

# 위치 정보 검색을 위한 모바일 벡터 지도 일반화 연산 알고리즘 연구

김현우\*, 최진오\*

\*부산외국어대학교 컴퓨터공학부

## Study on Algorithms of Mobile Vector Map Generalization Operators for Location Information Search

Hyun-woo Kim\*, Jin-oh Choi\*

\*School of Computer Engineering, Pusan University of Foreign Studies

E-mail : jochoi@pufs.ac.kr

### 요 약

모바일 환경에서 벡터 지도 서비스를 위해서는 지도 일반화(map generalization) 과정을 통한 지도 간소화 작업이 지도 인지도를 높이는 데 큰 도움이 된다. 일반화 연산은 selection, aggregation, simplification, displacement 등 다양하지만 아직 정형화된 연산 알고리즘이 확립되지 못하고 있다. 그 이유는 지도 일반화 알고리즘을 위해서는 자동화되기 어려운 상당한 전문적 지식과 판단이 필요하기 때문이다.

본 논문은 기존의 시도된 일반화 연산 알고리즘들에 기초하여 모바일 환경에 적합한 알고리즘을 고찰하고 제시한다. 복잡한 공간 객체를 제한된 자원의 모바일 기기에 효율적으로 출력하기 위한 다양한 접근 방법을 모바일 환경에 적용하는 과정을 보인다.

### ABSTRACT

In the mobile environments for the vector map services, a map simplification work through the map generalization steps helps improve the readability of the map on a large scale. The generalization operations are various such as selection, aggregation, simplification, displacement, and so on, the formal operation algorithms have not been built yet. Because the algorithms require deep special knowledge and heuristic, which make it hard to automate the processes.

This thesis proposes some map generalization algorithms specialized in mobile vector map services, based on previous works. We will show the detail to adapt the approaches on the mobile environment, to display complex spatial objects efficiently on the mobile devices which have restriction on the resources

### 키워드

mobile vector map, map generalization, location information search

### 1. 서 론

지도 일반화 기법은 지도 제작 과정에서 사용자의 인지도를 높이기 위한 필수적인 요소이며, 전자 지도 서비스를 포함한 다양한 분야에서 자동 지도 생성을 위한 일반화 알고리즘들이 최근까지 연구되고 있다. 하지만, 일반화의 원칙들을 규정하는 명확한 법칙은 없다[1]. 그 이유는 지도 일반화 과정이 인간의 고도로 전문화된 경험과

판단력에 의존하기 때문이다. 따라서 지금까지 특정 분야에 특화된 일반화 연산 알고리즘들을 제시하는데 주력하고 있다.

모바일 벡터 지도 서비스 환경에서는 이동 클라이언트 기기의 리소스 제약과 전송 대역폭의 제한으로 서버에서 전송하는 지도 데이터를 한정된 시간 안에 클라이언트로 전송할 수 없을 뿐만 아니라 클라이언트에서 그대로 출력할 수 없는 문제점이 있다. 따라서 이 분야에서 지도 일반화 연산을 도입한다면 지도 간소화 작업을 통한 효

과를 크게 기대할 수 있을 것이다.

그러나 지도 일반화 연산에 대한 표준화 시도는 1991년 ICA(International Cartographic Association)[2] 단체에서부터 지금까지 이어지고 있지만 일반화 연산들에 대한 분류와 정의 정도가 개념적 수준에서 정립 되었을 뿐이다. 세부 알고리즘은 적용분야에 따라 상당한 차이를 보일 수밖에 없다. 예를 들어, 기존의 위치 지도로부터 인구 센서스(census) 정보 출력을 위해 일반화 연산을 적용할 때의 판단 기준과, 상하수도 개념도를 출력하기 위해 일반화 연산을 적용할 때의 판단 기준은 상이하다. 즉, 일반화 연산의 상세 알고리즘은 적용분야별로 휴리스틱(heuristic)과 전문적 판단을 필요로 하는 특징을 가진다.

본 논문에서는 기존의 유선 서비스를 위한 벡터 지도 데이터베이스로부터 위치 정보 검색을 위해 모바일 벡터 지도를 생성할 때 적용 가능한 일반화 연산의 알고리즘들을 고찰하고 제안하고자 한다. 이를 위해 필요한 것은 모바일 환경이 가지는 특성이 일반화 연산의 적용 기준에 어떠한 영향을 미치는지 파악할 필요가 있다. 또한 위치 정보 검색이라는 일반화 연산 적용 분야가 지니는 특성과 이에 따른 알고리즘의 판단 기준을 정립이 필요하다.

