STORING AND QUERYING SPATIALLY VARYING DATA QUALITY INFORMATION USING AN INTEGRATED SPATIAL RDBMS

MGSM Zaffar Sadiq [a,*], Matt Duckham [b]

[a]Cooperative Research Centre for Spatial Information, Dept. of Geomatics,
The University of Melbourne, Victoria - 3010, Australia,   s.mohamedghouse@pgrad.unimelb.edu.au

[b]Dept. of Geomatics, The University of Melbourne, Melbourne, Victoria - 3010, Australia,
mduckham@unimelb.edu.au

KEY WORDS: quality, spatial variation, metadata, sub-feature variation, spatial database

ABSTRACT:

Current representations of spatial data quality (SDQ) do not adequately represent the spatial variation of SDQ. For example, a user who wants to know the positional accuracy of a feature usually has to rely on metadata statements, which usually refer to the entire data set. In reality, SDQ varies spatially; quality may be higher in some locations and lower in others perhaps due to different data collection and acquisition methods. To represent spatially varying data quality, SDQ may need to be stored for individual features and parts of features in a dataset. This paper proposes a new, flexible data model for efficient storage and retrieval of spatially varying quality information in a spatial database. The model can store quality information in different ways according to the requirements of the data set. SDQ can be stored alongside individual features or parts of features in the database, or as an independent spatial data layer. The paper reports on an extension to Oracle spatial RDBMS, used to implement this model of spatially varying SDQ. An investigation into the different querying mechanisms required to support this model of SDQ shows that the different model allows a flexible representation of spatially varying quality, including modeling sub-feature variation in quality.

1. INTRODUCTION

1.1 Overview

Data quality is crucial to assess the "fitness for use" of spatial data and therefore key to achieve interoperability (Frank, 2004). Standards organizations have recognized the importance of data quality and have made provisions to report the quality in the form of "data quality statements" (Qiu, 2002), which are widely accepted and used in the industry. In reality, spatial data quality (SDQ) varies according to location. For example, cadastral data produced by the Victorian Department of Sustainability and Environment (DSE, the government department which is responsible for maintaining and updating spatial data for the state of Victoria) can have positional accuracy that varies *within* individual land parcels. Current spatial data quality storage mechanisms are inadequate for representing such sub-feature variation. As a result, the actual positional accuracy of DSE data in some locations can be *higher* than the reported accuracy (Ramm, 2005).

1.2 Motivation

Spatial variation in data quality is a common feature of spatial data sets. Hunter (2001) uses the example of metadata that reports positional accuracy ranging from 100m to 1,000m or ±1.5m (urban) to ±250m (rural). The metadata does not report *where* this variation occurs. The information would be more meaningful if the location of the different positional accuracies were reported. Knowing the locations where different data qualities exist is clearly important to decision making using a dataset. With the advent of new positioning technologies used for cadastral upgrading, updated data often has better accuracy than the data it replaces (Hunter, 2005). Although Goodchild (2006) notes that quality is difficult to attach to individual features in database, to enable proper communication of data quality it is necessary to store quality information for individual features and parts of features in the database

1.3 External and integrated representation of SDQ

SDQ storage in spatial databases can be broadly classified into two representations: *external* and *integrated*. In the *external* representation, the SDQ information is stored separately from the spatial database. Using the external representation, it is harder to maintain the link between spatial data and externally stored SDQ information, and harder to update, and query that information. Moreover, the SDQ information is often aggregated for the entire spatial database, ignoring spatial variation. Devillers et al. (2005) have criticized the linked metadata approach as: (1) there are limitations in the type of metadata that can be stored and (2) the level of detail of the metadata information. In the *integrated* representation the SDQ is integrated with the spatial database. By adopting an integrated representation the user can more easily represent and query spatially varying quality.

Several existing studies have adopted an integrated approach to storing SDQ. Duckham (2001) has developed a formal model of object-based variation in spatial data quality, using object calculus. Gan et al. (2002) have developed an Error Metadata Management System (EMMS) to represent quality at feature level. Similarly, Qiu et al (2002) have developed a model to represent spatial variation of quality in a database up to the feature-level. Devillers et al. (2005) have designed a Quality Information Management Model (QIMM), which has access to different level of details on spatial data quality. QIMM is an extension of Qiu's (2002) work, based on multidimensional spatial database analysis tools (SOLAP).

However, these approaches suffer from two important drawbacks. Firstly, by storing quality information for each object in the database, some spatial variation can be represented across the layer. However, these approaches are limited in their representational capabilities, since they cannot adequately represent variation in quality *within* an object otherwise can be called *sub-feature variation* (see section 1.4).

