INT. J. GEOGRAPHICAL INFORMATION SYSTEMS, 1995, VOL. 9, NO. 6, 621–635

Research Article

Text attributes and processing techniques in geographical information systems

MARK J. CARLOTTO†
Pacific-Sierra Research Corp., 1400 Key Blvd., Suite 700,
Arlington VA. 22209, U.S.A.

(Received 18 December 1991; accepted 22 June 1994)

Abstract. Text is an important data source in many GIS applications. In some applications, text data are in a format (e.g. tables) that can be directly converted into database records or objects. Other kinds of data such as historical records, field reports, and intelligence messages are unformatted and thus more difficult to incorporate into a GIS. Instead of simply attaching text to geographical data, an alternative paradigm is introduced where non-spatial data (free text as well as relational data) are converted into textual attributes. A textual attribute can be a free-text description of a geographically-referenced feature or event, relational data converted into a set of domain-specific terms, or a combination of domain-specific terms and free-text. An advantage of using textual attributes and text processing techniques to describe and retrieve geographical information is their extendibility. As the database changes over time and new attributes are required, attribute-value pairs can be converted into text attributes and added to the system without otherwise altering the database. A vector-based representation is used to efficiently encode text attributes. Such a scheme is well-suited for incorporating text data into an object-oriented GIS. Text vectors are stored in slots local to the objects they describe. As new data are added to the system, existing vectors are not affected. Thus the database is easy to maintain. Methods can be easily implemented within the object-oriented framework to measure the similarity of object vectors to the vector of another object or a query. The similarity or score can be combined with other similarity measures (e.g. spatial/temporal proximity) to allow geographical information to be retrieved by location, time, and content in a uniform fashion. Two case studies are presented to illustrate the usefulness of textural attributes and text processing techniques in describing, accessing, and visualizing geographical information.

1. Introduction

Text is an important data source in many GIS applications. In some applications text data are in a format (e.g. tables) that can be easily converted into database records or objects. Other kinds of text data (e.g. historical records, field reports, and intelligence messages) tend to be unformatted and thus more difficult to incorporate into a GIS. If free text is simply attached to geographical entities (e.g. via pointers to files) the GIS retrieve these data indirectly. Alternatively, if geographical data are attached to text files in a similar way, text retrieval techniques can access geographical information. If the geographical and free text databases are cross-referenced then the text retrieval system and GIS are able to work together (Dangermond 1988).

Information about geographical features can be divided into two broad categories:

† Please send all correspondence to Mark J. Carlotto, PSR/Boston, 5 Ryans Place, Beverly MA 01915, U.S.A.

geometrical or spatial attributes which describe the location and spatial extent of the geographical feature; and non-geometrical attributes which provide nominal and scalar information about the feature (Nagy and Wagle 1979). Stated in another way, attributes are either spatial or non-spatial and attribute values are either numbers, symbols, or numerical/symbolic data structures. Spatial attributes are typically numbers or numerical data structures that describe the location and spatial extent of the geographical entity. Location can be specified indirectly by symbols (place names) whose actual locations are defined numerically in a gazetteer. Non-spatial attributes are either numerical (e.g. population, vegetative index, etc.) or symbolic (e.g. land cover type, water colour, etc.). Numerical attributes usually take on a large number of possible values where symbolic attributes take on a smaller (usually a predefined) number of possible values. Numerical data can however be divided into ranges or quantized into a smaller number of values.

This paper proposes the use of textual attributes and text processing techniques to describe and retrieve geographical information. A textual attribute can be a free-text description of a geographically-referenced feature or event such as an intelligence message:

> Probable Scud battalion CP in operation since 1215 Z. Additional units sited in vicinity of H-3 pumpstation in dispersed defensive formation at UTM coordinates 37SES681448

a series of domain-specific terms, for example describing water quality from a field report:

> CategoryWater, WaterColorBrown, Turbidity < 1ft, BottomContentUnknown, Depth < 3ft, BottomColorDark, AquaticVegetationNone, AquaticOdorChemical.

or a combination of domain-specific terms and free-text:

> CategoryWater, WaterColorBrown, Turbidity < 1ft, BottomContentUnknown, Depth < 3ft, BottomColorDark, AquaticVegetationNone, AquaticOdorChemical. Creek along railroad bed. 4 inch diameter steel piping runs along road. Pond south of here has been filled in. Test wells are visible.

There are several potential advantages in representating geographical information in this way. First, a variety of techniques that have been developed for efficiently encoding and retrieving text data can be used to retrieve geographical information as well. An advantage of text retrieval techniques over conventional relational and object-oriented database retrieval techniques is their extendibility. A text retrieval system has to be able to handle new words as they are encountered. On the other hand it is difficult to alter relational and object-oriented databases. Yet in a GIS, attributes often change over time as the domain changes. Vrana (1989) calls this 'attribute temporality'. If geographical information can be represented in a textual format, as new attributes are required, attribute-value pairs can be converted into new terms and added to the system as needed without otherwise altering the database.

The organization of this paper is as follows: § 2 reviews text processing techniques with particular attention to vector-based approaches. The methodology is contained in § 3; a vector-based technique known as overlap or surrogate coding is described first. Then an architecture for vector-based text retrieval that is well-suited to an object-oriented GIS is presented. Finally several methods for converting relational data into textual attributes are outlined. In § 4 a prototype text-based GIS for retrieving geographically-referenced multimedia data is described, and two case studies are presented to illustrate the usefulness of textual attributes and text processing techniques in describing, visualizing, and accessing geographical information.

2. Background

The development of text processing techniques and systems has been motivated largely by the need to quickly search large on-line text databases automatically. In early systems, texts were manually indexed by a small number of key words. Retrieval involved matching key words to those in a query. Advances in computing technology have made automatic full-text retrieval systems such as those discussed by Salton (1986) and Stanfill and Kahle (1986) feasible. In these systems, the texts are indexed automatically. Salton (1975) describes a variety of indexing techniques in which word frequency statistics are used to select the most informative words as indices. For example, one approach is to eliminate common words (articles, prepositions, and adverbs), rank remaining words in order of their frequency of occurrence, and select those in the middle percentile (neither too frequent nor too infrequent) as index terms.

