인터넷을 통한 벡터 공간 데이타의 효율적 전송을 위한 최적화 기법

(An Optimization Strategy for Vector Spatial Data Transmission onover the Internet)

양 신 †　　이 충 호 ††　　배 해 영 †††
(Liang Chen)　(Chung-Ho LEE)　(Hae-Young BAE)

요 약 일반적으로 공간 벡터 데이타는 래스터 데이타에 비해 많은 정보를 포함하고 있으므로, 좀 더 융통적이고 효율적으로 데이타에 대한 처리가 가능하다. 그러나 인터넷을 통한 공간 벡터 데이타의 조작 시 해결해야 할 문제로 좁은 대역폭을 갖는 인터넷에서 크기가 크고 복잡한 벡터 데이타를 어떻게 효율적으로 전송하는가 라는 문제이다. 본 논문은 좁은 대역폭을 갖는 인터넷을 통한 공간 벡터 데이타를 효율적으로 전송하기 위한 새로운 전송 기법인 스케일에 기반한 전송 기법을 제안한다. 제안된 기법의 아이디어는 보여질수 있는 것만을 전송하는 것이다. 특정 스케일에서 일부 피처만이 사용자에게 보여지므로, 자연히 스케일은 공간 피처와 연관된 요소이다. 제안된 기법은 웨이블릿에 기반한 지도 일반화 알고리즘을 통해 공간 객체 중에서 출력되는 스케일에 따라 보여질 필요가 없는 피처들을 필터링하고, 보여지는 피처만을 최종적으로 전송한다. 본 논문에서는 실험을 통해 제안된 기법을 사용하는 경우, 개개의 공간 연산들에 대한 응답 시간이 대체적으로 향상됨을 보인다.

키워드 : 웹 지리정보시스템, 전송 기법, 벡터 공간 데이타, 지도 일반화

Abstract Generally, vector spatial data, with richer information than raster spatial data enabledata, enables a more flexible and effective manipulation of the data sets. However, one of challenges against the publication of vector spatial information on the Internet is the efficient transmission of the big and complex vector spatial datadata, which is both large and complex, across the narrow-bandwidth of the Internet. This paper proposes a new transmission method, namely, the Scale-Dependent Transmission method, with the purpose of improving the efficiency of vector spatial data transmission on the narrow-bandwidthacross the Internet. Simply put, its main idea is "Transmit what can be seen"" . Scale is regarded as a factor naturally associated with spatial features so that not all features are visible to users at a certain scale. With the aid of the Wavelet-Wavelet-based Map Generalization Algorithm, the proposed method filters out invisible features from spatial objects according to the display scale and then to transmit onlytransmits only the visible features as athe final answer for an individual operation. Experiments show that the response times ofan individual operation has been reducedoperations were substantially by the usage ofreduced when using the proposed method.

Key words : Web GIS, *transmission method, vector spatial data, map generalization*

† 비 회 원 : 중경 우전대학교 GIS연구센터
liangchen76@yahoo.com
†† 학생회원 : 인하대학교 대학원 전자계산공학과
chlee@dblab.inha.ac.kr
††† 종신회원 : 인하대학교 전자계산공학과 교수
hybae@inha.ac.kr
논문접수 : 2000년 12월 21일
심사완료 : 2003년 2월 18일

1. Introduction

During the recent dozens of years, spatial information has been increasingly popular to a point that this kindtype of information has been as common as alphabetical data for people. Vector spatial data usually carrycarries much richer information than raster spatial data, even when corresponding to the same spatial data set. Vector spatial data can enableset, and enables a more flexible and complex manipulation and analysis. Meanwhile, the Internet, as the greatest publication medium, promotes the popularity of spatial information. The Internet is a favorable means for GISGeographic Information System (GIS) users to exchange GIS data, conduct GIS analysis and present GIS output [1]. Geographic information on the Internet has been rapidly evolving with the change ofchanges in the Internet and web technologies [2,3,4,5]. Unfortunately, such complex geographic phenomenaon the earth have made spatial data characterized of complexness and large volume. And there is nocharacteristically complex and a large number. Currently, there is not enough bandwidth forthe Internet users to transmit large-sized datalarge data sets, such as spatial data and multimedia, especially fora large number of modem users. These unique characteristics have already brought new challenges to the implementation of GIS. And now they haveGIS, and have now become more critical when brought ontointo the Internet.

Generally, there are two methods to publish geographic information classified by data format. One is raster-based whileand the other is vector-based. The raster-based method proves efficient for a number of reasons. The size of raster data is smaller and relatively constant regardless of feature complexity [6], and raster datatheir transmission algorithms are simple. This method works well for digital images, and satellite data products. Server-based GIS' usually employs this method to reduce the size of data transmission. Presently, spatial data transmission mainly relies upon the raster-based method. However,the raster data are not equipped with as much extensible information asthe vector data. Any sophisticated manipulation and analysis operation must be performed by the server on the related vector data sets. As a consequence, the server has become greatly worse in the response time as clients increase and also the system has becomeserver's response time suffers as the number of client's increases while the system also becomes less stable.

