ESTABLISHING CORRELATIONS IN MULTI-DIMENSIONAL GIS DATABASES

Peggy AGOURIS[*], Giorgos MOUNTRAKIS[*], Anthony STEFANIDIS[**]

University of Maine, USA
[*]Dept. of Spatial Information Science and Engineering
[**]National Center for Geographic Information and Analysis
{peggy, giorgos, tony}@spatial.maine.edu

Working Group WG IV/5

KEY WORDS: Multi-dimensional, GIS, databases, queries, metadata, multimedia

ABSTRACT

Modern geospatial databases are becoming increasingly complex, with multiple types of information (e.g. imagery, maps, vector data, video, and text), huge volumes of data (e.g. numerous satellite images continuously captured in the span of a mission), and distributed storage (e.g. various servers storing different types of information). Furthermore, spatiotemporal analysis is also becoming more complicated, with analysts making use of diverse datasets to make complex decisions. These trends make geospatial queries increasingly complex and challenging.

In this paper we introduce non-linear correlations within geospatial databases to better handle user queries in distributed environments. In order to support queries, datasets are typically indexed according to their metadata information. For example, an image may be indexed according to its metadata parameters (e.g. area, scale, time, sensor). This results into defining a multidimensional (MD) space and indexing individual datasets in this space. Each dimension of this space corresponds to an individual parameter in the metadata description.

1 INTRODUCTION

A major challenge in geospatial databases is the successful retrieval of information sources for further analysis. This operation can be seen as an extension of information retrieval in multimedia databases (Faloutsos et al 1994, Faloutsos 1996, Kingsley 1996, Lombardo and Kemp 1997, Subrahmanian 1998). For example a user might request an aerial photograph with specific attributes, some of which might be metric (e.g. resolution, temporal instance) and some qualitative (e.g. infrared, mission number). With quantitative attributes a common approach would be the creation of a multi-dimensional feature vector where each dimension would correspond to a metadata value (Hjaltason and Samet 1995, Roussopoulos et al 1995, Papadopoulos and Manolopoulos 1997, Ciaccia et al 1998, Berchtold et al 2000).

Queries using such multidimensional (MD) indices commonly follow some sort of a nearest neighbor-based approach: the query defines a point in this MD space and returns the datasets whose indices are nearest to the query point. This comparison is done by measuring the Euclidean distance between the MD vectors of the Query and the Database:

$$V_Q = [m_{1q}, m_{2q}, m_{3q}, \ldots, m_{nq}]$$
$$V_{DB} = [m_{1db}, m_{2db}, m_{3db}, \ldots, m_{ndb}]$$

$$E_{distance} = \{(m_{1q} - m_{1db})^2 + (m_{2q} - m_{2db})^2 + (m_{3q} - m_{3db})^2 + \ldots + (m_{nq} - m_{ndb})^2\}^{0.5}$$

The smaller the distance the highest the correlation between the two vectors. In more sophisticated approaches the user can predefine weights for each dimension. In this case the result would be:

$$E_{distance} = \{W_1 (m_{1q} - m_{1db})^2 + W_2 (m_{2q} - m_{2db})^2 + W_3 (m_{3q} - m_{3db})^2 + \ldots + W_n (m_{nq} - m_{ndb})^2 / SW\}^{0.5}$$

This approach allows the user to predefine the importance of each dimension, but still within each dimension the Euclidean distance is the basis of the similarity. These distances depend on the underlying assumption that orthogonality exists between dimensions as well as that MD space is isotropic. This is not the case though in our MD space.

In this paper we propose an alternative way for expressing correlation within each dimension of the qualitative space. This is accomplished through a set of continuous functions, namely correlation functions. Following we introduce the notion of a correlation function and we provide some general mathematical classes. Then we present functions that depend only on the difference within each dimension followed by functions that depend on the actual values. We conclude this paper with a summary of our findings and a glimpse of potential future work.

2 CORRELATION FUNCTIONS

We propose a new approach to facilitate the non-linearity within each dimension. We keep the weights to allow the user to predefine the importance of each dimension but we substitute the Euclidean distances with *correlation functions* (CF) in each dimension. So in our case the obtained result would be:

$$\text{Correlation}\{V_Q, V_{DB}\} = \{W_1 * CF_1[m_{1q}, m_{1db}] + W_2 * CF_2[m_{2q}, m_{2db}] + W_3 * CF_3[m_{3q}, m_{3db}] + \ldots + W_n * CF_n[m_{nq}, m_{ndb}]\}/SW$$

The correlation functions can be mathematical functions such a binary, gaussian, sigmoidal, etc. or arbitrary ones (e.g. resulting from a neural networks analysis). Also the correlation functions can depend only on the difference between query and database value in that dimension, or on the actual values described above.

