

GIS용 벡터링 도구의 설계 및 구현에 관한 연구

허 봉 식* · 김 민 환*

A Study on the Design and Implementation of Vectoring Tool for GIS

Pong-Sik Ho · Min-Hwan Kim

요 약

GIS를 성공적으로 구축하기 위해서는 매핑기술, 기본 S/W 기술, DB Tool 기술 등이 필수적으로 요구된다. 그 중에서도 매핑기술은 GIS 공간 데이터베이스 구축에 있어서 많은 시간과 노력이 요구되는 매우 중요한 기술이라고 할 수 있다. 이에, 본 논문에서는 GIS를 위한 효율적인 벡터링 도구를 체계적으로 설계하고 구현하였다. 벡터링 과정을 분석하여 자동화 가능한 부분과 나머지 부분을 분리하여 자동화 효율을 높임과 동시에, 편리한 사용자 환경을 구현하여 사용자에 의한 처리 과정의 효율 또한 높였다. 또 벡터 데이터의 다단계 표현을 제안하고 본 논문에서 개발한 벡터링 도구에 적용하여 그 유용성을 확인하였다.

ABSTRACT : To construct GIS successfully, mapping technology, primary S/W technology, and DB tool technology are necessarily required. Among them, the mapping technology is the most important one in constructing a GIS spatial database that needs much time and effort. In this paper, we designed and implemented a systematic and effective vectoring tool. The general vectoring process was analyzed and parted into an automated part and a remaining part for increasing overall efficiency. We also proposed a multi-level representation method for vector data and applied it to the developed vectoring tool. We could verify usefulness of the proposed representation method.

서 론

GIS(Geographic Information System)는

이미 선진국에서 1960년대부터 그 중요성을 인식하고, 이의 효율적인 구축과 활용 방법에 대하여 꾸준한 연구와 투자를 하여 왔다. 국내에서도 최근에 GIS에 관한 관심이

* 부산대학교 컴퓨터공학과(Dept. of Computer Engineering, Pusan National University, San 30, Jangjun-dong Guemjung-gu, Pusan, KOREA, 609-753, Tel)510-1548)

높아져 이에 관한 연구와 활용이 많이 이루어지고 있다. 효율적이며 실용적인 GIS를 구축하기 위해서는 매핑기술, 기본 S/W 기술, DB Tool 기술 등이 필요하며, 이 중에서 매핑기술은 기초적인 수치지도 작성에 필요한 기술로서 전체의 80% 이상의 개발 노력과 경비가 요구되는 GIS구축에 있어 매우 중요하다. 그러므로 성공적인 GIS의 구축과 활용을 위해서는 효율적이고도 편리한 매핑도구의 개발이 필수적으로 선행되어야 한다(정진완, 1995).

수치지형정보의 입력방법에는 지상측량, 항공측량, remote sensing, 기존 도면 등을 이용한 방법이 있으나, 상대적으로 기존 도면을 이용한 입력방법이 경제적이다. 이에, 국립지리원의 기본도 수치지도화 사업에서도 약 30%에 해당하는 자료를 기존 도면을 이용하고 있는 것으로 알려져 있다. 기존 도면을 입력하는 방법으로는 디지털자에 의한 수동 벡터링 방법, 스캐닝된 지도를 자동 벡터링하는 방법 및 사용자와의 대화식에 의한 반자동식 벡터링 방법이 있다. 수동 벡터링 방법은 현재 가장 많이 사용되어지고 있으나, 벡터링 방법중에서 가장 비효율적이며 경제적 부담이 많은 방법으로서 차후에는 점점 사용되지 않을 것이다. 한편, 자동 벡터링 및 반자동 벡터링 방법은 아직 미흡한 성능으로 인해 단독적인 입력방법이 아닌 보조 입력방법으로 현재 사용되고 있으나, 차후 기능 및 성능이 보완되면 주된 입력방법이 될 것으로 예상된다. 이에, 본 연구에서는 자동 및 반자동 벡터링 도구의 개발을 목표로 하고 있다.

자동 벡터링에 관한 연구로는 D. Dori,

1989, S.H. Joseph and T. P. Pridmore, 1992, A. Okazaki et al., 1988, A. Okazaki et al., 1988, L.A. Fletcher and R. Kasturi, 1988, R. Kasturi et al., 1990, A.J. Filipski and R. Flandrena, 1992 등이 있다. 그러나, 이러한 연구들은 GIS에서 활용하고자 하는 지도 도면이 아닌 기계 및 전자회로 설계도면 등을 자동으로 인식하기 위한 연구들이며, 또한 특정 분야의 지식을 인공지능적인 방법으로 응용하여 보다 차원높은 도면의 객체 인식에 목적을 두고 이루어진 연구들이다. 따라서 이러한 연구 결과들을 지도 도면의 인식에 직접적으로 적용하여 만족할 만한 성과를 얻기는 어려운 상태이다. 반면에, 지도 도면을 자동으로 벡터링하기 위한 연구는 아직 많이 이루어지지 않고 있는 상태이다.