On the other hand, the vector-based method has many significant advantages over the raster-based one on the Internet. Firstly, clients can reuse vector data repeatedly whenafter being transmitted just once, which has conservedconserves the valuable bandwidth of the Internet. Secondly, sophisticated operations can be performed by the client. Therefore, the system load is distributed over servers and clients, and the system has better scalability for the view of system performance.offers a better scalability. Thirdly, vector data are ready for complex analyses, which would certainly discoverprovide more information out offrom the spatial data for individual users. However, the complexness and largenesscombined complexity and size of vector spatial data and the low speed of the Internetcombined have posed a big challenge for the publication of vector spatial information. Predictably, an efficient transmission methodagainst this challenge would facilitate thesophisticated manipulation of spatial information onover the Internet.

This paper will proposeproposes a new method for vector spatial data transmission on the narrow-bandwidth of the Internet by examining the association between spatial features and scales and taking advantage of a map generalization method. It is empirically established that not all features are visible on a map withat a certain scale since features are associated with scales.the scale. According to [1], “As scale decreases, lines should become less irregular.” In cartography or map generalization, based on a desired scale of interest, cartographers select real world objects or concepts and determine an appropriate way to depict them in map form [7]. The resulting map is effectively a visualization of spatial information. Of course, it is extremely important to present spatial features as accurate and precise as possible. But this point by no means demands that all features on a map must be visible to users at all the time. Thereby, if a server transmits the complete data of spatial objects to a client without any consideration about the display scale on the client, theclient's system, invisible data cannot update the visual fidelity, but onlyfidelity but only serve to increase the transmission cost. To exploit the narrow bandwidth of Internet to the extreme, Internet, the efficiency of data transmission should be placed in the first place.highest regard. This paper defines the efficiency of data transmission as the ratio in size between the visible data and the transmitted data in accordance with the purpose of GIS. SoThus, to eliminate such superfluous data could promisinglywould improve the transmission efficiency.

In this paper, to improve the transmission efficiency and the response time, a Scale-Dependent Transmission method is proposed to filter out invisible data from the query answer, according to the display scale, with the aid of the Wavelet- based Map Generalization Algorithm and then transmits only the data that can contribute to the visual data quality. However, Internet or Web-based GIS systems are usually oriented to common people instead of GIS professionals and seldom offer more than read-only operations. Therefore, an Internet-wide revision will not be covered in this paper.

The remainder of this paper is organized as follows. Section 2 gives an overview of related work. Section 3 describes the Scale-Dependent Transmission method, the data structures and the implemented algorithms. Section 4 discusses the processing of spatial operations. Section 5 gives the performance evaluation model and discusses the performance evaluation and comparisons quantitatively. Finally, section 6 states the conclusions and future works.

2. Related Work

With the increase in the availability of the Internet and the application of spatial information, it has now become one of the major concerns of GIS professionals to manipulate vector spatial data across the Internet. So many studies have been conducted on how to efficiently transmit vector spatial data. Some methods have proposed the transmittance of vector data in a progressive process [6,8,9], and were inspired by a similar idea in progressive raster data transmission. The level of detail of a map gradually grows as the data stream flows from a server to a client until a user interrupts the transmission or the query answer is transmitted completely. To maintain the integrity of the spatial data set and to enable the client to perform operations on incompletely transmitted data, resulting from the user's interruption, spatial data are separated among various levels according to the scale of their objects. Usually, the procedure for map division, while preserving the integrity of the spatial information, is very complex. A common way to enable the progressive transmission of vector files is to pre-compute a sequence of consistent representations at lower levels of detail on the server site from the fully detailed representation and to transmit them in order of increasing detail [6].

However, the method proposed in this paper exploits another limitation of computer visualization. Because of the limit on the display scale of the client, some details of the map cannot be visually available, even when their data are transmitted. Therefore, such superfluous data increase the transmission cost without improving the visual quality of the resulting map. Different presentations of a map at scales can be pre-computed or be computed at the run time according to the display scale on the client. Therefore, this proposed method is of substantial difference from the progressive transmission, even though the considered situation — a slow communication link and particularly large-sized datasets — is the same. It takes advantage of the client's limitation of visualization and eliminates invisible data at the display scale and, thus, reduces the response time. Obviously, there is no such concept in progressive transmission. Analogically, the transmission of vector data is generally done by means of a long, one-step process [8] and the progressive transmission separates this one step into many smaller ones, while the method proposed in this study shortens this step.

The association between spatial features and the scale is far from original. Nowadays, most Internet GIS' use a display scale to determine the list of layers to be transmitted. However, the grouping of spatial object into a thematic layer is common. In this type of data organization, it is very likely that objects in one layer have a very different range of scale, and this combination of scales usually makes the visible scale range of the layer too wide to be effective for transmission cost reduction. If the visible scale range of one layer is narrowed down, some objects that should be presented at that scale are also eliminated along with other invisible objects. Therefore, to determine if either all the data or nothing within a whole layer are transmitted seems too arbitrary. In contrast, our method examines each object rather than the layer, thereby avoiding such arbitrary decisions.

3. The Scale-Dependent Transmission

This section proposes a scale-dependent transmission method to improve the transmission efficiency and reduce the response time of an individual operation.

3.1 "What is seen is what is transmitted"

It is believed that everyone understands that a map may have different representations at different scales. In GIS, spatial objects, such as lines and polygons, mainly consist of an array of vertexes, each of which is composed of X and Y coordinates. The significant vertices refer to those that are regarded as noticeable in the shapes and locations of the objects on a map [9]. At any scale, a simplified object should both preserve the main element of its shape, and recognize topological properties. Although not all features appear at a certain scale, users would feel comfortable when the recognizable features are preserved [1]. From the comparison between two representations of one map, in Figure 1, a lot of the detailed features in the left-hand representation do not appear in the other while the recognizable features are still preserved. Accordingly, the size of the spatial objects is much less. As well, if the simplified objects are only transmitted, the transmission cost will be decreased.