Secondly, this existing work does not address the efficiency of storage and retrieval mechanisms. Efficient storage and querying mechanisms are an important factor when representing SDQ for each object or sub-object in the database, since such an approach can potentially result in voluminous SDQ information. Chrisman (2006) has also raised the concern that the amount of quality information might take up as much space in data storage as geometric coordinates. One might argue that data storage costs are continually decreasing. However, irrespective of storage cost, efficient and effective storage and querying are essential for handling large volumes of SDQ.

1.4 Sub-feature variation

None of the existing research outlined in section 1.3 can claim to be able to efficiently represent spatially varying SDQ across a range of spatial data types. One of the critical questions that is not adequately addressed by these models is the representation of sub-feature variation (variation in quality within a geographic feature). Sub-feature variation is not an issue for raster-based data structures, since each cell in a raster represents an atomic unit of space for the purpose of a particular dataset. Similarly, sub-feature variation is not a problem for points in vector-based spatial data. However, potentially lines and polygons in vector data sets can exhibit sub-feature variation in data quality. It is this issue that we pay particular attention to in the remainder of this paper.

2. MODELS

2.1 Spatial variation models in database

Models of spatial variation in a database can be classified according to whether they are based primarily on object-based or field-based representations. An object-based model treats the space as populated by discrete, identifiable entities with a geospatial reference. A field-based model treats geographic information as collections of spatial distributions. Each distribution may be formalized as a function from a spatial framework to an attribute domain (Worboys and Duckham, 2004). Given the importance of features and sub-feature variation, three different models of spatial variation in data quality have been identified for storing spatial variation in data quality in a database. First, in the per-feature model (object-based), quality information can be stored against each feature in the database. Second, in the feature-independent model (field-based), quality information is stored independently of the features. Their, the feature-hybrid model is derived from a combination of object- and field-based models.

2.1.1 Per-feature model

The spatial data quality of a feature can be stored along with the feature as an additional attribute (Hunter and Qiu, 2003, United States Department of the Interior Bureau of Land Management, 2001). Per-feature (object-based) quality can be modeled as a function f: $O \rightarrow Q$ where O is the set of objects and Q is the quality codomain. Features can include points, lines and polygons. However, as discussed in section 1.3 the per-feature model cannot represent variation within a feature (sub-feature variation).

2.1.2 Feature-independent model

The feature-independent model store spatial variation in spatial data quality as a separate theme or layer in a spatial database (Maclean et al., 1993), termed the feature-independent model. Feature-independent (field-based) quality can be modeled as a function g: $S \rightarrow Q$ where S is the spatial framework and Q is the quality co-domain. The feature-independent model is also related to work by Heuvelink (1996), who proposed a model for analyzing spatial variation. In order to retrieve sub-feature variation, it is necessary to perform spatial joins with other stored spatial data.

2.1.3 Feature-hybrid model

The feature-hybrid model stores quality information on a per-feature basis, but augments the stored quality with additional spatial structure. Feature-hybrid quality can be modeled as a function h: $O \rightarrow Q^S$ where Q^S denotes the set of functions (fields) from S to Q, i.e., $Q^S = \{f \mid f: S \rightarrow Q\}$. Sadiq et al (2006) have investigated the use of linear referencing systems (LRS) as one example of a feature-hybrid model implementation. LRS are widely used in the transportation. In a LRS, additional tables are used to store information about *measures* within the linear features, like roads. A measure represents information for a segment of the linear feature, enabling the representation of sub-feature information. An explicit link between the feature attribute table and the additional table enables the retrieval of sub-feature information. This paper focuses on implementing a more general model, which can support many other types of feature-hybrid implementation, including storing sub-feature SDQ information about polygonal features, which was a limitation in this earlier work.

3. STORAGE

Rather than propose new architectures for storing SDQ (e.g., Duckham 2001), this research relies on a conventional spatial RDBMS (relational database management system) architecture. Oracle Spatial was used to implement the relational scheme for each of the three models of spatial variation in SDQ. To illustrate the different models of spatial variation in SDQ a few parcels (9) pertaining to Panton Hill suburb from the DSE's (the Victorian Department of Sustainability and Environment) Vicmap Property database were taken as sample data for developing a prototype. The parcels were represented as polygon features. The vertical accuracy of the corresponding parcels was generated hypothetically to test the spatial variation models in section 2.1.1, 2.1.2 and 2.1.3.

Let us name the parcel layer as base (Fig 1) which has nine parcels of the Panton Hill locality. The base table which stores the spatial data has the following relation scheme:

base (<u>fid</u>: number, geom: geometry)

where fid is a primary key and geom is the geometry of the parcel features in the base table. The following subsection show how spatially varying SDQ information can be stored in the three models. Note that as discussed in section 2.1.1, the representational limitations of the per-feature model mean that there is no sub-feature variation in the per-feature model.

