Supporting Quality-Based Image Retrieval Through User Preference Learning

Giorgos Mountrakis, Anthony Stefanidis, Isolde Schlaisich, and Peggy Agouris

Abstract

It is common for modern geospatial libraries to contain multiple datasets that cover the same area but differ only in some specific quality attributes (e.g., resolution and precision). This is affecting the concept of content-based geospatial queries, as simple coverage-based query mechanisms (e.g., declaring a specific area of interest) as well as theme-based query mechanisms (e.g., requesting a black and white aerial photo or multispectral satellite imagery) are rendered inadequate to identify and access specific datasets in such collections. In this paper we introduce a novel approach to handle data quality attributes in geospatial queries. Our approach is characterized by the ability to model and learn user preferences, thus establishing user profiles that allow us to customize image queries for improving their functionality in a constantly diversifying geospatial user community.

Introduction

GIS data in digital format are collected and stored at constantly increasing rates. Furthermore, the GIS user community is also expanding, with users of various levels of expertise aiming to access and use this information. Considering geospatial imagery in particular, it is now common for geospatial libraries to contain multiple datasets that cover the same area but differ only in some specific quality attributes (e.g., resolution and precision). This evolution is affecting the concept of content-based geospatial queries, as simple coverage-based (e.g., declaring a specific area of interest) and theme-based (e.g., requesting a black and white aerial photo or multispectral satellite imagery) query mechanisms are rendered inadequate to identify and access specific datasets in such collections. Thus, we find ourselves in need of query solutions to support complex content-based queries where the term *content* refers (beyond spatial and temporal coordinates) to quality attributes (e.g., resolution and precision) of geospatial datasets. In this paper we address this issue by introducing a novel approach to geospatial queries that takes into account quality attributes. Data quality in this case is used as an umbrella term. Concepts that play into our definition of data quality are, among others: accuracy, error, and precision (Buttenfield, 1993).

In typical image database queries users provide an *ideal* set of quality attributes that best satisfies their needs, and the database is searched to identify the dataset that best resembles the user request in terms of these attributes. This involves a comparison of attribute values of each dataset to the user-defined ideal, and the ranking of the datasets. The degree to which a dataset meets the specifications requested by a user to address a specific task at hand is termed *fitness of use*.

We keep this well-established approach, but modify the mechanism that compares quality attributes among datasets, to accomodate the fact that user preferences are much more complex than traditional nearest neighbor approaches. Indeed, common solutions make use of standard distance metrics (for example, nearest neighbor) that are rather simplistic (e.g., linear) and ignore the fact that attribute variations may have different importance to different types of users. For example, scale variations may affect dataset fitness differently for a geologist as compared to a cartographer. Current solutions ignore these particularities of different user communities, using the same distance function to evaluate datasets regardless of a user's preferences. To overcome this shortcoming we introduce a novel approach that allows users to express their own preferences, models these preferences in complex distance metric functions, and uses these functions to evaluate dataset similarities for each request. Our approach could be used to model various attributes of a geospatial dataset; it is particularly suitable for *quality* attributes like the above mentioned ones, as they are the ones that convey inherently application-dependent preferences that cannot be satisfied by existing dataset similarity metrics.

Essentially, our approach leads to the establishment of *user profiles* in geospatial queries, expressing the manner in which attribute variations affect a user's preferences. These profiles may be user dependent, application dependent, or both. A user profile is a set of parameters, describing the relationship between attribute variations and user preferences in a mathematical manner. We proceed by first defining, then training, and finally using these profiles in the query process. To accomplish our goal we introduce a novel user-adaptive learning algorithm that iteratively learns user preference regarding data quality. Within the process we assume that users have previously performed a filtering in non-quality attributes like space and time. Thus, our approach complements existing spatio-temporal queries, supporting the additional evaluation of quality attributes during the query process. It should also be mentioned that even though our motivation stems from geospatial image queries, the approach presented here could be applicable to other types of geospatial datasets as well.

The paper is organized as follows:

- Related Work: An overview of literature relevant to our work,
- Data Quality Attributes: A discussion of the data quality attributes that are used to convey image data quality,

Department of Spatial Information Science & Engineering, and National Center for Geographic Information and Analysis (NCGIA), University of Maine, 348 Boardman Hall, Orono, ME 04469-5711 (giorgos@spatial.maine.edu; tony@spatial.maine.edu; isolde@spatial.maine.edu; peggy@spatial.maine.edu).

Photogrammetric Engineering & Remote Sensing
Vol. 70, No. 8, August 2004, pp. 973–981.

0099-1112/04/7008–0973/$3.00/0

- Learning User Preference: A description of our learning algorithm,
- Training and Simulation: A demonstration of our training and simulation environment. We conclude with a discussion on the benefits of our approach and future work directions.

Related Work

During the last two decades the geospatial user community is becoming increasingly aware of the significance of modeling and communicating data quality information (McGranaghan, 1993; Buttenfield and Beard, 1994; Beard, 1997). The USGS identified data quality communication as an important issue and devoted a whole section to data quality in the U.S. Spatial Data Transfer Standard (STDS) (Fegeas *et al.*, 1992).

Discussion of Terminology

As mentioned above, the term *quality* is used in the context of this paper as an umbrella-term, embracing all aspects of the problem (Buttenfield and Beard, 1991; Buttenfield, 1993; Davis and Keller, 1997; Evans, 1997; Beard, 1997; Veregin, 1999).

The parameters that are most commonly used to express data quality include *error*, which is used to describe the difference between the true value and the value that is stored in the database (Hunter and Goodchild, 1995). *Accuracy* expresses the degree to which data conform to truth (Buttenfield, 1993), and, specifically, *spatial accuracy* refers to the accuracy of the spatial component of data (Veregin, 1999). *Precision* denotes the exactness with which a measurement is made (MacEachren, 1992). *Reliability* can be defined as the level of confidence a data provider has that the data are correct (Evans, 1997). *Uncertainty* describes a situation where the resolution of the data does not allow a user to make a confident decision about the content of the data. For example, pixels in remotely sensed images might contain uncertain information because of sub-pixel mixing or sensor sampling bias (Bastin *et al.*, 2002). Thus uncertainty can be viewed as the aggregate effect of several other concepts that are combined under the term of data quality, such as error, vagueness, and ambiguity (Fisher, 1999). Unwin (1995) introduced the concept of error-sensitive GIS, describing a GIS that can handle the uncertainty that is inherent in geographic information.

Within the context of our application, the aggregation of quality attributes is expressing a novel type of uncertainty: *uncertainty of fitness of a dataset to a user request*. This is the type of uncertainty we handle in this paper.