따라서 본 논문은 제시 분야의 특성을 고려하고 이에 따라 현장에 적용 가능한 일반화 알고리즘을 제시하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구를 소개하고, 3장에서 적용 분야의 특징과 이에 따른 일반화 연산 알고리즘들을 제시한다. 마지막으로 4장에서 결론과 향후 연구 과제에 대해 논의한다.

## II. 관련 연구

지도 일반화 기법에 대한 표준화 시도는, 1991년 ICA(International Cartographic Association) 산하의 지도 일반화 워킹 그룹이 구성되었고, 이후 지도 일반화 정형화 문제 등에 대한 연구 결과[2][3]가 보고되었다. ESF(European Science Foundation)는 GISDATA라는 연구 프로그램에서 전자지도(digital map) 일반화의 현황을 연구하였다. NCGIA(National Center for Geographic Information and Analysis)는 1993년 지도 일반화에 대한 전문가 모임(initiative 8 : Formalizing Cartographic Knowledge)을 구성하고 연구 결과[4]를 발표하였다. 이 보고서에서 지도 일반화 및 전자지도 제작 논점들을 중점적으로 다루었다.

이러한 노력과 이후 관련 연구[5][6]들이 이어지고 있지만 일반화를 위한 많은 기본적인 문제점들이 아직 해결되지 못한 채 남아있다. 일반화 연산들에 대한 분류와 정의는 개념적 수준에서

정립이 되어 있지만 세부 알고리즘은 상당한 노력과 전문적 판단을 필요로 하기 때문이다.

본 논문에서는 모바일 위치 정보 검색이라는 특정 분야에 대하여 벡터 지도 일반화 연산의 적용 여부를 고찰하고 이 분야에 특화된 다양한 접근 방법들을 고안하고 알고리즘으로 정립하여 제시한다.

## III. 일반화 연산 알고리즘

공간 데이터의 일반화 연산의 종류는 제안 연구별로 상이하고 다양하지만 본 논문에서는 모바일 위치 정보 검색 환경에 적합한 연산으로 *selection*, *simplification*, *filtering*, 그리고 *displacement*, 이렇게 4종류를 선정하였다. 그리고 각각의 세부 연산들은 출력 축척과 가상 계층 그룹에 따라 달리 수행된다. 예를 들어 *selection* 연산은 기본도 계층 그룹의 중축척 출력과, 이점표 계층 그룹의 대축척 출력을 위해 각각 다른 기준을 적용한다.

*selection* 연산은 GIS 서버로부터 검색된 원본 공간 데이터에서, 출력되어야 할 객체들만 선택하고 나머지는 삭제(elimination)하는 일을 수행한다. 선택의 기준은 현재의 출력 축척과 가상 계층 그룹의 종류이다. 본 논문에서 제안하는 *selection* 알고리즘은 그림 1과 같다.

```

selection(map, scale, newMap) {
  for i<map.numOfLayers {
    if map.layer[i] is in BasicLayers[]
      if map.layer[i] is in preSet[sacle][B]
        addLayer(newMap, map.layer[i])
      else if map.layer[i] is SearchLayers[] {
        for j<map.layer[i].numOfObjects
          if map.layer[i].object[j] < searchObject
            delObject(map.layer[i],j)
          addLayer(newMap, map.layer[i])
      }
    else if map.layer[i] is in MilestoneLayers[]
      if map.layer[i] is in preSet[sacle][M] {
        leaveSomeObjects(map.layer[i],
numOfMSPerLayer[sacle])
        addLayer(tempMap, map.layer[i])
      }
  }
  for k, total<numOfMSObjects[sacle] {
    x=getPriority(tempMap, k)
    addLayer(newMap, tempMap.Layer[x])
    total=total+tempMap.Layer[x].numOfObjects
  }
}
    
```

그림 1. *selection* 알고리즘

이 알고리즘에서 'preSet[sacle]'은 미리 결정된 특정 축척에서 출력할 계층들의 집합이다. 'searchObject'는 사용자가 검색하는 객체로서 누

락되어서는 안되므로 반드시 포함시킨다. 'numOfMSPerLayer [scale]'은 현재 축척에서 각 이정표 계층에 포함시킬 객체의 최대 수를 지정한 것이다. 소축척일 경우 그 개수가 가장 많다. 'numOfMSOjects[scale]'은 축척에 따라 지도에 포함될 이정표 객체의 최대 수를 제한하기 위해 설정된 값이다. 역시 소축척일 경우 그 수가 가장 많다. 즉, 이정표 계층 그룹은 계층별로 특정 개수의 객체만 선택하고, 다시 전체 최대 허용 객체만큼만 우선순위 계층 순으로 채운다. 'getPriority()'는 이정표 계층들 중 주어진 우선순위의 계층 번호를 반환한다. 'delObject()'은 한 계층에서 주어진 한 객체를 삭제하는 함수이고, 'leaveSomeObjects()'는 한 계층에서 주어진 개수의 임의 객체만을 남기고 나머지는 삭제하는 함수이다.

