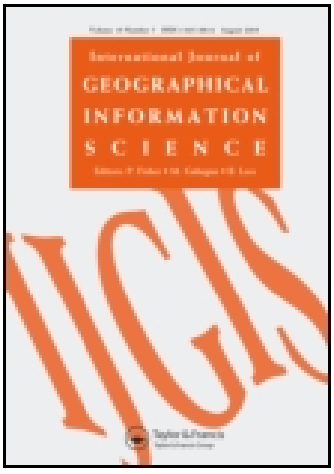

International Journal of Geographical Information Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tgis20

A feature-based approach to conflation of geospatial sources

Ashok Samal [a] , Sharad Seth [a] & Kevin Cueto [a]
[a] Department of Computer Science and Engineering , University of Nebraska-Lincoln , Lincoln, NE 68588-0115, USA E-mail:
Published online: 06 Oct 2011.

To cite this article: Ashok Samal , Sharad Seth & Kevin Cueto (2004) A feature-based approach to conflation of geospatial sources , International Journal of Geographical Information Science, 18:5, 459-489

To link to this article: http://dx.doi.org/10.1080/13658810410001658076

INT. J. GEOGRAPHICAL INFORMATION SCIENCE
VOL. 18, NO. 5, JULY–AUGUST 2004, 459–489

Research Article

A feature-based approach to conflation of geospatial sources

ASHOK SAMAL, SHARAD SETH and KEVIN CUETO[1]
Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588-0115, USA; e-mail: samal@cse.unl.edu

(*Received 17 December 2002; accepted 22 September 2003*)

Abstract. A Geographic Information System (GIS) populated with disparate data sources has multiple and different representations of the same real-world object. Often, the type of information in these sources is different, and combining them to generate one composite representation has many benefits. The first step in this conflation process is to identify the features in different sources that represent the same real-world entity. The matching process is not simple, since the identified features from different sources do not always match in their location, extent, and description. We present a new approach to matching GIS features from disparate sources. A graph theoretic approach is used to model the geographic context and to determine the matching features from multiple sources. Experiments on implementation of this approach demonstrate its viability.

1. Introduction

The process of collecting data for a GIS project from different sources often leads to problems of inconsistency, redundancy, ambiguity, and conflict of information in the collection. For example, a satellite image source may have an accurate shape of an object, a tourist map may have a less accurate shape but include a name, and a database may have the object name and the name of the architect. In GIS, conflation is defined as the problem of combining information from disparate sources such that accurate data are retained, redundancies are eliminated, and data conflicts are reconciled (Longley *et al.* 2001). Conflation manifests in many different forms; horizontal conflation refers to matching edges in adjacent maps to eliminate positional discrepancies; conflating vector GIS data with raster is also a common problem. See the work of Yuan and Tao (1999) for a more detailed classification of different types of conflations. As there are so many types and variations of the conflation problem, no single approach can solve the general problem. A type of conflation problem that is often called *feature conflation* refers to the problem of improving the features in one coverage by combining the features of another coverage (GIS/Trans 2003). This paper is concerned with the feature

[1]This work was done while a graduate student at the Department of Computer Science and Engineering at the University of Nebraska-Lincoln.

International Journal of Geographical Information Science
ISSN 1365-8816 print/ISSN 1362-3087 online © 2004 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/13658810410001658076

conflation problem for urban areas or other feature-rich regions. The first step in feature conflation is the determination of the correspondence between features in different sources, and this is the focus of this paper.

At first glance, the problem may appear to be trivial for reliable georeferenced sources. However, in practice, the sources may have different levels of accuracy and precision in attribute, spatial, and temporal dimensions (Goodchild 1995, Guptill and Morrison 1995, Bernhardsen 2002). Furthermore, they may have been created for different purposes. All of these lead to discrepancies in the representation of features in different sources. Thus, a simple overlay of the sources would not automatically reveal correspondence.

To illustrate this problem, consider two sources of geospatial data relating to Washington, DC: a digital orthophotograph (figure 1(*a*)) and a topographic map (figure 1(*b*)). The features from these sources have been extracted, registered to a common coordinate system, and stored in a GIS. When the features are overlaid (figure 1(*c*)), the extent of mismatch is evident. Translating the features in a source, say the orthophoto, does not solve the problem; it may align some features, but it will misalign others. The displacement vectors that will align a set of features from one source with those of the second source, while being similar, will rarely be identical. Names of many features are present in the topographic map, but not in the orthophoto. Many objects shown in the orthophoto are absent in the topographic map.

Interestingly, humans can reliably identify matched features in multiple sources, even with a seemingly large number of inconsistencies. They do this by direct comparison of shapes and positions of features in the source, and their relations to one another, and by using some domain knowledge. A similar approach is used for feature matching in this paper.

We assume that each source is georeferenced and is represented by a set of *features* derived by automated or manual means. As illustrated by figure 1, having two georeferenced sources does not imply perfect alignment or matching. Since the sources are georeferenced, we assume that the features may be aligned by small amounts of rotation and translation. In this research, we do not account for arbitrary rotations and translations. Over the years, manual methods have been used to derive features in many GIS data sources. More recently, researchers have developed many automated and semi-automated approaches to derive features such as roads, buildings, and airports from remotely sensed imagery (e.g. Geman and

(a) *Digital ortho-photo* (b) *Topographic Map* (c) *Overlay of the two sources*

Figure 1. Two GIS data layers for Washington, DC and their overlay.

Jedynak 1996). Commercial software such as ERDAS Imagine (Leica Geosystems 2003, Visual Learning Systems 2003) provides supervised learning based methods to extract roads and buildings as well. While the extracted features are not always complete, they provide a basis of reconciling recent information missing from more accurate sources that are perhaps manually coded or verified. As the automated techniques improve in accuracy and completeness, features will become more readily available for conflation.

In addition to the availability of features, we assume that each feature has a list of *attributes* that are known. These can be obtained by automated methods once the features are extracted. The initial matching is performed using the geo-coordinates and the attributes. The matching does not take into account the geographical context of the features, which can sometimes help disambiguate many uncertain matches. Therefore, we define a new construct, called the proximity graph, of a feature to capture the geographic context and use it in the matching process. The pairwise similarity between features of two sources is extended to multiple sources using a graph-theoretic framework. The goal is to find sets of similar features, where each set may include at most one feature from a single source. This problem is reduced to finding maximal cliques in the graph containing features from all the sources. Experimental results with a set of complex geospatial sources show that the approach is both effective and efficient.

1.1. *Motivation and potential applications*

Figure 2 shows a query-processing system in which the feature-matching module can be integrated. The schematic of the system consists of three parts as shown in

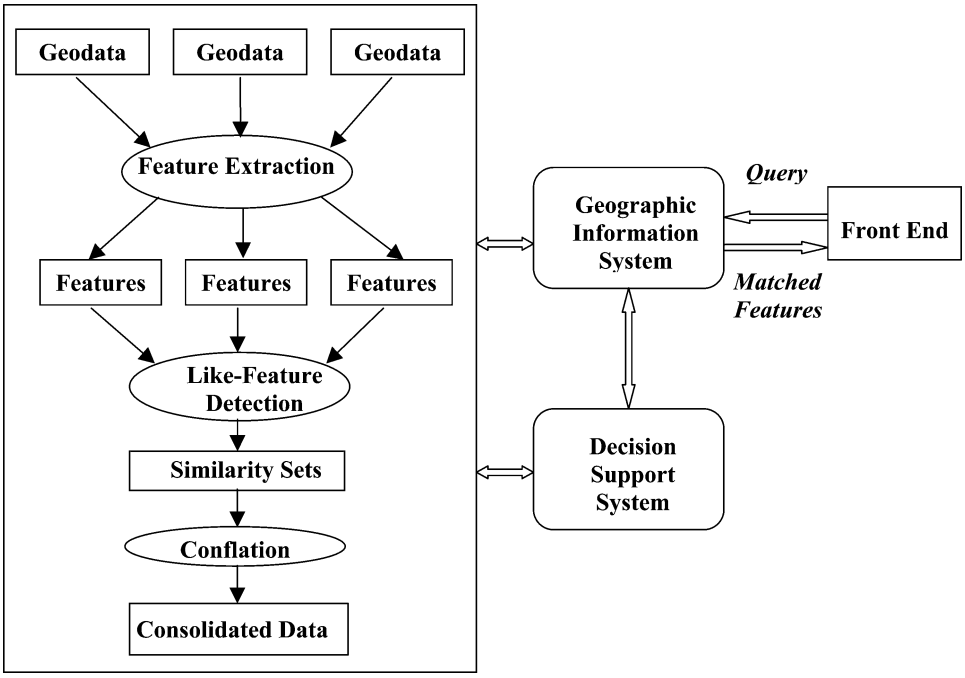

Figure 2. Role of like-feature detection module in a GIS and decision support systems.

the figure: (a) a GUI front-end, (b) a GIS and a decision-support system, and (c) a back-end processing engine. The front end uses an existing GIS and decision support system to display the results. The back-end processing engine supports the core operations needed to answer the queries submitted by the user. Examples of such operations include feature extraction, like-feature detection, and conflation. A set of geospatial sources is first presented to the feature extraction module. The extracted features are submitted to the like-feature detection module to find matching features from the sources. The similar objects identified by the like-feature detector (LFD) can be directly presented to the user. Alternatively, the conflation module can derive a unified view of the sources that can then be presented to the user. The like-feature detector, which is the heart of the system, is the focus of this paper.

Like-feature detection will find many applications in the domain of geospatial analysis, and a sampling of some applications is briefly described below.

Coverage consolidation

Gathering data is the largest expense of creating a GIS. This expense is directly correlated with data quality, which is a function of the standards used during data acquisition and data entry. A feature-matching system would reduce the cost of GIS data acquisition by combining inexpensive sources to build a better source. Such sources are now abundantly available on the Web, but are not tapped for the useful information they can add to the GIS because they are deemed to be untrustworthy.

Error detection

In our feature-matching system when features from two or more data sources are matched, the LFD lists groups of matching features and, for each feature in a group, provides a confidence measure for its membership in that group. This information can be used for interactive verification. For example, poorly matched features might indicate potentially inaccurate data.

Improving coverage registration

A non-georeferenced spatial data set must be registered before it can be stored in a GIS. Critical to registration is the choice of features for solving the affine transformations. To achieve good registration, the chosen features need to have accurate geo-positional information, and they need to be spatially accurate on their original source. A matching system could be used to refine registration by choosing spatially accurate features. This idea can be further extended to automatic registration. If an accurate feature-based source is available, one could use it as the basis of registration. Using our approach, one can derive many matching features and also identify prominent features that are more likely to have accurate geospatial coordinates. Thus, once a non-georeferenced dataset has been registered by traditional means with a few points, our system can provide additional points that can be used to obtain a more accurate registration.