According to the' display scale, an appropriate representation should be trimmed or generalized. The map generalization method is used to extract

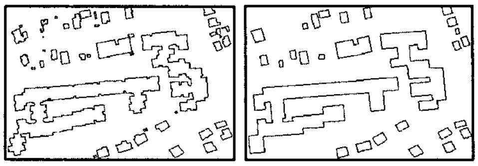


Figure 1 The Building Layer at Different Scales

visible and eliminate over-detailed data without contribution to the visual fidelity out of the initial answer to an operation. Then only the visible data is transmitted to the clients.

Because the visible data set is no larger than the complete data set in size, it is favorable for a slow communication link to transmit only the visible features or the data improving the visual quality of the map. Consequently, this method uses a generalization to reduce the transmission cost without damaging the visual fidelity. Figure 2 describes the system processing procedure where this method is applied. Upon receipt of a user's query with the client's display scale, a server generates a map as the original answer (Step 1) and thus generalizes it according to the received display scale (Step 2). Then, only the visible data is transmitted across the Internet (Step 3). When receiving the result map, the client displays it (Step 4) and replaces the corresponding cached objects with more detailed ones and, therefore, updates the scale of the cached data set (Step 5). Gradually, the size of the spatial objects sent by the server decreases while the scale of spatial objects increases.

Admittedly, it is quite time-consuming to generalize a map. Pre-computation is considered as an aid. The purpose of map generalization is to eliminate over-detailed features from objects and so there are more features in a complete map to eliminate than in a relatively less detailed one. Therefore, to generalize a map based on a pre-computed map, at a certain scale, is more efficient than directly using a complete map. On the other hand, to pre-compute many maps with different levels of detail demands a lot of time and space. Moreover, the defined levels of the details on a map in the pre-computation are quite subjective, not based on the client's requirements, so that further computations have to be done by the server at the run time. To avoid these two extremes, we pre-compute a sequence of maps at some scales with a relatively large gap between them in our simulated work. If the scale of the requested map is equal to one of these pre-computed maps, the pre-computed map is transmitted. Otherwise, the one closest to the current display scale is selected and the resulting map is computed on the basis of this map at the run time.

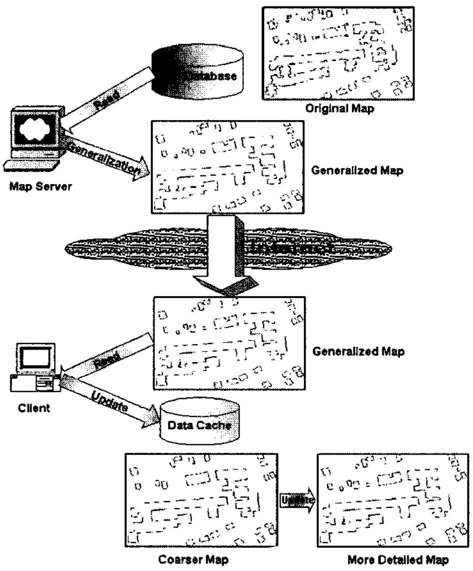


Figure 2 The System Processing Procedure

3.2 Map Generalization

Since over-detailed or invisible data damage the transmission efficiency, a map generalization algorithm is employed in this method to extract significant or recognizable features from an object according to the display scale of the map desired by the client.

Map generalization has been studied for many years. Some simple algorithms are purely based upon mathematical models and geometry, but more advanced algorithms take into account the characteristics of features, such as spatial relationships and patterns [10]. Possibly the most well known algorithm was proposed by Douglas and Peucker in 1973 and focuses on line simplification [11]. The simplification algorithm first joins the beginning and ending vertices of a line feature by a straight line, and then examines perpendicular distances to the individual vertices. Those closer than a selected threshold distance can be removed. The point furthest away is selected as a new end point for repetition of the process until there are no points closer to a line than the threshold. Recently, a wavelet-based line generalization technique was proposed [12]. This method, avoiding the conventional geometric computing of past methods, models the problem of linear feature simplification into that of signal processing and applies a wavelet to linear feature simplification. This new method has achieved O(n) in time complexity. Because it is feasible to organize most spatial objects as an array of points, the wavelet-based generalization algorithm can be used in the Scale-Dependent Transmission method.

Factors of deploying a generalization algorithm in our method will be described in section 5 in detail. With the purpose of improving the transmission efficiency and the response time, our method performs the map generalization only on the server, and never on the client.

3.3 The Structures for Transmission and Cache

This section defines the transmission and cache structures as defined for the Scale-Dependent Transmission method.

A transmission structure consists of a group of spatial objects. An object is composed of *Object Identifier* (OID), *Graphic Attribute* (GA), *Number Of Vertices and Vertexes*. The *Object Identifier* is an identifier of a spatial object while the *Graphic Attribute* is an attribute that is displayed, such as the line width, color, style and others. The *Number Of Vertices* are the number of vertices that are transmitted from the server to the client, and the *Vertex* is composed X and Y coordinates. The features of the object are represented using significant vertices, such as the turning point [10].

Moreover, an object has a field *flag* to indicate whether it is simplified or not.