한편, 스캐닝된 도면의 입력에서는 최근에 반자동 입력 방법이 많이 이용되고 있으며, 이러한 입력 방식을 채택하고 있는 상용 벡터링 도구로는 영국 Laser Scan사의 VTRAK, 미국 Intergraph사의 I/GEOVEC, 미국 ESRI사의 CADCore 등이 있다(쌍용정보통신(주), 1995, S. Aronoff, 1989). 이 도구들은 벡터화 방법으로서 중앙선 추적 방식에 의한 벡터링 방법 또는 외곽선을 벡터화한 후에 다시 중심선을 추출하는 방법을 사용하고 있으며, 처리하기 곤란한 부분에 대해서는 사용자가 개입하여 대화식으로 처리할 수 있도록 하고 있다. 이와 같이, 사용자의 인식능력을 활용함으로써, 완전 자동 입력방법의 단점 즉, 도면의 질 저하 및 잡음으로 인한 인식처리의 한계성을 극복하고 있으며, 아울러 전체적인 입력 효율도 높이고 있다. 그러나 너무 잦은 정지나 사용자

의 개입요구로 인해 도면의 입력효율을 저하시키는 경향이 있으므로 보다 개선될 필요가 있다. 이에 따라, 국내에서도 외부에 전혀 공개되지 않고 있는 상용 벡터링 도구들의 기본 기술을 시급히 확보하고, 여기에 보다 효율적이고 편리한 기능을 추가하여 국제 경쟁력이 있는 벡터링 도구를 개발할 필요가 있다.

이에, 본 논문에서는 보다 효율을 높이면서도 편리한 사용자 인터페이스 환경을 제공할 수 있는 체계적인 벡터링 도구의 구조와 벡터링 방법을 제시하고, 세선화(thinning)에 의한 자동 벡터링 기술, 선 추적(line tracing)에 의한 반자동 벡터링 기술, 벡터데이터의 수정 및 편집기술 등을 개발하였다. 또한, 기존의 벡터링 도구에서는 찾아 볼 수 없는 기능인 벡터 데이터의 다단계 표현법을 제시하고, 개발한 벡터링 도구에 적용하여 그 유용성을 확인 하였다.

벡터링 도구의 전체적 설계

본 연구에서 개발한 벡터링 도구의 설계 목표는 벡터링 대상인 이진 이미지(binary image)를 입력받아 GIS에서 활용 가능한 벡터 형식의 데이터를 출력하는 것이다. 이를 효과적으로 수행하기 위해서 본 논문에서는 Fig. 1과 같은 구조를 가지는 벡터링 도구를 설계하였다. 본 논문에서 설계 및 구현한 범위는 Fig. 1에서 점선으로 둘러싼 부분이다.

일반적으로 우리가 사용하는 종이 지도에서는 등고선, 도로, 가옥 등 여러 종류의 정보들이 혼합되어 표시되어 있다. 또한 국립

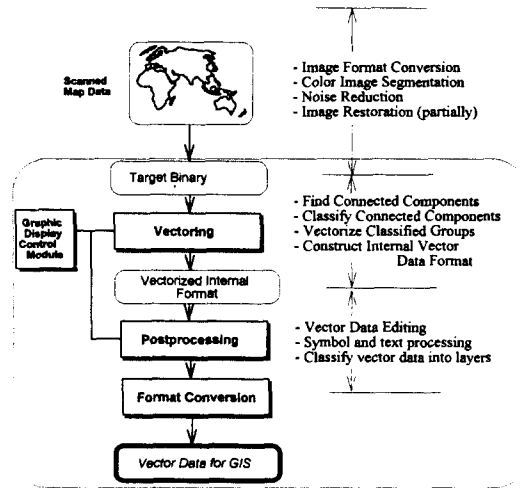


Fig. 1 Overall Structure of Vectoring System

지리원에서 사용하고 있는 하나의 원도에도 다양한 종류의 벡터링 대상물이 포함되어 있을 수 있으므로, 모든 벡터링 대상물을 한 가지 방법으로 벡터링 할 수 없다. 즉, 어떤 벡터링 대상물은 그 외곽선을 따라서 벡터 데이터를 추출해야 하고, 또 다른 벡터링 대상물은 그 중심선을 따라가면서 벡터 데이터화 해야 할 필요가 있다. 특히, 문자열이나 심볼 같은 정보는 그 외곽선이나 중심선을 따라서 벡터링 할 성질의 정보는 아니다. 따라서, 입력 도면에 있는 벡터링 대상물의 종류에 따라 적절한 벡터링 방법의 선택이 필요하다. 이를 위하여, 본 논문에서는 벡터링 대상물들을 먼저 연결성분들의 집합으로 표현하고, 각 연결성분을 분석하여 적용할 벡터링 방법에 따라 세 개의 연결성분 그룹으로 분류하였다.

연결성분들의 분류는 연결성분의 기하학적인 구성형태에 관한 정보들(예를 들면, 연결성분을 최소한으로 포함하는 bounding box의 크기, bounding box의 가로와 세로의

비율, 연결성분을 구성하는 점들의 수, 이 점들이 bounding box를 채우는 비율(filling ratio) 등)을 이용하여 이루어진다. 이에 따라, 연결성분들은 Fig. 2에서와 같이 세 가지 연결성분 그룹으로 나뉘어져 다음과 같이 서로 다른 방법으로 벡터링 된다.

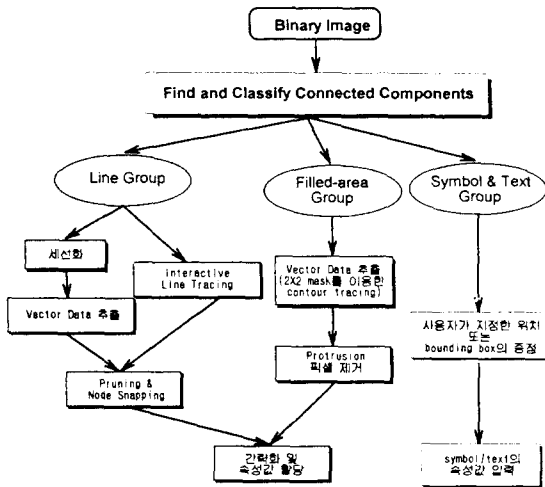


Fig. 2 Vectoring Process for Each Connected Component Group

(1) curved line type 연결성분 그룹

이 그룹의 연결성분들은 세션화 또는 선 추적에 의해 중심선을 벡터화 한다.