1.2. *Contributions*

In this paper, we present a systematic method to identify the features from multiple geospatial sources that represent the same real-world entity. This, of

course, is an essential step before conflation. We first formulate the problem of feature matching from multiple sources as finding similarity sets of features. We then show how existing similarity measures can be used to match features from diverse sources and how they can be integrated into a single context-independent similarity measure. Further, we develop a new approach to measure the similarity of features based on their geographic context, defined as a spatial proximity graph. Finally, we develop a graph-theoretic approach to determine the similarity sets based on context-independent and context-dependent similarity measures. The experimental results show that this approach is successful. Some of the results presented here have been reported previously (Cueto 1999, Cueto *et al.* 2000, Samal *et al.* 2001).

2. Previous work

The concept of similarity has been found to be useful in many different fields including pattern recognition, artificial intelligence, information-retrieval systems, and psychology. In each field of study, of course, the concept serves a different purpose. In information retrieval, the objective is to find the degree of match between different documents, while psychologists are concerned with how humans perceive similarity between both abstract ideas and concrete objects. It is not surprising that humans' view of similarity is subjective; however, automation demands objective (quantitative) measures.

Similarity and distance are complementary concepts. Hence, often in the literature, one is defined in terms of the other. A distance measure can be converted to a similarity measure by first normalizing it to a value between zero and one and then taking the complement:

$$\Psi(A,B) = 1 - \frac{\Delta(A,B)}{U}$$

where Ψ is the similarity function, A and B are variables(features), Δ is a distance measure, and U is a normalization factor. To complete the transformation, a value for U must be determined. It can simply be chosen as the maximum distance between two features that can occur in the data set.

Below, we review a number of different similarity measures appearing in the literature that are of interest to us.

2.1. *Tversky's measures*

In psychology, similarity is often represented by geometric models (Helmuth 1980). These are appealing because they relate easily to the notion of clustering. As most clustering algorithms require a metric distance function (d), the following three axioms are assumed to hold.

- Minimality: $d(a,b) \geqslant d(a,a) = 0$
- Symmetry: $d(a,b) = d(b,a)$
- Triangular inequality: $d(a,c) \leqslant d(a,b) + d(b,c)$.

Although widely used, the metric model has been challenged for its relevance by Tversky (1977). He found that the metric axioms did not always hold during human experiments. For instance, study participants more often said that Korea was more similar to China than China was to Korea, thus violating the symmetry axiom. To

better accommodate such anomalies, Tversky outlined a set theoretic approach in which similarity between two objects is measured by a function of three arguments: (a) the attributes those are common to the two objects, (b) the attributes that belong to the first object but not to the second, and (c) the attributes that belong to the second object but not to the first.

Tversky's method is not required to follow any of the metric axioms. His method is particularly well suited to binary attributes, and it was extended to fuzzy attributes by Santini and Jain (1999). Trees are an alternative model used for similarity (Corter 1996).

2.2. *Categorical similarity*

Assessing the similarity of categorical GIS data has been thoroughly investigated by a group at the University of Maine using semantic similarity of feature classes (Rodríguez and Egenhofer 1999, Rodríguez and Egenhofer 2003). Using Tversky's set theoretic similarity model as their foundation, they have built a knowledge base of spatial categories using both WordNet (Fellbaum 1998) and Spatial Data Transfer Standard (SDTS 1999). Essentially, each categorical item in their ontology has a set of characteristics, and the similarity of two items is a function of both shared and unique characteristics of the two items.

2.3. *String similarity*

String similarity has been studied in many different contexts such as information retrieval, word processing, databases, and natural language processing. A good summary of methods is available in the books by Sankoff and Kruskal (1983) and Aoe (1994). The method chosen for this research is based on the Levenshtein distance for strings (Hall and Dowling 1980, Ukkonen 1983). This metric is defined as the minimum number of insertions, deletions, and substitutions needed to transform one string to the other. To make it more flexible, each of the operations can be given different weights. Thus, the Levenshtein distance is particularly well suited to accommodate minor spelling errors.

In GIS, these errors could be due to typos or OCR errors during data acquisition. An alternate approach to measuring string similarity is Soundex (Hall and Dowling 1980). Soundex represents strings phonetically, and therefore, it is well suited to errors in transcription. This may work well with non-traditional data sources, e.g. data collected by audio recording of a tourist guide.

2.4. *Shape similarity*

There is a large body of literature on different measures of shape similarity. Fortunately, several textbooks provide good introductions to the common framework implicit in the published work (Duda and Hart 1973, Ballard and Brown 1982, Nadler and Smith 1993, Sonka *et al.* 1999, Shapiro and Stockman 2001). First, the shapes are represented in canonical forms, and then the forms are compared. Special care is necessary to accommodate shapes that are not rotated and translated properly. In a GIS, however, most shapes are rotated and translated correctly by the registration and rectification process. This makes the problem easier, and thus a simpler similarity measure such as that used by Goodchild for polylines may be effective (Goodchild and Hunter 1997).

2.5. *Image-retrieval systems*

Image retrieval is related to GIS feature matching. In an image-retrieval system, a query is executed by scanning for images that closely match the given set of characteristics. Likewise, feature matching uses characteristics to determine whether two features match. One of these systems, Query by Image Content (QBIC) (Flickner *et al.* 1995) has influenced our work. In QBIC, a database is searched for objects that closely match the query constraints. Query results are matched by the similarity of attributes of the images in the database. This is very similar to the similarity of features in a GIS. A main difference between QBIC and the approach presented in this paper is that QBIC only allows a query to retrieve items based on one attribute, and this research combines all similarities for matching. Additionally, the GIS databases considered are multi-layered, which increases the complexity of the task. Del Bimbo discusses other approaches to various similarity measures that are viable for image retrieval (Del Bimbo 1999).

2.6. *Other feature-matching approaches*

Researchers at the University of Maine have done work related to feature matching. Much of their research focuses on topological relations between GIS objects. Their applications include spatial query by sketch (ESRI 1998) and similarity of spatial scenes. They model context using a matrix of possible topological relationships (Bruns and Egenhofer 1996). For instance, topological possibilities for aerial features include disjoint, meet, overlap, covering, etc. A scene is characterized by a matrix which is represented by a set of topological relationships. The number of gradual changes required to morph one scene into another quantifies the similarity between the two. Included in the measurement of gradual change is the similarity of distance and directional relations. This is accomplished by modelling these relations with discrete values such as very close, close, and far. Direction is discretized into cardinal directions. Their method is well suited to querying by sketch because it can measure how alike two scenes are. However, measuring the similarity of scenes is not equivalent to feature matching. Two scenes of different places can be quite similar, but obviously their features represent different real-world objects. In fact, their research assumes that common features are known so that the direction in which to change is known. Additionally, they consider only two sources at a time, whereas the system presented in this research attempts to match features from an arbitrarily large number of sources, which greatly increases the complexity of the task. Another approach to measure scene similarity that is scale- and orientation-invariant is proposed by Stefanidis *et al.* (2002). Spatial reasoning is used to identify similar features in two scenes based primarily on direction and orientation relationships. This approach does not use other ancillary attributes of objects and considers two sources at a time.

3. Feature matching

We now describe a new approach to matching features from multiple sources. The features of interest might include buildings, monuments, and parks. Matching of features is accomplished in three steps (figure 3). An initial matching step compares the common attributes of the features. The similarity between features from different sources is measured based on these individual attributes without

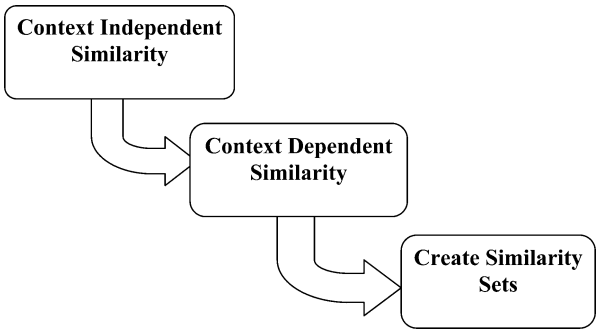

Figure 3. Steps in matching features from disparate sources.

regard to their relationships with other features. This context-independent similarity is computed as a weighted average of the similarities of the attributes of the features. However, if any one of the attributes differs significantly, we determine that the two features cannot match, irrespective of the similarity between other attributes. The method to compute the similarities of individual attributes as well as the procedure for combining them is described in this section.

In the second step, we examine the features in relation to their neighbouring features, i.e. the geographic context. We describe a model to represent the geographic context of a feature and a mechanism to compute the similarity of two geographic contexts. Both the context-independent and context-dependent similarities are then integrated into one similarity score. In the final step, we group features into sets based on their similarity. Each similarity set includes at most one feature from each source and represents one physical entity. A graph-theoretic approach to determine the similarity sets is also described in this section.

It may seem counter-intuitive to match features without using context, since context is central to any geographic information. However, as will become clear later on, the context-dependent processing we do can become computationally very expensive and does not scale up unless we limit the context. The purpose of the context-independent matching is precisely to limit the context to a small number of easily identified landmarks. Thus, instead of using all the neighbouring features in the context, we only use the more significant features, the landmarks.

3.1. *Problem formalization*

We assume that there are several *sources* of data stored within a GIS, for example, tourist maps, satellite images, digital orthoquads, and topographic maps. Furthermore, we assume that a set of features have been derived for each source. As described before, they can be derived by automated methods or manually. Examples of features include buildings, parks, lakes, roads, and airports. Clearly, such features are more easily found in urban or populated areas than in wilderness areas. A feature is represented by a set of descriptive *attributes* such as latitude, longitude, name, height, width, length, and type. By allowing a null value field for an attribute, we can assume that all features in all sources share the same set of

attributes. Formally, we will use the following notation to describe these entities:

GIS data $= \{S^1, S^2, \cdots, S^n\}$

Source $S^i = \left\{F_1^i, F_2^i, \cdots, F_{pi}^i\right\}$, where p_i is the number of features in the ith source

Feature $F_j^i = \left\{A_{j,1}^i, A_{j,2}^i, \cdots, A_{j,q}^i\right\}$, where q is the number of attributes in any feature

Attribute $A_{j,k}^i = \langle$value, confidence$\rangle$

Confidence values for attributes can be derived from precision and accuracy values of sources, or they can be filled during GIS data population. The values can be computed using data-quality elements (positional accuracy, attribute accuracy, completeness, logical consistency, lineage, semantic accuracy, and temporal information) that may be embedded in the metadata descriptions for the data sources using some metrics described in Guptill and Morrison (1995).