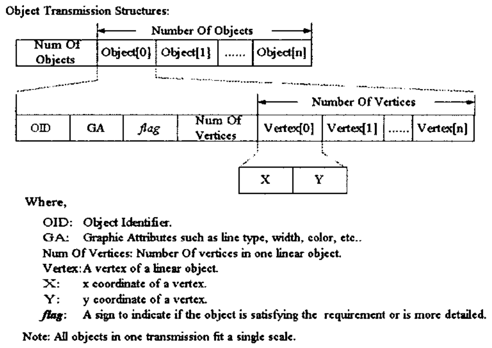


Figure 3 The Object Transmission Structures

Compared with the objects transmission structures, the cache structures have one additional field -Scale- through which the features of the spatial objects are associated with the scale. The following vertices are significant or visible at this scale. Figure 4 further describes the structure of an objects cache.

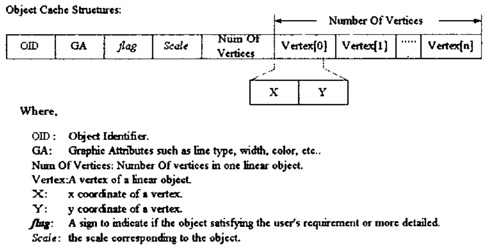


Figure 4 The Object Cache Structures

3.4 "Transmit What Can Be Seen"

In this paper, a scale is specifically defined as the number of pixels to describe one unit distance in the map. When a scale decreases, the number of apparent features lessens. Therefore, some features are not visible at a certain scale. The scale changes when operations such as ZOOMIN and ZOOMOUT are performed or the client's window is resized. At that time, the appropriate number of features should be extracted from the objects. The display scale is calculated with the following equation.

Scale = Max (ClientWidth/MapWidth, ClientHeight / MapHeight)

where, ClientWidth and ClientHeight are the size of the client window, in pixels, and MapWidth and MapHeight are the width and height of the map to be read.

When receiving a query, the client first determines the objects overlapping with the user's query MBR through the duplicate R-tree at the client. Because no update operation is considered, this duplicate copy does not cause problems with inconsistencies. Then it examines which´objects are cached and detailed enough to offer the presentation at the current display scale, and finally requests those objects that have not been cached. If a user demands a more detailed map (*NewScale* > *object.scale*), the generalization has to be performed on the complete objects by the server. If a less detailed map is requested, it can be rendered with the clients' cached objects with no additional generalization. If the scale is not changed, for example, when a user pans the map, they are certainly still meaningful. The following algorithm describes the above procedure.

Algorithm 1. QueryObjs (QueryMBR, ClientWindow)

Input: QueryMBR: the user's query MBR or the size of a map to be read.
ClientWindow: the size of the client window.

Output: RequestedOIDs: a set of object IDs to be requested from a server.
CurrentScale: the current display scale.

```
CurrentScale = Max ( ClientWindow.Width / QueryMBR.Width,
  ClientWindow.Height / QueryMBR.Height)
for "all objects overlapping with QueryMBR"
  if "object has not been cached at the client" then
      Add object.OID into RequestedOIDs
  else if "object.flag == 1 and CurrentScale !=
          NewScale" then
      if "NewScale > object.scale" then
```

```
07:             Delete object from Cache
08:             Add object.OID into RequestedOIDs
09:         end if
10:     end if
11: end for
12: Send RequestedOIDs and CurrentScale to the server
```

End Of Algorithm 1

In Algorithm 1, RequestedOIDs and CurrentScale serve as the input for Algorithm 2 (below), which is executed by the server. In addition, line 01 determines the current display scale at the client. Lines 02 through 11 record into the array RequestedOIDs the OIDs of the uncached objects (line 03) and those objects with a larger scale than the current one (line 05).

When receiving a request from the client, a server tries to generalize the requested spatial objects according to the received scale. If any practical generalization may damage an object's visual fidelity, its complete spatial objects have to be transmitted. Otherwise, the generalization is performed before transmission to the client. Algorithm 2 describes the server's response to the client's query.

Algorithm 2. MakeResult (RequestedOIDs, CurrentScale)

Input: RequestedOIDs: an array of object IDs to be requested from server.
CurrentScale: the current map scale on a client.

Output: ResultObjs: an array of spatial objects, either actual or simplified.

```
01: for "all OIDs in RequestedOIDs"
02:     Get object by the OID
03:     Clear object.flag with 0
04:     WaveletGeneralize (object) /*[12] described
        an algorithm implementation in a full detail*/
05:     Set flag with 1
06:     Add object into ResultObjs
07: end for
08: Send ResultObjs to client
```

End Of Algorithm 2

In Algorithm 2, ResultObjs are an array of generalized spatial objects whose structures are described in Figure 3. In addition, line 04 calls on the Wavelet-based Generalization Algorithm [12] to generalize an object. The *flag* of an object is set to 1 to indicate that it is generalized (line 05); otherwise it is complete (line 03). All processed objects are added to ResultObjs for transmission (line 06).

Because different objects have different scale ranges, the resulting map may contain both generalized and complete objects, which may also happen in the cache. When the display scale is reduced, the map becomes less detailed. Consequently, the number of generalized objects is larger and that of complete objects is smaller. With an increase in the display scale, the map becomes more detailed. Moreover, the client replaces objects with fewer features with their corresponding ones that have a greater number of features. Therefore, the number of generalized objects decreases while that of complete objects increases. Gradually, the objects will be displayed in an increasingly higher scale and with richer features. When a lower detailed map is demanded, the cached, higher-scale objects can be used directly to display the lower-scale representation and, therefore, no object is needed from the server. Of course, these complete objects can be exploited repeatedly once they are cached.