The performance of text retrieval systems is typically characterized by recall rate (number of relevant texts actually retrieved divided by the total number of relevant texts in the database) and precision (number of relevant texts retrieved divided by the total number of texts retrieved). A common problem that tends to reduce recall is words that actually appear in a text may be different than (although similar in meaning to) words used in the query. Salton (1975) suggests the use of a thesaurus to improve recall by adding similar words to the query to broaden the search. However by increasing the recall rate the precision is often reduced. One method of improving the precision is through the use of relevance feedback (Stanfill and Kahle 1986) where the user identifies texts that are most relevant to the query at hand and the text retrieval system uses those texts to find other similar texts from the database.

Three basic approaches are used for full-text retrieval: string matching, inverted index, and vector encoding. For string matching the search time is proportional to the number of words in the query times the average number of words per text file times the number of text files. A more efficient approach is to build an inverted index or table which lists texts by the words contained in the texts (Knuth 1973). The retrieval time is related to the number of words in the query and the number of texts in which each word appears. A disadvantage of using an inverted index for text retrieval is that it must be updated each time a new text is added to the database. It is a centralized data structure and is thus not well-suited to object-oriented systems.

A third approach is to represent the text by a vector and use various similarity measures to rank order and retrieve texts on the basis of their corresponding vector representations. In the simplest scheme, each element of the vector represents a word. Clearly this is not efficient since most domains involve thousands of words and most texts contain only a small percentage of all possible words. A more practical technique suggested by Sammon (1969) represents the relevance of words to a fixed set of categories (the domain) in a matrix that is used to compute a vector which represents the relevance of the text to those categories. The length of the vector is equal to the number of categories. Sammon's method is highly domain-dependent and thus not suitable for domains that may change over time. A third technique based on statistical properties of text (Shannon and Weaver 1949) represents texts by vectors that describe the frequency of character sequences known as N-grams (e.g. D'Amore and Mah 1985). An advantage of N-grams is that they are relatively insensitive to spelling errors since character sequences and not words are encoded. A disadvantage is that the transformation is not reversible, i.e., the encoded words cannot be recovered from an N-gram. The last type of technique considered uses overlapping or surrogate codes (Knuth 1973, Roberts 1979, Christodoulakis and Faloutsos 1984, Stanfill and Kahle 1986) where words are encoded as bit patterns in a binary vector. Although surrogate coding is sensitive to spelling errors, the words that were originally encoded can be recovered from the vector. A major advantage of surrogate coding is its simplicity—requiring little if any knowledge of the domain or language. The latter can also be seen as a disadvantage in certain applications since surrogate coding cannot be 'tuned' for particular domains or languages as in the previous two techniques.

In general all of the vector-based approaches are well-suited for incorporating text data into an object-oriented GIS. Text vectors can be stored in slots local to the objects they describe. As new data are added to the system, existing vectors are not affected. Thus the database is easy to maintain. Methods can be easily implemented within an object-oriented framework to measure the similarity of object vectors to the vector of another object or a query. The similarity or score can be combined with other similarity measures (e.g. spatial/temporal proximity) to allow geographical information to be retrieved by location, time, and content in a uniform fashion.

3. Methodology

3.1. *Converting free text into vectors*

Let $\mathbf{W} = \{w_n\}$ denote a text containing a set of N words where w_n is the n-th word. In overlap or surrogate encoding, each word is assigned a unique code that is associated with that word in a dictionary. A code consists of Q integers $\{k_q\}$ selected at random from the interval $0 \leqslant k_q < K$ which is equivalent to a K bit binary vector $\boldsymbol{\omega}$ with Q bits k_1 through k_Q set to one. The number of words that can be uniquely encoded in this way is equal the number of possible arrangements of K items taken in quantities of Q items at a time (regardless of order):

$$\frac{K!}{Q!(K-Q)!} \tag{1}$$

For typical values, $K = 1024$ and $Q = 4$, this number is quite large, about 50 billion. For efficiency, codes are usually generated as needed and stored in a dictionary $\mathbf{D}$. The dictionary transforms words into codes $\mathbf{D}$: $w \rightarrow \boldsymbol{\omega}$. Since the mapping is one-to-one, words can be recovered from their codes, $\mathbf{D}^{-1}$: $\boldsymbol{\omega} \rightarrow w$. Using a thesaurus, words with similar meanings can be assigned the same code.

Lexically speaking, a text is the sum of its words. The vector code of a text is thus the sum (bit-wise logical or) or the individual word codes:

$$\mathbf{W} = \bigcup_{n=1}^{N} w_n \tag{2}$$

$$\Omega = \bigcup_{n=1}^{N} \boldsymbol{\omega}_n = \boldsymbol{\omega}_1 + \boldsymbol{\omega}_2 + \ldots \boldsymbol{\omega}_N \tag{3}$$

where $\boldsymbol{\omega}_n$ is the vector code of the n-th word.

The number of words M that can be encoded in a vector of given length is limited by the probability that the codes of two or more words overlap and give rise to the code of another word. For example, let $K = 16$ and $Q = 4$ and assume the codes for 'tree',

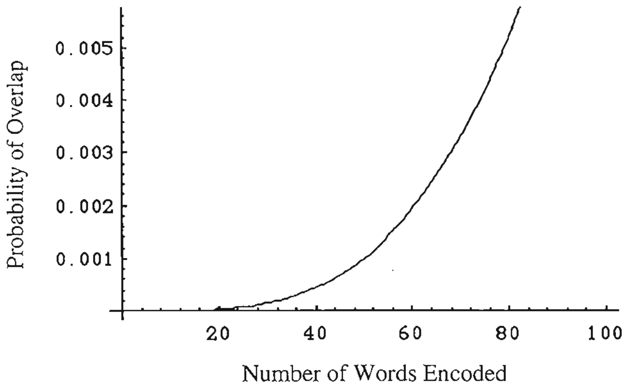


Figure 1. Probability of word overlap as a function of the number of words per text.