In the description of data sources, two types of 'null' attribute values may occur:

- *Missing*—when the source contains information about the attribute type, but the feature lacks data about the attribute. An example of a missing attribute is a tourist map that includes building names and footprints, but has a feature with only footprint and no name. In this case, the source contains name information, but the attribute of a feature lacks a value for name, so it is missing.
- *Undefined*—when the source contains no information about an attribute type. An example of an undefined attribute is a street-name attribute for a satellite image. Since the image contains no information about the names of streets, all street features derived from the image will have undefined values for the attribute.

The distinction between the two null attributes is important because the two cases should be treated differently when comparing attributes. For instance, it would be incorrect to compare a name attribute from a tourist map with a name attribute from a satellite photo because the latter carries no label information. However, if two tourist maps with name data are compared, and a name is missing, this may indicate a falsely detected feature. The details of comparing attributes are discussed later.

Our goal in feature matching is to identify features in different sources that represent the same real-world object. This is equivalent to finding a set of features that have a high degree of similarity. These sets are called similarity sets. Thus, given a set of sources, the goal is to construct similarity sets, $\{E_1, E_2, \ldots, E_m\}$, that have the following properties:

1. A similarity set E_i may not have more than one feature from any source.

$$\forall A, B \in E_i \text{ if } A \in S^i \text{ then } B \notin S^i$$

2. Every feature belongs to at least one similarity set (collectively exhaustive sets). In the extreme case, the features may belong to singleton sets, i.e. have no match.

$$E_1 \cup E_2 \cup \ldots \cup E_m = S^1 \cup S^2 \cup \ldots \cup S^n$$

3. A feature may belong to more than one similarity set because it may have ambiguous characteristics.

3.2. *Context-independent similarity measures*

As described earlier in this section (figure 3), in our approach, the first step in matching is to compute the similarity between features solely based on their individual attributes. Attributes of features may be of different types; the area of a feature is a number, but its name is a string. We briefly describe the different similarity measures used in this research to compute context-independent similarity.

String similarity

Good string similarity measures are tolerant of the anomalies shown in table 1, without inferring truly different labels as similar. It is assumed that the strings being compared are usually fewer than five words long. This is because the strings of interest are usually proper names of objects that rarely exceed a length of five words, but this is not, however, a fundamental limitation of the method.

The strings to be compared are first preprocessed. The preprocessing steps in comparing strings include tokenizing, stripping punctuations, stemming, and removing words in a stop list. Use of a lexicon comprising synonyms and abbreviations for words can be used to normalize a token into a standard form. For example, two sample input strings: $S_1 =$ 'The Portrait Gallery (Smithsonian)' and $S_2 =$ 'National Gallery of Portraits' yield the token sets: $T_1 = \{$'Portrait', 'Gallery', 'Smithsonian'$\}$ and $T_2 = \{$'National', 'Gallery', 'Portrait'$\}$, respectively.

After preprocessing is complete, a string similarity function measures the similarity of the two token sets. A simple measure is computing the ratio of the number of elements in the intersection of the two token sets to the size of the sets. Although efficient algorithms exist for this computation (Aoe 1994), it has a major drawback. This method does not tolerate character inaccuracies, e.g. transposition, and omission. Thus, if the word 'Gallery' was misspelled as 'Galery' in one string, the word would not be in the intersection, and the similarity measure would be significantly different.

A second approach to measuring string similarity is counting the minimum number of character deletions, additions, and substitutions that are required to transform one token into another. This count is known as the Damerau–Levenshtein metric (Hall and Dowling 1980). Thus, similarity using the Damerau–Levenshtein metric accommodates the character errors in table 1. After the distance between tokens is computed, a word–word matrix is configured using the following:

$$\Lambda_{i,j} = 1 - \frac{DamLev(i,j)}{Max(Length(i), Length(j))}$$

Table 1. String errors that matching should accommodate.

	Examples	
Error type	Sample 1	Sample 2
Word omission	Abraham Lincoln Memorial	Lincoln Memorial
Word substitution	Reagan National Airport	Washington National Airport
Word transposition	National Art Gallery	National Gallery of Art
Word abbreviation	National Archives	Nat'l Archives
Character omission	Washingtn Monument	Washington Monument
Character substitution	Frear Gallery	Freer Gallery

where *DamLev* (*i, j*) is the Damerau–Levenshtein distance between the two tokens in the word–word matrix, and *Max* (*Length*(*i*), *Length*(*j*)) is the character count of the larger token. Each element of the matrix, $\Lambda_{i,j}$, represents the similarity of the token pair (token in row *i* and token in column *j*) and is a value between zero and one that represents the similarity of the token pair. To find the overall similarity of two strings, the word–word matrix is searched for an optimal solution. This involves finding best matches between tokens along the rows and columns with the constraint that a token (either along a row or a column) cannot match with more than one token. In case of multiple solutions, we choose the one that has the best match as computed by the sum of the similarities between the matched tokens in the two strings. In the worst case, this search may become exhaustive but is justified because of the small search space. Once the best match is computed, the token similarities are summed and normalized as follows:

$$\sigma(S_1,S_2) = \frac{\sum_{k=1}^{l_{\max}} \Lambda_{\text{optimal}}(k)}{\mu}$$

where $l_{\max}$ is the size of the larger token set, Λ_{optimal} denotes the cells that produce the optimal solution, and μ is the mean length of the sets of tokens derived from the two strings. Either the geometric or the arithmetic mean may be used. This method has the advantage that it will accommodate word transpositions as well as character errors. Table 2 shows the word–word matrix between the two strings S_1 and S_2.

In this example, there is an exact match between two pairs of tokens ('Portrait'–'Portrait' and 'Gallery'–'Gallery') yielding a match of 1.0. The best match of the token 'National' occurs with 'Smithsonian' with a similarity of 0.3636. Since there is no overlap between the matches, they constitute the optimal match. The overall similarity between the two strings is the average similarity between the matched tokens (1.0, 1.0, and 0.3636), which results in a similarity measure of 0.7879; if, however, the word 'Gallery' were misspelled as 'Galery', the similarity value would have been 0.7403. Thus, this technique yields a much more intuitive value.

Scalar similarity

Scalars are quantities such as area and length that have a magnitude describable by a real number. Comparing scalars is a relatively simple task. In general, the distance between two scalars is simply the difference between their values. The distance can then be converted to a similarity value. A good similarity measure should match intuition. For example, one would say that a pair of numbers (10, 20) is less similar than the pair (123,010, 123,020). This is because many people assume that the range of values is close to the given numbers. Thus, the similarity of two

Table 2. Matrix showing the similarities of the tokens (cells that produce an optimal solution are shown in bold).

	Portrait	Gallery	Smithsonian
National	0.0	0.125	**0.3636**
Gallery	0.0	**1.0**	0.0
Portrait	**1.0**	0.0	0.1818

scalars with values a and b can be computed as:

$$\sigma(a,b) = 1 - \frac{|a-b|}{|U|}$$

where $|U|$ denotes the magnitude of the range of values defined for the scalar variables a and b.

Positional similarity

A function that differentiates geographic information systems from database management systems (DBMS) is the ability to use positional information about objects. When comparing features, it is important to consider their positions. We assume that the coordinate system is UTM. Because UTM coordinate pairs represent distances on a plane, we can use well-known planar distance measures to measure the similarity. Two common measures of distance are Euclidean and Manhattan (or city block) distance.

Given two coordinate pairs on a plane, $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ the Euclidean and Manhattan distances are given by:

$$\Delta_{\text{Euclidean}}(P_1,P_2) = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$$
$$\Delta_{\text{Manhattan}}(P_1,P_2) = |x_2-x_1| + |y_2-y_1|$$

Assuming that the points are in the same UTM zone (see Jones 1997 or Dana 1999) to modify this if they are not), we compute the positional similarity as:

$$\sigma(P_1,P_2) = 1 - \frac{\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}}{U}$$

where U is the maximum distance. As with scalar similarity, the maximum distance can be derived from either the data set or the pre-set for a domain. It should also be noted that use of positional similarity implies that each feature has a clear and unambiguous representative point. For an areal feature, its centroid and the centre of its bounding box provide such points.

Shape similarity

An important attribute of many polygonal features is their shape. Because shapes in GISs are generally represented as polygons, the terms 'shape' and 'polygon' are used interchangeably in this paper. The field of computer vision provides a cornucopia of published works about comparing shapes. Good shape-similarity measures are invariant to rotation, translation, and scale, and they also conform to human perception of shape similarity. Some examples of representations are moment descriptors, chain codes, and Fourier descriptors (Nadler and Smith 1993). We use the measure given by Goodchild and Hunter (1997). Although they describe a linear measure, it can easily be generalized to compare two-dimensional shapes. First, a veto is imposed if the aspect ratios are significantly different. Otherwise, the shapes are scaled to match the lengths of their major axes. Then, a buffer space is placed around one of the shapes, and they are overlaid.

Overlaying is accomplished by aligning the centre points of the bounding boxes. The similarity of the shapes is simply the percentage of a shape that is within the buffer zone of the other shape (figure 4). To ensure a symmetrical measure, the

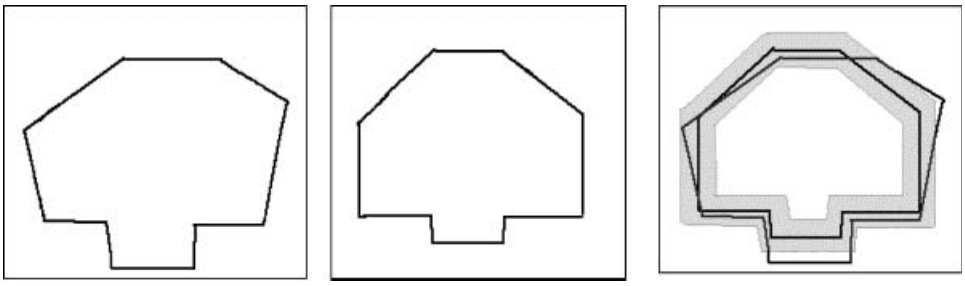

Figure 4. Two polygons and the buffered intersection.

computation is done twice: once for the first shape's distance to the second, and again for the second shape's distance to the first. The two measurements are averaged to determine the similarity of the pair.