(2) filled area type 연결성분 그룹

이 그룹의 연결성분들은 그 외곽선이 벡터화 할 대상이 된다. 예를 들어, 지도에서 호수와 같이 특정 영역을 나타내는 이미지 영역에 대해서는 그 중심선을 찾아서 벡터링 한다는 것이 의미가 없으므로, 그 외곽선을 추출하여 벡터 데이터를 생성하여야 한다. 이미지 영역의 외곽선을 추출하기 위한 방법으로는 형태학(morphology)을 이용하거나, 래스터 스캔(raster scan)방식을 이용

하거나, 또는 마스크를 이용한 외곽선 추적 방법 등이 있다. 본 논문에서는 2x2 마스크를 영역의 외곽선을 따라 움직이면서 지나간 점들의 자취를 벡터 데이터로 만드는 방법을 사용하였다.

(3) text type 및 symbol type 연결성분 그룹

이 그룹의 연결성분들은 앞의 두 가지 벡터링 방법으로는 벡터화가 불가능하고, 그 의미나 내용을 입력하는 것이 벡터화라고 할 수 있다. 이러한 연결성분들은 패턴인식 기술을 이용하여 자동입력이 가능하다. 그러나, 스캐닝 과정에서의 오류나 잡음으로 인한 데이터의 훼손과 크기 및 회전변형 등에 의한 기하학적인 변형으로 인해 자동인식이 쉽지 않다. 따라서, 이러한 기능은 현재 사용되고 있는 상용 벡터링 도구에도 제대로 구현되어 있지 않다. 이에, 본 논문에서는 후처리 과정에서 이러한 연결성분들의 위치 및 모양을 사용자에게 자동으로 보여주고, 사용자가 그 의미 및 내용을 입력하는 대화식 입력방법을 사용하였다.

Line Type 연결성분의 벡터링

Line type 그룹에 속하는 연결성분들의 벡터화 대상은 이미지의 중심선이다. 그 중심선을 찾아내는 방법은 크게 세션화에 의한 방법과 선 추적에 의한 방법으로 나눌 수 있다.

세션화에 의한 벡터링

화상처리 분야에서 어떤 물체의 중심선을 추출하기 위해 세션화 방법(R. C. Gonzalez

and R.E. Woods, 1992, L.O Gorman, 1990, V. K. Govindan and A. P. Shivaprasad, 1987, 이성환, 1993, 상공부, 1993)이 널리 이용된다. 본 연구에서는, line type의 연결성분에 대해 (R. C. Gonzalez and R.E. Woods, 1992)의 세선화를 행한 후, 그 결과인 골격선을 추적하면서 벡터 데이터를 생성하는 방법을 개발하였다. 이에 따라, 벡터링의 전 과정이 사용자의 간섭 없이 자동으로 이루어 질 수 있었다.

그러나 원래의 이미지에 존재하던 잡음으로 인해 세선화 과정에서 기생 성분(parasitic component)이 생성되기도 하며, 특히 교차점 영역에서 필요 이상의 교차점을 생성시키기도 하였다. 이에, 본 연구에서는, 기생 성분을 제거하기 위한 절단 과정(pruning)과 불필요한 교차점을 삭제 또는 통합하기 위한 노드 병합(node grouping)과정을 두어, 정선된 데이터를 생성하였다. 또한, 경우에 따라서는 교차점의 위치가 세선화 과정에서 왜곡되는 경우가 발생하였다(Fig. 3). 이러한 왜곡에 대해 자동적으로 수정해 주는 방법론을 구하기가 쉽지 않아 후처리 과정에서 사용자에게 의해 편집하도록 하였다.

기본적으로 도면의 완전 자동 벡터링은 상당히 어렵다. 이에 따라, 후처리 과정에서의 사용자 부담이 많아 현실적으로 유용한 자동 벡터링 도구의 개발에 어려움이 많다. 이에, 본 연구에서는 벡터링 과정에서 사용자가 개입하여 어려운 문제점들을 해결해 나가는 대화식의 반자동 벡터링 도구의 개발에 보다 치중하였다.

