Optimal Spatial Data Transmission for Mobile Terminals based on Object Oriented Geographical Information Systems

Shohei MIYAGI and Morikazu NAKAMURA
University of the Ryukyus

This paper proposes an optimal spatial data transmission in a network-base object-oriented GIS. Since spatial data are represented as collection of feature objects in object-oriented GISs, flexible selection can be allowed for the spatial data transmission. Our propose is to reduce the size of spatial data with quality keeping, therefore, we try to select spatial data by taking account of usefulness of data and the band-width and display size of user's terminal. In this paper, we formalize the selection problem as an integer programming, and then develop an exact algorithm based on branch and bound (B&B) approach and a heuristic algorithm.

オブジェクト指向地理情報システムにおける モバイル端末のための最適空間データ転送

宮城尚平　名嘉村盛和
琉球大学工学部情報工学科

本論文はオブジェクト指向地理情報システムにおける最適空間データ転送法を提案する．オブジェクト指向地理情報システムでは，地理データがオブジェクトの集合として表現されるため，転送空間データの構成において柔軟な地理情報選択が可能となる．提案手法の目的は，情報の質をできるだけ落さずに転送する空間データのサイズを削減することである．すなわち，地理情報の価値，通信のバンド幅，利用者端末の表示能力等を考慮して転送空間データセットを構成する．本稿では最適空間データ転送問題を整数計画問題として定式化し，分枝限定法に基づく厳密解法と貪欲法に基づく近似解法を提案する．

1 Introduction

Information including geographical positioning data is called spatial data. Geographic Information Systems (GIS) are a database and also an information processing system for spatial data. Recently, GISs are regarded as an software infrastructure for computerized society[3]. A mobile GIS is a style of network-base GIS in which mobile terminals are connected to GIS servers by wireless communication systems. In this style, the spatial data to be transmitted from a GIS server seems to be very heavy for mobile terminals because of narrow band-width, not enough memory, and small display capability.

To overcome these problems, we propose a method to reduce automatically the size of spatial data to be transmitted to mobile terminals. In the mobile GIS, generally users use not all of spatial data sent from the server, just a small amount of information. Therefore, we try to select spatial data to transmit to a user by taking account of the usefulness of the data. Though "What is useful?" is a difficult question, we just consider the amount of the social common value and user's personal interests of spatial data and then regard the selection problem as a combinatorial optimization.

In this paper, we formulate the selection problem for the spatial data transmission as an integer programming and show that the problem is a knapsack-type optimization problem. This formalization can be easily done since we employ the object-oriented GIS in which we can treat flexibly spatial data because the data are managed as a collection of feature objects. On the other hand, layer files are managed in traditional GISs.

2 System Outline

Traditional GISs treat spatial data as a collection of layers. A layer is constructed with *a subject* such as "city-boundary", "road networks", "building location", and so on. By overlaying layers, various maps are generated in GISs. On the other hand, in object-oriented GISs we treat spatial data as a collection of objects.

2.1 Feature Class

A object is a feature on the earth such as "a building", "a part of road", "a part of river", and so on. Collection of objects constructs a map in object-oriented GISs. We are developing a GIS based on object oriented technology[1]. In our system, a feature object is the fundamental unit of spatial data. We refer to the object design for the feature in OGM(Open Geodata Model) proposed by Open GIS Consortium[2]. Figure 1 illustrates simply the feature architecture designed in our system. A feature consists of Name, ID, Schema ID, Geometry, Property and also includes other features as child elements.

A Geometry expressing the geospatial information (including position data) is represented by a set of Point, Curve, Surface objects. A Surface is composed of some Curve objects, A Curve is composed of some Point objects, and a Point object is the most fundamental element of the coordinate geometry.

A Property is a set of Item objects composed of pairs of Key and Value. An Item is used to express attribute information.

2.2 Multilayer Structure

Our GIS system is composed of three layers, the presentation layer, the function layer and the data layer. Figure 2 shows the structure of the system. The role of each layer is shown as follows:

- The Presentation Layer: This layer is to give interface to users, that is, displaying spatial objects and receiving user's request and pass to the function layer.
- The Function Layer: The function layer processes request sent from the presentation layer and gives/takes spatial data to/from data layer. The proposal of this paper, the spatial data selection, is implemented in this layer. The function layer may have several function sub-layers.