The equation for computing the similarity of two shapes A and B is given by

$$\sigma(A,B) = \frac{A\Phi B + B\Phi A}{2}$$

where $A\Phi B$ is the percentage of A that is within the buffer zone of B. An assumption of the measure above is that the shapes are rotated correctly. This is a sound assumption because the shapes of interest are from geographically rectified maps.

3.3. *Context-independent similarity of features*

An overall similarity between two features is computed by taking a weighted average of the similarities between their individual attributes. Weighted averages have added flexibility of giving some attributes more importance. For instance, the area attribute of a building is often inaccurate and may be assigned a lower weight than the name attribute. Thus, the overall context-free similarity between two features is given by:

$$\Psi_{\mathrm{CF}}(F_1,F_2) = \frac{\sum_{i=1}^{q} w_i \cdot \sigma_i(A_{1,i},A_{2,i})}{\sum_{i=1}^{q} w_i}$$

where w_i is a weighting factor for the ith attribute, and $\sigma_i(a,b)$ is the attribute similarity of the ith attribute between features F_1 and F_2. Note that a veto of any attribute similarity indicates a veto in the combined similarity. The weight vector $[w_1, w_2, \ldots, w_q]$ is pre-set for a given set of sources. It can be obtained by training, or set by a human expert.

The combined measure indicates the similarity between two features. The next step is to extend it to measure the similarity between multiple features. Prior to taking this step, the similarity between each feature pair needs to be organized in a convenient manner. A similarity matrix is a structure that provides this organization. It contains all pairwise feature similarities for two sources. Figure 5 shows a similarity matrix for two sources, each with six features. The shaded cells indicate vetoes implying that the corresponding features cannot possibly match.

Similarity matrices are produced for each pair of sources and are used in the

	Source 1					
Source 2	12	34	35	45	56	66
123	0.00	0.12	**0.76**	0.10	0.00	0.00
145	0.23	0.45	0.10	**0.88**	0.00	0.11
234	0.00	0.00	0.00	0.00	0.40	0.40
255	0.23	0.10	0.30	0.48	0.00	0.54
267	**0.65**	0.00	0.00	0.20	0.10	0.00
315	0.10	0.04	0.15	0.05	0.00	0.05

Figure 5. A sample similarity matrix.

next step of the matching process to compute contextual similarity. The definition of the cell value for a similarity matrix for two sources S^1 and S^2 is given by:

$$SM^{1,2}[i][j] = \Psi_{\mathrm{CF}}\left(F_i^1, F_j^2\right)$$

Note that the similarities of features are independent of one another, so neither rows nor columns must sum to one.

3.4. *Context-dependent similarity*

Context-independent similarity measures alone are not always sufficient to determine matching features unambiguously. This is reflected in the lack of a clear maximum in each row and column of a similarity matrix. We propose the use of geographic context to assist the process of disambiguation. Geographic context refers to the spatial relationships between objects in an area. Some common relationships used by humans are topologies, distances, and directions. Point or small area features lack interesting topological relationships; hence, only the distance and angle are used for this research to define the context of a feature. The geographic context of a feature is formally defined as its distance and the orientation to other features in the source. Our hypothesis is that the geographic context of a feature is invariant and is independent of the representation chosen. It is then logical to boost the degree of match between two features with a similar geographic context, i.e. similar distance and directional relationships with other features in their respective sources. An example of two ambiguous features (figure 6) shows that features A and B both have similar shape and geographic coordinates.

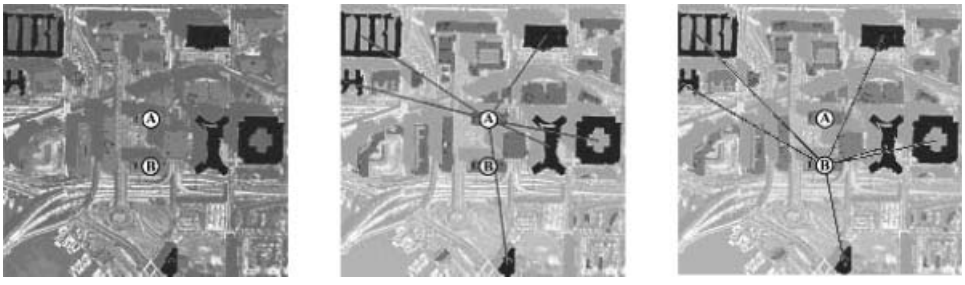

Figure 6. Two similar features and their contexts.

However, their contexts (represented as graphs) are quite different. Contextual similarity measure captures this difference as a function of their context-free similarity and the geographic contexts of the two features in the two sources.

Proximity graphs

Maps and other geographic sources of geospatial data are abstractions of the real world and suffer from various inaccuracies. Depending on the intent of the cartographer and potential uses by an audience, the map may distort or displace features to different degrees. While these sources may not be very useful in determining the exact geocoordinates or measuring the distances between two features accurately, they are still very valuable for a variety of applications. The primary reason is that they maintain relationships among their features. A map may have the Washington Monument at the wrong distance from the White House but is unlikely to change its relationships, i.e. approximately south of the White House, east of the reflecting pool, and west of the US Capitol. All useful maps will maintain this adjacency information, and we capture these relationships in the form of a graph, called a proximity graph (figure 7 is an example).

A proximity graph is a star graph, which represents the context of the feature at the centre. All other features in the source are also represented as nodes and are connected to the centre node by edges. Each edge is weighted with the distance and angle of the node's feature relative to the centre node, i.e. each edge is a vector. The nodes in the proximity graph are all features that belong to a source and have attributes associated with them. Formally, a *proximity graph of a feature*, F_j^i, is defined as a weighted directed graph as follows:

$$G\left(F_j^i\right) = \left(F_j^i, V_i, E_i, \delta, \theta\right)$$

where: $V_i = S^i$, the set of all features in the ith source; $E_i = \left\{\left(F_j^i, F_k^i\right) \forall k \neq j\right\}$, the set of edges; $\delta : E_i \rightarrow \Re$, where $\delta\left(F_j^i, F_k^i\right)$ is the Euclidean distance between the two features; $\theta : E_i \rightarrow \Re$, where $\theta\left(F_j^i, F_k^i\right)$ is the angle of the line segment between the two features with respect to the horizontal (figure 7).

In principle, a proximity graph contains edges from the central feature to all other features in the source. However, in practice, we constrain its spatial extent. Only the set of features within a certain radius are used in the proximity graph. This

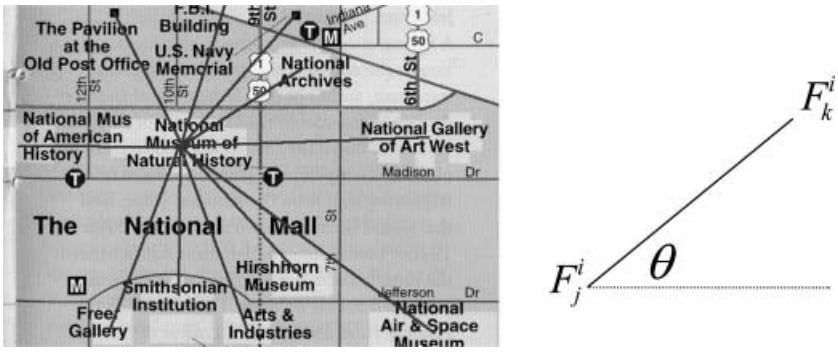

Figure 7. A partial proximity graph for a feature.

allows the size of the graph to be relatively independent of the size or scale of the source.

Similarity of proximity graphs

We now outline a method to compare proximity graphs and quantify their similarity. First, we present a brief motivation for our approach. If we overlay the proximity graphs for a feature from two perfect sources, they would match exactly. In general, however, it is rare to have perfect matches, and one has to use more sophisticated methods. Missing and imperfectly located features contribute to this problem. In such cases, one can use an environment-centred approach, where significant markers (features) guide the matching process. This approach is often used in navigation.

These significant features, which are likely to be present in all sources, are called landmarks in our approach. An algorithm to automatically determine landmarks from similarity matrices is given in the next section. The total vector offset between the landmarks in the two sources determines the degree of similarity between two proximity graphs. Figure 8 (left) shows two overlaid proximity graphs with landmarks circled. The landmarks that match are shown in boxes. Figure 8 (right) shows the offset vectors for the landmarks.

The total offset vector is computed as the vector sum of all the offset vectors for the landmarks. The similarity between two proximity graphs is computed as a function of this measure as described below. Given a set of offset vectors (Λ) derived from landmark matching given by

$$\Lambda = \left\{ \vec{\lambda}_1, \vec{\lambda}_2, \ldots, \vec{\lambda}_n \right\}, \text{ where } \vec{\lambda}_i = (\Delta_i, \Theta_i)$$

the magnitude of the summary offset vector is calculated by:

$$\mathrm{Y} = \left| \sum_{i=1}^{n} \left(\vec{\lambda}_i \right) \right| = \sqrt{\left(\sum_{i=1}^{n} (\Delta i \cdot \cos(\Theta i)) \right)^2 + \left(\sum_{i=1}^{n} (\Delta i \cdot \sin(\Theta i)) \right)^2}$$

Finally, the similarity of two proximity graphs is given by:

$$\sigma\left(G\left(F_1^i\right), G\left(F_2^j\right) \right) = 1 - \frac{\mathrm{Y}}{U \cdot n}$$

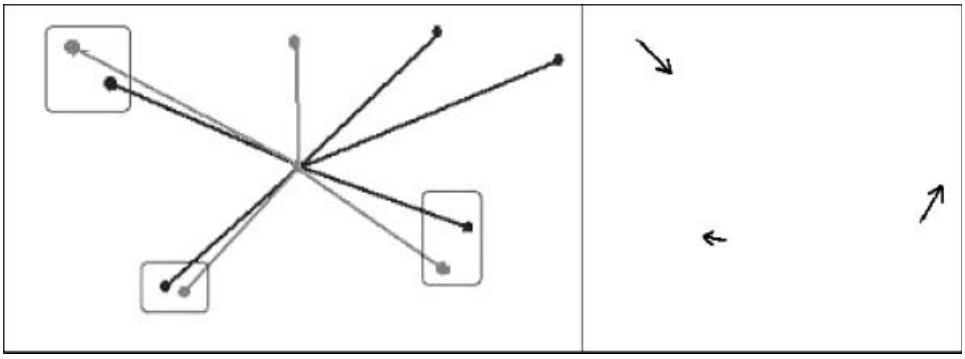

Figure 8. A matched proximity graph (left) and landmark offset vectors (right).