'house', and 'moon' are {1,6,9,11}, {2,6,10,13} and {1,2,6,13}, respectively. Encoding the first two words will create the same code as encoding all three words;

'tree'	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0
'house'	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0
'moon'	0	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0

The probability of overlap is derived by Roberts (1979) and is approximately equal to

$$P_{overlap} = \left(1 - \left(\frac{K-1}{K}\right)^{MQ}\right)^{Q} \tag{4}$$

For a given Q and M, this number can be made arbitrarily small by increasing K but at the expense of increased storage and computation. Figure 1 plots the probability of word overlap as a function of the number of words per text M for $K = 1024$ and $Q = 4$.

The words that are encoded in a text vector can be recovered by comparing the text code with each word code in the dictionary. If the Q bits of the n-th word code are set in the text code then the word is present. This can be written compactly in vector notation as

$$\text{If } \Omega^T \boldsymbol{\omega}_n = Q \text{ then } w_n \text{ is present} \tag{5}$$

where $\bullet^T$ denotes vector transpose. The actual complexity in recovering all the words $\{w_k\}$ from a text code Ω is equal to the number of words in the dictionary times the number of bits per word.

3.2. *Vector-based text retrieval*

An advantage of surrogate coding is that once a text has been converted into a vector, the similarity between the text and a query or another text can be computed using vector operations. The similarity can then be used to rank and retrieve the texts with the highest similarity or score.

Often binary vectors are compared using the Hamming distance. For two binary vectors $\mathbf{\Omega}$ and $\mathbf{\Psi}$ the Hamming distance $\mathbf{\Omega} \oplus \mathbf{\Psi}$ is the number of bits that are different. It can be shown that

$$\mathbf{\Omega} \oplus \mathbf{\Psi} = \mathbf{\Omega}^T\mathbf{\Omega} + \mathbf{\Psi}^T\mathbf{\Psi} - 2\mathbf{\Omega}^T\mathbf{\Psi} \tag{6}$$

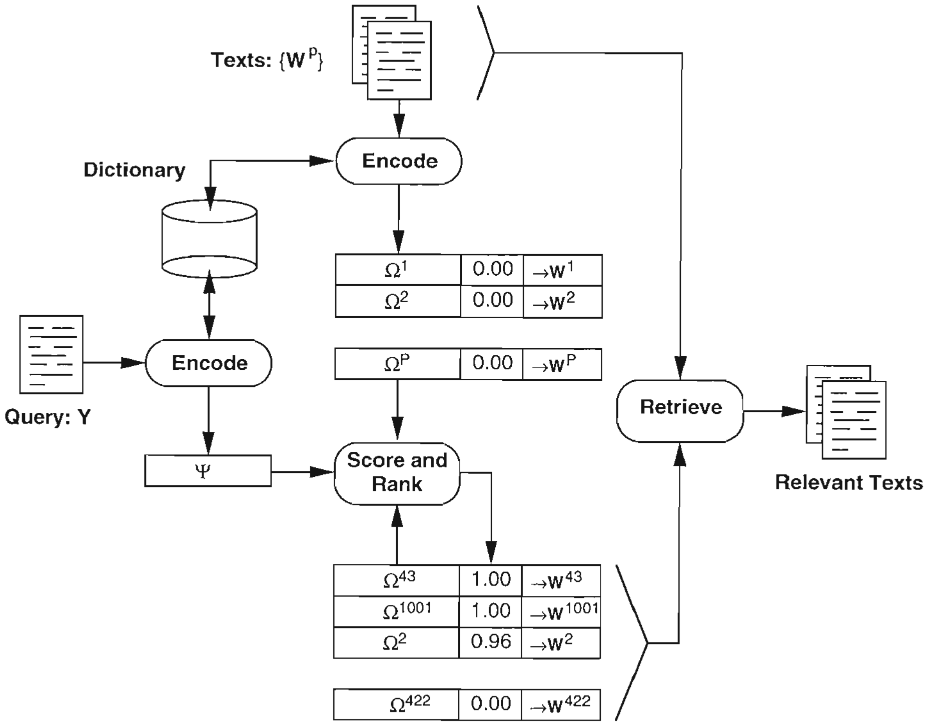


Figure 2. Overview of text encoding and retrieval process.

where $\boldsymbol{\Omega}^T\boldsymbol{\Omega}$ and $\boldsymbol{\Psi}^T\boldsymbol{\Psi}$ are the number of ones in $\boldsymbol{\Omega}$ and $\boldsymbol{\Psi}$ respectively, and $\boldsymbol{\Omega}^T\boldsymbol{\Psi}$ is the inner product or correlation of $\boldsymbol{\Omega}$ and $\boldsymbol{\Psi}$. The number of ones in a vector is related to the number of words in the corresponding text. The correlation between two vectors is thus related to the number of words common to the two corresponding texts. When the number of ones in $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ is fixed, the Hamming distance is inversely related to the correlation. If the number of ones varies between text files, the Hamming distance will not be a good similarity measure because shorter texts will tend to have a smaller Hamming distance between them than longer texts. The correlation depends on the number of words common to both texts and is affected to a lesser degree by differences in size between texts.

Let $\mathbf{Y} = \{y_m\}$ be a query of M words and $\{\mathbf{W}_p\}$ be a database of P texts. If $\boldsymbol{\Psi}$ and $\{\boldsymbol{\Omega}_p\}$ are their corresponding vector codes, the similarity between the query and the p-th text is defined as

$$s_p = S(\mathbf{Y}, \mathbf{W}^p) = \frac{\boldsymbol{\Psi}^T\boldsymbol{\Omega}^p}{\boldsymbol{\Psi}^T\boldsymbol{\Psi}} \tag{7}$$

If $\mathbf{W}^p$ contains all of the terms in the query then $s_p = 1$. Values less than one indicate fewer terms are present.