4. Query Processing with Scale-Dependent Tra

Spatial operations can be generally classified into two groups. One is basic operations, such as ZOOMIN, ZOOMOUT, and MOVE, while area measurements, polygon overlap and buffering belong to the other group — spatial analysis.

If the client has cached enough information, some basic operations can be performed on the client. Otherwise, they have to be performed on the server.

ZOOMIN — When a more detailed map is demanded, it is necessary to again simplify complete objects. After this operation, the client will replace objects shown at a smaller scale with ones at a larger scale.

ZOOMOUT — Before this operation is offered, objects at a larger scale have to be cached on the client; a less detailed map can be rendered with these cached, higher-scaled objects without generalization at the client, even if these cached objects might not be complete.

MOVE (map) — The characteristic of this operation is that the scale does not change. The client requests the objects that have not been cached and the server simplifies the requested objects according to the current scale and returns the results.

After generalization, some features may disappear from the map, so that the spatial analysis performed on generalized objects may be not correct or accurate. Therefore, it is better to perform such operations on complete objects in order to gain the correct and precise results. However, such operations do not always have to be performed on the server. They can be performed locally when the complete objects have been cached on the client.

Area Measurement — If the relevant objects are not complete on the client, this operation needs to be processed by the server. In such a case, only the results will be returned to the client, rather then the relevant objects. Of course, if the complete objects have been cached on the client, there is enough data for the client to perform these operations.

Polygon Overlap — In the simplified map, some objects with missing features are distorted. Therefore, it is possible for the spatial relationships to change, especially when an unsophisticated generalization tool is employed. To avoid this problem, this operation should be performed on complete objects.

Buffer — The creation of boundaries, inside or outside of an existing polygon offset by a certain distance, and parallel to the boundary, needs precision and accuracy. Therefore they have to be computed with complete objects rather than their simplified versions.

5. Performance Evaluation

We did a series of experiments to evaluate the method proposed in this study. In each experiment we measure and compare the proposed Scale-Dependent Transmission method with the nonScale-Dependent Transmission one in their response times with various network speeds representing different network environments. In addition, we also compare these two methods with the Progressive Transmission method.

5.1 Experiment Environment

The simulated system uses a part of the Seoul map as the experimental data set. It contains 5116 spatial objects and its total size is 3.753 megabytes, giving an average size of about 769 bytes per object. Each vertex occupies 16 bytes.

The experimental hardware environment is described in Table 1. The network bandwidth is assumed to be 144 - 256Kbps in the experiment, allowing for variations in the Internet network bandwidth.

Table 1 The Experimental Hardware Environment

	CPU	Memory	OS
Server	1GHz	256MB	Windows 2000
Client	233MHz	64MB	Windows 98

The server communicates with the client through WinSocket. The client was developed with an ActiveX control. Thus, a user can uses a Web browser plugged into the ActiveX control to explore a map and performs spatial operations.

5.2 Performance Analysis

To quantitatively evaluate the performance of this proposed method, several factors were developed, and these and their corresponding equations were employed to guarantee that the map generalization be beneficial to this method. In the Scale-Dependent Transmission, it is too costly and not necessary for the scale to grow continuously. Based on the size of the client window, the minimal increment step or scale step can be defined. Therefore, the scale will increase or decrease in quantum by the scale step. Among these factors, the scale (**Scale**) and scale step (SS) describe the display scale on the client and its minimal increment range. The Average Generalization Time

(**AGT**) is determined by the nature of the generalization algorithm, while that of the sample data set determines the array of the Generalization Ratio (GR). Furthermore, using the following factors, the policy for generalization is defined in the method.

The definitions of these factors are specified as follows.

Scale — The display scale on the client as defined in section 3.4.

SS — Scale Step, that is, the minimal change in scale causing no damage to the visual quality.

Diff — Difference between the new and old scales.

AGT — Average Generalization Time refers to the performance of the generalization algorithm measured by the average time spent on processing a byte. The cost of generalizing an object is proportional to the size of the object.

GR — Array of Generalization Ratio at different scales. GR is the ratio in size between the generalized and complete objects. They are determined by the nature of the dataset.

SCO — Size of Complete Objects in bytes, that is, the size of the data set input for generalization.

NB — Network Bandwidth determined by the network environment.

ST — the time needed for preparing the data set for generalization, which is assumed to be independent of the data transmission.

Generally, the response time is defined as the sum of the processing time on the server and the transmission time over the network. There are two transmission cases to satisfy the users' requirements. One is to transmit complete objects and the other is to transmit generalized objects. The response time is described with the following equations in these two cases.