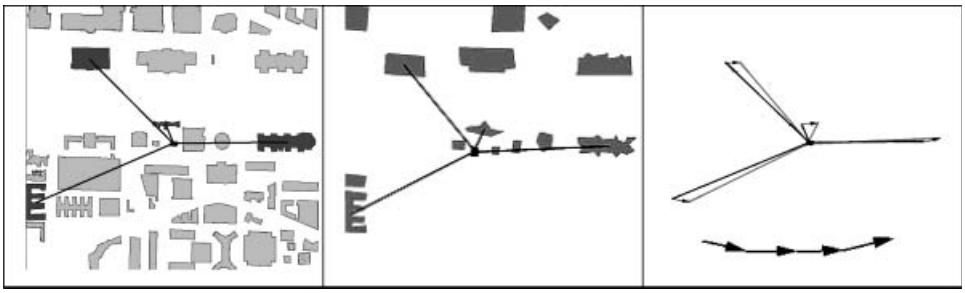

Figure 9. An ambiguous feature and its potential match.

where U is the maximum possible distance between two landmark features, and n is the number of landmarks. The value of U is bounded by the intersection of the areas that the proximity graphs cover. If, for example, one source completely covers another source, the maximum diameter is simply the diameter of the smaller source. Likewise, if the areas overlap, the maximum distance is the diameter of the area of intersection.

Although features in two sources will share the same set of landmarks, the magnitude of the summary offset vector (Y) will be small only if the features match well. This is illustrated with an example shown in figures 9 and 10. On the left of the figures are two features that have similar position and shape attributes. The middle pictures show another source with a potential match for the two features, i.e. a high context-independent similarity. The right pictures show the effect of overlaying the proximity graphs. It can be seen that the vectors of figure 9 all point east, so when they are summed, the magnitude of the resulting vector is large. In contrast, the offset vectors of figure 10 are very small and randomly oriented. The summary offset vectors shown at the bottom clearly show that the degree of match in figure 10 is significantly higher.

Selection of landmarks

Landmarks are significant features that are likely to be found in any representation of a geographic area. As described before, they play a significant role in the determination of a geographic context of a feature and are used to match proximity graphs. Here, we describe an approach to automatically determine

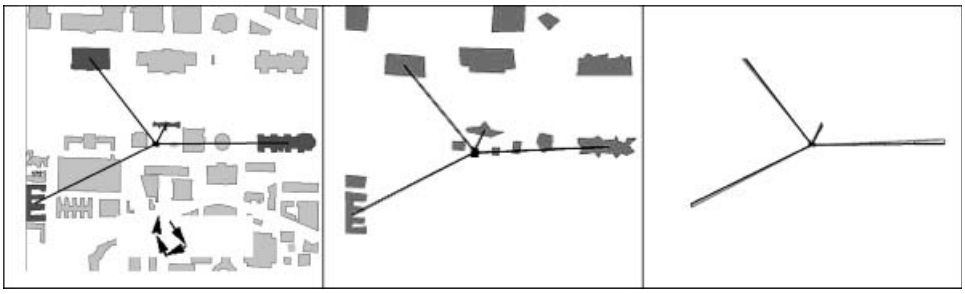

Figure 10. Second ambiguous feature and its potential match.

landmarks in a set of geospatial sources. Intuitively, the landmarks will be more prominently and accurately represented in any representation. Consequently, the features corresponding to the same landmark in different sources will be relatively easy to match, i.e. they will have a high similarity. This can be easily determined from the similarity matrices. A match (i.e. a cell in the similarity matrix) would manifest as an unambiguous maximum in both the row and the column. This idea is coded as a simple function that will discover landmarks and is given below.

Given two sources S^i and S^j and their similarity matrix ($SM^{i,j}$), we define the following auxiliary functions:

- $MR_1(l)$: the row in column l holding largest value;
- $MR_2(l)$: the row in column l holding second-largest value;
- $MC_1(k)$: the column in row k holding largest value;
- $MR_2(k)$: the column in row k holding second-largest value.

A cell in row k and column l of $SM^{i,j}$ is chosen as a landmark if:

- $MR_1(l)=k$ and $MC_1(k)=1$, and
- $(MR_1(l) - MR_2(l) >$ Threshold) and $(MC_1(k) - MR_2(k) >$ Threshold).

In other words, these conditions require that a landmark be the maximum value in its row and column. Furthermore, the difference between the landmark's similarity and the next highest similarity is greater than a threshold for both its row and column. The similarity matrix shown in figure 5 would yield three landmarks (shown in bold) if we choose the landmark threshold to be 0.60. The similarity between the features in source 2 with feature 35 in source 1 is shown along the third column. The similarity between feature 35 in Source 1 and feature 145 in source 2 is shown as 0.76. This is significantly greater than the next highest score (0.30) in that column. Similarly, the second row shows the similarity between feature 145 of Source 2 with the features of Source 1. It shows that the similarity between feature 145 (Source 2) and feature 45 (Source 2) is 0.88 and is significantly higher than the similarity between 145 and all other features in Source 2. Thus, this match, i.e. Feature 145 in Source 2 and Feature 45 in Source 1, represents a feature that can be used as a landmark.

3.5. *Feature matching with two sources*

In the previous sections, we have described context-independent and context-dependent similarities between features of different sources. Now, we describe how both are integrated to derive the set of matching features in a pair of sources. A simplistic approach that computes the overall similarity by averaging the two may lead to the creation of a new similarity matrix that invalidates some landmarks or discovers others. To avoid this problem, a relaxation-based iterative approach is proposed here. The algorithm (figure 11) starts by initializing an overall similarity matrix with the context-independent values. The landmarks are identified, and cell values are updated by averaging their value with their contextual similarity. After all cell values are updated, the matrix is again searched for landmarks, and the process is repeated. The algorithm stops when the set of landmarks for the two sources stabilizes. Updating cell values is an iterative process similar to that posed

Input: $SM^{i,j}[][]$ – A matrix initialized with context independent similarities.

Output: $CSM^{i,j}[][]$ – A context dependent similarity matrix.

Algorithm:

$CSM^{i,j}[][] = SM^{i,j}[][]$

Repeat

Landmarks = DetermineLandmarks($CSM^{i,j}[][]$)

For each row x

For each column y

$$CSM^{i,j}[x][y] = \frac{w_1 \cdot SM^{i,j}[x][y] + w_2 \cdot \sigma_{proximity}(x, y)}{w_1 + w_2} n$$

End

End

Landmarks2 = DetermineLandmarks($CSM^{i,j}[][]$)

Until(Landmarks2 = Landmarks)

Figure 11. Algorithm for integrating context for similarity matching.

in relaxation algorithms (Sonka *et al*. 1999). Because these algorithms converge to a solution iteratively, the pattern of updates does not affect the solution. Therefore, a simple raster scan of the cell values will work well.

3.6. *Feature matching with more than two sources*

The matching process just described for two sources can be extended if the number of sources is greater. The concept of grouping similar features is captured with a similarity set. A similarity set is simply a group of features whose pairwise total similarity is very high. Ideally, similarity sets are mutually exclusive, because each set should represent a single real-world entity. However, it is possible for a feature to be ambiguous and thus be a member of more than one similarity set.

Finding similarity sets can be reformulated in graph theoretical terms as follows. The similarity for each pair of features in a source is represented by a weighted edge between the two features. A veto is represented by omitting an edge. There are no edges between features of the same source because it is assumed that all features from a source represent distinct objects. A total similarity graph that represents similarity relationships between all features in sources is built (figure 12 shows a schematic). The total similarity graph is defined by:

$TSG = < V, E >$ is a graph comprising vertices and edges.

$V = \bigcup_{i=1}^{n} S^i$ is a set of vertices that represent all features in all sources.

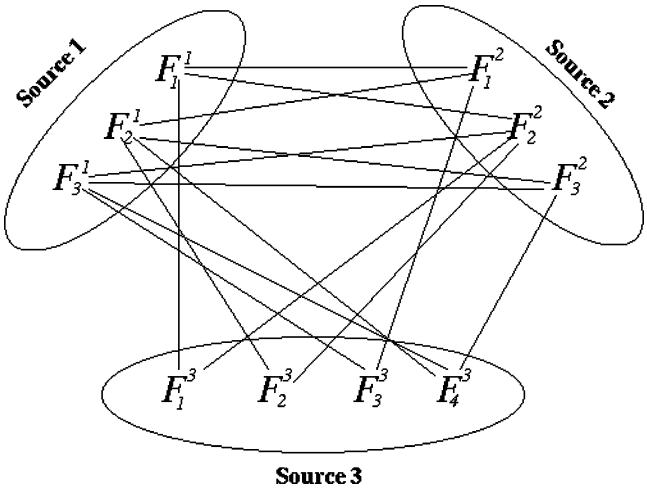

Figure 12. Total similarity graph for three sources. Weights are not shown along the edges.

$E = \{u,v\}$ where $u, v \in V$;

$u \neq v$;

if $v \in S^i$ then $u \notin S^i$;

$\Psi(u,v) \neq$ veto

$w(E) = \Psi(u,v)$ Edges are weighted with the similarity between two features.

The goal is to locate, in the total similarity graph, groups of features (at most one from each source) that have a high similarity between them. This implies that there must be an edge between each pair of features in the group, i.e. there can be no vetoes within a group. In graph terminology, this group is a fully connected sub-graph or a clique (Cormen *et al.* 1992). The problem of finding similarity sets then is equivalent to finding maximal cliques in the total similarity graph that have a group similarity above a certain threshold.[2]

Clique finding is an NP-complete problem, but efficient approximate algorithms for finding maximal cliques can be found in the literature (Ballard and Brown 1985; Sonka *et al.* 1999). Once the similarity sets are found, we determine their goodness by using an approach described in the next section.

3.7. *Similarity of a clique*

The obvious approach to represent the similarity of a clique by the average of the weights of the edges is not always desirable. It penalizes the overall group similarity if one edge is weak, which is not appropriate. Consider the clique on the left of figure 13. The similarity between nodes B and D is relatively low, but similarities between A, B and A, D are high. The fact that this happens may not be intuitive, but similarities are a function of common attributes. Thus, if D and B have few attributes in common, their similarity may be low. However, node A may have information in common with some features that fills those gaps. Thus,

[2]A clique is maximal, if it is not a proper subset of any other clique.