As an example we could examine the "temporal granularity" dimension. When the DB value is smaller than the query then the correlation is 100%. When it is larger than the query the correlation depends on the query value, so when we request high resolution information (small temporal granularity value, say 1 min) then the correlations should more rapidly decrease than when we request low resolution information. Cases like this, that could not be addressed through an Euclidean distance approach are easily expressed through our functions. Here we should mention that these functions are just a subset of potential others since the users can define custom functions at query time or even the system itself might be able to grasp these relations through a relevance feedback training approach.

2.1 Delta dependent functions

First we present functions that depend only on the delta value within each dimension, in essence depend on $\nabla(m_{nq}, m_{ndb}) = (m_{nq} - m_{ndb})$. In this general class of functions we could identify four mathematical functions being used in our approach: step, linear, gaussian, and sigmoidal functions.

- Step Functions

With this type of class a binary step function is defined based on a threshold value. This function shows that up to a certain point query and DB values are completely correlated and beyond that they are not related at all (Fig. 1). An example could be the "temporal value" where we create a binary buffer zone around the query value. Note that the step in constant throughout this mapping (e.g. 15 min).

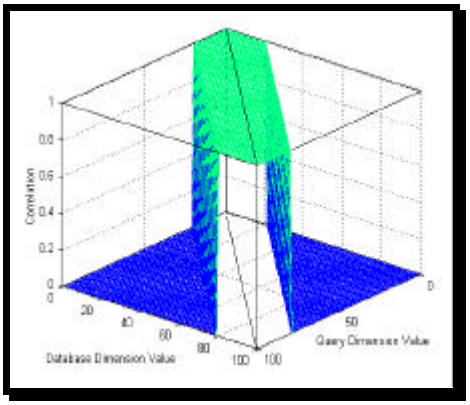

Figure 1. Delta dependent Step Function

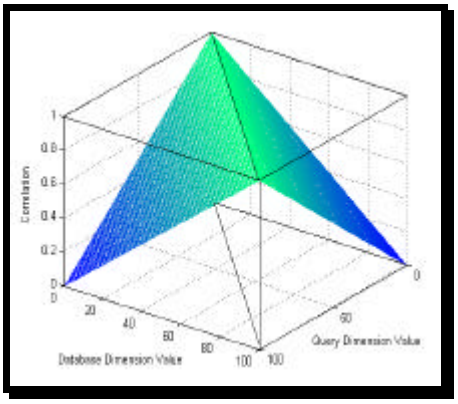

Figure 2. Delta dependent Linear Function

• Linear functions

Through this set of functions we are able to provide compatibility with the traditional Euclidean distance approach. This is done by defining CF_i as a simple distance between the two vectors (DB and Q respectively). As an example we could use this function to correlate the "temporal value" dimension. In this case we assume uniform linear distribution through that dimension. The graph in figure 2 describes the correlation in that dimension.

• Gaussian functions

Another function class that can be used is a gaussian distribution (Fig. 3). Close values result to high correlations where the further away we get the less correlated they are in a non-linear way. As an example we could use this function to correlate again the "temporal value" dimension. By using this function we create a non-linear gradual mapping between DB and Query.