Learning Algorithms

In order to customize our results based on user preference we had to follow a learning approach, a widely used Artificial Intelligence (AI) technique. Applications of AI concepts in GIS include efforts to identify relationships between GIS objects (Walker and Moore, 1988), perform feature extraction from DEM (Bennet and Armstrong, 1996), support texture matching in aerial photographs (Ma and Manjunath, 1998) and perform road extraction from multispectral imagery (Doucette *et al.*, 2001).

Specifically, in learning spatial data quality information there are a few approaches concentrating on capture and production (Campbell *et al.*, 1994; Goodchild, 1999; Duckham *et al.*, 2000), but so far there is no technique that addresses the communication of spatial data quality information between user and computer through a learning algorithm. The current method employed is the Nearest Neighbor (Cover and Hart, 1967). Classification of the results' relevance is performed by storing examples as points in the feature space and then using a distance metric to measure correlation to these points (Aha *et al.*, 1991; Cost and Salzberg, 1993; Wilson and Martinez, 2000). These methods do not provide the flexibility necessary for advanced user customization of the results. On the other hand there is significant work done in developing learning systems for image retrieval. Some examples include applications of complex non-linear systems using neural networks (Mandl, 2000; Lim *et al.*, 2001; Müller *et al.*, 2001; Carkacioglu and Fatos, 2002). However, none of these approaches takes into consideration the specific characteristics of data quality attributes and types of geospatial user preference, therefore their applicability is limited. This is a gap that we attempt to fill with our approach.

Data Quality Attributes as Input for Our Learning Process

As mentioned above, we use quality attributes to determine the fitness of a dataset to a user request. We assume that such attributes are available for image datasets in the form of metadata. To decide which attributes we use to convey the quality of image data, we started from the data quality attributes that are specified in the data quality section of the SDTS which are: positional accuracy, attribute accuracy, completeness, logical consistency, and lineage. These attributes apply to spatial data in general. Since we are concentrating on the quality of spatial imagery, we chose a subset of the STDS quality attributes and added some more that we feel are important for quality representation geared towards image data. Note that even though we identify the following attributes for data quality assessment, we want to emphasize that the selection of quality attributes is neither exhaustive nor unique; it is rather used as a demonstration example for our learning process. Our approach does not depend on the selection of a specific set of attributes; it can be applicable to any selected set of such attributes.

It is worth mentioning here that we chose not to include lineage, logical consistency and attribute accuracy measures for the following reasons: Lineage information is not used because, according to the SDTS, lineage information is to "include a description of the source material from which the data were derived and the methods of derivation, including all transformations involved in producing the final digital files". Image data are not subjected to as much transformation as other types of spatial data though, and therefore we feel that it is not a top priority to include lineage in our quality analysis. Inconsistency is a big problem for spatial databases that include a variety of data, but when we work with single images, consistency within the image is a given. Therefore, we are not including logical consistency. We are also not addressing attribute accuracy, as we assume that such information is available without error in image datasets.

The attributes we use to describe data quality are positional accuracy, spatial completeness, resolution, accessibility, and cost. Following, you may find a more detailed escription of our selected attributes:

Positional Accuracy

For spatial data this attribute describes how much the data deviates from the ground truth (e.g., ± 5 cm). In image data we perceive as positional accuracy of an image the digression from ground truth due to an aggregation of sensor quality, calibration, measuring errors, and orientation modeling accuracy.

Spatial Completeness

In our case spatial completeness describes how much of the image contains useful data. Satellite images and aerial photographs can be partially obstructed by cloud coverage. These will be listed as incomplete. The range of completeness is given as the percentage of un-obstructed parts of the image.

Resolution

For image data resolution is an important quality measure. It describes how much ground is covered by one pixel. Common sensors of satellite images for instance can have a resolution of 0.8 m (QuickBird) to over 50 m (Landsat). For spatial data in general, scale is another indicator of fitness for use. Nevertheless, we chose not to include information about scale as a quality measure, because in digital raster images, the resolution is the deciding factor of quality. In a digital image one can zoom in and out, seemingly changing the scale, without changing the quality of the data. The resolution in which a picture is taken, however, certainly affects the quality of the information.

Accessibility

This measure is a combination of image size and connection speed of the server. In other words it translates into the amount of time it takes to download the data. For raster representations such as images, due to their potentially large file size, the amount of download time is an important piece of quality information.

Cost

Data might be free to download, but sometimes one has to pay a fee for accessing spatial images. Both the accessibility and the cost of the data can influence whether the user is able to use the data or not and is therefore playing a role in the fitness of use of the data. Consequently, we add them as a factor of data quality.

Learning User Preference On Image Quality

In order to provide high classification accuracy to user queries, we have created a multi-dimensional, data quality, preference-learning algorithm. In the following segments of this section we will provide a description of our approach supported by a detailed mathematical definition. After the specifics are set, we present our learning method in detail. Finally, we display the overall information flow within the algorithm.

Approach Overview

In a typical image database query process (Figure 1) users describe an *ideal* set of attributes for their application. This set is then compared to the corresponding set of every image (or some images if a filtering takes place) in the database. A ranking is created based on which image is *closer* to the original request. Then, the best result(s) is (are) displayed to the user for further evaluation.

In the above process there is an important piece missing. The ranking of results is not adjustable to specific users or applications. The *comparison* metrics used to provide the ranking are not user-adaptable, rather they remain the same.

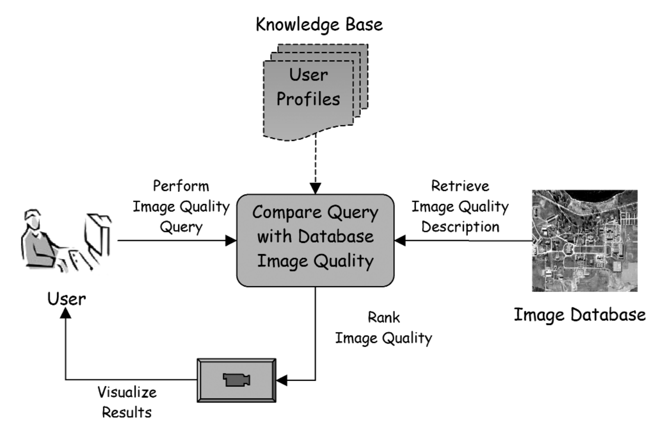

Figure 1. Process flow within Image Quality Queries.

To improve this we introduce another component in the query process for image retrieval, the *User Profiles* (Figure 1). These profiles express how user preference in data quality varies in the anticipated results of a query. They are used in every query to rank the results. Using such profiles is not a new idea in the database field. For example, in (Mitaim and Kosko, 1997) a neuro-fuzzy approach with agent profiles using *if-then* rules was proposed to optimize the query process. Each profile is composed of a set of parameters describing a mathematical relation. This set is trained over a preference sample that the user has provided. Once the training is complete, the variables of the relation are adjusted to user preference.