S can thus be used to rank order texts for retrieval. Figure 2 depicts the text encoding and retrieval process. Texts $\mathbf{W}^p$ are converted into vectors $\boldsymbol{\Omega}^p$ which are stored in objects and organized in a queue. Associated with each $\boldsymbol{\Omega}^p$ vector is a score s_p between 0 and 1 and a pointer to the corresponding text $\rightarrow \mathbf{W}^p$. The query $\mathbf{Y}$ is converted into the vector $\boldsymbol{\Psi}$. To retrieve those texts that are most similar to the query $\mathbf{Y}$, the scores between the

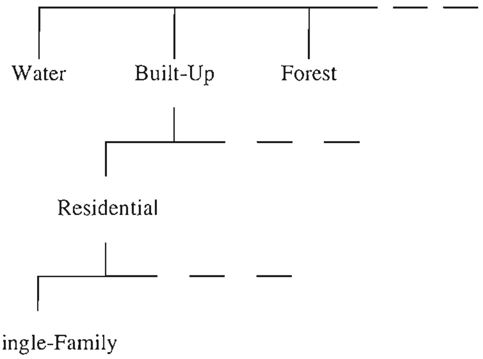


Figure 3. Portion of Anderson's land use/land cover classification system.

vectors are computed and used to sort the corresponding objects (vectors and associated file pointers) in descending order. Objects containing texts that are most similar to the query are placed at the top of the queue where they can be most easily reviewed and selected by the user.

The above architecture can also handle multiple queries. Let $s_{p,t} = S(\mathbf{W}^p, \mathbf{Y}^t)$ be the score of the p-th text with respect to the t-th query. To find texts that match all T queries; i.e.,$\mathbf{Y}^1 \cap \mathbf{Y}^2 \cap \ldots \mathbf{Y}^T$, the minimum score

$$\min_{t=1}^{T} \{s_{p,t}\} \tag{8}$$

can be used as a similarity measure. To find texts that match at least one query; i.e., $\mathbf{Y}' \cup \mathbf{Y}^2 \cup \ldots \mathbf{Y}^T$ the maximum score

$$\max_{t=1}^{T} \{s_{p,t}\}$$

can be used. If a text matches the p-th query $s_{p,t} = 1$. Thus, if the text matches all T queries the min and max scores will be one. If it matches at least one of the queries the min score will be less than one and the max score will be one.

3.3. *Converting relational data into free-text attributes*

A relational database can be thought of as a table of data (the rows) organized into fields (the columns). Fields contain symbols (strings) or numbers. For example the field 'LandCoverType' might have possible values 'Water', 'Forest', 'Built-up', and so forth while the field 'CanopyClosure' would contain numbers between 0 and 100 (per cent).

Symbolic fields are easily converted into free-text attributes by concatenating the field name with the field value; e.g., 'LandCoverTypeWater', 'LandCoverTypeVegetation', 'LandCoverTypeBuiltUp', etc. For the query 'LandCoverTypeVegetation' all texts that contain that term would have a score of one.

If symbols are organized in a hierarchy, their position in the hierarchy can be encoded using multiple terms. For example, consider the portion of Anderson's land use/land cover classification system (Anderson *et al.* 1976) shown in figure 3. Level I categories are encoded as described above. A Level II category would be encoded with two terms; e.g., 'LandCoverTypeResidential' and 'LandCoverTypeBuiltUp', while a Level III category would be encoded with three terms; e.g., 'LandCoverTypeSingle-Family', 'LandCoverTypeResidential' and 'LandCoverTypeBuiltUp'. By using mul-

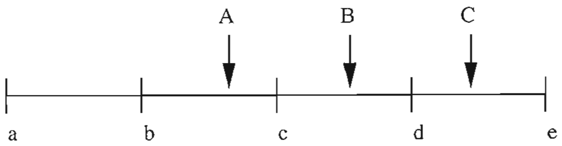


Figure 4. Encoding order.

tiple terms the ability to generalize the search is explicitly built into the representation. In other words one can retrieve features that are built up area, or residential areas, or areas containing single family dwellings.

Numeric fields can be discretized and represented in the same way as symbolic fields; e.g., 'CanopyClosure' can be divided into ranges to produce the terms 'CanopyClosure0–25%', 'CanopyClosure25–50%', 'CanopyClosure50–75%', and 'CanopyClosure75–100%'. However queries involving numerical fields often use predicates other than 'is equal to' (which was implied above) such as 'greater than', 'less than', and 'between'. In order to represent the order in numerical data we divide the range of possible values into intervals and use multiple terms to encode the position of the value with respect to the intervals. With reference to figure 4, the position of A on the real number line satisfies the following relations: $A > a$, $A > b$, $A \leqslant c$, $A \leqslant d$, and $A \leqslant e$. Let the attribute canopy closure be divided into $L = 4$ ranges (0–25%, 25–50%, 50–75%, and 75–100%). The value A would be represented by the following $L + 1$ terms:

$\mathbf{P}_A = \{$'CanopyClosure $> 0\%$', 'CanopyClosure $> 25\%$', 'CanopyClosure $\leqslant 50\%$', 'CanopyClosure $\leqslant 75\%$', 'CanopyClosure $\leqslant 100\%$'$\}$.

The following three queries:

$\mathbf{Y}_{>25} = \{$'CanopyClosure $> 25\%$'$\}$,
$\mathbf{Y}_{\leqslant 100} = \{$'CanopyClosure $\leqslant 100\%$'$\}$, and
$\mathbf{Y}_{>25, \leqslant 75} = \{$'CanopyClosure $> 25\%$', 'CanopyClosure $\leqslant 75\%$'$\}$

would find any geographical feature with this value of canopy closure since $S(\mathbf{P}_A, \mathbf{Y}_{>25}) = S(\mathbf{P}_A, \mathbf{Y}_{\leqslant 100}) = S(\mathbf{P}_A, \mathbf{Y}_{>25, \leqslant 75}) = 1$. It can be shown that the difference between attribute values encoded in this way is inversely related to the similarity between the corresponding vectors; i.e., with respect to figure 4, $S(\mathbf{P}_A, \mathbf{P}_A) > S(\mathbf{P}_A, \mathbf{P}_B) > S(\mathbf{P}_A, \mathbf{P}_C)$.