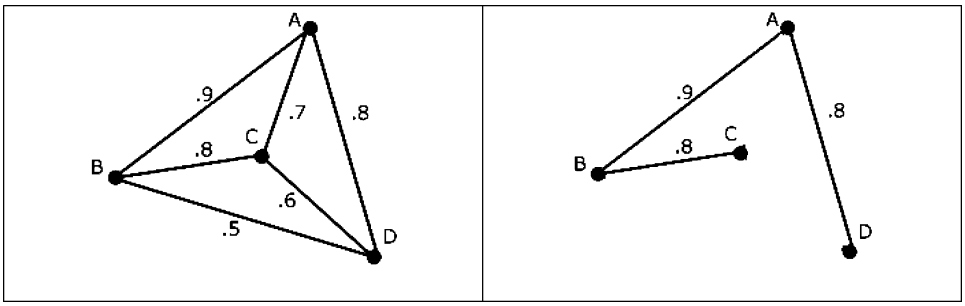

Figure 13. A clique and corresponding maximal spanning tree.

the overall similarity of the clique should not degrade because of incomplete information.

A graph theoretic approach can be used to overcome this. First, a maximum spanning tree is built using the edges of the clique using standard algorithms (Cormen *et al.* 1992). The spanning tree ensures that all features will have an adjacent edge that contributes to the group similarity. The overall similarity of a set is then defined as the average of the weights of the edges in the spanning tree.

4. Experimental results

The prototype system uses Arcview (ESRI 1998) for a front end. A point of interest (POI) query capability was implemented on top of Arcview using the Avenue scripting language. In essence, the front end allows users to click the mouse on a POI, and it retrieves from each source all features within a certain radius of the point. The retrieved features are passed to the LFD for processing.

The LFD module, as described in the last section, is implemented using the Java programming language. The LFD included a user interface, but it was only used to visualize the data during various processing stages.

4.1. *Data sets*

The test data set comprised several sources of data covering the Washington, DC area (figures 14–20). All non-georeferenced sources were registered and rectified using ARC/INFO. These sources were chosen because of their variety and free availability. The remainder of this section describes each source and the methods that were used to extract the information from each one. All figures are cropped to show their common area, and scaled for printing; the images used for research are of a higher quality. The digital orthoquadrangle and the topographic maps have the standard scale of 1:1:24 000. Other sources, e.g. tourist maps, did not have an associated scale, since they are not standard map products.

1. *Digital Orthoquadrangle:* This is a USGS aerial photograph that has a resolution of 1 m. The image was registered and rectified by the USGS prior to its acquisition. Features extraction was done manually using Arcview's on-screen digitization. The attributes defined for each feature are coordinates, shape, category, area, and perimeter. This source is the most accurate, and it was used for registration of all non-georeferenced images.
2. *Topographic Map:* The topographic map was obtained from USGS. It was

pre-registered and rectified. As with the orthophoto, the features were extracted manually using Arcview. The defined attributes for this source are coordinates, shape, category, area, perimeter, and name.

3. *Geographic Names Information System:* This was also obtained from USGS. It is simply a gazetteer that is distributed via a comma delimited text file. The data were imported into ARC/INFO, the coordinate system was changed to UTM, and the data were then exported to Arcview. The defined attributes for the features are coordinates, category, and name.
4. *World Wide Web Tourist Map:* This source is a tourist map found on the web. It is a non-georeferenced GIF file with an unknown origin. The image was registered to the orthophoto using ARC/INFO, and the features were extracted manually. This source's defined attributes are coordinates, shape, category, area, perimeter, and name.
5. *Magazine Tourist Map 1:* This tourist map source is an image scanned from a tourist guide found in a Washington, DC hotel lobby. The image was scanned at 600 dpi. The source was registered to the orthophoto, and the features were extracted manually. Defined attributes for the source are coordinates, shape, area, perimeter, category, and name.
6. *Magazine Tourist Map 2:* This was obtained from a Thrift Car Rental tourist booklet. The image was scanned at 600 dpi. The source was registered to the orthophoto, and the features were extracted manually. Defined attributes for the source are coordinates, category, and name.
7. *Magazine Tourist Map 3:* This also was found in a Thrift Car Rental tourist booklet. The image was scanned at 600 dpi. The source was registered to the orthophoto, and the features were extracted manually. Defined attributes for the source are coordinates, shape, area, perimeter, category, and name.

Figure 14. Source 1: digital orthoquadrangle.

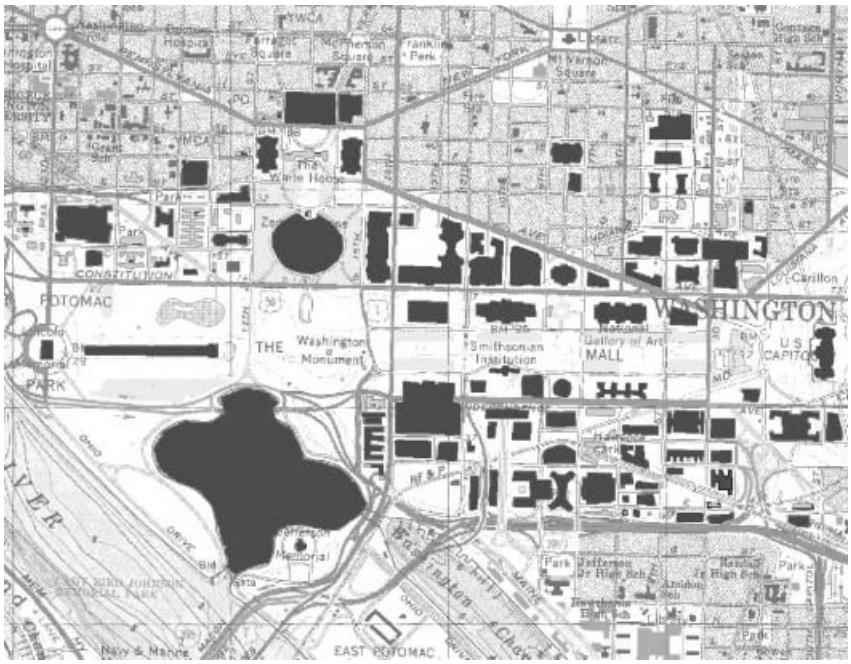

Figure 15. Source 2: topographic map.

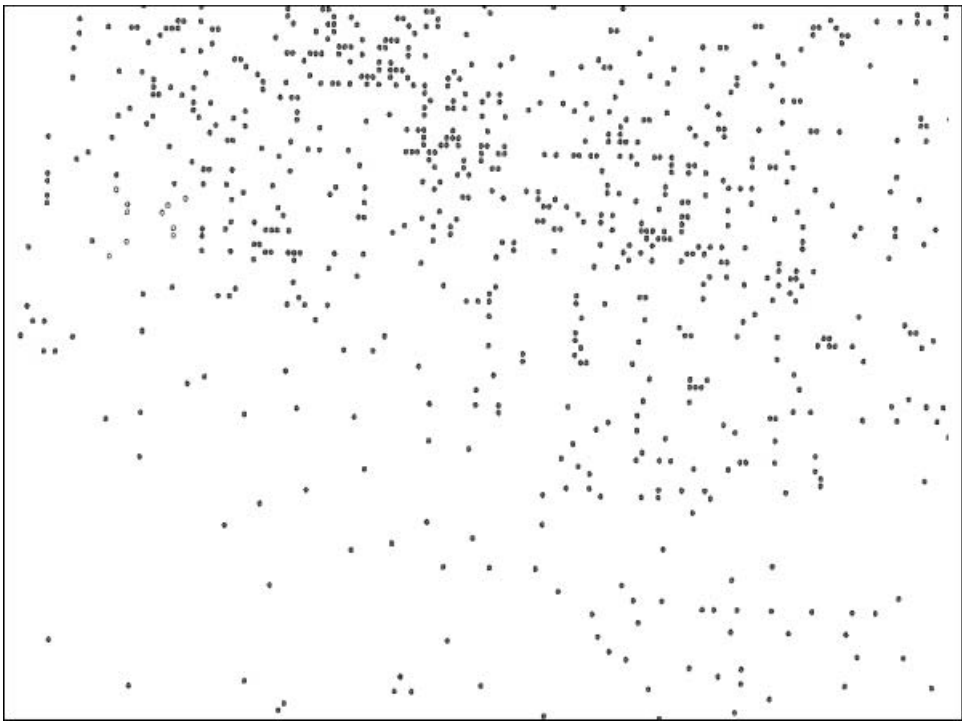

Figure 16. Source 3: plot of geographic names information system points.

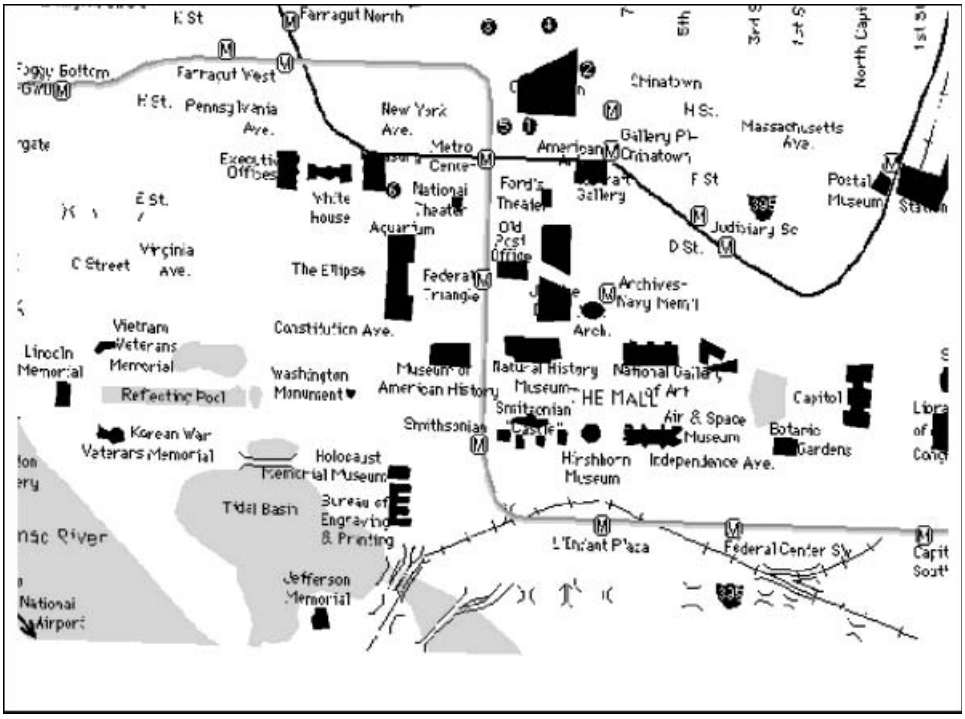

Figure 17. Source 4: World Wide Web tourist map.