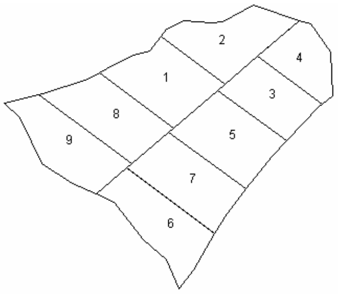

Fig 1: BASE

3.1 Per-feature relation scheme

The per-feature quality model (see section 2.1.1) has a relation scheme of the general form:

pf (qid: int, quality: int, …)

where qid is a quality identifier and quality stores the quality information, which relates qualities to features in a specific feature relation. In our example, the per-feature quality relation is implemented as table pfq (table 1), which stores corresponding quality information (vertical accuracy in meters) for each parcel in the base table. The per-feature quality table pfq we have the following relation scheme:

pfq (qid: number, vacc: number)

QID	VACC
1	2.75
2	2.75
3	2.75
4	2.5
5	2.5
6	2.5
7	2
8	2
9	1

Table 1: PFQ (Per-Feature Quality)

3.2 Feature-independent relation scheme

A feature-independent quality relation fi has the relation scheme of the general form:

fi (qid: int, geom: geometry, quality: number,…)

where qid is a quality identifier, geom is a geometry data type that describes the spatial characteristics of each quality feature and quality stores the quality information (section 2.1.2). The feature-independent quality model in this example is implemented as table fiq (table 2, fig 2). The quality information is stored along with the quality geometries independent of base's (parcel) geometries. Thus, for the feature-independent quality table fiq we have the following relation scheme:

fiq (qid: number, geom: geometry, vacc: number)

QID	GEOM	VACC
1	MDSYS.SDO_GEOMETRY	2
2	MDSYS.SDO_GEOMETRY	2.5
3	MDSYS.SDO_GEOMETRY	2.75
4	MDSYS.SDO_GEOMETRY	3

Table 2: FIQ (Feature-Independent Quality)

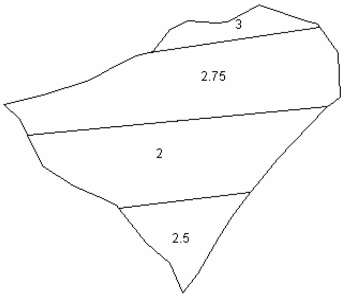

Fig 2: FIQ (Feature-Independent Quality)

3.3 Feature-hybrid relation scheme

A feature-hybrid quality model (see section 2.1.3) has a relation scheme of the general form:

fh (qid: int, fid: int, geom: geometry, quality: int,...)

where qid and fid are composite keys (quality identifiers), which relate qualities to features in a specific feature relation; geom is a geometry data type that describes the sub-feature spatial characteristics of each quality feature; and quality stores the SDQ information.

The feature-hybrid quality model is implemented as table fhq (table 3, fig 3). The fhq table stores sub-feature quality information for each parcel in the base table. Thus, for the feature-independent quality table we have the following relation scheme of the form:

fhq (qid: number, fid: number, geom: geometry,
vacc: number)

QID	FID	GEOM	VACC
1	6	MDSYS.SDO_GEOMETRY	2.5
1	7	MDSYS.SDO_GEOMETRY	2.5
2	1	MDSYS.SDO_GEOMETRY	2
2	3	MDSYS.SDO_GEOMETRY	2
.	.	.	.
4	9	MDSYS.SDO_GEOMETRY	2.75

Table 3: FHQ (Feature-Hybrid Quality)

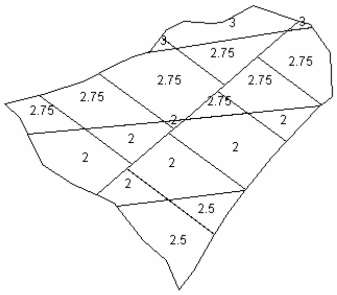

Fig 3: FHQ (Feature-Hybrid Quality)

Note that in this case, the (hypothetical) quality information in the fhq table is derived from the fiq table (simply the intersection of the base and fiq tables). Of course it need not be the case that the fhq be derived from the fiq table: this is purely for illustrative purposes.

3.4 Summary

To summarize:

1. In the per-feature model, the quality is stored in an additional table pfq. For each parcel of the base table a corresponding quality value can be stored in the pfq table. The limitation of the model is that sub-feature variation cannot be represented. For example, the model cannot represent the case where different parts of parcel no.1 have different quality values.

2. In the feature-independent model, the geometry of spatial variation in quality is stored independently of the parcel geometry. While this is simple, it can increase the storage and maintenance costs when compared to the per-feature model. The model does allow the representation of sub-feature variation when it is queried in conjunction with the base table.

3. In the feature-hybrid model, the qualities represented in the model are associated with geometries that are independent, but contained within, the corresponding parcel's geometry. To link the quality and parcel information, the feature id of each parcel is stored along with its quality geometries. As a result, each parcel may have more than one quality value, making the fhq table capable of representing sub-feature variation.

