Spatiotemporal Aggregate Functions for Temporal GIS*

*Jin Soo Kim, **Hyun Ho Shin, **Jeong Hee Chi, **Keun Ho Ryu
Database Laboratory, ChungBuk National University
Cheongju Chungbuk, 361-763, Korea
*kjs9990@yahoo.co.kr ; **{shinhh, jhchi, khryu}@dblab.chungbuk.ac.kr

Abstract: Aggregation is an operation that returns a result value through a computational process on the data which satisfy a certain condition. Recently many applications use aggregation to analyze spatiotemporal data. Although spatiotemporal data change its states over time, previous aggregation works have only dealt with spatial or temporal aspect of object. In this paper we propose spatiotemporal aggregate functions that operate on spatiotemporal data. The proposed algorithms are evaluated through some implementation results. The experiment results show that the proposed aggregate functions are applicable to spatiotemporal data efficiently.
Keywords: Spatiotemporal aggregate functions.

1. Introduction

Most of applications such as geographic, agriculture, and weather management systems have a common characteristic that the semantics of spatial and time-varying data are the major elements. A wide range of applications manage spatial, time-varying, or spatiotemporal data[1]. These applications can handle the user's various queries for spatiotemporal data analysis. In particular, one of the query operations, aggregation is an operation that returns a summarized data through a computational process on the data which satisfy a certain condition.

As an example consider an agriculture management system. Users can query on spatiotemporal data, such as "given an arbitrary spatial region compute the total precipitation of rain falls in this region in 1999". Query has both temporal and spatial constraints and focuses on the past and current time. However, previous works have been attached only to temporal or spatial so that are not directly available in real world.

This paper addresses these problems by proposing spatiotemporal aggregate functions. The proposed aggregate functions operate on spatiotemporal data. And these can represent query briefly than other application.

The rest of this paper is organized as follows: Section 2 describes related works in the spatial and temporal aggregate functions. Section 3 defines spatiotemporal aggregate functions. Section 4 describes algorithms of the proposed spatiotemporal aggregate functions. In section 5, the proposed spatiotemporal aggregate functions will be implemented and evaluated with the estate management application. Finally, Section 6 present conclusion of our work and future directions.

* This work was supported by University IT Research Center Project and KOSEF RRC Project(Cheongju Univ. ICRC) in Korea

2. Related Works

Many approaches have been recently proposed to address temporal or spatial aggregate functions. The aggregation in temporal databases, that is, temporal aggregation is an extension of conventional aggregation on domain and range of aggregates to include time concept. In [2,3], they proposed the temporal aggregate functions that are performed on the historical information of time-varying data such as TIMEFRIST, TIMELAST, and TIMEMIN and so on. Also spatial aggregation performed on the spatial data. [4] proposed spatial aggregate function such as MAXgeo and MINgeo. The MAXgeo returns the max object satisfying some condition.

However, these aggregate functions have a problem that can't apply to spatiotemporal data directly. Therefore we propose the spatiotemporal aggregate functions which are directly applied to the spatiotemporal data to solve the above problem.

3. Definition

In this section, we propose the spatiotemporal aggregate functions which operate spatiotemporal data and define the proposed aggregate functions. Table1 shows notations for definition our spatiotemporal aggregate functions.

Table 1. Notations for Spatiotemporal Aggregate Function.

Notation	Descriptions
A	an attribute of spatial object
SC	a construction factor of spatial object(MBR)
W_L	a layer of given window area
F_L	a retrieved layer of spatial data
CVT	common valid time
‘	a current relation

Definition 1. *Spatial* (F_L) *stCOUNT Spatial* (W_L)

$$= [COUNT\ (\forall\ F_L)\ with\ \exists\ SC\text{¢}(F_L) \cap \exists\ SC\text{¢}(W_L) \wedge CVT(F_L,\ W_L)]$$

Definition 2. *Spatial* (F_L) *stSUM Spatial* (W_L)

$$= [SUM\ (\forall A\text{¢}(\ F_L))\ with\ \exists\ SC\text{¢}(F_L) \cap \exists\ SC\text{¢}(W_L) \wedge CVT(F_L,\ W_L)]$$

Definition 3. *Spatial* (F_L) *stAVG Spatial* (W_L)

$= [AVG\ (SUM\ (A(F_L))\ /COUNT\ (F_L))$
$with\ \exists\ SC(F_L) \cap \exists\ SC(W_L) \wedge CVT(F_L, W_L)]$