- The equation for the Response Time when complete objects are transmitted has one variable item, that is, the time spent on transmission over the Internet.

$$RT_c = ST + \underbrace{SCO * 8 \ / \ NB}_{transmission\ time} \quad (1)$$

- The equation for the Response Time when generalized objects are transmitted consists of two variable items, i.e., the time spent on map generalization and that spent on transmission of the generalized objects, respectively. Firstly, we define $\mathbf{COST_g}$ as the cost of generalization and $\mathbf{COST_t}$ as the cost of transmitting generalized objects.

$$COST_g = \underbrace{SCO \ * \ AGT}_{generalization\ cost} \quad (2)$$

$$COST_t = \underbrace{(SCO*8) \ * \ GR[\lceil Diff/SS \rceil]}_{size\ of\ gineralized\ objects} / NB \quad (3)$$

Then we can define the response time in the Scale-Dependent Transmission method as the sum of the time spent in preparation for generalization, the generalization time and the transmission time.

$$RT_G = ST + COST_g + COST_t \quad (4)$$

- The Benefit of Generalization (BG) is defined as the time saved by using the Scale-Dependent Transmission method.

$$BG = RT_c - RT_g \quad (5)$$

The difference between costs in the two cases indicates whether or not generalization can contribute to the reduction of the response time. The result BG > 0 means that the cost of generalization outweighs the reduced transmission cost by generalization. In such a case, it is more beneficial to perform generalization. On the other hand, it is a better to transmit more detailed objects when BG < 0 or BG = 0, which indicates that the gain from generalization are offset by its own overhead. Accordingly, the Favorable Generalization (FG) policy is defined based upon the above equation. Generalization will be applied only when it improves the response time so that its negative aspects can be avoided. Because the request can be satisfied by objects of equally or more detailed features, it is not necessary to insist on the requested scale accurately.

To highlight the determinant factors in this policy, Equation (5) was formulated using Equation (1) and Equation (4). Thus,

$$\begin{aligned} BG &= RT_c - RT_g \\ &= SCO * AGT + SCO * GR[\lceil Diff/SS \rceil] * 8/NB - SCO * 8/NB \\ &= SCO * (AGT + GR[\lceil Diff/SS \rceil] * 8/NB - 8/NB \\ &\sim AGT - \frac{1 - GR[\lceil Diff/SS \rceil]}{NB} * 8 \end{aligned}$$

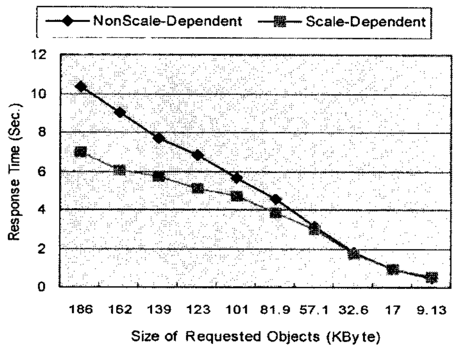
(Here the symbol '~' indicates that two numbers share a common sign. For example, A~B means A and B have the same sign.)

Here it is observed that the value of BG is determined by five factors— *AGT*, *GR*[], *Diff*, *SS* and *NB*. *SCO* is irrelevant to *BG*. From the definitions of these parameters above, *AGT*, *GR*[], *SS* and *NB*. can be measured before the run time. Only *Diff* has to be calculated for every request. Consequently, the FG policy can be evaluated instantly at the run time.

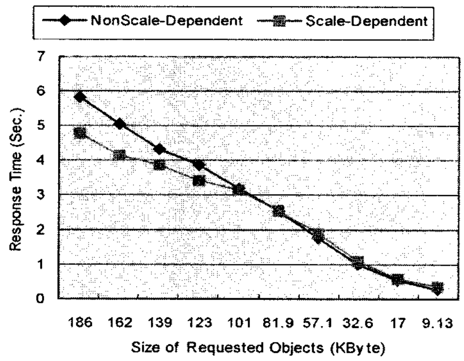
In our experiments, six steps in the scale were defined within the experimental data set. The values for *GR*[1..6] are as follows: 0.89, 0.83, 0.75, 0.66, 0.56, 0.49. The *AGT* of the Wavelet-based Generalization Algorithm used in Algorithm 2 was measured and found to be $1.0*10^5$ sec. Since the transmission speed of the Internet is not fixed, the range of *NB* (144-256 Kbps) was examined. In the following section, the experimental results will demonstrate that this method varies in its performances under different network environments.

5.3 Performance Experiments

Initially we measured the effects of the requested objects' size on the response time. To eliminate the interference of other factors, we assume the scale difference to be constant and set four scale steps as the most frequent case. Figure 5 shows the influence different sized objects have on the response times under different network environments. From these results, the new method clearly demonstrates its effectiveness when the size of requested objects is large. In Figure 5(a), when the size of the requested objects is under 32.6KB, there is little difference between the two methods since the complete objects have to be transmitted according to the FG policy. However, when the object size exceeds 32.6KB, the response time of the Scale-Dependent Transmission is less than that of the nonScale-Dependent Transmission. When the size reaches 186KB, a reduction of 36.4% is seen. Although the breakdown point in Figure 5(b), which is for a wider network bandwidth, the proposed method improvesis different, the trend is still similar with that seen in Figure 5(a). Although with increases in the network bandwidth the improvement in the response time is not as great, the results of these tests are still favorable for spatial data since they are usually large in size.



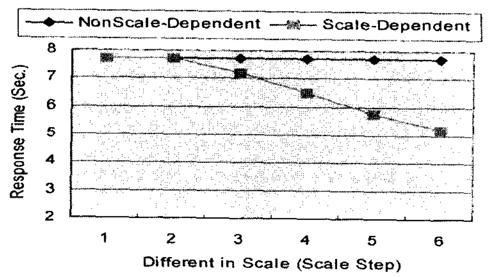
(a)



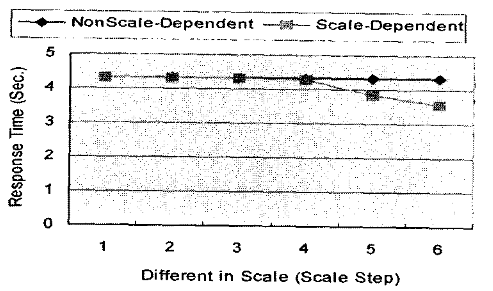
(b)

Figure 5 The Effect of the Size of Requested Objects on the Response Time (a) **NB**=144 Kbps; (b) **NB**=256 Kbps.