In order to get an insight into what a user profile should contain, we first examine how image quality is represented when a query is accessing a database. A set of attributes is chosen and a value is inserted in every field. For query purposes this information is presented in a multi-dimensional vector:

$$X_{Source} = \left[X^1_{Source}, X^2_{Source}, \ldots, X^n_{Source}\right]. \tag{1}$$

The dimensionality of the vector (value n) corresponds to the number of attributes that are chosen. An example of a data quality vector would be:

$$X_{Source} = [Positional\_Accuracy, Resolution, \ldots, Spatial\_Completeness]. \tag{2}$$

In every query request, the users would create a corresponding vector describing their *ideal* values of the data quality attributes. Formally, this would be expressed as:

$$X_{User} = \left[X^1_{User}, X^2_{User}, \ldots, X^n_{User}\right]. \tag{3}$$

For the same example as above, this would translate into a vector like:

$$X_{User} = [Position\_Accuracy_{Desired}, Resolution_{Desired}, \ldots, Spatial\_Completeness_{Desired}]. \tag{4}$$

Our goal is to find a mathematical path to compare the user request vector (X_{User}) and the corresponding image data quality vector (X_{Source}) and produce a scalar result ($DQ_{Preference}$). This scalar expresses how similar the request and the image source are in terms of data quality. By using a standard metric (e.g., distance metrics) to compare and rank multi-dimensional indexes, existing solutions fail to capture the particular needs of users. In order to have this preference scalar adapt to different users and applications, we introduce a learning algorithm. This allows us to overcome a significant drawback of existing data quality estimators, namely their lack of adaptability to user preference.

From the learning perspective we attempt to model a function $P(\bullet)$ that estimates $DQ_{Preference}$. As inputs of the process, we have the two vectors, $\overrightarrow{X}_{User}$ and $\overrightarrow{X}_{Source}$. Formally, this corresponds to:

$$DQ_{Preference} = P[\overrightarrow{X}_{Source}, \overrightarrow{X}_{User}]. \tag{5}$$

To find function $P(\bullet)$ we decompose it to a subset of functions that perform certain tasks. This is a traditional methodology employed in database query matching techniques (Lim *et al.*, 2001). The decomposition is based on the specific elements of each vector. A preference function is assigned separately for each dimension of the vector. Then, the preference results from each dimension are combined into a total metric, the $DQ_{Preference}$. If we define function $F^i(\bullet)$ as the preference function for dimension i, and $G(\bullet)$ as the aggregation function, then function $P(\bullet)$ can be substituted by:

$$DQ_{Preference} = G\left[F^1\left(X^1_{User}, X^1_{Source}\right), F^2\left(X^2_{User}, X^2_{Source}\right), \ldots, F^n\left(X^n_{User}, X^n_{Source}\right)\right]. \tag{6}$$

For the example of Equation 4 this would translate into:

$$DQ_{Preference} = G\begin{bmatrix} F^1(Positional\_Accuracy_{Desired}, Positional\_Accuracy), \\ F^2(Resolution_{Desired}, Resolution), \ldots, \\ F^n(Spatial\_Completeness_{Desired}, Spatial\_Completeness) \end{bmatrix}. \quad (7)$$

Function $F^1(\bullet)$ expresses the degree of satisfaction in the *Positional_Accuracy* dimension, function $F^2(\bullet)$ expresses the *Resolution* preference, and so on. Function $G(\bullet)$ accumulates the results of these functions to the $DQ_{Preference}$ metric. The corresponding *User Profile* to this example would be the values of the parameters required to define functions $F^i(\bullet)$ and $G(\bullet)$.

Preference Calculation Within Dimensions

In order to provide an estimation of user preference in every dimension of the *DQ* vector, we make use of a variety of fuzzy membership functions. The original idea was introduced in Mountrakis and Agouris (2003), and in this section, we expand it to address data quality preferences.

To explain our approach in detail, let X^i_{User} represent the query request, and X^i_{Source} be the corresponding quality metric in a specific dimension of a stored image in the database. The objective is to find function $F^i(\bullet)$ that expresses user preference for a given pair $[X^i_{User}, X^i_{Source}]$. Mathematically,

$$F^i: \Re^2 \rightarrow [0,1]. \quad (8)$$

Function Characteristics

After investigating typical dimensions used to describe data quality, we identified three important characteristics for function $F^i(\bullet)$; characteristics that are not currently incorporated in existing query mechanisms that follow the popular Nearest Neighbor method and its variants:

Asymmetry

Asymmetry translates into possibly different preference values in symmetrical candidates around the desired value. For example, for a positional accuracy request of 1 meter, symmetrical returns of 0.5 and 1.5 meters will not result in the same user preference, respectively, as a dataset of 0.5 meter resolution satisfies the requested 1 meter accuracy, while 1.5 m falls short of the request. A visualization of the differences in establishing similarity metrics between existing nearest neighbor solutions (top) and our proposed approach (bottom) can be seen in Figure 2.

Non-Linearity

Regardless of symmetry, the underlying user preference may not follow a linear function (as dictated by nearest neighbor solutions). In the same scenario as above for a request of 1 meter positional accuracy, assume returns of 2 meters and 4 meters positional accuracy metrics from two candidate sources. Also, assume that there might not be the same linear preference decrease of interest due to, for example, application accuracy constraints. These requests could lead to a disproportional lower preference for the 4 meter source than the 2 meter one. A linear function could not model this sufficiently; therefore, we propose the use of non-linear functions, as is the case in Figure 2 (bottom).

Value Compensation

Another assumption based on existing techniques is that the calculated preference is only dependent on a distance metric. In other words, the difference (when discussing a one-dimensional attribute) between the query $[X^i_{User}]$ and the candidate $[X^i_{Source}]$ value is the only input necessary to calculate the corresponding preference. In data quality attributes, preference might not be solely dependent on the distance between these two values, but the actual values as well.