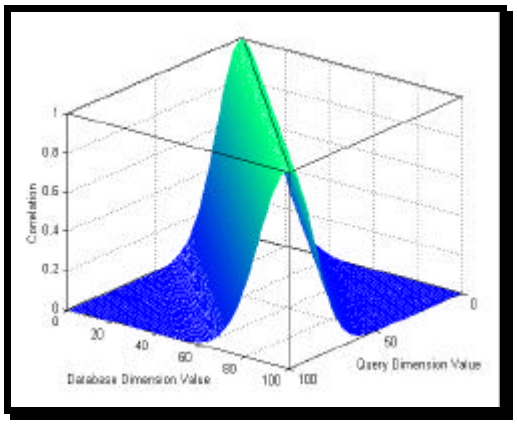

Figure 3. Delta dependent Gaussian Function

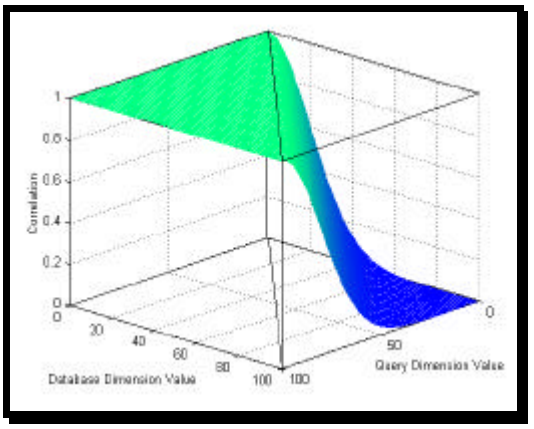

Figure 4. Delta dependent Sigmoidal Function

• Sigmoidal Functions

With this type of class a non-linear distribution is assumed for one half of the graph while a plane is assigned for the second half. A meaningful example for this function would be the "spatial granularity" dimension. If the spatial resolution of the database is better than the query's then a 100% correlation is assigned. If it is worst then a gradual correlation is returned.

2.2 Value dependent functions

In this subset of correlation functions we introduce cases where the difference ($m_{nq} - m_{ndb}$) is not sufficient to describe the correlation in that dimension. We make use of the actual (m_{nq} , m_{ndb}) values. The necessity of this kind of functions is shown below.

• Step functions with variable width

In this case we allow the user to specify in a dimension a variable step function, by assigning a scaled-width factor based on the Query Dimension Value (Fig. 5). This function implies that the width closer to the origin is much smaller than further away. By using this function we can express uncertainty within our model. Assuming that measurements closer to the origin were more reliable then a smaller search (high correlation) window should be assigned. Here we should note that the step functions are not limited to the one shown in figure 5, but can be substituted by others depending on the case (e.g. an elliptical-shape plane might be more appropriate sometimes).

• Gaussian functions with variable sigma

This function type allows the same flexibility with the above example, only it provides a gradual slope of correlation change. This way a threshold value is not necessary and a more meaningful (statistically) result can be obtained.

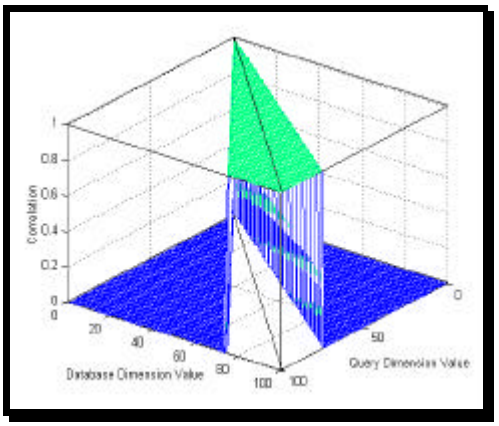

Figure 5. Value dependent Variable Step Function

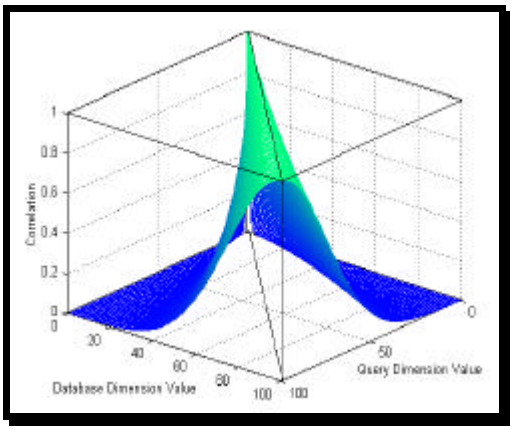

Figure 6. Value dependent Variable Sigma Gaussian Function

• Sigmoidal functions with variable slope

A "spatial resolution" example could help understand the use of such function. Assume that each axis represent the pixel size in meters both the Query (y axis) and the Database (x axis) domain. If the query would be "return all images with pixel size *20* meters and a confidence of 80%" following the mapping of this function all values < 20m would be returned since better resolution is acceptable. By examining the other half of the graph we would see that pixel size up to 24 meters would have a confidence more than 80%. What is important here is the difference of 24-20 =4m.
Now let's assume that the query is rephrased into "return all images with pixel size *60* meters and a confidence of 80%". In this case all images with pixel size <60 will be returned as expected. But due to the variable slope of the sigmoidal images up to 70m will be returned. In this case the difference would be 70-60 =10m for the same confidence percentage. This variable difference is desirable since the finer the resolution (smaller pixel size) the smaller the acceptable margin of error and the other way around.
Another example would be the "temporal granularity" dimension. In this case as well the correlation range closer to the origin is much smaller than as we go further away. So when we are dealing with seconds the acceptable return range is much smaller than when we are querying on months.

