ABSTRACT
Counts of objects are important for big data analytics. However, spatial objects do not work well with counts. We present the latest developments on distinct counting problem. In particular, we explain Euler Histograms, which are a category of spatial data structures that address the distinct counting challenges. Euler histograms support traditional counting queries as well as other query types.

Categories and Subject Descriptors
E.1 [Data]: Data Structures

General Terms
Algorithms

Keywords
Aggregate Query; Spatial Histogram.

1. INTRODUCTION
With the increasing need to find solutions to various data mining problems, big data analytics has become an important research field. The development of big data analytics brought the need to address an important obstacle, the computational costs associated with manipulating large amounts of data. As the size of raw data can be significantly high, e.g., several petabytes, queries that need to retrieve raw data can consume long periods of time, which can be impractical. Another issue is the trade-off between utility and privacy. Big data applications may need to store highly detailed personal information, such as the trajectories of users, in order to maintain a high level of utility. This leads to the risk that comprehensive profile of individuals can be obtained by an adversary.

One way to address these problems is through the use of aggregates, such as counts, which are important to a wide range of applications. A system that only maintains counts can answer queries significantly faster than the systems that need to retrieve individual objects using aggregate data structures. Another benefit of using counts is that it provides a high level of privacy protection as the count data cannot be easily used to fully identify individuals or the behaviour of individuals.

In many traditional approaches, counts of spatial objects are stored in spatial histograms, which are virtual buckets that partition a space. Although these histograms can be suitable for counting points, they do not work well with objects with extents. The reason is that counts in different buckets do not show relationships between each other. In other words, the histograms cannot maintain the connectivity between different parts of the same object. When the counts in a query area are summed up, the aggregate value can be significantly higher than the correct answer. This is known as the distinct counting problem [1,3,7,8]. Since histograms are mainly designed to store counts, it is difficult to identify which counts have originated from the same object so that the duplicated counts can be eliminated from the result. For example, if we sum up the counts in the query window shown in Figure 1, the result is 3, which is incorrect. We should note that it is possible to attach counts to certain traditional spatial data structures designed for individual object retrieval, e.g., aR-tree [5]. However, this only improves the query processing time. The problem with privacy risks cannot be addressed as individual objects still need to be stored. We investigate a new category of spatial histograms, Euler Histograms, which provide efficient solutions to a range of counting queries and address the problems mentioned above.

2. EULER HISTOGRAMS
Euler Histogram is a spatial data structure for counting rectangular objects that intersect with rectangular query areas [1]. Euler Histogram maintains three types of counts in any space that can be partitioned into grid cells. Counts are maintained for the inner faces of cells, edges between adjacent cells and vertices. The answer to a query can be computed as $F - E + V$, where $F$ is the count from faces inside query area, $E$ is the count from edges between the
faces and $V$ is the count from vertices between the faces. Figure 2 shows an example Euler Histogram, where the answer to the query can be computed as $11 - 10 + 2 = 3$. Although this data structure addresses the distinct counting problem for rectangular objects, it does not work well with concave objects, objects with holes and objects with disjoint parts.

The original Euler Histogram counts objects that intersect with a query area. The data structure can also be used for counting objects based on fine-grained spatial relations between the objects and the query area, e.g., counting objects that are fully contained in an area [4,6]. Proposed techniques collect partial results, such as the number of objects intersecting with the interior of query area, the number of objects intersecting with the exterior of query area and the total number of objects. By manipulating the partial results, the number of objects can be deduced for specific spatial relations. A challenge to these solutions is that they work well only if all the objects have similar width and height. To address this challenge, the existing work proposes multi-scale/multi-resolution methods that construct different histograms for objects with different combinations of height and width. A drawback of such methods is that the storage cost can be high.

**Distributed Euler Histogram** [10] is developed for counting the number of non-simple (i.e., self-intersecting) curves, rather than rectangular areas. This is achieved by incrementing face counts and edge counts whenever an object enters a face or crosses an edge. The data structure is particularly suitable for distinct entry queries on moving objects, which are different to distinct object queries as an object can make multiple entries to a query area.

One approach that can address the distinct counting problem is attaching partial ID information with the counts [9]. This helps to achieve a balance between accuracy, privacy and storage cost. It also enables solutions to a range of innovative queries in traffic monitoring, such as counting the number of vehicles that move on the boundary of an area.

Yet another approach that helps to address the distinct counting problem uses a hierarchy of histograms, called **Euler Histogram Tree** [11]. Different levels of the tree correspond to space partitions at different resolutions. Counts of an object are updated at all the levels of the tree. The proposed solution tries to use counts from low-resolution histograms as much as possible. Figure 3 shows an example, where the answer is computed as $3 - 2 = 1$. This approach can achieve a high level of accuracy in counting arbitrary spatial objects that do not work well with the original Euler Histogram.

In recent work, we have looked into counting trajectories with virtual counts that can eliminate the concaveness and holes from objects [2]. When the virtual counts and the actual counts are combined, a trajectory is turned into an object that is convex without holes. This also effectively reduces errors caused by the distinct counting problem.

### 3. CONCLUSION

We present the development of Euler Histograms that are highly efficient and privacy-aware in solving counting queries. Euler Histograms have advantages over many traditional data structures as the connectivity between different parts of a spatial object can be preserved in the counts to a high level of confidence. Euler Histograms enable a range of innovative spatial queries. The distinct counting problem that are common to spatial histograms can be addressed by different variations of this data structure.

### 4. REFERENCES