Secondly, we measured the effect of changes in the scale. In this experiment, the sizes of the requested objects were assumed to be constant. Figure 6 shows that the response time decreases substantially when the difference in the scale is large, that is, more superfluous data are eliminated. There is no concept of scale in the nonScale-Dependent transmission so the response time does not change with the scale, shown as a horizontal line. When the difference in the scales increases, the size of the eliminated data is substantial and, therefore, the transmission cost is been reduced further. In Figure 6(a), when the scale difference is six scale steps, a reduction in the response time of 32.6% is seen. However, when the difference is small, according to the FG policy, complete objects are transmitted and the response time is the same as a nonScale-Dependent transmission. As in Figure 5, Figure 6(b) follows a similar trend as Figure 6(a) but improvements in the response time lessen as the transmission speed increases.



(a)



(b)

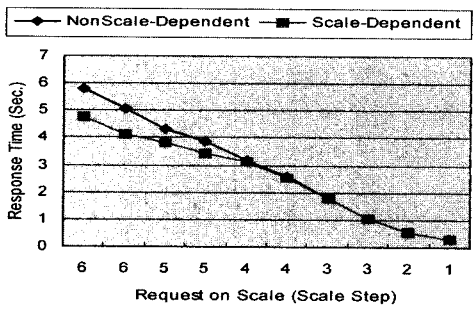
Figure 6 The Effect of Change in Scal on Response Time (a) **NB**=144 Kbps; (b) **NB**=256 Kbps.

Thirdly, we have done the simulated work with a series of operations on the sample map in order to fully evaluate this method. The operation pattern is as follows: All View, Zoom In, Move, Zoom In, Zoom Out, Zoom Out, Move, Move, Zoom In and Zoom Out. From this we can perceive the combined effects of these two factors on the response time of individual operations.

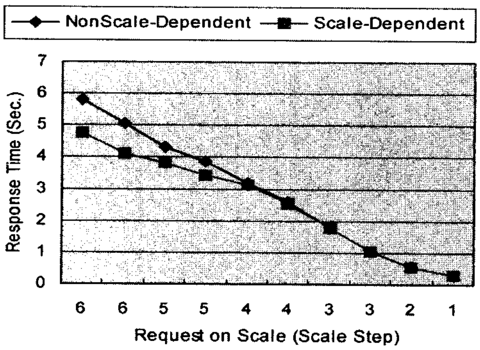
In addition, we demonstrate the effect of pre-computation on the response time. As mentioned in section 3.1, we could pre-compute maps at a set of certain scales in order to improvethe response time. To avoid the abundant computation, we can compute the maps at different levels in an incremental fashion because the level of detail diminishes while the scale of a map decreases. That is, the present map is computed based on the previously computed map. The first one is computed from the original map. The space cost is the amount of the space occupied by all pre-computed maps. Based on the experimental parameters, we can draw out the cost on time and space in this experiment. We pre-compute to maps at three different levels (scale = 5, 3, 1). The time costs are as follows (in sec.): 357, 310 and 249,and the time cost of preprocessing is their sum 915 seconds. The space costs are as follows (in MB): 3.34, 2.8 and 2.1,and the space cost of pre-computation is that 3.34 + 2.8 + 2.1 = 8.24 MB.

Figure 7 shows the response time improves when the combined effects of the size of the requested objects and the difference in the scales are favorable, especially for the initial operations. Even without the pre-computed maps, the reduction in the response time is nearly 20% for the best case. In the worst case scenario, that is, when the differences in the scales and the sizes of requested objects are really small, the response time is equal to that of the nonScale-Dependent transmission since the FG policy is applied to avoid the generalization processing. Using the pre-computed data for each of the three scales (scale = 5, 3, 1), the response times of the requests for these three scale steps have been further reduced.

Lastly, we compared the proposed method with the conventional nonScale-Dependent Transmission method and the Progressive Transmission method. The progressive transmission method is implemented based on [6]. The pre-computation of the progressive transmission is relatively long, 1154 seconds, while its space cost is 5.84 MB, which is less than that of the Scale-Dependent Transmission. Figure 8 illustrates the comparison of these three methods in terms of their response times. When the client requests a more detailed map (for example, Scale = 6), the Progressive Transmission gave a better performance since the data redundancy is lower than in the Scale-Dependent Transmission. However, when a less detailed map is requested (for example, Scale = 1), the Progressive Transmission takes a longer time mainly because the division of the features must be reintegrated on the clients system, while the Scale-Dependent Transmission simply replaces the coarser features with the more detailed ones.



(a)



(b)

Figure 7 The Combined Effect of Both Two Factors (**NB** = 256Kbps). (a) does not use the pre-computed maps but (b) does.