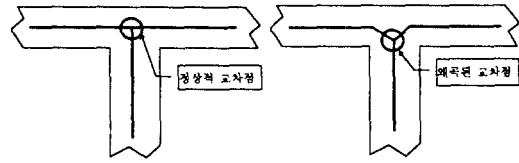


Fig. 3 Example of Distortion of Intersection Point During Thinning Process

선 추적에 의한 벡터링

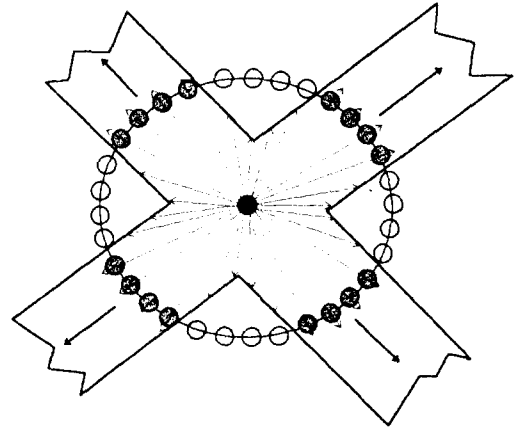
세선화에 의한 자동 벡터링 방법의 보완 방법으로서 선 추적 방식의 벡터링 방법을 개발하여 병행하여 사용할 수 있도록 하였다. 본 논문에서 개발한 선추적 방법은 사용자와 대화식으로 하되 최대한 자동화시켜 입력 효율을 높였다. 우선 line type으로 분류된 연결성분 하나를 사용자가 선택한 다음, 추적을 시작할 위치를 사용자로부터 입력 받는다. VTRAK과 같은 기존의 벡터링 도구는 시작 위치 뿐만 아니라, 초기 추적 방향, 추적 시 고려할 영역의 범위 등을 역시 사용자로부터 입력을 받아야 하지만, 본 논문에서는 그 시작 위치만을 입력 받고 나머지 필요한 정보는 내부적으로 결정을 하여 사용하도록 하였다.

선을 추적하는 동안에 거치는 일련의 점들이 기본적인 벡터 데이터가 되는데, 여기에는 추적과정에서 불가피하게 생성된 점들도 많이 포함되어 있으므로, 좌표 평활화(coordinate thinning)를 하여 얻은 데이터가 실질적인 벡터링 결과라고 볼 수 있다. 또한 선 추적 방법에 의한 벡터링 과정에서 중요한 것은 추적 과정에서 만나게 되는 교차점을 어떻게 처리를 할 것인가 하는 것이다. 즉 교차점 자체의 인식과 교차점에 연결된 갈래의 탐색 방법, 또한 이미 추적한 부

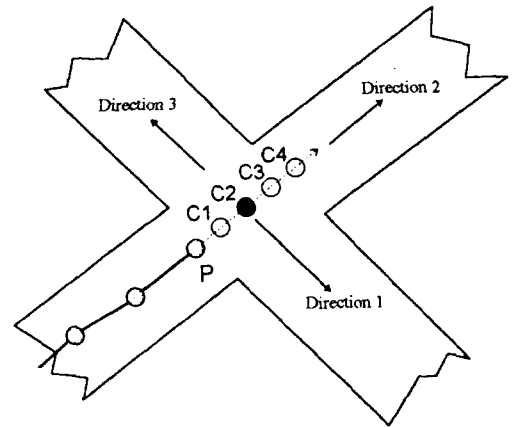
분에 대한 이중 추적 방지와 이미 발견한 교차점을 새로운 교차점으로 이중 인식하지 않도록 하는 것이 중요한 문제이다. 이런 문제는 경우에 따라서 사용자의 잦은 개입을 요구하게 되어, 전체적인 입력 효율을 저하시키는 요인이 될 수 있다. 본 논문에서는 꼭 필요한 경우가 아니면 사용자의 개입을 줄여 자동화율을 높일 수 있는 선 추적 방법을 개발하였다.

본 논문에서 제안하는 교차점 분석 방법은 다음과 같다. 우선 이미지 위의 한 점으로부터 진행 가능한 방향을 탐색하는 과정을 Fig. 4-(a)에 나타내었다. 한 점을 중심으로 하는 반지름 R의 원을 그렸을 때 원호상에 위치하게 되는 각 점과 중심점을 잇는 방향이 각 탐색 방향이 되며, 중심점으로부터 탐색 방향으로 빛이 나아가듯이 일직선으로 나아 갈 수 있는 최대 거리가 그 방향의 탐색 거리가 된다. 이렇게 모든 탐색 방향에 대해서 탐색 거리를 구한 다음, 그 탐색 거리의 배열을 분석함으로써 한 점으로부터의 진행 가능한 방향들을 조사 할 수 있다. 여기서 원의 반지름 R은 입력 도면에 따라 사용자가 미리 지정한 값을 사용한다.

실질적인 교차점을 구하는 과정은 Fig. 4-(b)이다. 점 P는 현재 까지 추적해 온 마지막 점이며, 교차점 분석을 시작하는 지점이라고 할 수 있다. 현재까지의 진행 방향을 고려하여 앞으로 진행 가능한 점들을 C1, C2, C3, C4와 같이 계산한다. 다음에는 각 점 C1, C2, C3, C4에서 위에서 언급한 진행 가능한 방향 탐색을 한다. 만일 진행 가능한 최대 개수의 방향을 가진 점이 하나이면 그 점이 교차점이 되며, 두 개 이상이면 그 중



(a) Searching the Directions which can be traced from the given point



(b) Intersection Point Selection Process

Fig. 4 Intersection Point Analyzing Process

간점을 실질적인 교차점으로 결정한다. Fig. 4-(b)에서는 C1, C2, C3에서 진행 가능한 방향의 개수가 4개로 서로 같으므로 그 중간 위치에 해당하는 C2가 최종적인 교차점이 된다.

이미 탐색한 교차점을 이중으로 탐색하는 것을 방지하고, 더 이상 추적이 어렵거나 끝

난 경우에 새로운 추적 출발 위치를 자동으로 선택하기 위한 구조가 내부적으로 필요하다. 그렇지 않으면 추적이 중단될 때 마다 계속 사용자에게 다음 출발 위치를 요구하여야 하기 때문이다. 그래서 본 논문에서는 이 과정을 효율적으로 하기 위해 Fig. 5와 같은 자료 구조를 제시하고 구현하였다.