Often there is an implicit similarity in a set of symbols. Similarity can also be represented in an explicit way using multiple terms. Consider wind direction (figure 5). North is more similar or closer to North-east than it is to East. Let the symbols 'WindDirectionNorth', 'WindDirectionNorth-east', and 'WindDirectionEast' be represented by the following three sets of terms:

$\mathbf{P}_N = \{$'WindDirection315', WindDirection0', WindDirection45'$\}$,
$\mathbf{P}_{NE} = \{$'WindDirection0, WindDirection45 WindDirection90$\}$, and
$\mathbf{P}_E = \{$'WindDirection45', WindDirection90', WindDirection135'$\}$,

where each is in turn encoded in a vector $\mathbf{\Phi}_N$, $\mathbf{\Phi}_{NE}$, and $\mathbf{\Phi}_E$. North is closer to North-east than it is to East since $S(\mathbf{P}_N, \mathbf{P}_{NE}) > S(\mathbf{P}_N, \mathbf{P}_E)$. The number of terms used defines a neighbourhood. For the above case with $L = 3$ terms, $S(\mathbf{P}_N, \mathbf{P}_N) > S(\mathbf{P}_N, \mathbf{P}_{NE}) > S(\mathbf{P}_N, \mathbf{P}_E) > S(\mathbf{P}_N, \mathbf{P}_{SE})$. Outside the neighbourhood all symbols are equally dissimilar, $S(\mathbf{P}_N, \mathbf{P}_{SE}) = S(\mathbf{P}_N, \mathbf{P}_S) = S(\mathbf{P}_N, \mathbf{P}_{SW}) < S(\mathbf{P}_N, \mathbf{P}_E)$.

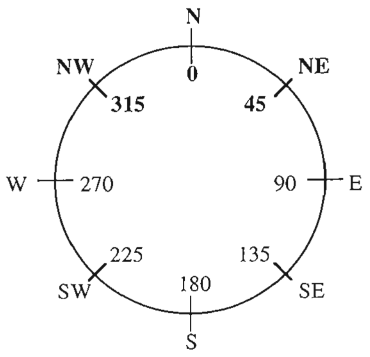


Figure 5. Encoding similarity or closeness.

Since there is a circular order to wind direction, $S(\mathbf{P}_N, \mathbf{P}_{NE}) = S(\mathbf{P}_N, \mathbf{P}_{NW})$; $S(\mathbf{P}_N, \mathbf{P}_E) = S(\mathbf{P}_N, \mathbf{P}_W)$; $S(\mathbf{P}_N, \mathbf{P}_{SE}) = S(\mathbf{P}_N, \mathbf{P}_{SW})$.

4. Application and case studies

This section describes a software system known as *HyperMap* for accessing geographical data using free-text attributes. Two case studies are then presented. The first illustrates the combined use of spatial, temporal, and textual queries for interpreting a small database of simulated intelligence messages. The second case study presents results from an environmental monitoring study where ground truth in the form of a set of relational database tables are converted into free-text files and processed within *HyperMap*.

4.1. *HyperMap*

HyperMap is a prototype software system for organizing, accessing, and analysing multimedia data (text, pictures and video) in a geographical context. It is currently implemented in Common Lisp on a Macintosh and consists of several thousand lines of code.

The user interface or workspace consists of a collection of tools and windows for accessing and displaying multimedia object data and maps. Maps in this context refer to multiple coregistered images over a given geographical area (e.g., digitized topographic maps, Landsat imagery, digital elevation models, and land cover and environmental feature images derived from the Landsat data). Objects are points of interest or events within this area and are described by geographic or UTM coordinates, date/time designators, and free-text descriptions. Multimedia data such as pictures, sound, and video may be associated with any object. All data are stored as standard Macintosh files. The workspace is configured according to the size of the screen and the display requirements of the individual media. A dialogue window provides tools for quickly finding objects of interest based on their location, date/time, and content.

Figure 6 is an overview of *HyperMap*. The text processor converts ASCII files into a list of objects. Geographical coordinates and date/time information are extracted by a structured text parser. The parser matches user-defined templates that specify alternative ways the data might appear in the text; e.g., for date/time group

```
DD XXXYY
DDHHMM XXX YY
DDHHMM  XXX YY
```