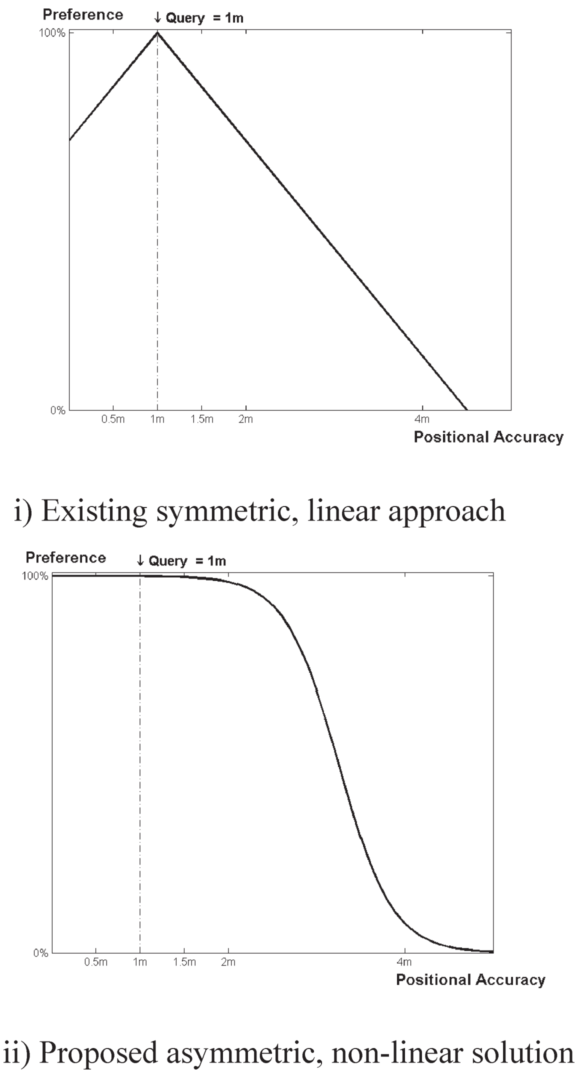

Figure 2. Asymmetry and non-linearity example.

For example, let us expand on the previous discussion: a request of 1 meter positional accuracy. We add two more requests, one for 20 meters and another for 50 meters. Assume now three candidate sources of 6 meters, 25 meters and 55 meters, each one attempting to satisfy one of the above requests. User preference could return the following results shown in Table 1.

A quick observation on the values of Table 1 is that even though the distance ($X^i_{User} - X^i_{Source}$) remains the same throughout the examples (5 meters), user preference varies significantly. A typical way to overcome this would be to normalize the difference based on the X^i_{User} value. In complex

Table 1. User Preference Example

X^1_{User}	X^1_{Source}	User Preference
1 m	6 m	20%
20 m	25 m	65%
50 m	55 m	80%

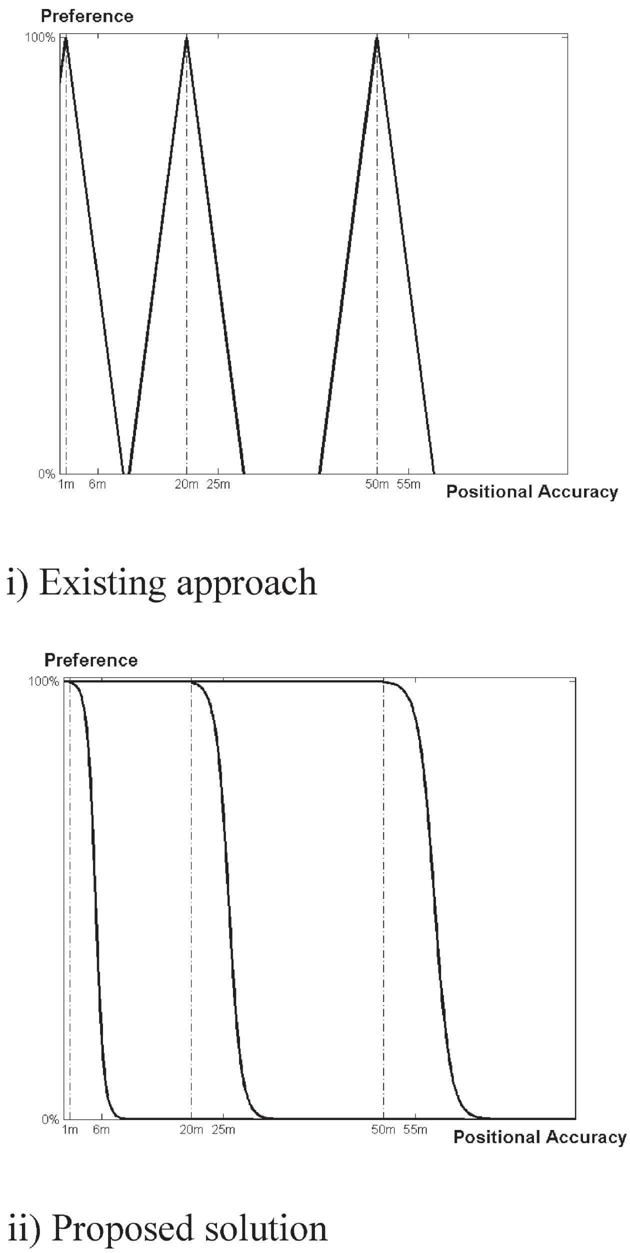

Figure 3. Value compensation example.

preference as in the example of Table 1, no (linear) normalization could address the issue. Therefore, the preference function $F^i(\bullet)$ should be able to support non-linear dependency on the actual values of $[X^i_{User}, X^i_{Source}]$ and not just their distance. An example of the existing and proposed functions is shown in Figure 3.

Function Example

In Mountrakis and Agouris (2003) a learning system was proposed composed of fuzzy functions of adaptable complexity. Initially, linear functions are interpolated to get an estimate of the underlying complexity. If the desired accuracy is not achieved, they are replaced by more sophisticated non-linear, sigmoidal functions. In this paper we manipulate these functions accordingly to capture the characteristics of the dimensions we are addressing. The collection and justification of the chosen functions, as well as an intelligent learning technique, where less complex functions act as approximations for more complex ones to follow can be found in Mountrakis and Agouris (2003). Here, we demonstrate a representative example of their functionality on data quality attributes.

So, investigating user preference in the dimension of positional accuracy of the data quality vector reveals important aspects of the anticipated preference behavior described in the previous section. After examining these characteristics, user preference is represented in Equation 9.

$$F^{Pos.Acc.}(X_{Source}, X_{User}) = \left\{ \begin{array}{l} \dfrac{1}{1+e^{-a_R(X_R)}} \quad if\ X_{User} \le X_{Source} \\ a_R = k_1 + \dfrac{(X_{User})}{k_2} \\ X_R = (X_{User} - X_{Source} - c_R)\cos\varphi_R \\ \qquad + (X_{User} + X_{Source} + c_R)\sin\varphi_R \\ c_R = k_3 + \dfrac{(X_{User})}{k_4} \\ \dfrac{1}{1+e^{-a_L(X_L)}} \quad if\ X_{User} > X_{Source} \\ X_L = (X_{User} - X_{Source} - c_L)\cos\varphi_L \\ \qquad + (X_{User} + X_{Source} + c_L)\sin\varphi_L \\ c_L = k_5 + \dfrac{(X_{User})}{k_6} \end{array} \right\}. \tag{9}$$

The input to the function is the $[X_{User}, X_{Source}]$ pair and the parameters for the system to learn are $[k_1, \ldots, k_6, a_L, \varphi_L, \varphi_R]$. Note that the function is composed of two sub-functions, each one applicable in half of the input space (e.g., $X_{User} > X_{Source}$) to compensate for asymmetrical cases. A major factor for choosing a sigmoidal function comes from its superior modeling capabilities. The parameters c_R and c_L specify the translation along the X_{Source}-axis, which is especially useful in specifying the highly active portion of the function (close to 100 percent). The slope of each sigmoidal function is expressed through a_R and a_L, respectively. Efficient manipulation of the slope can result in representing a variety of cases, ranging from a linear relationship up to a step-like behavior. A result of this trained function can be seen in Figure 4.