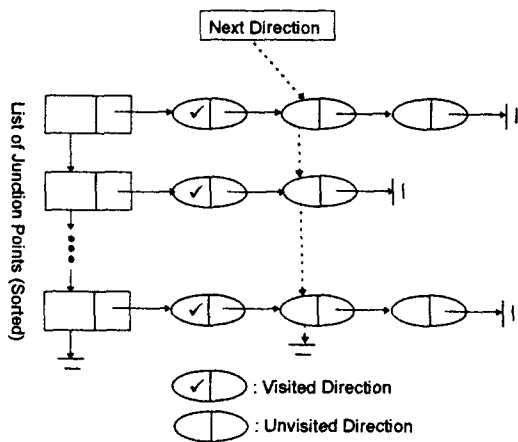


Fig. 5 Data Structure for Intersection Point and Search Directions

선 추적 과정에서 분석된 교차점들은 정렬하여 관리되며, 선 추적 도중에 교차점으로 추정되는 지점을 만나면 이미 발견된 교차점 리스트를 검색하고, 만일 발견되면 그 교차점에 등록된 진행 방향들 중에서 현재 진행하고 있는 방향과 일치되는 방향을 조사하여 제거한다. 그리고 새로운 교차점 하나를 발견하면, 연결 리스트(linked list)에 등록하고, 그 교차점에서 진행 가능한 방향들 역시 발견된 교차점에 등록된다. 이때, 이미 지나온 방향은 표시를 하여 등록하고, 아직 진행해야 할 방향들은 다음 진행을 위해서 방향 큐(direction queue)에 등록이 된

다. 일단 교차점과 해당 방향들에 대한 등록이 끝나면 그것으로 현재의 선 추적이 끝나게 된다. 다음에는 새로 추적을 시작할 위치와 추적 방향을 방향 큐로부터 얻어서 시작하며, 만일 방향 큐가 비어 있으면 더 이상 추적할 방향이 남아 있지 않은 것이 되므로 현재의 연결성분은 추적이 끝났음을 알 수 있고, 사용자에게 새로운 연결성분을 선택하도록 한다.

벡터 데이터의 다단계 표현

이상의 두 방법으로 벡터링 하여 얻은 벡터 데이터는 아크(arc), 노드(node), 버텍스(vertex)로 표현되고, 벡터링 도구 내부의 데이터 형식으로 저장이 된다. 벡터링 과정에서 생성되는 내부 형식의 벡터 데이터는 입력 도면과의 정확도를 유지하기 위해 충분한 개수의 점들로 구성된다. 그러나 외부 GIS에서 필요로 하는 데이터를 생성하는 경우에, 내부 데이터에 대해 좌표평활화(coordinate thinning) 과정을 거쳐 폴리곤 근사화(polygon approximation)를 하게 된다. 이 과정에서 벡터 데이터 생성에 사용되는 평활화 기준이 일률적으로 적용되어, 새로 생성된 벡터 데이터가 부분적으로는 원래의 도면을 제대로 표현하지 못하는 경우가 발생하게 된다. 이러한 경우에는 이미지를 다시 벡터링 하고 평활화 기준을 재설정하여 원하는 결과를 얻어야만 하는 문제가 있다. 이러한 문제는 기존의 벡터링 도구에서 사용하는 벡터 데이터의 표현방법을 약간만 변형함으로써 해결이 가능하다. 즉, 내부 벡터 데이터를 구성하는 각 점들에 대해 속성값을 계산하여 같이 저장하여 두고, 외

부에서 요구되는 정밀도에 부합되는 점들만을 선택하여 새로운 벡터 데이터로 생성하여 줄 수 있게 함으로써, 벡터링을 다시 해야 하는 문제점을 해결할 수 있다.

이를 위해 본 논문에서는 벡터 데이터의 다단계 표현을 제안하였다. 벡터 데이터의 각 점들에 대해서 속성값을 결정하는데, 이 값은 각 점이 해당 아크의 전체적인 모양 형성에 기여하는 정도에 비례하여 결정된다. 속성값을 결정하는 방법은 아크의 양 끝 점인 두 노드를 잇는 직선까지의 수직 거리를 내부의 각 버텍스들에 대해서 구하고, 가장 큰 값을 가지는 버텍스를 중간점으로 하여 원래의 아크를 분리하고, 이렇게 분리된 두 개의 아크에 대해서 앞의 방법을 계속 재귀적으로 적용해 나간다. 이 과정에서 분리 버텍스로 결정될 당시의 수직거리에 비례하여 그 버텍스의 속성값이 결정되며, 이 과정을 Fig. 6-(a)~(f)에 나타내었다. 그리고 각 버텍스의 수직거리는 미리 정의된 속성표(Fig. 6-(g))에 따라 실제 속성값으로 변환된다.

이상과 같이 내부 벡터 데이터의 각 점들에 대한 속성값들을 결정하고 나면, 벡터 데이터가 응용될 정도에 따라 적절한 속성값에 대한 임계값을 설정함으로써, 필요한 벡터 데이터를 생성할 수 있게 된다. 그 예를 Fig. 7에 나타내었다. 그리고 속성값에 대한 임계값을 하나의 아크에 대해서도 필요에 따라 구간별로 다르게 지정하여 벡터 데이터를 생성할 수 있다면, 도면의 상황에 따라, 또는 사용자의 요구에 따라 벡터 데이터의 표현 정밀도를 편리하게 할 수 있다.