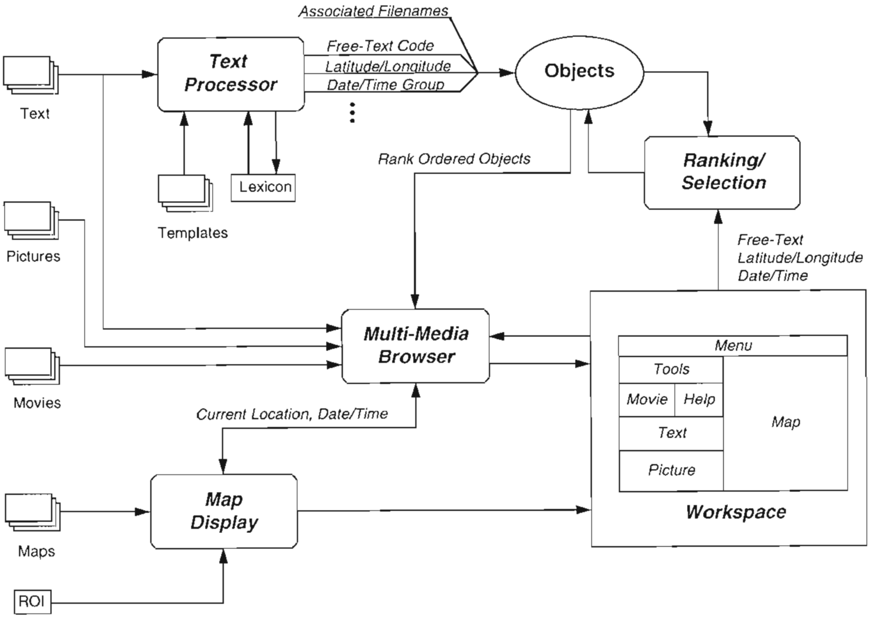


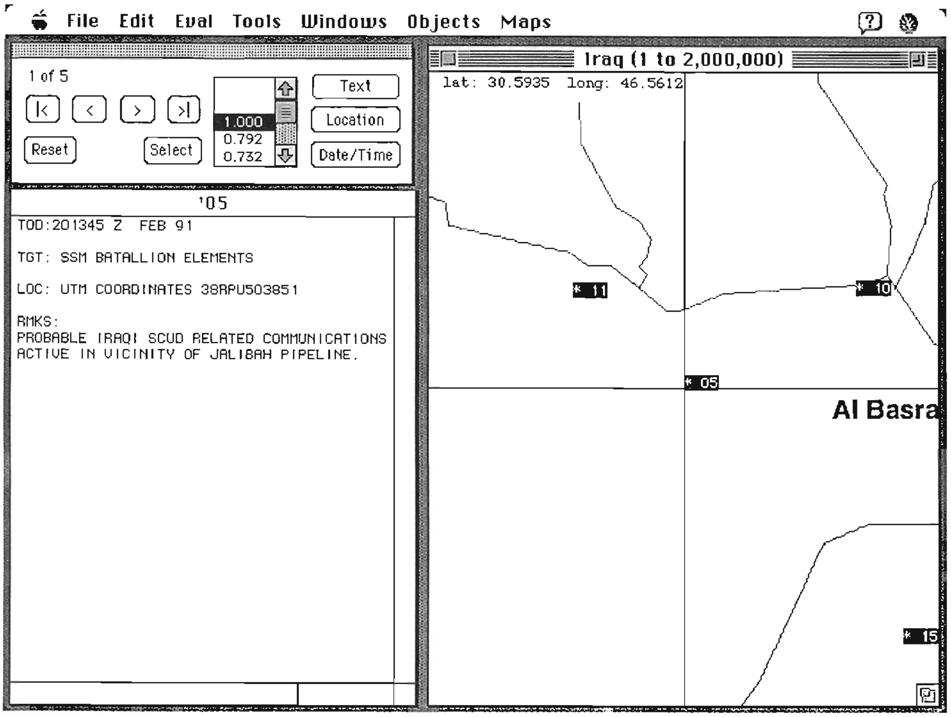
Figure 6. Functional architecture of *HyperMap* system.

DDHHMM XXX YY
DDHHMM XXX YY

where D, H, M, X, and Y are place holders for the day, hour, minute, month, and year. After the structured portion of the text has been parsed, the full-text is converted into a binary vector as was described in § 3. An alphabetical listing of all encoded words and terms is stored in a lexicon which can later be used to initiate key-word search.

Ranking and selection functions process spatial, temporal, and textual queries. A score is assigned to each object based on its proximity to a given point in space or time, or its similarity to a list of key words or an entire text. For text the score is equal to one when all of the terms in the query are present in the text and less than one otherwise. For spatial queries the range of distances between the query object and all other objects is normalized so that objects nearer to the query object have scores closer to one and those farther away have scores closer to zero. Temporal queries are handled in a similar fashion with objects/events nearer in time to the query object/event being assigned scores closer to one with others that are much earlier or later assigned scores closer to zero. The scores from successive queries can be combined in two ways. To find objects that satisfy the current query and preceding query(s), the scores are combined using a min operation. To find the objects that satisfy the current query or preceding query(s) the scores are combined using a max operation. In the first case study presented in the next section we search for intelligence messages that contain the terms: 'SCUD' or 'SSM' or 'TELS'. Messages that contain one or more of these terms will have a score of 1 while messages that contain none of these terms will have a score that is less than one. Those objects with the highest scores, in this case those equal to 1 can then be selected for display and further processing.

The multimedia browser is a tool for examining selected objects in the queue in order of their score. Objects with the highest score are at the top of the queue. Buttons move forward, backward to the beginning and to the end of the queue. The browser



```
'05
TOD:201345 Z  FEB 91

TGT: SSM BATALLION ELEMENTS

LOC: UTM COORDINATES 38RPU503851

RMKS:
PROBABLE IRAQI SCUD RELATED COMMUNICATIONS
ACTIVE IN VICINITY OF JALIBAH PIPELINE.
```

Figure 7. Message processing example.

automatically coordinates the display of information from text, picture, and movie files directly to the workspace including any updates to map displays.

The map display supports the interactive scroll-zoom of multiple linked map windows and display of geographical pixel coordinates of objects and other features in maps. Several display options are provided: in auto-scroll mode map displays are updated so that the position of the current object in the queue is in the centre of the screen; in overlay mode object locations and names are displayed over the map layer. Objects can also be selected interactively via the map display.

The workspace is automatically configured at start-up based on the screen size and display requirements. All windows support standard Macintosh 'cut-and-paste' operations. The menu-bar contains commands to open and close workspaces, to set different processing options, to select map layers, and to create multiple map displays. All window operations are event-driven by the mouse and/or key clicks within the active window. Other operations are initiated by buttons in tool dialogues. On-line help information is also provided.

4.2. *Message processing example*

The first case study considers the use of free-text attributes in processing military intelligence messages. A small database of 16 messages was generated over a 7-day period during the Iraqi conflict. A 1:2 000 000 map of Iraq and the surrounding area was used as the map background. Figure 7 shows a display from the *HyperMap* system. One of the messages is shown in the text window. The objective of the experiment was to search for groups of messages that relate to mobile missiles. To start the search, messages containing the terms 'SCUD' or 'SSM' or 'TELS' were found as described above. Display of these messages revealed a group of several messages in southern Iraq near the Kuwaiti border. The central message (05) was then selected and used to rank the surrounding messages by their proximity in space and time. Three of the four nearest messages in space and time that pertain to mobile missiles are shown overlaid on the map display.

4.3. *Remote sensing example*

The second case study explores the use of free-text attributes to represent relational data in situations where the database may change over time and/or contain unformatted (i.e., free-text) information. In this example *HyperMap* was used in an environmental remote sensing study to assess the utility of multi-spectral imagery in detecting environmental anomalies such as stressed vegetation, water pollution, and atmospheric plumes (Carlotto *et al.* 1992). A variety of data were involved: multi-spectral satellite imagery (Landsat-TM), image processing algorithm results, map data (elevation and land use), and ground truth (digitized photographs, video, and field reports). *HyperMap* was used to organize and analyse this information, in particular to retrieve ground truth over specific areas to evaluate image processing results, to view raw imagery, processing results, to view raw imagery, processing results, and maps together, and to assess the consistency of processing results from different dates over the same area.