In Figure 5 we have the corresponding contour plot of Figure 4. Also included are two sections for specific user

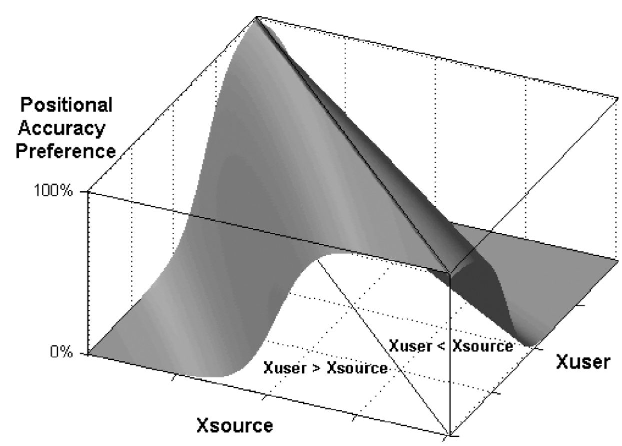

Figure 4. Training example of a positional accuracy preference function.

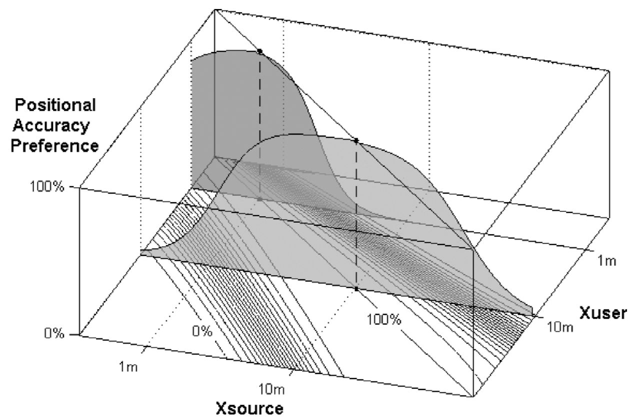

Figure 5. Contour plot and query examples of positional accuracy preference function.

requests for positional accuracy of 1 meter and 10 meters. By examining these two sections, we can conclude the following:

1. In the $X_{User} > X_{Source}$, half the dependency of shift (c_L) on the X_{User} input is able to express the gradual decrease of user's interest because the returned positional accuracy is better (smaller) than that requested. Analyzing the reasons of such a preference pattern is beyond the scope of this paper, however, it is easily understood that there exist numerous reasons for such patterns (e.g., high acquisition cost for better positional accuracy). Note in Figure 5 that user flexibility increases as the positional accuracy request X_{User} gets larger. No normalization could encapsulate this dependency, and
2. at the right half, more complicated modeling is necessary. A dependency on the X_{User} input exists for both slope (a_R) and shift (c_R). This powerful combination expresses user decrease of interest when the returned accuracy is worse (larger) than the requested one. It is dependent on the X_{User} input since the larger the requested value is the larger the range of highly acceptable values increases (through shift manipulation), and the interest to decrease rate is smaller (done through slope modification). In other words, when users request 1 meter positional accuracy, they are less flexible in accepting similar results than when querying for a 10 meter request, and this is what is expressed with the displayed function.

Aggregation of Preference Results into a Single Metric

After the preference is calculated for every dimension, the next step is to combine the preference results into one metric. This metric represents the overall correlation between the user request and the images of the database in terms of quality.

We chose to represent the global metric as a weighted summation of the preference results from every dimension. This method is fast, and, most importantly, can be solved using a linear, least-squares solution. The mathematical expression of this aggregation function would correspond to function $G(\bullet)$ presented as Equation 10:

$$DQ_{Preference} = G\left[F^1\left(X^1_{User}, X^1_{Source}\right), F^2\left(X^2_{User}, X^2_{Source}\right), \ldots, F^n\left(X^n_{User}, X^n_{Source}\right)\right]. \quad (10)$$

After substituting function $G(\bullet)$ with a weighted summation results in:

$$DQ_{Preference} = W_1F^1\left(X^1_{User,} X^1_{Source}\right) + W_2F^2\left(X^2_{User}, X^2_{Source}\right) + \cdots + W_nF^n\left(X^n_{User}, X^n_{Source}\right), \quad (11)$$

where $\sum_1^{i=n} [W_i]=1$. Or, more concisely: $DQ_{Preference} = \sum_1^{i=n}[W_iF^i(X^i_{User}, X^i_{Source})]$.

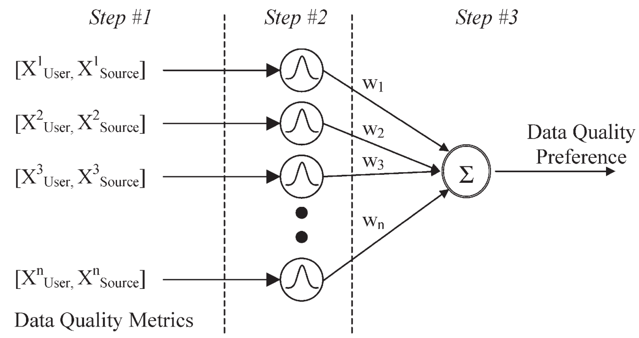

Figure 6. Data Quality Learning System.

The proposed aggregation function can model user preference as to each dimension's contribution (importance) to the overall solution. But, it neither captures dependencies between dimensions nor adjusts the weights based on the actual dimension values. Tests performed using a quadratic function to compensate for dependencies between dimensions have added a significant computational complexity without a proven gain. Therefore, we assume independence between dimensions, but if users are aware of dependencies, we support their incorporation by giving the users the option to manually adjust the non-diagonal elements of the covariance matrix of the quadratic function. Adjustment of the weights based on the corresponding actual dimension values is reserved for future work.

Algorithm Design

The design of our learning algorithm for data quality preference is presented in Figure 6. Three steps compose the process:

Step Number One
The multi-dimensional vectors X_{User} and X_{Source} are paired together separately in every dimension. This way we create n two-dimensional inputs, where n is the number of dimensions. Each two-dimensional input has the user-desired value, and the available source image value in the same corresponding dimension.