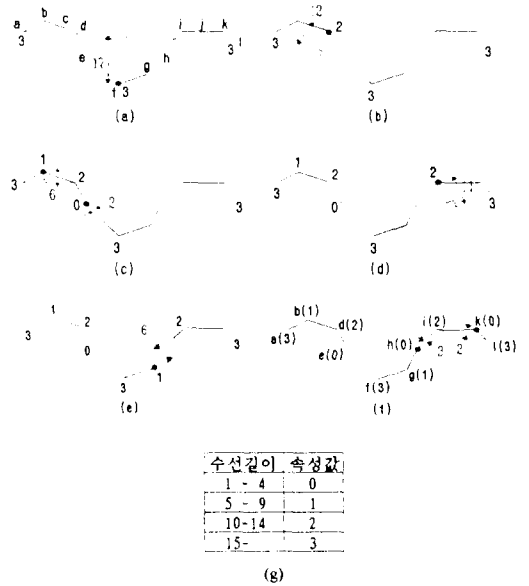


Fig. 6 Process of Attribute Value Decision for Each Vertex

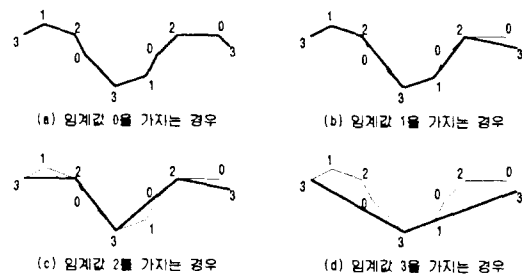


Fig. 7 Vector Data Display According to the Variation of Attribute Value Threshold

그리고 본 논문에서 제안한 벡터 데이터의 다단계 표현 방법은 많은 양의 데이터를 화면에 출력하기 위해 처리해야 하는 GIS 소프트웨어의 경우에 그 처리부담을 적응적으로 덜어 줄 수 있게 하는데 유용하게 응용될 수 있다. 즉, 다량의 벡터 데이터를 좁은 화면에 축소하여 출력하는 경우에 응집

성 증가로 인하여 실질적인 벡터 데이터 판독이 어려워질 뿐만 아니라, 화면에 표시하기 위해 필요 이상의 연산을 함으로써 처리 속도를 저하 시킬 수 있다. 그러므로, 축소되는 정도에 따라 아크의 모양 형상에 보다 중요한 역할을 하는 점들만 선택적으로 출력함으로써, 사용자에게 의한 벡터 데이터의 판독력을 높임과 동시에 처리 속도도 높일 수 있다. 본 논문에서는 벡터 데이터의 다단계 표현 방법을 개발한 벡터링 도구에 적용하여 구현해 봄으로써 그 유용성을 확인하였다.

후처리 과정 및 데이터 형식 변환

후처리 과정의 필요성은 다음 세가지로 요약할 수 있다. 첫째는 벡터링 단계에서 발생할 수 있는 오류를 수정하거나, 보다 지능적인 벡터 데이터 생성을 위한 편집 기능으로서 필요하고, 둘째는 입력된 이진 이미지에 여러 종류(또는 레이어)의 정보가 혼합되어 있을 경우, 벡터링 단계에서 생성된 벡터 데이터들을 사용자가 정의하는 레이어별로 분류할 필요가 있다. 마지막으로 text type과 symbol type으로 분류된 연결성분들에 대한 사용자에게 의한 벡터링 과정으로서 필요하다.

벡터 데이터 편집 및 수정 기능

먼저 편집 기능으로는 잘못 분류된 연결성분의 재 벡터링 기능, 이진 이미지 자체에 포함된 오류를 보정하기 위한 편집 기능(예를 들면, 아크의 수정, 추가, 삭제, 이동, 폐곡선화(closing), 병합(merging) 등), 벡터

링 과정 자체의 오류에 대한 보정 기능(예, Fig. 3에서의 교차점 왜곡 현상을 보정하는 기능) 등이 있다.

여러 레이어의 벡터 데이터들이 혼재되어 있는 경우에, 사용자의 판단에 의해 레이어 별로 연결성분별 벡터 데이터들을 효율적으로 분류하는 방법을 제공하며, 연결성분별 선택은 영역 지정에 의하거나, 아니면 개별적 선택에 의해 이루어 질 수 있다. 또한 점선이나 철도와 같이 지도상에 특수한 형태로 표시되어 있는 벡터 데이터를 의미 있는 데이터로 자동으로 변환하여 등록하는 기능도 구현하였다.

그리고 그 외의 유용한 보조 기능으로는 벡터링 결과인 벡터 데이터와 이진 이미지를 중첩 시켜 검증해 보는 기능, 레이어별 벡터 데이터를 선택적으로 출력 또는 다중 중첩 출력 기능, 래스터 및 벡터 데이터의 확대 및 축소 기능, 화면의 원활한 위치 이동을 지원하기 위한 패닝 윈도우(panning window) 기능 등이 있다.

Text type과 Symbol type 연결성분의 벡터링

이 두 개의 그룹으로 분류된 연결성분들은 앞의 두 가지 방법으로 벡터링 하는 것이 의미가 없고, 문자열 또는 심볼을 인식하여 그 내용을 기록하는 것이 적절한 벡터링이라고 할 수 있다. 그러므로 문자나 심볼을 자동으로 인식하여 벡터링 하는 것이 가장 이상적인 자동 벡터링이라고 할 수 있다. 그러나 잘못 인식한 부분에 대한 수정 및 확인 과정에 대한 오버헤드(overhead)도 무시할 수 없다. 그러므로 본 논문에서는 우선

사용자에 의해 필요한 정보를 입력하는 방법이 훨씬 효율적이라고 판단하여 후처리 과정에서 일괄적으로 처리할 수 있도록 하였다. 그리고, 문자열이나 심볼이라고 추정된 연결성분들의 위치로 화면을 자동으로 옮겨 사용자가 편리하게 정보를 입력할 수 있도록 하였다.