Definition 4. *Spatial* (F_L) *stMAX Spatial* (W_L)

$= [MAX\ (F_L)\ with\ \exists\ SC(F_L) \cap \exists\ SC(W_L) \wedge CVT(F_L, W_L)]$

Definition 5. *stMAX* $\hat{U}$ *stMIN*

4. Aggregate Algorithms

We describe algorithms of spatiotemporal aggregate functions that are able to analyze data statistically to generate summarized information through computation on the past and current information. The proposed aggregate functions can be classified into two different categories: the computational aggregate function and the selective aggregate function.

1) Algorithms of Computational Aggregate Functions

The value of the computational aggregate is obtained by accumulating certain computations such as stSUM, stCOUNT, and stAVG.

Table 2. Algorithm of stSUM.

```
Input : Source Object, Target Object
Output : sum of feature attributes
Method :
Identify input features from the window;   /* query window */
While(Check current feature schema) {
/*extract spatiotemporal satisfying spatiotemporal conditions */
 If (feature fid in current feature schema equal feature fid in input
   tuple){
      if(valid time of feature fid in current feature schema <=
           valid time of feature fid in input tuple) {
             for(Retrieve all tuple != NULL) {
                   stSUM+ = aggregated attributes;
                         }
                    }
               }
    return stSUM
   }
End stSUM  /* result of stSUM */
```

The algorithm of stSUM returns sum value of spatiotemporal attribute satisfying spatiotemporal conditions.

Table 3. Algorithm of stCOUNT.

```
Input : Source Object, Target Object
Output : count of features
Method :
Identify input features from the window     /* query window */
While (Check current feature schema) {
/*extract spatiotemporal satisfying spatiotemporal conditions */
 If (Feature fid in current feature schema equal feature fid in input
   tuple ) {
      If (Validtime of feature fid in current feature schema <=
           validtime of feature fid in input tuple) {
             for (Retrieve all tuple != NULL) {
                   stCOUNT+ = aggregated features;
                         }
                    }
               }
     return stCOUNT;
    }
  End stCOUNT
```

Table 3 shows stCOUNT that returns numbers of counted objects satisfying spatiotemporal conditions.

Table 4. Algorithm of stAVG.

```
Input : Source Object, Target Object
Output : average of feature attribute
Method :
Identify input features from the window;    /* query window */
While (Check current feature schema) {
/*extract spatiotemporal satisfying spatiotemporal conditions */
 If (Feature fid in current feature schema equal feature fid in input
   tuple ) {
     If (Validtime of feature fid in current feature schema <=
          validtime of feature fid in input tuple) {
           for (Retrieve all tuple != NULL) {
                  SUM+ = aggregated attributes;
                  COUNT+ = aggregated features;
                 }
            stAVG = SUM / COUNT;
             }
          }
   return stAVG;
    }
  End stAVG
```

The algorithm of stAVG returns average value that is obtained by dividing stSUM by stCOUNT.

2) Algorithms of Selective Aggregate Functions

The value of the selective aggregate is obtained by selecting a representative value among the values of tuples. stMAX and stMIN are the selective aggregate functions. The algorithm of stMAX returns the object which has the max attribute value satisfying the given interval and query window.

Table 5. Algorithm of stMAX.

```
Input : Source Object, Target Object
Output : feature of Max attribute
Method :
Identify input features from the window;     /* query window */
While(Check current feature schema) {
/*extract spatiotemporal satisfying spatiotemporal conditions */
 If (Feature fid in current feature schema equal feature fid in input
   tuple ) {
     If (Validtime of feature fid in current feature schema <=
          validtime of feature fid in input tuple) {
            for (Retrieve all tuple != NULL) {
                 MAX = feature attribute;
                 stMAX = MAX(fid);
                If (next feature attribute > MAX)
                     MAX = feature attribute;
                     stMAX = MAX(fid);
                            }
                 return stMAX;
                      }
                  }
              }
End stMAX
```

Also the stMIN returns the object of min attribute value among entire attributes of objects satisfying some conditions.