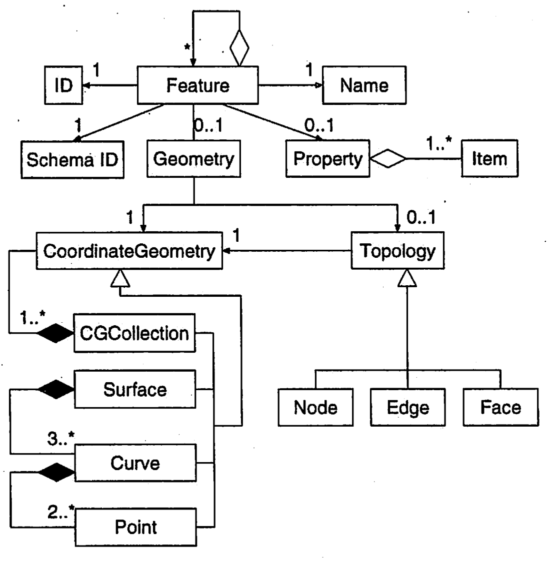

図 1: Feature Class

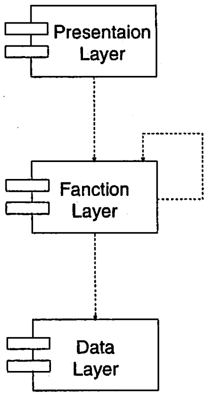

図 2: Multilayer Architecture

- The Data Layer: The data layer plays a role of the database in this system. This layer stores and manages all the data.

2.3 Mobile Terminal Environment

In the mobile GIS environment, the presentation layer is run on a mobile terminal. The user requests for the GIS server to search spatial data with some conditions.

The function layer receives these requests and performs spatial data retrieval with the data layer. And then the function layer returns the retrieval profile to the presentation layer(user). The profile data includes the number and size of the spatial data. After referring the profile, the user can indicate how the presentation layer transfer the retrieval result to the user, that is, the total size of spatial data, interest spatial data type, attribution information, and so on.

The function layer selects spatial data set to be transfered to the presentation layer according to the indication from the user.

3 Spatial Data Selection

Before transmission, the function layer needs to select spatial data. In this section, we consider spatial data selection but we exclude here road objects, which is treated in Section 4.

3.1 Integer Programming Problem

Let $F = \{f_0, f_1, f_2, \cdots, f_n\}$ be the set of spatial objects (excluding road objects) retrieved from the data layer in accordance with user's request. Object f_i includes its position, geometric information, name, established date, and so on. We call these information except for position *attribute information*. That is, an object is composed of "position data" and a set of "attribute information". The data size and value of object f_i are denoted by $s(f_i), v(f_i)$, respectively. Moreover, simplified objects by reducing attribute information are also stored and managed in the database. A simplified object f_i^k is an object removed some attribute information from f_i^{k-1}. Note that f_i^0 is the original object.

Let us consider here the case that there are just two simplified objects, that is, f_i^1 and f_i^2 for easier explanation. So, now we denote the spatial data set including simplified objects by $\hat{F} = \{f_0^0, f_0^1, f_0^2, f_1^0, f_1^1, f_1^2, f_2^1, \cdots, f_n^0, f_n^1, f_n^2, \}$. Without losing generality, the following relations hold.

- $s(f_i^j) > s(f_i^k)$ if $j < k$.
- $v(f_i^j) > v(f_i^k)$ if $j < k$.

The upper-bound of the transmitted size are indicated by the user or can be automatically calculated from the band-width, memory size, and display size of user's terminal. Let us denote the bound by C. Therefore, we can formulate now the selection problem as follows:

$$V = \max \sum_{i,k} v(f_i^k) \cdot x_i^k \quad (1)$$

$$s.t. \quad \sum_{i,k} s(f_i^k) \cdot x_i^k \leq C \quad (2)$$

$$\forall i, \sum_{k} x_i^k \leq 1 \quad (3)$$

$$\forall i, k, x_i^k \in \{0, 1\} \quad (4)$$

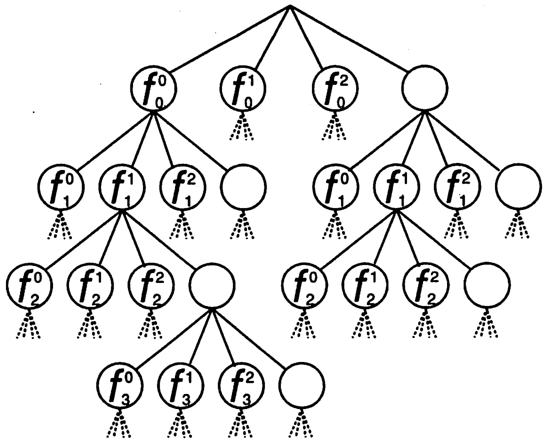

図 3: Search Tree

Variable x_i^k represents whether Feature f_i^k is transmitted to a user or not, that is, if x_i^k is 1, f_i^k will be transmitted. We can observe that this problem is a kind of 0-1 knapsack problem, which is known as NP-hard.

In this paper, we present two types of algorithms. One is a B&B algorithm to obtain the exact solution and the other is a heuristic algorithm with (very efficient) polynomial time complexity.

3.2 Selection Algorithm

The searching tree of the problem can be drawn as shown in Figure 3. The depth of the tree is the number of original objects, that is, $|F|$ and each node has four children, each corresponding to f_i^0, f_i^1, f_i^2, and no selection. Therefore, solution candidates mean here a path from the root to a node in the tree. All we have to do is to find a path such that the total weight of the objects on the path is no more than C and the total value is larger than any other candidates. However, such a path can be found only by searching all paths. That is, getting the optimum path requires $O(4^n)$ time complexity.

For efficient searching, we construct the tree by ordering the objects from the root with respect to the descending order of the value per size of the object, that is, $v(f_i^k)/s(f_i^k)$.

3.2.1 Branch & Bound Approach

We use the B&B method as a technique to get the exact solution more efficiently than the simple depth first search. Firstly, we can get some solution candidates after a few searching. And we evaluate at an intermediate node the upper bound of the objective value of the subtree before the next searching. If the estimation value is smaller than the value of the best solution candidate found so far, we do not continue searching on this subtree because we can not get any better solutions under the node. In this way, we can reduce scale of the search tree by the B&B way. Reducing size of the search tree means shortening search time. This approach may need exponential order time complexity in the worst case, but in the best case it can require reasonable computation time.

For the estimation, we employ the linear relaxation technique. That is, x_i^k relaxes from {0,1} to (0, 1). Firstly, the objects are ordered with respect to the descending order of the value per size of the objects, $v(f_i)/s(f_i)$, and renumber the subscript index from 0 by incrementing by one such that $h < k$ if $v(f_h)/s(f_h) < v(f_k)/s(f_k)$. We find j satisfying the following expression:

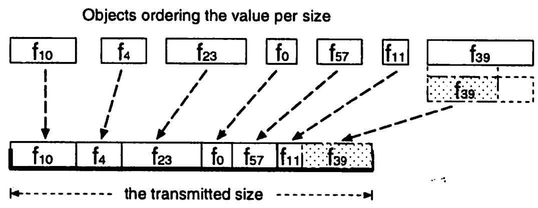

図 4: Linear Relaxation

$$S = \sum_{i=0}^{j} s(f_i) \leq C < \sum_{i=0}^{j+1} s(f_i) \tag{5}$$

That is, j is the largest number such that the sum of the object size from f_0 to f_j is less than C. In the linear relaxation, the objective value V is maximized when the values of x_i^k are as follows:

$$x_i = \begin{cases} 1 & (0 \leq i \leq j) \\ \frac{C-Size}{s(f_i)} & (i = j+1) \\ 0 & (j+2 \leq i < n) \end{cases} \tag{6}$$

where $Size = \sum_{i=0}^{j} s(f_i)$. That is, f_{j+1} is partiality selected in this solution (See Figure 4). Clearly the objective value V can be a upper bound of the original problem. Therefore, the relaxation can be used as estimation for bounding.

3.2.2 Approximate Solution Approach

The purpose of approximate solution approach is to get approximate solution in reasonable time with a little sacrifices of solution quality. The approximate solution approach proposed in this paper is based on *greedy-algorithm.* In the approach, only one path is searched instead of searching all paths. And such a path is treated as approximate solution.