벡터 데이터 형식 변환

본 논문에서 구현한 벡터 데이터의 형식은 내부적인 형식으로 표현되고 관리된다. 그러나 본 논문에서 개발한 도구는 GIS를 위한 벡터링 도구이므로 외부의 GIS S/W에서 이용할 수 있는 형태로 변환해 주어야 한다. 그래서 본 논문에서는 많이 이용되는 지형정보시스템중의 하나인 ARC/INFO와 CAD 데이터 형식 중의 하나인 DXF 형식으로 변환하는 기능을 구현하였다. 그 외의 데이터 형식으로는 위의 두 가지 형식으로부터의 변환이 가능하다. 그리고 벡터 데이터의 좌표계를 단순한 이미지 좌표계로부터 다른 실세계 좌표계로의 변환도 필요한데 현재는 지도 제작에 널리 통용되고 있는 UTM(Universal Traverse Mercator) 좌표계로의 변환을 지원한다.

실험 및 토의

본 논문에서 설계한 벡터링 도구는 SUN SPARC II 워크스테이션에서 X Window와 Motif를 활용하여 구현되었다. 실험 이미지는 VTRAK에서 사용한 실험용 이미지를 벡터링에 필요한 부분을 추출하여 만든 이진 이미지이며(Fig. 8-(a)), 벡터링 결과는

Fig. 8-(b)와 같다.

Fig. 8을 보면 이미지 중에서 검게 나와 있는 세 부분은 filled area type의 연결성분이므로 그 외곽선을 따라서 벡터링이 되어 있고, 나머지 부분들은 그 중심선을 따라서 벡터링 되어 있음을 볼 수 있다. 그리고 중앙 부분에 굽어져 있는 철도를 나타내는 부분을 보면 철도 양쪽으로 돌출해 있는 가지들은 절단한 상태로 벡터링 되어 있음을 볼 수 있다. 이것은 후처리 과정에 포함되어 있는 지능적 편집 기능으로 처리된 것이다. 그 외에도 그림에서는 나와 있지 않지만, 점선과 같은 부분도 자동적으로 인식하여 하나의 아크로 벡터 데이터화 하는 과정, 이미 생성된 벡터 데이터들에 대해서 수정 및 편집하는 기능들을 구현하여 실험하였다.

설계된 벡터링 도구의 문제점으로는 벡터링 과정에서 자주 만나게 되는 교차점 분석 과정에서 이미지에 포함되어 있는 잡음에 의해 교차점 분석이 잘못 되는 경우가 발생하였다. 이를 무조건 사용자에게 넘기기 보다는 좀 더 잡음에 강한 분석 방법을 찾거나, 현재 사용하고 있는 방법을 보완하여야 할 것으로 본다. 그리고, 전처리 단계에서 구한 연결성분들을 분류하는 과정에서 잘못 분류되는 연결성분들이 생겼다. 그 이유는 두 가지 이상의 그룹으로 분류될 만큼 서로 비슷한 분류특성을 가지고 있었기 때문이며, 결과적으로 재 벡터링 해야 하는 문제가 있었다. 이것은 연결성분이 두 개 이상의 연결성분 그룹으로 분류될 수 있는 특성을 가지고 있었기 때문이다. 그러므로 연결성분의 분류 과정을 좀 더 정확하게 할 수 있는 방법을 연구하여야 할 것으로 생각된다. 그