Text files were created from ground truth data. Forms were developed to record information about the air, water and vegetation (table 1).

Table 2 shows how we convert water quality measurements into text attributes. The first two attributes are numeric, the rest are symbolic. Figure 8 is a screen shot from this application showing part of a text file and corresponding picture (left). The lower right window plots ground truth locations where water quality measurements were taken and where the turbidity (optical depth) was less than 1 foot. These points are displayed over Landsat band 5. The upper-right window shows a subset of these measurements where the depth was less than 3 feet overlaid on a turbidity image derived from the Landsat data. Two of the four water bodies in the database with a depth of less than 3 feet and a turbidity of less than 1 foot (sites *t* and *r*) have a relatively high turbidity as computed from Landsat. The other two water bodies were too small to be resolved in the imagery.

Each row in a relational table can be thought of as a point in a multi-dimensional attribute space. Traditional clustering and data visualization techniques can be used to identify patterns in the data, e.g., groups of geographical features that are similar to one another. Collections of text data can be similarly visualized (Carlotto 1994). Geographical features described by text attributes are represented by binary vectors of extremely high dimensionality. Non-linear mapping techniques can be used to project these vectors to a lower dimensional space. Figure 9 shows the entire set of ground truth data displayed in an information map. In this map points that are close together are similar in content. Unlike spatial maps the axes have no special significance. The different kinds of ground truth data are evident in the map. Those pertaining to air quality are at the top, water quality to the left and vegetation condition to the right. Those in between may contain measurements in more than one regime. The two sites with shallow and highly turbid water (*r* and *t*) are near the bottom-left.

5. Summary

Text is an important data source in many GIS applications, particularly those that must handle intelligence messages, field reports, and historical data. Instead of simply

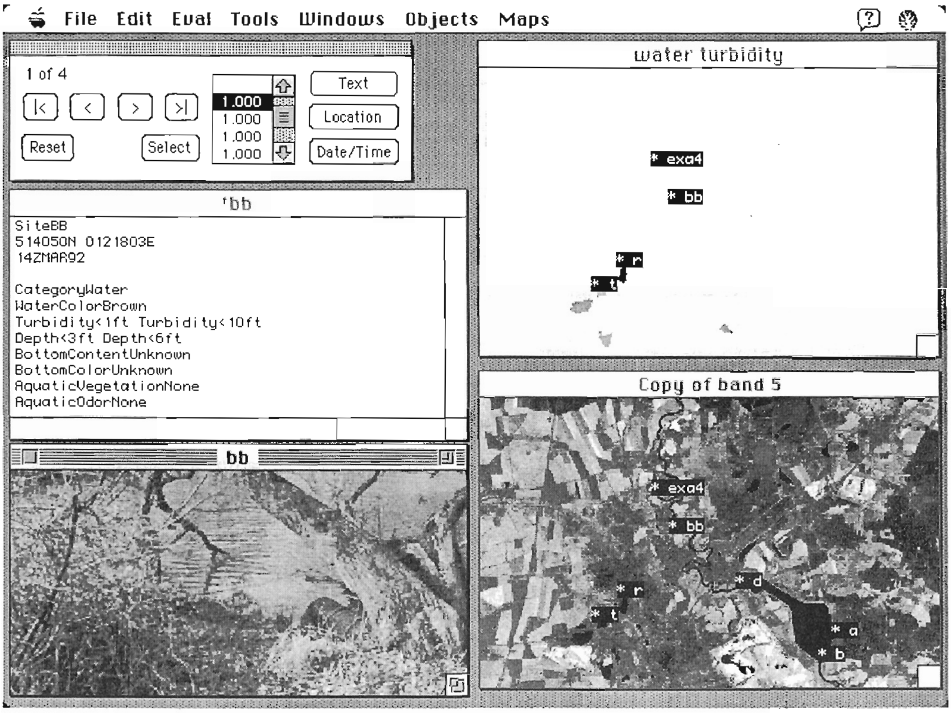
Table 1. Ground truth data collected for environmental remote sensing application.

Water quality	Air quality	Vegetation condition
Water colour	Visibility	Type
Bottom content	Smoke emissions	Health
Bottom colour	Air odour	Density
Aquatic vegetation	Wind direction	Closure
Aquatic odour		Height
Turbidity		Managed?
Depth		

Table 2. Water quality converted into text attributes.

Attribute	Values	Text attributes
Turbidity	Colloidal	Turbidity < 1 ft, Turbidity < 10 ft
	Turbid	Turbidity ≥ 1 ft, Turbidity < 10 ft
	Clear	Turbidity ≥ 1 ft, Turbidity ≥ 10 ft
Depth	Shallow	Depth < 3 ft, Depth < 6 ft
	Moderate	Depth ≥ 3 ft, Depth < 6 ft
	Deep	Depth ≥ 3 ft, Depth ≥ 6 ft
WaterColour	Blue	WaterColourBlue
	Green	WaterColourGreen
	Brown	WaterColourBrown
	Black	WaterColourBlack
BottomContent	Bedrock	BottomContentBedrock
	Cobbles	BottomContentCobbles
	Sand	BottomContentSand
	Muck	BottomContentMuck
	Unknown	BottomContentUnknown
BottomColour	Dark	BottomColourDark
	Moderate	BottomColourModerate
	Light	BottomColourLight
	Unknown	BottomColourUnknown
AquaticVegetation	None	AquaticVegetationNone
	Algae	AquaticVegetationAlgae
	Weeds	AquaticVegetationWeeds
AquaticOdour	Decay	AquaticOdourDecay
	Chemical	AquaticOdourChemical
	Sewerage	AquaticOdourSewerage
	None	AquaticOdourNone

attaching text to geographical data and using conventional database management and retrieval techniques, an alternative paradigm has been introduced where free text and non-spatial relational data are converted into textural attributes. The advantage of using textual attributes and text processing techniques to describe and retrieve geographical information is their extendibility; i.e., as the database changes over time and new attributes are required, text attributes can be added as needed to the system without otherwise altering the database. We have outlined a vector-based representation to efficiently encode text attributes that is particularly well-suited for object-oriented GIS. Text vectors are stored in slots local to the objects they describe. As new data are added to the system, existing vectors are not affected. Thus the database is easy to maintain. A prototype software system known as *HyperMap* was developed that embodies the



```
SiteBB
514050N 0121803E
14ZMAR92

CategoryWater
WaterColorBrown
Turbidity<1ft Turbidity<10ft
Depth<3ft Depth<6ft
BottomContentUnknown
BottomColorUnknown
AquaticVegetationNone
AquaticOdorNone
```

Figure 8. Environmental remote sensing example.