4. QUERYING

Querying is the process of retrieving stored information from the database. SQL (Structured Query Language) is the standard language for querying relational databases, and used to query the SDQ information stored in the three models.

A **non-spatial query** retrieves information based on non-geometric attributes. A point query finds the feature, which is associated with a particular attribute value. A range query searches for features whose attribute values fall within a particular range. Many SDQ queries can be considered non-spatial, for example:

- "What is the vertical accuracy of parcel no. 5?" (point query)
- "List those parcels with a vertical accuracy between 1m and 2.5m?" (range query)

A query, such as "Show the distance between two points," deals with geometries (spatial data). A **spatial query** is defined as one that manipulates location data in addition to attribute data (Adam and Gangopadhyay, 1998). Shekhar and Chawla (2003) have classified spatial queries as: point (find all features that intersect a particular point), range (find all features that intersect a particular region), nearest neighbor (finds the geometries residing close to a given feature), and spatial join queries (involves retrieving information based on the spatial relationships between features in two or more relations). Some examples of spatial queries in SDQ include:

- "What is the vertical accuracy at coordinate (x, y)?" (spatial point query).
- "List the vertical accuracy of the region within 50m of parcel with vertical accuracy 10m?" (spatial range query).
- "List two nearest vertical accuracies to any region with 1m vertical accuracy?" (nearest neighbor query).
- "List the feature ids for parcels where the recorded per-feature and feature-independent vertical accuracy differ by 0.25m or more?"

The queries above are amongst many that have been tested on the different data quality storage models described above. The following section discusses a few queries and their results.

4.1 Non-spatial point query (Query1)

"What is the vertical accuracy of parcel no. 5?"

4.1.1 Per-feature model

In order to retrieve the quality information on per-feature model a natural join has to be performed on relation base and pfq as: $\text{base} \bowtie_{\text{fid} = 5} \text{pfq}$.

Query 1.1

```
SELECT    a.fid, b.vacc
FROM      base a, pfq b
WHERE     a.fid = b.qid AND a.fid = 5;
```

Results:

```
FID   VACC
-----  --------
5      2.5
```

The results of query 1.1 retrieves vertical accuracy of 2.5m for the parcel no 5 upon performing a natural join between base and pfq tables.

4.1.2 Feature-independent model

The non-spatial point query on feature-independent model requires a spatial join has to be performed on relation base and fiq as: $\text{base} \bowtie_{\text{base.geom = fiq.geom}} \text{fiq}$

Query 1.2

```
SELECT   a.fid, b.vacc
FROM     base a, fiq b
WHERE    base.geom ∩ fiq.geom AND a.fid = 5;
```

Results:

```
FID   VACC
-----  --------
5      2
5      2.75
```

We notice from the results of query 1.2 that the parcel fid 5 has two vertical accuracy values 2m and 2.75m within the feature. Thus, the feature-independent model represents sub-feature variation. The spatial join needed to find the spatial intersection of records in base and fiq is computationally expensive when compared to the natural join performed in the query 1.1.

4.1.3 Feature-hybrid model

The non-spatial point query for the feature-hybrid model requires a natural join base $\bowtie_{\text{base·fid = fhq.fid}}$ fhq to retrieve vertical accuracy of parcel no 5.

Query 1.3

```
SELECT   fid a, vacc b
FROM     base a, fhq b
WHERE    a.fid =b.fid AND a.fid = 5;
```

Results:

```
FID   VACC
----- --------
5     2
5     2.75
```

As expected, the result of the feature-hybrid query shows the same as the results of feature-independent query 1.2.

4.2 Spatial point query (Query2)

"What is the vertical accuracy of the parcel at coordinate x, y?"

For this query a separate table poi is created to store x and y as point geometry. Depending on the quality model, a natural or spatial join is required between the base and quality (pfq, fiq and fhq) table to retrieve its corresponding vertical accuracy.

4.2.1 Spatial point query on per-feature model

In the per-feature model, a natural join base $\bowtie_{\text{fid = qid}}$ pfq is performed to retrieve the vertical accuracy of each parcel. In the next step the x, y value is retrieved using a spatial join between the geometries of the point (x, y) and the results of the natural join.

Query 2.1

```
SELECT  a.vacc
FROM    (SELECT  i.geom, j.vacc
         FROM    base i, pfq j
         WHERE   i.fid = j.qid) a, poi b
WHERE   a.geom ∩ b.geom;
```

Results:

```
VACC
---------
2.75
```

4.2.2 Spatial point query on feature-independent model

Querying quality of a point using the feature-independent model involves two spatial join as:

(base $\bowtie_{\text{base.geom = fiq.geom}}$ fiq) $\bowtie_{\text{geom = poi.geom}}$ poi

Query 2.2

```
SELECT  a.vacc
FROM    (SELECT  (i.geom ∩ j.geom) AS geom, j.vacc
         FROM    base i, fiq j ) a, poi b
WHERE   a.geom ∩ b.geom;
```

Results:

```
VACC
---------
3
```

4.2.3 Spatial point query on feature-hybrid model

The feature-hybrid model's spatial point query is similar to that of query 2.1. First, a natural join base $\bowtie_{\text{fid = fid}}$ fhq is performed to retrieve the vertical accuracy of each parcel. In the next step the x, y values are retrieved by performing a spatial join between the geometries of the point (x,y) and the results of the natural join.