(a) Input Raster Data

(b) Extracted Vector Data

Fig. 8 Input Binary Image and Vectoring Result

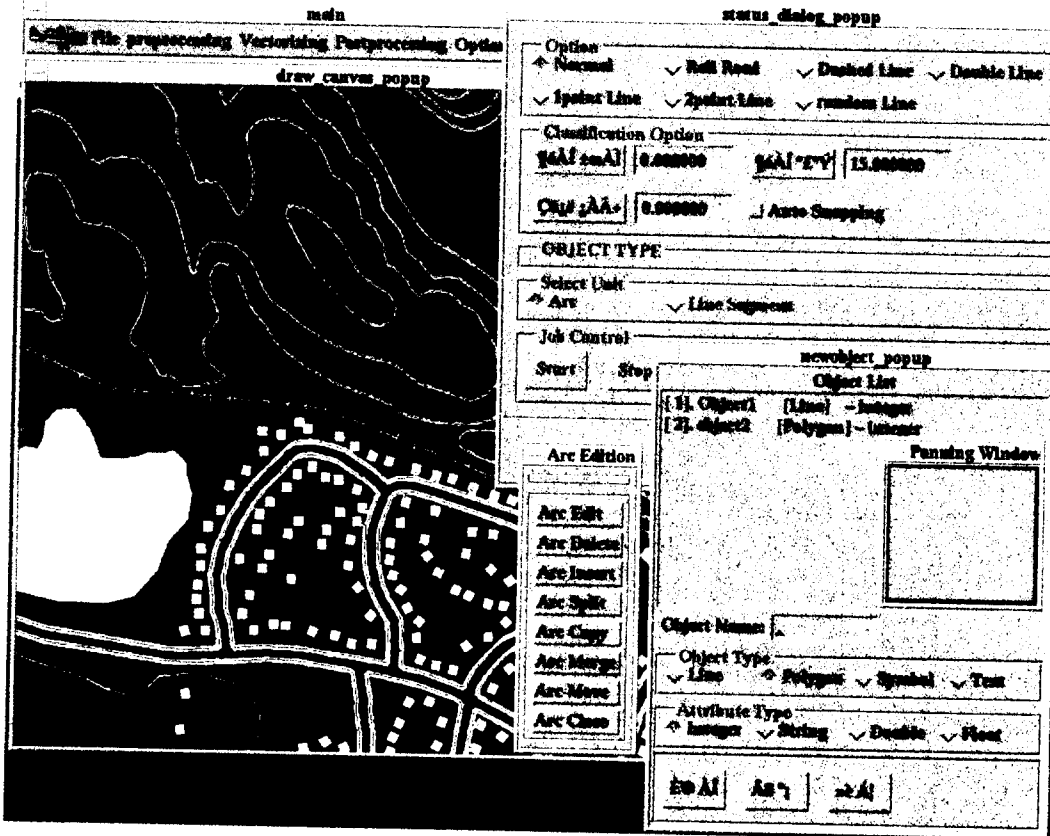


Fig. 9 Postprocessing Screen of Vectoring Tool

리고, Fig. 9에는 본 벡터링 도구의 후처리 과정의 화면을 나타내었다.

결 론

본 논문에서는 기존의 벡터링 도구개발에 대한 현황과 문제점을 알아 보았다. 그리고 GIS를 위한 효율적이고 체계적인 벡터링 도구를 제안하고 구현하였다. 벡터링 도구의 자동화율과 처리 효율을 높이기 위해서 우선 입력 이미지를 연결성분의 집합으로 다시 표현하였고 이를 벡터링 방법에 따라서 세 개의 연결성분 그룹으로 분류하였다. 그리고 연결성분 그룹중에서 line type 그룹의 연결성분들은 우선 세선화에 의한 벡터링 기법을 적용하여 사용자의 개입없이 자동으로 벡터링 할 수 있도록 하였고, 도면의 상태에 따라 자동 벡터링이 어려운 것은 사용자와 대화식으로 처리할 수 있도록하는 반자동 벡터링 방법도 제안하여 구현하였다. Filled area type 그룹의 연결성분은 그 윤곽선을 따라 자동으로 벡터링하고, 나머지 symbol이나 text type의 연결성분들은 후처리 과정에서 사용자가 대화식으로 처리할 수 있게 하였다. 또한 후처리 과정에서는 벡터링 과정에서 발생할 수 있는 오류, 입력 이미지에서의 잡음에 기인하는 오류 등을 지능적이면서도 편리하게 수정 및 편집할 수 있도록 하였다.

그리고 기존의 벡터링 도구가 벡터 데이터를 일률적인 기준에 의해 생성함으로써 발생하는 여러가지 문제점을 살펴보고, 이를 해결하기 위한 벡터 데이터의 다단계 표현방법을 제안하였다. 또 이 방법은 많은 양

의 벡터 데이터를 화면에 출력하였을 때, 응집성 증가로 인해 발생하는 판독력 저하를 어느 정도 해결하고, 또한 데이터 처리 부담도 줄이는데 유용하게 쓰일 수 있음을 본 논문에서 개발한 벡터링 도구에 적용하여 봄으로써 확인하였다.

앞으로의 연구 방향은, 연결성분의 분류 과정을 좀 더 정확하게 할 수 있는 방법과 문자열이나 심볼의 자동인식 방법에 대해 연구하여, 벡터링 도구가 좀 더 지능화 될 수 있도록 하는 것이다. 그리고 좀 더 사용자에게 편리한 인터페이스를 설계하는 것도 중요한 연구가 될 것이다.

감사의 글

본 연구는 1992년 한국과학재단 협력연구과제의 연구비 지원을 받아 수행되었습니다.

참 고 문 헌

- 정진완, 1995, 국가 GIS S/W 기술 개발 계획, '95 GIS S/W 개발에 관한 워크샵, 데이터베이 스연구회, 제11권 특집호, pp.201-212.
- 이성환, 1993, 문자인식 -이론과 실제-, 홍릉과학 출판사
- 상공부, 1993, 소규모 지도정보 시스템 기술 개발에 관한 연구(제 1차년도 중간 보고서)
- 쌍용정보통신(주), 1995, GIS 2001, 여름호
- R. C. Gonzalez and R.E. Woods, 1992, Digital Image Processing, Addison-Wesley

- L.O' Gorman, 1990, $k \times k$ Thinning, Computer Vision, Graphics, and Image Processing, Vol. 51.
- V. K. Govindan and A. P. Shivaprasad, 1987, A Pattern Adaptive Thinning Algorithm, Pattern Recognition, Vol. 20, No. 6, pp. 623–637
- S. Aronoff, 1989, GEOGRAPHIC INFORMATION SYSTEMS : A Management Perspective, WDL Publications.
- D. Dori, 1989, A Syntactic/Geometric Approach to Recognition of Dimensions in Engineering Machine Drawings, Computer Vision, Graphics, and Image Processing, Vol. 47, pp. 271–291.
- S.H. Joseph and T. P. Pridmore, 1992, Knowledge-Directed Interpretation of Mechanical Engineering Drawings, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 9, pp. 928–940.
- A. Okazaki et al., 1988, An Automatic Circuit Diagram Reader with Loop-Structure-Based Symbol Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, No. 3, pp. 331–341.
- L.A. Fletcher and R. Kasturi, 1988, A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, No. 6, pp. 910–918.
- R. Kasturi et al., 1990, A System for Interpretation of Line Drawings, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 10, pp. 978–992.
- A.J. Filipiski and R. Flandrena, 1992, Automated Conversion of Engineering Drawings to CAD Form, Proceedings of the IEEE, Vol. 80, No 7, pp. 1195–1209.