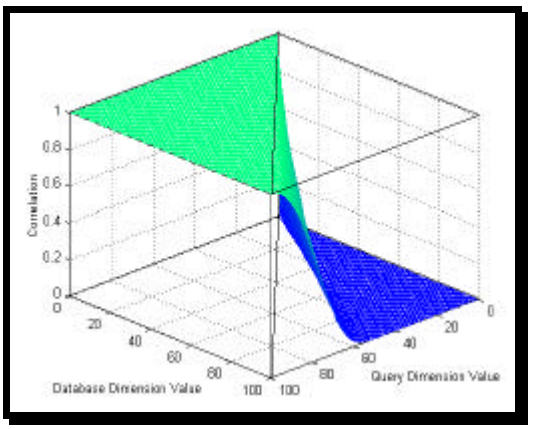

Figure 7. Value dependent Variable slope Sigmoidal Function

Here is the mathematical expression of this sigmoidal correlation function, where k is a slope rate constant:

$$CF_1[m_{1q}, m_{1db}] = \begin{cases} 1 & \text{if } m_{1q} >= m_{1db} \\ e^{\frac{-(dm^2)}{(k \cdot m_{1db})^2}} & \text{if } m_{1q} < m_{1db} \end{cases}$$

3 CONCLUSIONS

In this paper we addressed the modeling of non-linear correlations that exist within each dimension (metadata attribute) in a GIS information source database. This is accomplished by the use of correlation functions instead of an Euclidean distance calculation. The advantages of our approach were shown through a variety of examples. The computational burden is minimal since we express the correlation through continuous functions, while the benefit is substantial in the expressiveness of our model. Further expansion of this model would include correlations in qualitative dimensions (e.g. plane type) through the use of a correlation matrix, a discrete representation in this case. Also the expansion of this model will facilitate correlations between dimensions (e.g. sensor type and resolution) or groups of not directly comparable dimensions (e.g. "How close is an aerial photograph to a DEM ?"). Finally a training process can be introduced so that the correlation functions will result from a relevance feedback operation (e.g. neural network training).

ACKNOWLEDGEMENTS

This work is supported by the Advanced Research and Development Activity (ARDA) through a subaward by BAE Systems. The work of Profs. Agouris and Stefanidis is further supported by the National Science Foundation through grants number CAREER IRI-9702233, DG-9983445, and ITR-0121269.

REFERENCES

Berchtold, Keim, Kriegel, Seidl., 2000. Indexing the Solution Space: A New Technique for Nearest Neighbor Search in High-Dimensional Space. In: IEEE Transactions on Knowledge and Data Engineering (TKDE), Vol. 12, No. 1, 2000, pp. 45-57

Ciaccia P., Patella M. and Zezula P., 1998. Processing complex similarity queries with distance-based access methods. In Proc. 6th International Conference on Extending Database Technology (EDBT'98).

Faloutsos C., 1996. Searching Multimedia Databases by Content. Kluwer Academic Inc.

Faloutsos C., Barber R., Flickner M., Hafner J., Niblack W., Petkovic D., and Equitz W., 1994. Efficient and effective querying by image content. Journal of Intelligent Information Systems, 3:231–262.

Hjaltason G. R. and Samet H., 1995. Ranking in Spatial Databases. Proc. 4th Int. Symposium on Large Spatial Databases (SSD`95). Lecture Notes in Computer Science, Vol. 951. Springer Verlag, Berlin Heidelberg New York (1995) 83-95.

Kingsley C., Nwosu, Bhavani Thuraisingham, and P. Bruce Berra, 1996. Multimedia Database Systems: Design and Implementation Strategies. Kluwer Academic Publishers.

Lombardo D. and Kemp Z., 1997. Toward a Model for Mulitmedia geographical Information Systems. In Innovations in GIS 4. Taylor and Francis. pp. 56-69.

Papadopoulos A. and Manolopoulos Y., 1997. Performance of nearest neighbor queries in R-trees. 6th Int. Conf. on Database Theory (ICDT '97).

Roussopoulos N., Kelley S., and Vincent F., 1995. Nearest neighbor queries. In Proceedings of the 1995 ACM SIGMOD Conference on the Management of Data, pages 71--79, San Jose, CA.

Subrahmanian V.S., 1998. Principles of Multimedia Database Systems. Morgan Kaufman Press.