algorithm for skyline queries. In *Proceedings of the 28th International Conference on Very Large Data Bases*, pp. 275–286. VLDB Endowment.
- Lawal, M. M., H. Ibrahim, N.F.M. Sani and R. Yaakob (2020), "Analyses of Indexing Techniques on Uncertain Data with High Dimensionality," in *IEEE Access*, vol. 8, pp. 74101-74117
- Lawal M. M., Hamidah Ibrahim, Fazlida Mohd Sani and Razali Yaakob (2018): Skyline Query Algorithms for Computing Uncertain Databases: A survey, A chapter in a book: *The landscape of Computing and Informatics Research Edition (2016-2017)*. In *Proceedings of the 6<sup>th</sup> 2017 International Conference on Computing and Informatics* (pp. 173-191), Published by School of Computing, Sintok.

- Lawal M.M, Hamidah Ibrahim, Fazlida Mohd Sani and Razali Yaakob (2017): Skyline Computation of Uncertain Databases: A survey, 6<sup>th</sup> 2017 International Conference on Computing and Informatics, April 25-27, 2017, Sepang, Malaysia.
- Lee, K. C., Lee, W.-C., Zheng, B., Li, H., and Tian, Y. (2010). Z-sky: an efficient skyline query processing framework based on z-order. *The VLDB Journal*, 19(3):333–362.
- Li, X., Wang, Y., Li, X., and Wang, G. (2012). Skyline query processing on interval uncertain data. In 2012 IEEE 15th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops, pp. 87–92. IEEE.
- Liu, X., Yang, D.-N., Ye, M., and Lee, W.-C. (2013). U-skyline: A new skyline query for uncertain databases. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):945–960.
- Papadias, D., Tao, Y., Fu, G., and Seeger, B. (2005). Progressive skyline computation in database systems. *ACM Transactions on Database Systems (TODS)*, 30(1):41–82.
- Pei, J., Jiang, B., Lin, X., and Yuan, Y. (2007). Probabilistic skylines on uncertain data. In Proceedings of the 33rd International Conference on Very large data bases, pp. 15–26. VLDB Endowment.
- Saad, N. H. M., Ibrahim, H., Alwan, A. A., Sidi, F., and Yaakob, R. (2014). A framework for evaluating skyline query over uncertain autonomous databases. *Procedia Computer Science*, 29:1546–1556.
- Saad, N. H. M., Ibrahim, H., Sidi, F., Yaakob, R., and Alwan, A. A. (2016). Computing range skyline query on uncertain dimension. In International Conference on Database and Expert Systems Applications, pp. 377–388. Springer.
- Saad, N. H. M., Ibrahim, H., Sidi, F., and Yaakob, R. (2018). Non-index based skyline analysis on high dimensional data with uncertain dimensions. In International Baltic Conference on Databases and Information Systems, pp. 272 – 286. Springer.
- Saad, N. H. M., Ibrahim, H., Sidi, F., and Yaakob, R. (2019). Skyline probabilities with range query on uncertain dimensions. In Advances in Computer Communication and Computational Sciences, pp. 225–242. Springer.
- Tan, K.-L., Eng, P.-K., Ooi, B. C., et al. (2001). Efficient progressive skyline computation. In VLDB, volume 1, pp. 301–310.