simplification 연산은 selection 연산에서 선택된 객체들의 구성 포인트 개수를 줄임으로써 공간 데이터 간소화를 수행한다. 그리고 simplification 연산은 각도 조사, 클리핑(clipping), 그리고 심블화(symbolization)의 3종류 세부 연산으로 구현된다.

```
simplification(map, scale) {
  for i<map.numOfLayers //degree check
    if map.layer[i].type=(LINE | POLYGON)
      for j<map.layer[i].numOfObjects
        for k<map.layer[i].object[j].numOfPoints
          if checkDegree(p[k],p[k+1],p[k+2],deg[scale])=DELETE
            delPoint(map.layer[i].object[j],k+1)
  for i<map.numOfLayers //clipping
    if map.layer[i].type=(LINE | POLYGON)
      for j<map.layer[i].numOfObjects
        for k<map.layer[i].object[j].numOfPoints
          if checkVisiable(p[k])=FALSE
            delPoint(map.layer[i].object[j],k)
  for i<map.numOfLayers //symbolization
    if map.layer[i] is in symTarget[]
      if map.layer[i].type=POLYGON
        for j<map.layer[i].numOfObjects {
          p=getCenterPoint(map.layer[i].object[j])
          delObject(map.layer[i],j)
          addPolyObject(map.layer[i],p.x,p.y)
        }
}
```

그림 2. simplification 알고리즘

각도 조사는 세 포인트를 조사하여 180도에 근접한 각도일 경우 중간 포인트를 삭제하는 것이다. 클리핑은 한 객체를 구성하는 포인트 중 화면의 뷰포인트(viewpoint)를 벗어나는 것들을 삭제하는 것이다. 끝으로 심블화는 복잡한 모양의 객체를 단순한 모양의 객체로 변형하는 것이다. 이 연산의 알고리즘은 그림 2와 같다.

이 알고리즘에서 'p[]'는 한 객체를 이루는 포인트이며, 'deg[scale]'은 미리 설정한 축척에 따른 임계각도 값이다. 이 값은 소축척일수록 커야 한

다. 'symTarget[]'은 심블화할 대상 계층들을 미리 저장하고 있다. 그리고 'checkDegree()'는 주어진 3 포인트가 이루는 각도가 180도에서 임계값 내에 포함되는지 조사하며, 그럴 경우 'DELETE'값을 반환한다. 'checkVisiable()'은 주어진 포인트가 뷰포인트 내에 포함되는지 조사하며 그럴 경우 'TRUE'를 반환한다. 심블화 연산은 한 객체의 중심점을 구하고 삭제한 후 그 중심점을 가지는 간단한 폴리곤 객체를 추가하여 처리한다. 'addPolyObject()'가 이 역할을 수행한다.

'Filter'에 의해 수행되는 filtering 연산은 객체를 선택한다는 점에서 selection 연산으로 볼 수 있지만, 저수준 일반화 과정이 모두 끝난 후 선택적으로 수행되기 때문에 별도의 일반화 연산으로 취급하기로 한다. 이 연산은 서버에서의 일반화 연산으로 간소화한 지도가 여전히 전송 대역폭 이상의 객체를 포함할 경우 강제로 특정 계층의 특정 객체들을 삭제하는 것이다. 삭제 대상 계층은 지도 판독에 큰 지장을 주지 않으면서 객체 수가 많은 것을 택하며(예:가속 계층), 삭제 대상 객체는 검색 객체에서 먼 거리에 있는 것을 높은 확률로 택한다. 이 알고리즘은 그림 3과 같다.