Step Number Two
At this stage we calculate preference within each quality dimension. To achieve that we make use of the functions $F^i(\bullet)$ described earlier.

Step Number Three
In the final step, we combine the preference results from each dimension into one total data quality metric. This is performed by using aggregation function $G(\bullet)$.

Training and Simulation Environment

Training Process

A major concern when training a complex non-linear system such as ours is how convergence can be achieved often and training can be done fast and accurately. To accomplish this we kept the training process in mind during the design of the algorithm. Our training is performed in multiple stages to break up complexity. First, we train the algorithm to understand user preference within each dimension and then combine dimensions into a single metric based on their relevant weight. Here we should mention that due to the chosen modular design of our learning algorithm not all functions have to be trained. Their expressiveness and specific functionality allow manual

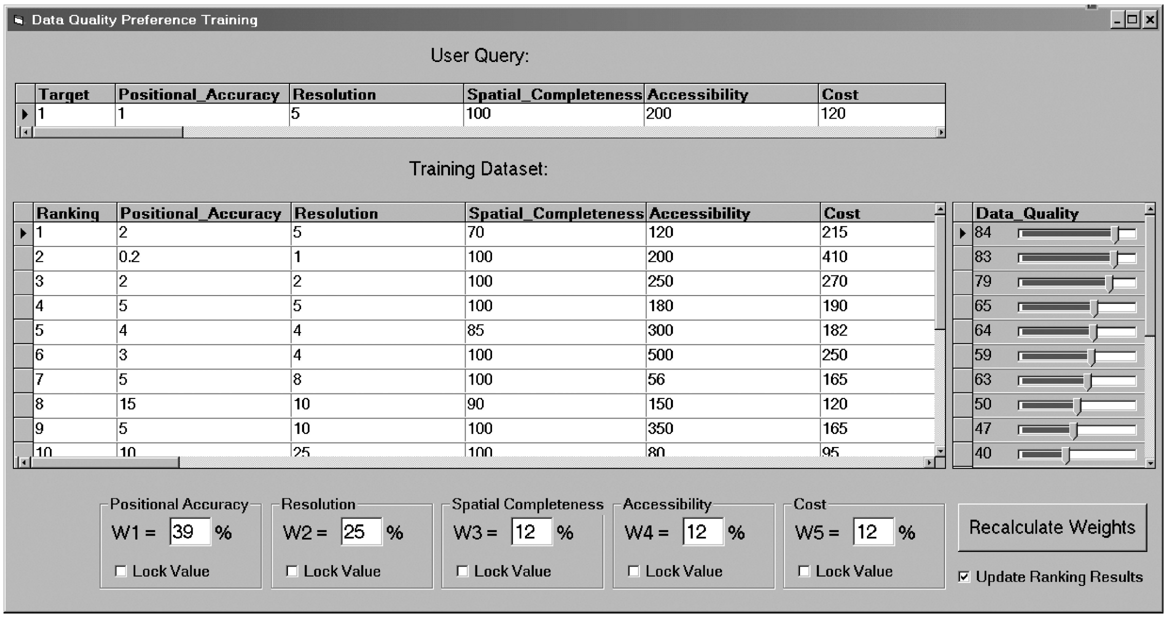

Figure 7. Training environment.

(external) specification of any of the $F(\bullet)$, $G(\bullet)$ functions and some or all of their parameters. This could speed up the learning process and allows users with different levels of expertise to use the learning algorithm to its best potential.

Preference Training Within Each Dimension—Functions *F*( )

In the first stage preference learning takes place within each dimension separately. Several pairs of $[X_{User}, X_{Source}]$ values for a specific dimension are presented to the user, and a preference percentage is requested. This is performed in every dimension independently. The training itself takes place using the dynamically adapted methodology of (Mountrakis and Agouris, 2003). Initially the user is presented with a relatively small set of training samples. A fast linear solution is obtained through the interpolation of a number of planes. The algorithm can decide automatically whether to proceed with more complex non-linear functions based on a predefined error estimate. Alternatively, the results can be communicated back to the user and let them do the evaluation. Nonetheless, if the desired accuracy is not achieved more complex non-linear functions are adapted, and the training sample size is expanded. Through this gradual training the complexity of the interpolated function grows as the problem requires. The result of this process is the identification of the parameters for functions $F^i(\bullet)$.

Preference Training in Combining Dimensions—Function *G*( )

In this step we solve for the weights of function $G(\bullet)$, the aggregation function. An important characteristic of the training dataset is that it is the same as the one used in *Step Number One.* The difference is that now all these individual datasets from each dimension are combined together (i.e., we have the whole X_{User} and X_{Source} vectors, not just elements of them as in *Step Number One*).

The user inputs a percentage that does not correspond to preference within an individual dimension but to their combination. Since we already know the preference result for each dimension from *Step Number One*, we can substitute these F^i values directly in Equation 11. This way we get a linear least squares solution for the weights.

In order to assist users of different expertise defining the training dataset, we provide them with the interface seen in Figure 7. There the user can see the target request ("User Query" in the Figure) and compare it with some samples ("Training Dataset"); in other words, train the algorithm by example. The users provide the data quality values on the right side of the figure. They can either enter a percentage directly or use the sliding bar. The purpose of the sliding bar is twofold: to provide a user-friendly interface for novice users and also to express user expertise though the predefined step of the slider (e.g., a smaller step is allowed for expert users).

The whole process is iterative and is performed until the user is satisfied with the results. The training dataset grows with the user expertise or desire for refinement. In the initial stage all weights are set to equal contribution. After the user is satisfied with a set of given percentages they request the recalculation of weights. They have the option of excluding weights from the solution if they are satisfied with their values (by selecting "Lock Value"). Following the recalculation, we offer the option of displaying another set of candidate sources to evaluate the last weight solution even further (by selecting "Update Ranking Results"). In the next step the candidate values from the previous step (before the recalculation) are presented with the new adjusted percentages. In addition, a supplementary number of training samples is included that are randomly chosen from the database and provide a representative sample of the data quality space (e.g., one sample with zero percent preference, another with ten percent, and so on). The user evaluates the results again, and if not convinced, they initiate another iteration. This iterative process allows users to explore their preference and make the necessary adjustments as they see the results of their input. The process is carried out until the user is satisfied with the returned data quality results.

Simulation Process

After the training is finished, a preference profile is created with all the parameters of the trained functions. This profile is then used every time a new query is introduced. By doing so, the results are adjusted to user or application preference.