Query 2.3

```
SELECT  a.vacc
FROM    (SELECT  i.geom, j.vacc
         FROM    base i, fhq j
         WHERE   i.fid = j.fid) a, poi b
WHERE   a.geom ∩ b.geom;
```

Results:

```
VACC
---------
3
```

Comparing per-feature, feature-independent and feature-hybrid models, the per-feature and feature-hybrid require only a natural join to retrieve accuracy at the x and y coordinates of geometry stored in base table. The feature-independent model retrieves vertical accuracy at coordinates x and y by performing a spatial join. The results of querying the feature-hybrid and feature-independent models are identical and exhibit sub-feature quality (more than one quality for a single feature). In this case, the feature-hybrid query achieves a desirable balance of greater efficiency than the feature-independent model, while still being able to represent sub-feature quality, unlike the per-feature model.

4.3 Spatial range query (Query3)

"List the vertical accuracy of parcels based on a region (south eastern part of Panton Hill suburb)?"

4.3.1 Spatial range query on per-feature model

The spatial range query for the per-feature model is developed by performing a spatial join (fig 4) over the region's geometries as follows:

Query 3.1

```
SELECT  geom, vacc
FROM    region a,
        (SELECT  c. geom
         FROM    base c, pfq d
         WHERE   c.fid = d.qid ) b
WHERE   a.geom ∩ b.geom;
```

Results:

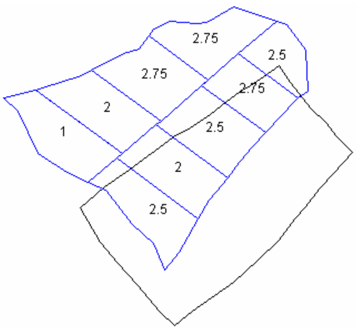

Fig 4: Spatial query of base $\bowtie_{\text{base.fid = pfq.qid}}$ pfq

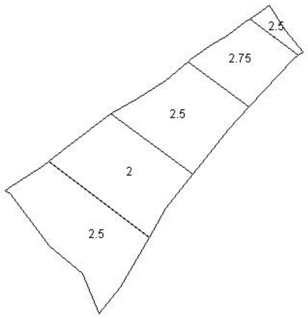

Fig 5: Results of query 3.1

The query results in fig.5 show 5 vertical accuracies across the southern region of Panton Hill locality.

4.3.2 Spatial range query on feature-independent model

Using the feature-independent model for a spatial range query requires two spatial joins when compared to the per-feature model. Fig 6 shows the results of the first spatial join performed between base $\bowtie_{\text{base.geom = fiq.geom}}$ fiq on region.geom

Query 3.2

```
SELECT  b.geom, b.vacc
FROM     region a,
         (SELECT  (i.geom ∩ j.geom) AS geom, j.vacc
          FROM     base I, fiq j ) b
WHERE   a.geom ∩ b.geom;
```

Results:

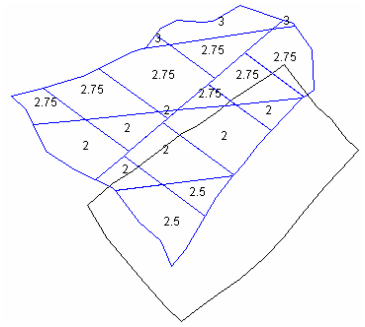

Fig 6: Intersect operation on (base $\bowtie_{\text{base.geom = fiq.geom}}$ fiq).geom, region.geom

Results:

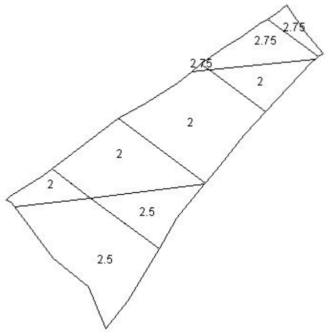

Fig 7: Results of query 3.2

The query results in fig.7 have retrieved 9 vertical accuracies across the southern region of the study area. The results reveal that except for parcel 4 (accuracy value 2.75m) all other parcels have vertical accuracies recorded in sub-features.

4.3.3 Range query on feature-hybrid model

The range query for feature-hybrid model involves one natural and spatial join. Geometries of the fhq table are intersected with the region's geometries and then a natural join is performed on the results of the intersection and the base table.