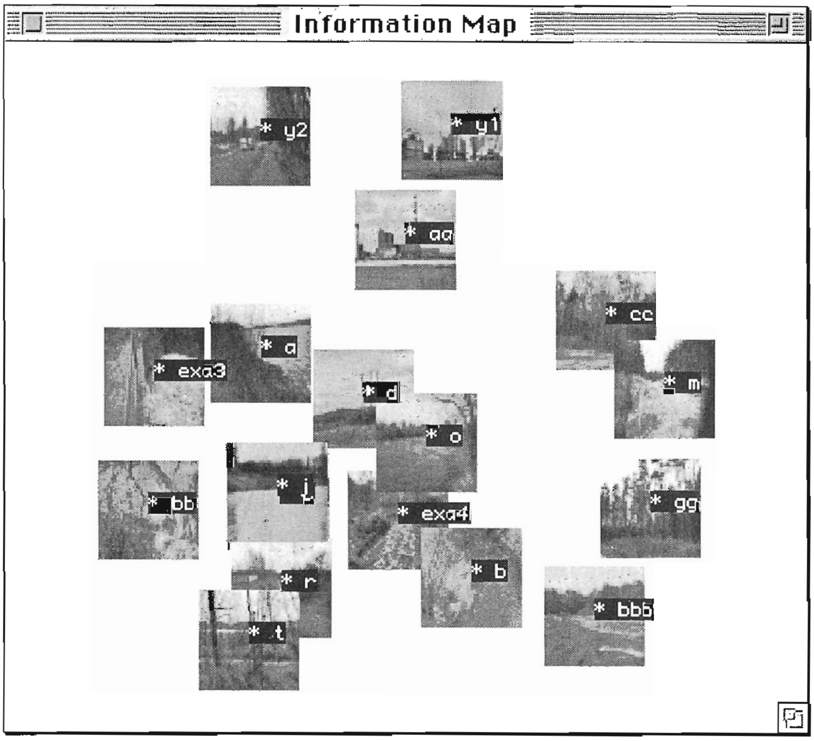


Figure 9. Information map visualization of database content.

above ideas. It provides interactive tools for quickly accessing geographical information based on an object's location, date/time, and content. *HyperMap* objects can contain text, pictures, and video information. The system supports a variety of map displays for visualizing geographical data spatially and by content. The content-based displays are computed automatically from textual attributes and provide an overview of the information content in a collection of geographical objects.

Two case studies were presented to illustrate the usefulness of textual attributes and text processing techniques in describing, accessing, and visualizing geographical information. Future work will concentrate on expanding the scope of these two experiments to larger data sets containing larger numbers of objects and on exploring other applications of the approach.

References

ANDERSON, J., HARDY, E., ROACH, J., and WITMER, R., 1976, A land cover and land classification system for use with remote sensor data. *U.S. Geological Survey Professional Paper 964*, U.S. Government Printing Office.

CARLOTTO, M., 1994, Nonlinear mapping algorithm and applications for multidimensional data analysis. *Journal of Visual Communication and Image Representation*, **5,** pp. 127–138.

CARLOTTO, M., LAZAROFF, M., and BRENNAN, M., 1992, Multispectral image processing for environmental monitoring. *Digital Image Processing and Visual Communications Technologies in the Earth and Atmospheric Sciences II, SPIE*, **1819,** pp. 113–124.

CHRISTODOULAKIS, S., and FALOUTSOS, C., 1984, Design considerations for a message file server. *I.E.E.E. Transactions of Software Engineering*, **SE-10,** pp. 201–210.

D'AMORE, R., and MAH, C., 1985, One-time complete indexing of text: theory and practice. *Proceedings 8th International ACM Conference on Research and Development in Information Retrieval*, pp. 155–164.

DANGERMOND, J., 1988, A review of digital data commonly available and some of the practical problems of entering them into a GIS. In *Fundamentals of Geographic Information Systems: A Compendium*, edited by W. Ripple, (Bethesda, MD: American Society of Photogrammetry and Remote Sensing).

KNUTH, D., 1973, *The Art of Computer Programming: Sorting and Searching* (Vol. 3), (Reading, MA: Addison-Wesley).

NAGY, G., and WAGLE, S., 1979, Geographic data processing. *Computing Surveys*, **11,** pp. 559–563.

ROBERTS, C., 1979, Partial-match retrieval via the method of superimposed codes. *Proceedings of the I.E.E.E.*, **67,** pp. 1624–1642.

SALTON, G., 1975, *A Theory of Indexing* (Philadelphia: Society of Industrial Applied Mathematics).

SALTON, G., 1986, Another look at automatic text-retrieval systems. *Communications of the ACM*, **29,** pp. 648–656.

SAMMON, J., 1969, A nonlinear mapping algorithm for data structure analysis. *I.E.E.E., Transactions on Computers*, **C-18,** pp. 401–409.

SHANNON, C., and WEAVER, W., 1949, *The Mathematical Theory of Communication* (Urbana: University of Illinois Press).

STANFILL, C., and KAHLE, B., 1986, Parallel free-text search on the Connection Machine system. *Communications of the ACM*, 29.

VRANA, R., 1989, Historical data as an explicit component of land information systems. *International Journal of Geographical Information Systems*, **3,** pp. 33–49.