In this approach, we first construct a priority list *PL* with respect to the descending lexicographical order of $(v(f_i^0), s(f_i^0))$. And then the original objects f_i^0 are selected until the total size of the selected objects reaches α_0% of C, simplified object f_i^1 are α_1%, and so on. The detailed description is shown in Figure 5. Determining the values of α_i is various. We employ the values: $\alpha_0 = 50$, $\alpha_1 = 30$ and $\alpha_2 = 20$ without special reason. The time complexity is of $O(n)$ though it can not ensure the optimality of the solution.

3.3 Numerical Example

We apply our approaches to spatial data of a small town in Okinawa, Chatan-cho. Table 1 shows numerical data of the exact and approximate solutions. While the original data set is 3.2MB and 11683 objects, the total transmitting data is set as 50KB and then 837 and 960 objects are selected. The approximate rate from the exact solution is just less than 13%. The computation time is quite reasonable.

4 Road Network Objects

As explained in Section 2, a road object is a part of a road network. Therefore a road in the real life is represented as a sequence of road objects in object-oriented GIS. For the road object selection, we can not

procedure ApproximateSelection;
begin
  var $S := 0, \alpha := 0, V := 0, j := 0$
  construct **PL** w.r.t. the lexicographical
  order of $(v(f_i^0), s(f_i^0))$;
  for $i := 0$ to m **do begin**
    $\alpha := \alpha + \alpha_i$;
    while $S < \alpha * C$ **do begin**
      $j := j + 1$;
      $S := S + s(f^i_{PL(j)})$;
      $V := V + v(f^i_{PL(j)})$
    end
  end
end;

図 5: ApproximateSelection

apply directly the same algorithm as the one in Section 3 since the connectivity of road networks can be easily broken. Such situation should not be allowed in many GIS applications. Therefore, we need to introduce a different treatment for road objects.

The objective of the spatial data selection is to reduce the data amount. Therefore, for road objects, all we have to do is to perform the data reduction with keeping the connectivity. The easiest way is to construct *road network objects* by combining neighboring road objects and treat them in the selection process instead of road objects, that is, we regard a road network object as just a spatial object. In this way, the connectivity problem is overcome since road network objects are not separable.

The reduction should be performed in the road network object construction process. For example, we can omit narrow roads, or roads on redundant paths, and so on. Let r^0 be the original network object including geometry of all the road objects. Then, simplified road network objects $r^k, k = 1, 2, ...$, can be defined with keeping connectivity. For example,

- r^1: excludes road objects on paths whose width is less than 3[m],
- r^2: excludes road objects on paths whose width is less than 5[m],
- ..... ,

or r^j can be the network object includes road objects on the shortest path network with respect to the main cross-points, and so on. So now we can prepare a set $\hat{R} = \{r^o, r^1, r^2, ..., r^k\}$ and apply the algorithm in Section 3 for $\hat{F} \cup \hat{R}$ for the spatial data transmission.

5 Conclusion

In this paper, we proposed selection algorithms for the spatial data transmission in Mobile GIS environments. By taking account of social value of the feature object and personal interests, we try to reduce the size of them. We presented both of the B&B algorithm and greedy-algorithm for the spatial data selection.

Acknowledgement

This work was partially supported by Grants-in-Aid for Scientific Research, No.13450165 and No.13780250, of the Ministry of Education, Culture, Sports, Science and Technology of Japan

表 1: Comparison of Solutions (C=50KB)

	Original Set	Approx. solution	Exact solution
Comp. Time	-	1s	12s
Total Objective Value	11683	837	960
Total Data Size	3.2Mb	50Kb	50Kb
Approx. Rate	-	12.81%	0
# of f_i^0	3551	22	17
# of f_i^1	0	19	38
# of f_i^2	0	42	0

参考文献

[1] Y. Nie, M. Nakamura, H. Miyagi, K. Onaga, "A Design and Implementation of Spatial Data Transfer System with Object Oriented Technology -Ryutan," Theory and Applications of GIS, Vol.8, No.2, pp.51-58, 2000.

[2] OGC Web Site, http://www.opengis.org/.

[3] S. Machida, "Geographic Information System initiation & master," Sankaido, 1998.

[4] T. Ibaraki, M. Fukushima, "Technique of optimization," , 1993.