여기서, 'filterThreshold'는 허용 임계 포인트 개수이며, 전송 지연시간과 휴대폰 단말기의 최소 메모리 용량을 고려하여 결정한다. 'fliterTarget[]'은 필터링할 대상 계층들을 미리저장하고 있다. 그리고 'getDistance()'는 주어진 객체와 검색 객체와의 거리를 반환하며, 'getDelProb()'는 주어진 거리값이 클수록 높은 확률로 'TRUE'값을 반환한다.

```
filtering(map) {
  for countPoints(map) > filterThreshold
    for i<map.numOfLayers
      if map.layer[i] is in filterTarget[]
        if map.layer[i].type=POLYGON
          for j<map.layer[i].numOfObjects {
            dist=getDistance(map.layer[i].object[j])
            if getDelProb(dist)=TRUE
              delObject(map.layer[i],j)
          }
}
```

그림 3. filtering 알고리즘

'displacement'는 지도 간소화 작업보다는 판독도를 높이기 위해 객체들의 위치를 임의로 이동시키는 연산을 말한다. 그러나 본 연구에서는 이정표 계층 객체들의 속성정보 출력이 이 연산을 적용하는 것이 오히려 휴대장치에서 지도 판독도 향상에 큰 도움이 된다는 것을 실험에서 고찰할 수 있었다. 따라서 이 연산은 객체의 좌표 이동보다는 속성정보 출력 위치 이동을 대상으로 한다. 이 연산의 알고리즘은 그림 4와 같다.

이 알고리즘은 먼저 이정표 객체들만 'msLayer'로 모아서 y좌표 기준으로 정렬한다. 그리고 상위 절반은 'findUpBestPoint()'로 위에서부터

터 최적의 속성정보 출력 위치를 찾는다. 이것은 출력될 속성정보가 화면 바깥으로 잘려나가지도 않고, 이미 결정된 영역과 충돌도 없는 시작점과 문자가 차지할 영역을 찾는 것이다. 'findDownBestPoint()'는 충돌 확률을 줄이기 위해 가장 아래에서부터 하위 영역을 우선 탐색한다.

```

displacement(map) {
  for i<map.numOfLayers
    if map.layer[i] is MilestoneLayers[]
      for j<map.layer[i].numOfObjects
        addObject(msLayer, map.layer[i].object[j])
      sortByYCoord(msLayer)
      for 0<=i<msLayer.numOfObjects/2
        p=findUpBestPoint(i,msLayer)
        for msLayer.numOfObjects>i>=msLayer.numOf-
          Objects/2
          p=findDownBestPoint(i,msLayer)
}
    
```

그림 4. displacement 알고리즘

#### VI. 결론 및 향후 과제

본 논문은 모바일 위치 정보 검색을 위해 벡터 지도 서비스를 하는데 있어서 공간 데이터 간소화 목적을 위해 지도 일반화 연산을 제안하였다. 지도 일반화 연산은 적용 분야에 따라 상이한 판단 기준을 가지고 있기 때문에 제시한 환경에서 사용될 수 있는 독자적인 지도 일반화 알고리즘이 필요함을 강조하였다.

본 논문에서 제안하는 공간 데이터 일반화 알고리즘은 모바일 위치 정보 검색 환경에 적합한 연산으로 *selection*, *simplification*, *filtering*, 그리고 *displacement*, 이렇게 4종류를 선정하였다. 이 연산들의 알고리즘은 선행 연구들에서 제안되었던 기법들에 기초하여 모바일 환경에 적합하도록 실험을 반복하여 확장 개발한 것이다. 제안한 알고리즘은 지도의 계층 종류별로 3개의 가상 계층 그룹으로 나누고, 이 그룹과 출력 축척에 따라 저수준 일반화 연산 기준을 달리 적용하는 기법을 사용하였다.

향후 각 연산별도 보다 정밀하고 다양한 접근 방법을 실험을 통하여 추가로 고찰할 필요가 있다. 또한 모바일 위치 정보 검색을 위해 지도 일반화 연산에서의 정확성(*correctness*) 판단 기준에 대한 기초 연구가 필요할 것으로 판단된다.

#### 참고문헌

- [1] ESRI, "Automation of Map Generalization : The cutting-Edge Technology", An ESRI White Paper, Environmental Systems Research Institute, Inc., 1996
- [2] Weibel, R., "Summary report : Workshop on progress in automated map generalization", Technical Report ICA Working Group on Automated Map Generalization, Barcelona, September, 1995
- [3] Mackaness, W. A. and Weibel, "Report of the ICA Workshop on Map Generalization", Technical Report ICA Working Group on Automated Map Generalization, Sweden, June, 1997
- [4] Barbara P. Buttenfield, Catherine Dibble, "Research Initiative 8 Formalizing Cartographic Knowledge: Scientific Report for the Specialist Meeting", NCGIA, 1995
- [5] Dettori, G. and Puppo, E., "Designing a Library to Support Model-Oriented Generalization", ACM Int. Symp. on Advances in GIS, United States, 1998
- [6] ESRI, "Map Generalization in GIS : Practical Solutions with Workstation ArcInfo Software", An ESRI White Paper, Environmental Systems Research Institute, Inc., 2000