During the simulation each image from the database is compared to the query properties, and a data quality

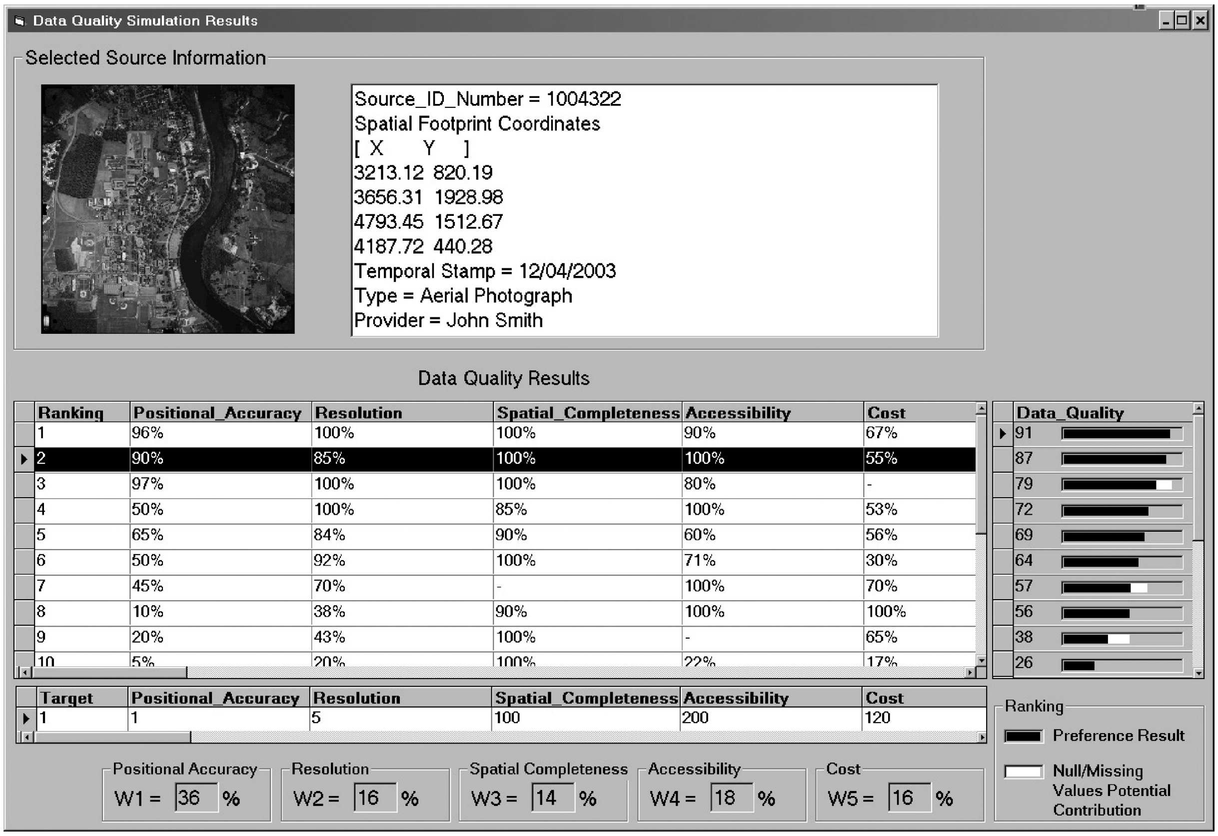

Figure 8. Simulation Environment.

metric is computed based on the profile used. A different interface is implemented to visualize these calculated metrics (Figure 8). The user can see the corresponding matching percentages in every dimension, as well as their aggregated metric. When an image is selected, additional information including a thumbnail, spatial coverage, and timestamp is presented to the user.

Handling Missing Metadata Values

A significant issue within data quality modeling is the fact that some dimensions (attributes) might have null or missing values in the database. A modeling method for data quality metrics would not be complete without compensating for such cases. The correction we apply is the assignment of a zero preference value in the $F^i(\bullet)$ function for the corresponding dimension when the X^i_{Source} value would be null or missing. This way we endorse the worst-case scenario in our solution. But at the same time we want to be able to inform the user that the overall preference metric is limited due to the lack of information (as compared to a mismatch to their request) and has the potential to increase if a value is introduced or updated in that field. We do that by associating an ambiguity value to the returned data quality metric. This value expresses the maximum range that could be added to the already calculated metric if the null or missing value is updated by the best-case scenario (a 100 percent match for this specific attribute). So, the overall result would be:

$$DQ'_{Preference} = DQ_{Preference} + \sigma^{Null/Missing}_{Preference}, \quad \text{where}$$

$$\sigma^{Null/Missing}_{Preference} = W_M \Big/ \sum_{1}^{i=n} [W_i], \qquad (12)$$

where $DQ_{Preference}$ is the preference calculated with the penalized value of zero preference for the missing or null attribute, $\sigma^{Null/Missing}_{Preference}$ is the maximum additional potential contribution by the missing/null value in dimension M, and $DQ'_{Preference}$ is the metric expressing the potential best-case scenario.

Within our display environment the user can choose and rank the results either according to the worst or the best case (i.e., without or with the $\sigma^{Null/Missing}_{Preference}$ value). In the lower right corner of Figure 8, the first four lines of the data quality report show this handling of missing information. The ranking is done based on the worst case, but where it applies, the user can see in *white* the $\sigma^{Null/Missing}_{Preference}$ value associated with the calculation next to the $DQ_{Preference}$ value (e.g., rows three, seven and nine in Figure 8). An interesting case can be seen when comparing the first and the third ranked images. The first has a higher $DQ_{Preference}$ value, but the third has a better $DQ'_{Preference}$. By supporting visualization and ranking of both values, the users can make their choice as to which source fits their quality preference best.

Conclusions

In this paper we introduced a novel approach to handle data quality attributes in geospatial queries. Our approach is characterized by the ability to model and learn user preferences, thus establishing user profiles that allow us to customize image queries, and improving their functionality in a constantly diversifying geospatial user community. Essential to our approach is a novel learning algorithm to model user preferences in terms of data quality attributes. The algorithm accommodates the particularities of quality attribute preferences, such as asymmetry, non-linearity, and value dependence in addition to distance dependence. This is a substantial improvement over existing solutions that fail to capture the particularities imposed by different applications in geospatial queries.

We presented the training process of our algorithm along with the corresponding environment. Users of various levels of expertise can train the system through a training-by-example method. Our training is an iterative process that adjusts its complexity based on user satisfaction about the given results. After training is performed, a profile is created for each user. Such profiles can be used later on in the communication process. Within the simulation environment a user can browse through image sources of varying quality. Furthermore, our result-ranking supports the important aspect of quality attributes that they might have missing or null values. Specific care was given to inform users how this missing information may affect the solution. We also introduced a set of data quality attributes that serve as input for the user query. However, they are not an exhaustive set that describes the data's fitness for use. They are demonstrative examples of how such attributes can be handled in our approach. Other descriptive attributes that may be used in our approach include for example, complementary sensor characteristics, differentiating visible from infrared imagery, and sensor calibration information.