Query 3.3

```
SELECT  geom, vacc
FROM     base a,
         (SELECT  geom, vacc
          FROM     region c, fhq d
          WHERE   c.geom ∩ d.geom ) b
WHERE   a.fid =  b.fid;
```

The results of query 3.3 are the same as the result of feature-independent model query 3.2. The feature-hybrid query retrieves sub-feature quality by performing one natural join and a spatial join when compared to the following:

- Per-feature query 3.1 performs one spatial and one natural join but does not retrieve sub-feature quality.
- Feature-independent query 3.2 performs two spatial joins to retrieve sub-feature quality.

4.4 Nearest Neighbor (NN) query (Query4)

"List two nearest vertical accuracies to any region with 1m vertical accuracy?"

The NN query fetches features from its nearest proximity. A spatial operator SDO_NN is used in the NN query, which requires the geometries to be spatially indexed. The geometries, which result from a query, cannot be used by the SDO_NN operator as the geometries are not spatially indexed. In Oracle, a spatial index cannot be created on the fly for the geometries whilst querying. Hence, the interim results of the query have to be stored as a table.

4.4.1 Nearest Neighbor (NN) query on Per-Feature model

The per-feature NN query is executed in two steps. The first part is to create a table pfqn based on the relation base $\bowtie_{\text{base.fid = pfq.qid}}$ pfq and create a spatial index on the geometries. The second part is to find the vertical accuracies adjacent to the parcel's region with 2m vertical accuracy.

Query 4.1

Step 1

```
CREATE TABLE  pfqn AS
   SELECT  a.fid, a.geom, b.vacc,
   FROM     base a, pfq b
   WHERE   a.fid = b.qid

CREATE INDEX  pfqn_idx ON pfqn(geom)
INDEXTYPE IS    mdsys.spatial_index
```

Step 2

```
SELECT  a.vacc, b.geom
FROM     pfqn b,  pfqn a
WHERE   b.vacc = 1 AND
         SDO_NN(a.geom, b.geom) = 'TRUE'
           AND ROWNUM <= 2;
```

Results:

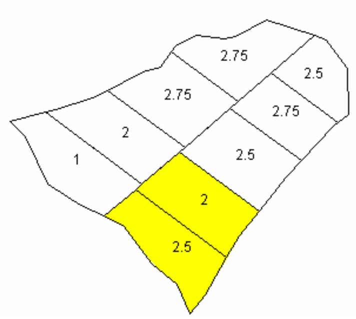

Fig 8: Results of query 4.1

The query results in fig.8 shows that there are 2 accuracies of 2m and 2.5m are in close proximity to 1m vertical accuracy.

4.4.2 NN query on feature-independent model

The feature-independent query model uses SDO_NN operator directly to find accuracies adjacent to 2.5m vertical accuracy.

Query 4.2

```
SELECT  a.vacc, b.geom
FROM     fiq b,  fiq  a
WHERE   b.vacc = 2.5 AND
            SDO_NN(a.geom, b.geom) = 'TRUE'
            AND ROWNUM <= 2;
```

Results:

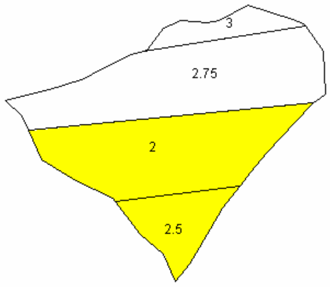

Fig 9: Results of query 4.2

The results of the query 4.2 retrieve 2.5 m and 2m accuracies.

4.4.3 NN query on feature-hybrid model

Unlike the other models on the NN query, the feature-hybrid model need not store an additional table. Thus, there is no need to create a spatial index as the fhq geometries are already spatially indexed. The SDO_NN query operator is directly used to retrieve the neighbors.

Query 4.3

```
SELECT  a.vacc, b.geom
FROM     fhq b,  fhq  a
WHERE   b.vacc = 2.5 AND
            SDO_NN(a.geom, b.geom) = 'TRUE'
            AND ROWNUM <= 2;
```

Results:

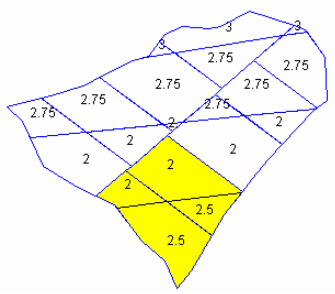

Fig 10: Results of query 4.3

The NN query on the spatial variation models retrieves the adjacent accuracies of the region requested. The per-feature model requires the query to execute in two steps due to index creation when compared to other two models.

4.5 Complex spatial join query (Query 5)

"List the feature ids and vertical accuracies for parcels where the recorded per-feature and feature-independent vertical accuracy differ by 0.25 m?"