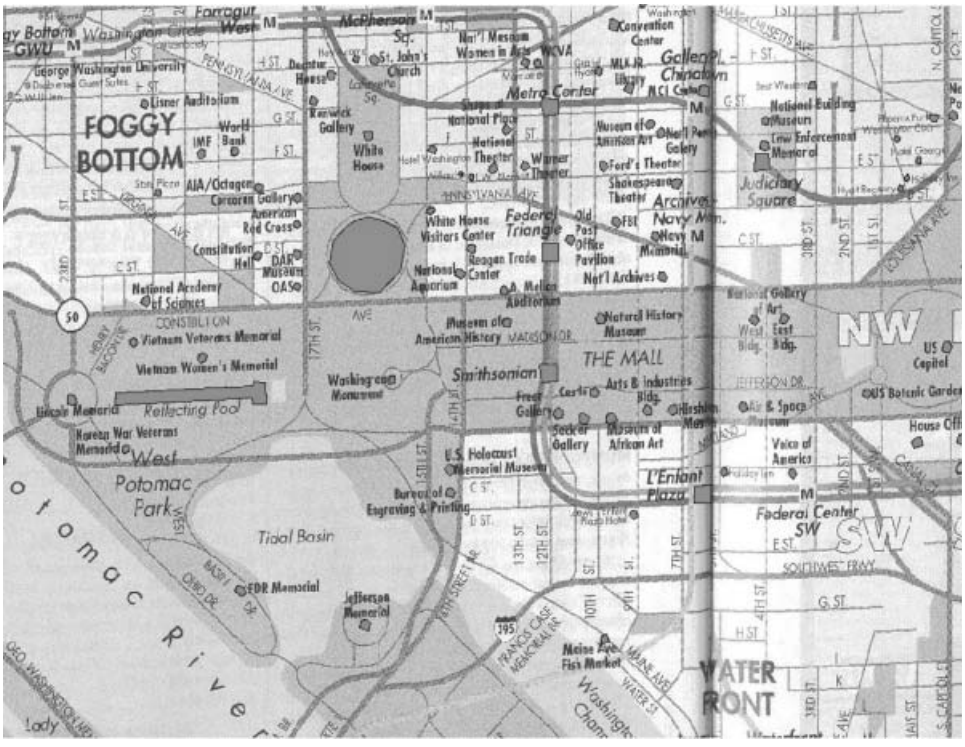

Figure 18. Source 5: magazine tourist map 1.

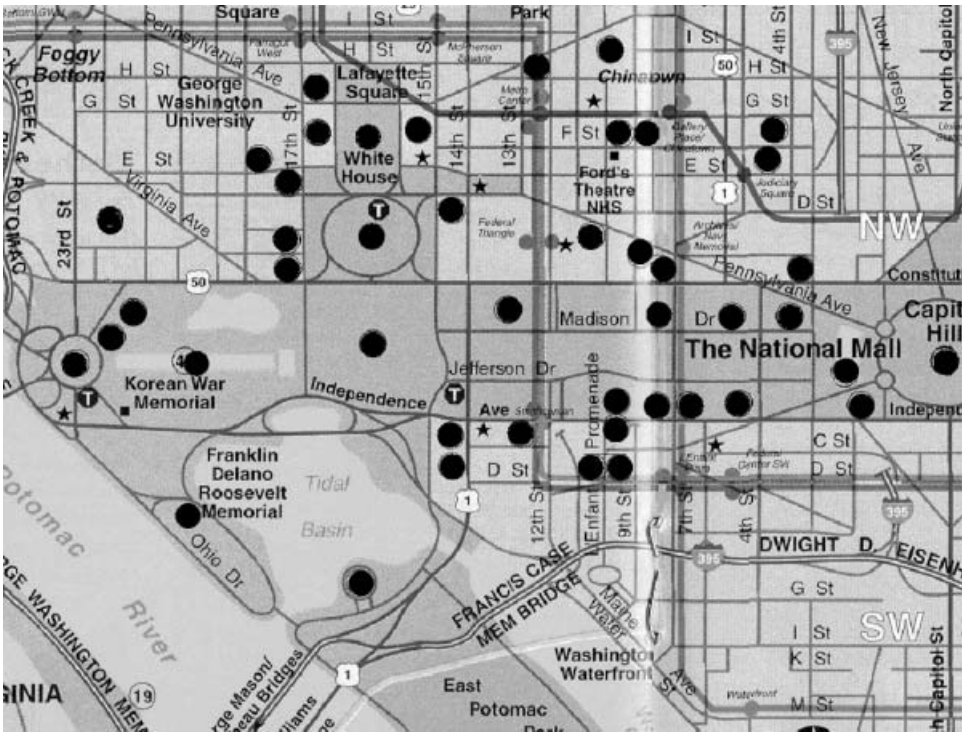

Figure 19. Source 6: magazine tourist map 2.

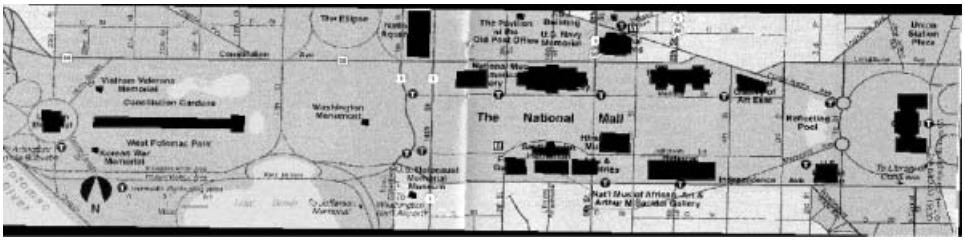

Figure 20. Source 7: magazine tourist map 3.

4.2. *Experiments*

The goal of the experiment is to evaluate how well the system matches features from disparate sources. This is not a simple task, since ground truth data are not readily available. Instead, the tests used a human (expert) generated set as ground data to measure the recall and precision of the automatically matched set.

The first step in the experiment is configuring an Arcview interface that allows the subject (i.e. human expert) to view all available data. The interface includes functions that allow sources to be zoomed, panned, and overlaid. Attributes for each feature are also easily accessible. For the experiment, the subject selects a feature of interest (FOI) from an arbitrary source. The subject then inspects all other sources and identifies any features that match the FOI. This set of features is the truth set that is used to evaluate the system.

The matching system is used to find matches for the same FOI. To accomplish

this, features from all sources within a pre-set radius of the FOI are considered for matching. The system processes the information and builds similarity sets with the input features. All similarity sets that contain the original FOI are used for system evaluation.

4.3. *Performance metrics*

The performance of the system is a function of two measures: precision and recall. They are defined for a similarity set with the following:

$$\text{Precision} = \frac{Number\_Correctly\_Matched}{Number\_In\_MatchingSet}$$

$$\text{Recall} = \frac{Number\_Correctly\_Matched}{Number\_In\_TruthSet}$$

where *Number_Correctly_Matched* is simply the number of features in the intersection of the two sets.

Feature ambiguity

Feature ambiguity is another metric that helps determine the effectiveness of the system. Ideally, the system would place each feature in a single similarity set. This is because each feature should represent only one real-world object. How well the system accomplishes this can be measured by counting the number of times a feature is a member of different similarity sets. The sum of the total number of features in each similarity set divided by the total number of features being matched gives an indication of the systems' confidence of feature matches. It should be noted that this metric does not evaluate whether the sets are correct. It simply indicates how well the sets cover the data independent of any truth set.

Clique confidence

Another metric is the confidence in the similarity sets. Simply computing the clique similarity for each similarity sets gives a sense of how well each clique matches. As with the feature ambiguity metric, this only gives a sense of the system's confidence in the output. It does not measure how well the output achieves the goal of matching features.

4.4. *Results and discussion*

Table 3 shows the performance of the system in terms of precision and recall. The left section of the table indicates the results without using contextual similarity, and the right side shows the results after the context is factored into the system. The confidence value associated with the best clique returned from the query is also listed.

The results show that the average recall is approximately 80%. The average precision is 73% without context and 84% with context, which indicates that using context improves the system performance. Note that the parameters used for these tests were chosen by experimentation. It is likely that different parameter values would slightly improve both recall and precision. The confidence measures do show a general correlation with the accuracy and precision. For most queries, higher confidence values indicate a higher accuracy and precision. The number of cliques

Table 3. Test results of the prototype system.

	No context				Context			
	Precision	Recall	Conf.	No. of cliques	Precision	Recall	Conf.	No. of cliques
Query 1	2/2	2/2	0.89	1	2/2	2/2	0.89	1
Query 2	2/3	2/3	0.89	2	2/3	2/3	0.79	2
Query 3	1/3	1/1	0.83	1	1/3	1/1	0.81	1
Query 5	3/7	3/6	0.82	2	5/5	5/6	0.81	3
Query 6	6/6	6/7	0.93	4	5/5	5/7	0.90	2
Query 7	5/6	5/7	0.93	1	6/6	6/7	0.81	2
Query 8	4/6	4/6	0.89	3	3/4	3/6	0.77	2
Query 9	5/5	5/5	0.91	2	4/4	4/5	0.86	2
Query 10	4/6	4/5	0.89	3	4/5	4/5	0.86	3

does not seem to show any correlation between the accuracy and precision of the query results.

4.5. *Effectiveness of contextual similarity*

Results in the previous section do not clearly show the benefits of using context, so further experiments were designed in which only context is used to measure the similarity between feature pairs. First, two features that are known to match are identified. Next, one of those features is compared with all other features in the other source. The results are ordered by their similarity values. If the similarity measure is meaningful and valid, then the known match to the feature should rank high on the list. The following describes the steps in the experiment.

1. Two sources, S^1 and S^2, are chosen for the experiment.
2. A truth set is manually produced, as described in the last section. It is a set of features pairs that are known to match. $T=\{\tau_1,\tau_2,\tau_3\cdots\tau_n\}$, where $\tau_i=\left\{F_j^1,F_k^2\right\}$.
3. A set of landmarks is determined. The landmark set is a subset of the truth set, and ideally its features are equally distributed throughout the intersection of the two sources areas. $\Lambda\subseteq T$.
4. A feature pair is chosen from the truth set less the landmark set. $\tau_i\in(T-\Lambda)$.
5. The similarity is measured between the feature from Source 1, in the chosen pair and all features from Source 2 in the set $(T-\Lambda)$.
6. The similarities are ordered by value, and the rank of the matching feature is determined.