Our future plans include the extension of our model to handle more general types of spatial datasets, especially addressing the lineage of such datasets. Towards this goal, our current work is focusing on the propagation of uncertainty information through various transformations, and the collection and analysis of user feedback for GUI design considerations.

Acknowledgments

This work was supported by the National Science Foundation through grants ITR-0121269 and DG-9983432, and by the National Imagery and Mapping Agency (now the National Geospatial-Intelligence Agency) through NURI Award NMA 401-02-1-2008.

References

Aha, D., D. Kibler, and M. Albert, 1991. Instance-Based Learning Algorithms, *Machine Learning*, 6:37–66.

Bastin, L., P. F. Fisher, and J. Wood, 2002. Visualizing Uncertainty in Multi-Spectral Remotely Sensed Imagery, *Computers & Geosciences* 28(3):337–350.

Beard, M. K., 1997. Representations of Data Quality, *Geographic Information Research—Bridging the Atlantic* (M. Craglia and H. Couclelis, editors), Taylor & Francis, London, pp. 280–294.

Bennet, D., and M. Armstrong, 1996. An Inductive Based Approach to Terrain Feature Extraction, *Cartography and Geographic Information Systems* 23(1):3–19.

Buttenfield, B., and M. K. Beard, 1994. Graphical and Geographical Components of Data Quality, *Visualization in Geographical Information Systems* (H. M. Hearnshaw and D. J. Unwin, editors), Wiley & Sons, Chichester, England, pp. 150–157.

Buttenfield, B. P., 1993. Representing Data Quality, *Cartographica* 30(2–3):1–6.

Buttenfield, B. P., and M. K. Beard, 1991. Visualizing the Quality of Spatial Information, *Auto-Carto 10*, *American Congress on Surveying and Mapping*, Baltimore, MD, pp. 423–427.

Campbell, G., L. Carkner, and P. Egesborg, 1994. A GIS-Based Multi-purpose Digital Cadastre for Canada Lands, *FIG Congress XX*, Melbourne, Australia.

Carkacioglu, A., and Y.-V. Fatos, 2002. Learning Similarity Space, *Intl Conference in Image Processing*, pp. 405–408.

Cost, S., and S. Salzberg, 1993. A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features, *Machine Learning* 10:57–78.

Cover, T., and P. Hart, 1967. Nearest Neighbor Pattern Classification, *IEEE Transactions on Information Theory* 13(1):21–27.

Davis, T. J., and C. P. Keller, 1997. Modeling and Visualizing Multiple Spatial Uncertainties, *Computer & Geosciences* 23(4):397–408.

Doucette, P., P. Agouris, A. Stefanidis, and M. Musavi, 2001. Self-Organized Clustering for Road Extraction in Classified Imagery, *ISPRS Journal of Photogrammetry and Remote Sensing* 55(5–6):347–358.

Duckham, M., J. E. Drummond, and D. Forrest, 2000. Spatial data quality capture through inductive learning, *Spatial Cognition and Computation* 2(4):261–282.

Evans, B. J., 1997. Dynamic Display of Spatial Data-reliability: Does it Benefit the Map User?, *Computers & Geosciences* 23(4):409–422.

Fegeas, R. G., J. L. Cascio, and R. A. Lazar, 1992. An Overview of FIPS 173, The Spatial Data Transfer Standard, *Cartography and Geographic Information Systems* 19(5):278–293.

Fisher, P. F., 1999. Models of Uncertainty in Spatial Data, *Geographic Information Systems: Principles and Technical Issue, Volume 1* (P. A. Longley, D. J. Maguire, and D. W. Rhind, editors), John Wiley & Sons, New York, 191–205.

Goodchild, M. F., 1999. Measurement-Based GIS, *International Symposium on Spatial Data Quality*, (W. Shi, M. F. Goodchild, and P. F. Fisher, editors), pp. 1–9.

Hunter, G. J., and M. F. Goodchild, 1995. Dealing with Error in Spatial Databases: A Simple Case Study, *Photogrammetric Engineering & Remote Sensing*, 61(5):529–537.

Lim, J.-H., J.-K. Wu, S. Singh, and A. D. Narasimhalu, 2001. Learning Similarity Matching in Multimedia Content-Based Retrieval, *IEEE Transactions on Knowledge and Data Engineering*, 13(5): 846–850.

Ma, W., and B. S. Manjunath, 1998. A Texture Thesaurus for Browsing Large Aerial Photographs, *Journal of the American Society of Information Science*, 49(7):633–648.

MacEachren, A. M., 1992. Visualizing Uncertain Information, *Cartographic Perspectives*, 13:10–19.

Mandl, T., 2000. Tolerant Information Retrieval with Backpropagation Networks, *Neural Computing & Applications*, 9(4):280–289.

McGranaghan, M., 1993. A Cartographic View of Spatial Data Quality, *Cartographica*, 30(2+3):8–19.

Mitaim, S., and B. Kosko, 1997. Neural Fuzzy Agents that Learn a User's Preference Map, *4th Intl Forum on Research and Technology Advances in Digital Libraries*, pp. 25–35.

Mountrakis, G., and P. Agouris, 2003. Learning Similarity with Fuzzy Functions of Adaptable Complexity, *SSTD'03—Lecture Notes in Computer Science*, Vol. 2750, pp. 412–429.

Müller, H., W. Müller, D. McG. Squire, S. Marchand-Maillet, and T. Pun, 2001. Performance Evaluation in Content-Based Image Retrieval: Overview and Proposals, *Pattern Recognition Letters*, 22(5):593–601.

NIST, 1992. *Federal Information Processing Standard*. Technical Report, Washington, DC, U.S. Department of Commerce, National Institute of Standards and Technology.

Paradis, J., and K. Beard, 1994. Visualization of Spatial Data Quality for the Decision Maker: A Data Quality Filter, *URISA Journal*, 6(2):25–34.

Unwin, D. J., 1995. Geographical Information Systems and the Problem of Error and Uncertainty, *Progress in Human Geography*, 19(4):549–558.

Veregin, H., 1999. Data Quality Parameters, (P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind, editors). *Geographical Information Systems—Principles and Technical Issues*, 1:177–189. New York, John Wiley & Sons.

Walker, P., and D. Moore, 1988. SIMPLE: An Inductive Modeling and Mapping Tool for Spatially-Oriented Data, *International Journal of Geographical Information Systems*, 2(4):347–363.

Wilson, D. R., and T. R. Martinez, 2000. An Integrated Instance-Based Learning Algorithm, *Computational Intelligence*, 16(1):1–28.