The complex spatial query involves a spatial join between different spatial variation models. The complex query is developed to retrieve the vertical accuracies with a difference of 0.25m in per-feature and feature-independent models. First a natural join is performed on base $\bowtie_{\text{base.fid = pfq.qid}}$ pfq. Secondly, a spatial join is performed between the geometries of relation base $\bowtie_{\text{base.fid = pfq.qid}}$ pfq with geometries of fiq. Lastly, the vertical accuracies are selected from the results of the spatial join, where the accuracies differ by 0.25m

Query 5

```
SELECT   a.fid  parcelid, a.vacc pfqvacc,
             b.vacc  fiqvacc
FROM       (SELECT  i.fid,i.geometry, j.vacc
              FROM      base i, pfq j
              WHERE   i.fid = j.qid ) a, fiq b
WHERE    a.geom ∩ b.geom
              AND b.vacc - a.vacc = 0.5;
```

Results:

```
PARCELID  PFQVACC  FIQVACC
--------------  -------------  -------------
7               2              2.5
4               2.5            3
```

The results of the complex spatial query enable comparison between the quality stored in per-feature and feature-independent model.

Other queries such as, non-spatial range and spatial range queries (buffer), which are not discussed in the query section, were implemented for all the three models. The non-spatial range query's performance was similar to that of non-spatial point query of section 4.1. The buffer query implementation reveals that the query pattern of region queries of section 4.3 was similar to that of buffer query.

5. SUMMARY AND CONCLUSIONS

This paper discusses the need to represent spatial variation in data quality in a spatial database. Three models based on object and field based approach were identified: per-feature , feature-independent, and feature hybrid. The per-feature model is easiest to implement, simply storing quality as an additional attribute. However, the model is unable to represent sub-feature variation.

The feature-independent model represents quality information independent of its spatial features. As a result, the feature-independent model can represent sub-feature variation. As the quality has geometries associated with it, additional storage and increase in processing time during querying affects the model's efficiency.

To achieve a balance of efficiency in storage and retrieval, the feature-hybrid model is derived from a combination of per-feature and feature-independent quality models. Both non-spatial and spatial joins may be required in querying this model.

Query	Per-feature			Feature-independent			Feature-hybrid		
	NJ	SJ	SF	NJ	SJ	SF	NJ	SJ	SF
Query 1	1	0	N	0	1	Y	1	0	Y
Query 2	1	1	N	0	2	N	1	1	N
Query 3	1	1	N	0	2	Y	1	1	Y
Query 4	additional table		N	no additional table		Y	no additional table		Y

Table5: Comparison table of query execution between the spatial variation models. NJ-Natural join, SJ-Spatial join, SF-Sub-feature variation, Y-Yes, N-No.

Table 5 summarized the key operations involved in the different queries across the three models. Based on these comparisons and analysis of the different models, the following conclusions are made:

a) From table 5, queries 1, 2 and 3 for the per-feature and feature-hybrid models have a similar pattern on performing the joins.

b) The per-feature and feature-hybrid query implementations use a greater number of natural joins than spatial joins in table 5. Natural joins (joins based on attributes) are computationally much less expensive when compared to spatial joins (joins based on geometries). Thus, the computation cost is less expensive for both the models as they use natural joins.

c) The feature-independent model query implementation on queries 1, 2 and 3 in table 5, uses only spatial joins to retrieve quality. Queries 2 and 3 have two spatial joins as the model retrieves quality based on geometries, making the models computation cost higher on retrieval when compared to the other two model's query performance.

d) The commonality between feature-independent and feature-hybrid model are that they store independent quality geometries and have the ability to retrieve sub-feature variation quality.

In general, each model of spatial variation is different in its representational and querying capabilities. However, no model is entirely superior in storing and retrieving spatially varying quality. Hence, we are developing an integrated approach, which combines per-feature, feature-independent, and feature-hybrid quality models. By developing an integrated approach, the user can retrieve quality irrespective of which data model is used to store it.

5.1 Further work

Current work is using real data from the Victorian Department of Sustainability and Environment (DSE). Hume, a Local Government Area (LGA) has prominent variation in SDQ, with precision ranging from 0.1m to 20m among other LGAs. The data also has sub-feature variation in quality (see fig 11). The models of spatial variation discussed have considered the sub-feature variation for retrieving and storing data quality. The point (quality) data of the Hume LGA has been used in the developed integrated model.

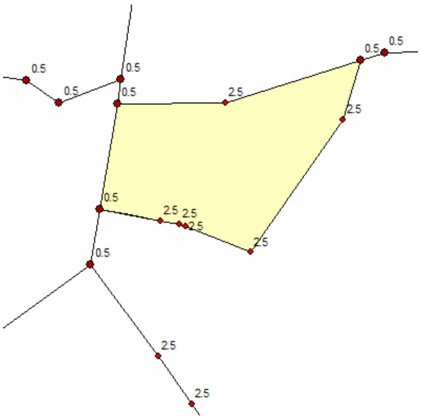

Fig 11: The parcel boundary depicting sub-feature variation with precision ranging 0.5m to 2.5m, obtained from the Vicmap Property data from DSE, for Hume LGA.