The experiment is repeated for several features and several pairs of sources, with each taking turn to be the first and second source. If the context is useful, then the actual match for a feature should rank high in the list. The results are summarized in table 4. It shows the number of matches in the top 1, 3, and 5, respectively, by using location only and then by using both location and context.

The results show that the location information alone is usually inadequate. However, by adding just the context, one almost always gets all the matches in the

Table 4. Summary of results of experiments showing the benefits of context.

Landmark set no.	First source	Second Source	No. of features	Match in top 1		Match in top 3		Match in top 5	
				Location	Context	Location	Context	Location	Context
Set 1	Tourist map	Orthophoto	26	4	18	8	23	16	26
Set 1	Orthophoto	Tourist map	26	3	16	6	25	16	26
Set 2	Tourist map	Orthophoto	27	4	18	8	23	16	26
Set 2	Orthophoto	Tourist map	27	3	16	6	25	16	26
Set 3	Tourist map	Orthophoto	28	4	9	8	24	16	27
Set 3	Orthophoto	Tourist map	28	3	17	7	27	17	27
Set 4	Tourist map	Orthophoto	29	4	20	8	25	16	28
Set 4	Orthophoto	Tourist map	29	3	18	7	27	18	28

top 3. The results convincingly show the benefits of using geographic context in the feature-matching process.

5. Conclusions and future work

This paper presents a method for feature conflation, i.e. to match features from a set of disparate GIS sources. We assume that the sources are georeferenced and hence do not account for arbitrary rotations and translations. The approach attempts to use all available information about features for matching. It is based on a series of similarity measures for common GIS attribute types. The geographical context is modelled using proximity graphs, and the similarity of contexts is used as additional evidence for matching. Finally, a graph representing all possible matches is created and searched for groups that most likely represent the same real-world object. A prototype system demonstrates both the benefits of using the context and the effectiveness of the overall approach. For our analysis, we assume that the features have already been derived either manually or in an automated fashion. Many existing sources of data already have features extracted manually, and automated systems are gradually becoming more effective. For our experiments, we used manual extraction methods. The quality of results is comparable to those obtained by good automated systems, but lower than those derived manually by trained operators. Clearly, the quality of matching will be directly correlated with the accuracy of feature extraction.

Although the designed system achieved the basic goal, it has several shortcomings that we describe next for future exploration. The matching system uses a form of context based on distances and directions between point-like features. This is only one of several types of context. Other potentially very useful contexts can be used, for example, topological relations between features, such as A covers B, or A touches B, or A is within B. These relations are particularly useful for areal features (Bruns and Egenhofer 1995). Topological relationships could be useful for feature-matching data that are a mixture of point-like and areal. Relationships between polygonal features to polylinear features (i.e. buildings in relation to roads) and multi-source context relationships, instead of two sources, may also be beneficial. It may also be possible to chain contexts from multiple sources, e.g. when building A is adjacent to building B in *source 1,* and building B is south of building C in *source 2.* We now have two relationships about feature B that could be used for matching in *source 3.* In general, context is a multi-faceted idea that needs to be explored more thoroughly for additional benefits. This could boost the accuracy of the system, especially for ambiguous features.

Because this system is designed to accommodate imperfect data sources, opportunities exist to explore the integration of non-traditional data. Examples of potential sources are audio annotated maps, data mined from the World Wide Web, and hand-drawn maps. These are likely sources of unique and potentially useful data, but they are currently not well suited for GIS integration. A huge opportunity exists for research into methods for the integration and application of these largely untapped resources. If the results of the feature-extraction process are robust (not necessarily perfect or error-free), this approach can adjust reasonably well. However, if the data quality significantly deteriorates, the results of this approach may also be questionable. Systematic evaluation of the results as a

function of data quality is beyond the scope of this paper and is left as a future research topic.

Another issue that has not been addressed in this research is the problem of one-to-many matching. This may arise, for example, when a cluster of related buildings in one source are generalized and shown as a single feature to represent the whole complex in another source. Here, we outline two possible ways in which our approach may be extended to solve the one-to-many matching problem.

Our approach calls for a relaxation-based algorithm to determine matching features in two sources (Section 3.5) before identifying matching features in multiple sources by clique similarity (Section 3.6). The one-to-many matching can be incorporated in either step. In the first step, if there are multiple features in one source that strongly match a single feature in another source, these may be collapsed into a single feature before forming the total similarity graph for finding the clique similarity. Alternatively, we relax the requirement that the cliques solution should be node-disjoint. This may be a better solution because it takes the global context of all the sources in determining the one-to-many matches.

References

AOE, J. (ed.), 1994, *Computer Algorithms: String Pattern Matching Strategies* (Los Alamítos, Mexico: IEEE Computer Society Press).

BALLARD, D., and BROWN, C., 1982, *Computer Vision,* 1st edition (Englewood Cliffs, NJ: Prentice-Hall).

BERNHARDSEN, T., 2002, *Geographic Information Systems,* 3rd edition (New York: Wiley).

BRUNS, H., and EGENHOFER, M., 1996, *Similarity of Spatial Scenes. Proceedings of the Seventh International Symposium on Spatial Data Handling,* edited by M. Molenaar and M-J Kraak (London: Taylor & Francis), pp. 31–42.

CORMEN, T. H., LEISERSON, C. E., and RIVEST, R. L., 1992, *Introduction to Algorithms,* 8th edition (New York: McGraw-Hill, MIT Press).

CORTER, J. E., 1996, *Tree Models of Similarity and Association.* Sage University Paper Series on Quantitative Applications in the Social Sciences, Thousand Oaks, CA.

CUETO, K., 1999, *Matching Features in Geographic Information Systems.* MS Thesis, Department of Computer Science and Engineering, University of Nebraska-Lincoln.

CUETO, K., SAMAL, A., and SETH, S., 2000, *Context-Based Similarity for GIS Feature Matching, GIScience 2000,* pp. 288–290, October 2000.

DANA, P. H., 1999, *Geographers Craft: Coordinate Systems.* <http://www.colorado.edu/geography/gcraft/notes/coordsys/coordsys.html>. Accessed: 15 November 2002.

DEL BIMBO, A., 1999, *Visual Information Retrieval* (San Francisco: Morgan Kaufmann).

DUDA, R., and HART, P., 1973, *Pattern Classification and Scene Analysis* (New York: Wiley).

ESRI (Environmental Systems Research Institute Inc.), 1998, *ARC/INFO (7.2.1) Arcview (3.1) Online References.* http://www.esri.com/

FELLBAUM, C. (ed.), 1998, *Wordnet: An Electronic Lexical Database* (Cambridge, MA: MIT Press).

FLICKNER, M., SAWHNEY, H., NIBLACK, W., ASHLEY, J., HUANG, Q., DOM, B., GORKANI, M., HAFNER, J., LEE, D., PETKOVIC, D., STEEL, D., and YANKER, P., 1995, Query by image and video content. *Computer*, **28**, 23–30.

GEMAN, D., and JEDYNAK, B., 1996, An active testing model for tracking roads in satellite images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**, 1–14.

GIS/Trans, Ltd, 2003, Conflation Background Information, http://www.gistrans.com/products/cf_info.html.

GOODCHILD, M. F., 1995, Attribute accuracy. In *Elements of Spatial Data Quality,* 1st edition, edited by S. Guptill and J. Morrison (Oxford: Elsevier Science).

GOODCHILD, M. F., and HUNTER, G. J., 1997, A simple positional accuracy measure for linear features.. *International Journal of Geographical Information Science*, **11**, 299–306.

GUPTILL, S., and MORRISON, J. (eds.), 1995, *Elements of Spatial Data Quality,* 1st edition (Oxford: Elsevier Science).

HALL, P., and DOWLING, G., 1980, Approximate string matching. *Computing Surveys*, **12**, 381–402.

HELMUTH, S., 1980, *Cluster Analysis Algorithms for Data Reduction and Classification,* 1st edition (New York: Wiley).

JONES, C. B., 1997, *Geographical Information Systems and Computer Cartography* (Harlow, UK: Addison-Wesley Longman).

Leica Geosystems, 2003, IMAGINE Professional, http://www.gis.leica-geosystems.com/

LONGLEY, P. A., GOODCHILD, M. F., MAGUIRE, D. D., and RHIND, D. W., 2001, *Geographic Information Systems and Science* (New York: Wiley).

NADLER, M., and SMITH, E. P., 1993, *Pattern Recognition Engineering* (New York: Wiley).

RODRIGUEZ, A., and EGENHOFER, M., 1999, Assessing similarity among geospatial feature class definitions. In *INTEROP'99, Lecture Notes in Computer Science 1580,* edited by A. Vckorski, K. E. Brasel, and H-J Schek (Berlin: Springer), pp. 189–202.

RODRÍGUEZ, A., and EGENHOFER, M., 2003, Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knowledge and Data Engineering*, **15**, 442–456.

SAMAL, A., SETH, S., and CUETO, K., 2001, Like-feature detection in geo-spatial sources. In *Proceedings of the SPIE, Geo-Spatial Image and Data Exploitation II,* Volume SPIE, pp. 62–73.

SANKOFF, D., and KRUSKAL, J. (eds.), 1983, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison,* 1st edition (Reading, MA: Addison-Wesley).

SANTINI, S., and JAIN, R., 1999, Similarity measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**, 871–883.

SDTS, 1999, SDTS homepage. http://mcmcweb.er.usgs.gov/sdts/

SHAPIRO, L. G., and STOCKMAN, G. C., 2001, *Computer Vision* (New York: Prentice-Hall).

SONKA, M., HLAVAC, V., and BOYLE, R., 1999, *Image Processing, Analysis, and Machine Vision* (Pacific Grove, CA: Brooks/Cole).

STEFANIDIS, A., AGOURIS, P., BERTOLOTTO, M., and CARSWELL, J. D., 2002, Scale- and orientation-invariant scene similarity metrics for image queries. *International Journal of Geographic Information Science*, **16**, 749–772.

TVERSKY, A., 1977, Features of similarity. *Psychological Review*, **84**, 327–352.

UKKONEN, E., 1985, Algorithms for approximate string matching. *Information and Control*, **64**, 100–118.

Visual Learning Systems, Feature Analyst, 2003, http://www.vls-inc.com/software/software.html

YUAN, S., and TAO, C., 1999, Development of conflation components. In *Proceedings of the International Conference on Geoinformatics and Socioinformatics, Ann Arbor, MI,* pp. 1–12, http://www.umich.edu/~iinet/chinadata/geoim99/Proceedings/yuan_shuxin.pdf