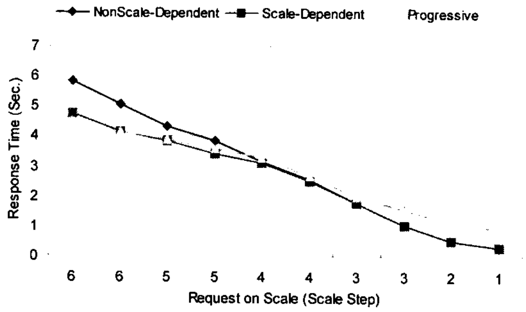


Figure 8 Comparison of three transmission methods in Response Time (**NB** = 256Kbps)

The only purpose of using generalization in this method is to reduce the transmission cost on the Internet. The scale, along with its request, is transparent to the user. The use of Scale Step (SS) is feasible in the method to avoid excessive computations on the server side. For every query, the data set is transmitted to the client as a result whose scale is closest to that requested, instead of exactly what the client requested. When generalization is profitable for the transmission on the Internet, the map is generalized and cached on the server. For further access, the cached data set is also available for all other users. Thus, this method is advisable for a Web-based GIS, a multi-user system with scalability, while the performance evaluations apply to both the single-user and multi-user environments.

6. Conclusions and Future Work

The method proposed in this study armed with the Wavelet-based Generalization Algorithm [12] fulfills the efficient transmission of vector spatial data on the Internet. Based on the display scale on the clients system, invisible spatial objects are filtered out to reduce the transmission cost and, thus, improve the response time. To evaluate two important factors, i.e., changes in the scale and the size of the requested objects, experiments were performed to analyze their individual and combined effects on the response time. The experimental results show that the response time was substantially reduced between a network bandwidth of 144 and 256 Kbps. It was also observed that the Scale-Dependent Transmission was more effective with lower-speed networks and when the size of the data set is larger. However, the proposed method survives as long as a compromise between the size of the vector spatial data and network bandwidth exists. In conclusion, these results clearly demonstrate that this proposed method positively contributes to the efficient transmission of vector spatial data on the Internet.

Future works include three separate studies. Firstly, high-performance map generalization may help in the efficiency of this method and broaden its range of application. Secondly, the quantitative definition of the scale step should be studied further to allow for its automatic determination based on the given map. Thirdly, future works should also look into how to select at which levels a map should be pre-computed to best aid in the computations.

Reference

[1] Esri White Paper, "Map Generalization in GIS: Practical Solutions with Workstation ArcInfo Software", *ESRI Inc.*, July 2000.

[2] ArcNews, "Scalable Internet Mapping Solutions from ESRI: Maps on the New Frontier", *ArcNews*, Fall 1997.

[3] Luis Manuel Callejas, "Web-Based GIS: A Brief History", http://blaze.trentu.ca/~ermlc/web-gis.html.

[4] Y.S. Cho, H.Y. Kim, J.H. Kim and H.Y. Bae, "The Design and Implementation of Componentized Web-enabled GIS", ICIMU 98, Sept. 1998.

[5] Z.R. Peng, "An Assessment of the Development of Internet GIS", *Proc. of the 1997 ESRI User Conference*, 1999.

[6] Michela Bertolotto, "Progressive Vector Transmission", *ACM GIS'99*, 1999.

[7] Leone Barnett and John V. Carlis, "Roads Data Model: A Necessary Component for Feature-Based Map Generalization," *GIS 96 Rockvillle MD USA*, 1996.

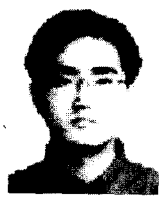
[8] Buttenfield B. P., "Progressive Transmission of Vector Data on the Internet: A Cartographic Solution", http://greenwich.colorado.edu/babs/ottawa/Bfield_Ottawa.htm.

[9] Oh Young-Hwan, "Advanced Progressive Transmission for Spatial Data in Web based GIS", *The Second AEARU Workshop on Web Technology*, October 1999.

[10] Ralf Hartmut Guting, "An Introduction to Spatial Database Systems", *VLDB Journal*, Vol.2, No.4, 1994.

[11] Robert Laurini and Derek Thompson, "Fundamentals of Spatial Information Systems", *Academic Press*, 1992.

[12] Chen Liang, Lee Chung Ho, et al, "Spatial Linear Feature Simplification Using Wavelets", *Proc. of World Science & Engineering Society — Simulation 2001*, June 2001.



양 신

1998년 중국 중경 우전대학교 컴퓨터공학과(공학사). 2001년 인하대학교 대학원 전자계산공학과(공학석사) 2001년~현재 중경 우전대학교 중-한 중경 GIS 연구 센터 연구원. 관심분야는 지리정보시스템, 공간 데이타베이스, 분산 데이타베이스

이 충 호

1997년 인하대학교 전자계산공학과 (공학사). 1999년 인하대학교 대학원 전자계산공학과 (공학석사). 2003년 인하대학교 대학원 전자계산공학과 (공학박사). 관심분야는 지리정보시스템, 시공간 데이타베이스

배 해 영

1974년 인하대학교 응용물리학과(공학사). 1978년 연세대학교 대학원 전자계산학과(공학석사). 1989년 숭실대학교 대학원 전자계산학과 (공학박사). 1985년 Univ. of Houston 객원 교수. 1992년~1994년 인하대학교 전자계산소 소장 1982년~현재 인하대학교 전자계산공학과 교수. 1999년~현재 지능형 GIS 연구센터 소장. 2000년~현재 중국 중경우전대학교 대학원 명예 교수. 관심분야는 분산 데이타베이스, 공간 데이타베이스, 지리정보 시스템, 멀티미디어 데이타베이스