6. REFERENCES

Adam, N. R., Gangopadhyay, A., 1998. Spatial queries. In: *Database Issues in Geographical Information Systems*. Kluwer academic publishers Boston/ Dordrecht/ London, pp.93–112.

Chrisman, N., 2006. Development in the treatment of spatial data quality. In: Devillers, R., Jeansoulin, R. (Eds.), *Fundamentals of spatial data quality*. ISTE Ltd, Ch. 1, pp. 21–30.

Devillers, R., B´edard, Y., Jeansoulin, R., 2005. Multidimensional management of geospatial data quality information for its dynamic use within gis. Photogrammetric Engineering & Remote Sensing 71 No. 2, pp. 205–215.

Duckham, M., 2001. Object calculus and the object-oriented analysis and design of an error sensitive GIS. *Geoinformatica* 5:3, pp. 261–289.

Frank, A. U., 2004. Foreword. In: Frank, A. U., Grum, E. (Eds.), Proceedings of the ISSDQ '04. Vol. 1. GeoInfo Series Vienna.

Goodchild, M. F., 2006. Foreword. In: Devillers, R., Jeansoulin, R. (Eds.), *Fundamentals of spatial data quality*. ISTE Ltd.

Heuvelink, G. B. M., 1996. Identification of field attribute error under different models of spatial variation. *International*

Journal Geographical Information Systems 10. No.8, pp. 921–935.

Hunter, G. J., 2001. Spatial data quality revisited. In: Bauzer-Medeiros, C. (Ed.), Proceedings of the 3rd Brazilian Geo-Information Workshop (Geo 2001), Rio de Janeiro, Brazil. pp. 1–7.

Hunter, G. J., Hope, S., Sadiq, Z., Boin, A., Marinelli, M., Kealy, A., Duckham, M., Corner, R. J., 2005. Next-Generation Research Issues in Spatial Data Quality. In: Proceedings of SSC 2005 Spatial Intelligence, Innovation and Praxis: The national biennial Conference of the Spatial Sciences Institute, September, 2005. Melbourne: Spatial Sciences Institute. pp. 865–872.

Hunter, G. J., Qiu, J., 2003. Automatic updating of spatial data quality information. In: Proceedings of the 2nd International Symposium on Spatial Data Quality ’03. Advanced Research Centre for Spatial Information Technology,The Hong Kong Polytechnic University, Hong Kong, pp. 210 – 214.

Maclean, A. L., D’Aveilo, Thomas, P., Shetron, G. S., 1993. Use of variability diagrams to improve the interpretation of digital soil maps in a gis. Photogrammetric Engineering and Remote Sensing 59 (2), 223–228.

Qiu, J., 2002. Managing spatial data quality information. Ph.D. thesis, University of Melbourne.

Qiu, J., Hunter, G. J., 2002. A gis with the capacity for managing data quality information. In: Shi, W., Fisher, P. F., Goodchild, M. F. (Eds.), *Spatial Data Quality*. Taylor & Francis, pp. 230–250.

Ramm, P., 2005. A question of accuracy. Position, 80–81.

Sadiq, Z., Duckham, M., Hunter, G., 2006. Modeling spatial variation in data quality using linear referencing. In: Caetano, M., Painho, M. (Eds.), 7th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences, Lisbon. pp. 225–235.

Shekhar, S., Chawla, S., 2003. Introduction to spatial databases. In: *Spatial Databases A Tour*. Prentice Hall, pp. 1–21.

United States Department of the Interior Bureau of Land Management (US DoI), 2001. Gcdb reliability diagram. Tech. rep., Land & Resources Project Office.

URL http://www.blm.gov/gcdb/Standards/reliability diagramdescr.pdf

Worboys, M., Duckham, M., 2004. *GIS: A computing perspective*. 2nd Edition. Boca Raton, FL: CRC press.

7. ACKNOWLEDGEMENTS

This work has been supported by the Cooperative Research Centre for Spatial Information, whose activities are funded by the Australian Commonwealth's Cooperative Research Centers Program.

APPENDIX A. EXPLANATION OF SYMBOLS

Symbol	Description
O	set of all geographic objects
Q	set of all quality objects
S	spatial framework (spatial point set)
$\bowtie$	relational join operation
$\cap$	spatial intersection operation
SDO_NN	spatial nearest neighborhood operation
pf	per-feature database table
fi	feature-independent database table
fq	feature-hybrid database table
pfq	per-feature quality database table
fiq	feature-independent quality database table
fhq	feature-hybrid quality database table
pfqn	per-feature quality database table