Table 6. Algorithm of stMIN.

```
Input : Source Object, Target Object
Output : feature of Min attribute
Method :
Identify input features from the window;      /* query window */
While (Check current feature schema) {
/*extract spatiotemporal satisfying spatiotemporal conditions */
 If (Feature fid in current feature schema equal feature fid in input
   tuple ) {
     If (Validtime of feature fid in current feature schema <=
          validtime of feature fid in input tuple) {
            for (Retrieve all tuple != NULL) {
                MIN = first feature attribute;
                stMIN = MIN(fid);
                If (next feature attribute < MIN)
                     MIN = next feature attribute;
                      stMIN = MIN(fid);
                           }
                return stMIN;
                       }
                   }
              }
End stMIN
```

5. Implementation and Evaluation

In this section, we implemented the proposed spatio-temporal aggregate functions that apply to the estate management application. We used the OpenGL Graphic Library, Visual C++6.0 and MS SQL Server for implementing the prototype. In the experiment, we showed that the proposed aggregate functions apply to spatio-temporal data directly in the fig. 1.

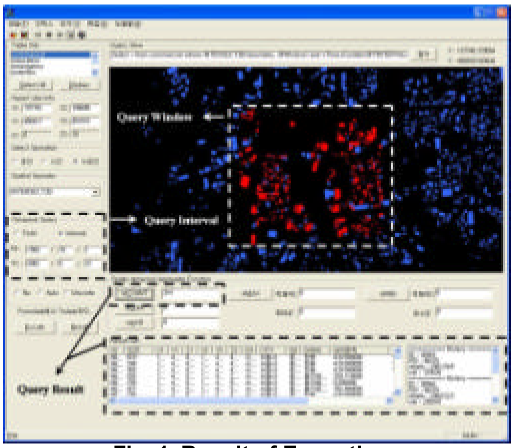

Fig. 1. Result of Execution.

We also compared our proposed aggregate functions with that of SDE(Spatial Database Engine). Table 7 is to represent query statement "given an arbitrary estate region compute the number of bank from March to September in 2000", in order to compare with the proposed aggregate function and SDE. The proposed aggregate functions show how any spatiotemporal aggregate query can be expressed very efficiently without any spatial analysis information.

Table 7. Comparison of query representations.

SDE(Spatial Database Engine)	Spatiotemporal Aggregate Function
Feature zonef, pesgf; Zonelayer = s100; Pescnt = 0; For (ret = SE_get_feature_by_layer (zonelayer, &zonef, Zone=ESTATE); ret = SUCCESS; Ret = SE_get_next_feature(&zonef)) { /* for */ SE_set_search_by_feature(&zonef); For returncode=SE_search(estlayer, SM_AI, &estgf, USE=BANK, VTs >= 01-Mar-00 and VTe < 31-Sep-00); ret= SUCCESS; ret=SE_next_search(&estgf)) Pescnt++; } /* outer loop */	SELECT S.***stCOUNT****(T)* FROM s100 S, s100 T WHERE S.zone = ESTATE and T.USE = BANK and T.tIntersects(S) and S.validtime PERIOD '01-Mar-00, 31-Sep-00' ;

6. Conclusion

Aggregate functions are essential to statistical tasks and decision support applications. The current demand of aggregate operations relates to spatiotemporal data which are contained both spatial and temporal data concurrently. However previous works just dealt with spatial or temporal aspect of object.

To overcome the limitations of previous work, we proposed the aggregate functions, such as stCOUNT, stSUM, stAVG, stMAX, stMIN, and described algorithms. The proposed aggregate functions are suitable for statistical analysis on temporal and spatial information of various spatiotemporal data which are stored in spatiotemporal database. We then implemented our proposed aggregate function by example of spatiotemporal query and compared query representation of proposed aggregate function with SDE. The estate management system can represent query briefly than other application by using the proposed aggregate functions.

In the future we plan to improve aggregate algorithms and index for process of aggregate query. We also will propose extended spatiotemporal aggregate functions.

References

[1] T.K.Sellis, "Research Issues in Spatio-temporal Database System," SSD, pp.5-11, 1999.

[2] DongHo Kim, InHong Lee and KeunHo Ryu, "The Design and Implementation of Aggregate Function of Temporal Databases in Main Memory," Journal of the KISS, Vol.21, No.8, Aug, 1994.

[3] N.Kline, and R.T.Snodgrass, "Computing Temporal Aggregates," International Conference on Database Engineering, p222-231, 1995.

[4] Erwig and M. Schneider, "Developments in spatio-temporal query languages," DEXA Workshop, 441-449, 2000.