

지적도면 수치화를 위한 정밀 벡터라이징 도구 개발

정재준* · 오재홍* · 김용일**

Development of Precise Vectorizing Tools for Digitization of Cadastral Maps

Jae-Joon Jeong* · Jae-Hong Oh* · Yong-Il Kim**

요 약

지적도면은 매 필지별로 토지에 대한 지번, 위치, 경계, 소유권 등을 규정하고 있는 토지에 관한 가장 기본적인 자료이다. 지적과 관련된 업무는 그 동안 거의 수작업에 의한 방법을 사용하여 효율성의 문제가 대두되었다. 따라서 정부에서는 지적도면을 전산화하려 하였으며, 토지 및 입야대장에 대한 속성정보를 모두 전산 입력하였다. 그러나 도형정보인 지적도면의 전산화가 이루어지지 않아 효율적인 토지정보시스템 구축에 많은 어려움이 있다. 따라서 본 연구에서는 오차가 허용될 수 없는 지적도면의 특성을 감안하여, 스크린 디지털라이징을 원형(prototype)으로 하고 작업의 효율성을 위해 선의 교점을 찾는 과정을 선추적 방식을 통해 자동화한, 혼합형(hybrid) 벡터라이징 방식을 개발하였다. 개발된 프로그램을 구동한 결과 벡터라이징의 정확도에 있어서는 스크린 디지털라이징 방법과 동일하였고, 효율성 측면에서는 본 프로그램에 의한 방법이 스크린 디지털라이징 방법보다 35분 정도 시간을 단축할 수 있었다.

Abstract : Cadastral map is the basic data that prescribe a lot number, the classification of land category, a boundary and ownership of the parcels. Because the analogue cadastral map is not appropriate for the Parcel Based Land Information System, computerization of cadastral map is needed. When considering other automatic vectorizing softwares, we conclude that they can not satisfy the accuracy needed in cadastral map. Also screen digitizing methods demand lots of time. So we developed semi-automatic vectorizing program that realized almost capacities, such as overlay display which is needed for screen digitizing, window link, vector file generation, and so forth. As comparing screen digitizing method using AutoCAD with our developed program, we could obtain not only almost same accuracy, but also 35 minute reduction in vectorizing.

* 서울대학교 지구환경시스템공학부 박사과정(Civil, Urban, Geo-System division, Seoul National University)

** 서울대학교 지구환경시스템공학부 부교수(Assistan professor at Civil, Urban, Geo-System division, Seoul Naional University)

1. 서 론

1.1 연구배경

지적도면은 매 필지별로 토지에 대한 지번, 위치, 경계, 소유권 등을 규정하고 있는 토지에 관한 가장 기본적인 자료이다. 정부의 업무, 특히 지방자치단체의 업무 중 토지와 관련된 업무는 전체 업무의 80%에 이르고 있으며(서울시정개발연구원, 1993), 그 내용은 토지의 위치정보, 토지특성, 토지가격, 이용제한 및 자원관리와 관련된 것이다.

지적과 관련된 업무는 그 동안 거의 수작업에 의한 방법을 사용하여 효율성의 문제가 대두되었다. 또한 일제시대에 처음 제작된 지적도면은 지금까지 사용되면서 정확도의 문제, 관리상의 문제가 존재하고 있었다. 따라서 정부에서는 지적도면을 전산화하려 하였으며, 이러한 정책의 일환으로 1982년부터 토지 및 임야대장에 대한 속성정보를 모두 전산 입력하였다. 그러나 도형 정보인 지적도면의 전산화가 이루어지지 않아 효율적인 토지정보시스템 구축에 많은 어려움이 있다.

1978년부터 서울시와 대한지적공사에서 Computer vision 사의 CGP/1000 시스템을 도입하여 컴퓨터를 이용한 지적도면 제작성 사업을 수행해 오고 있으나(백승철, 1994), 서울시와 대한지적공사에서 제작성하고 있는 방법은 디지털이징에 의한 수동입력방법으로 정확도 높은 결과를 산출하지만, 가장 문제가 되는 것은 작업속도의 한계로 축척 1/600의 지적도면을 좌표독취기를 이용하여 입력하는데 1일 정도가 소요되고 있으므로 72만 매에 달하는 지적도를 모두 전산화하기 위해서는 입력자를 50명이라 가정하고(좌표독취기를 50대가 가정하고), 1년간 작업일 수를 250일이라 하였을 때, 산술적으로 57년의 시간을 필요로 한다. 따라서 새로운 입력방법을 도입하지 않고서는 전체 지적도면의 전산화는 거

의 불가능하다고 할 수 있다.

이와 같은 수동입력방법의 단점을 보완할 수 있는 방법으로는 스크린 디지털이징방법과 자동입력방법이 있다. 이 두 방법은 지적도면을 먼저 스캐닝한 후 작업을 실행하므로 일단 지적도면이 스캐닝된 후에서 원시데이터가 변형되지 않는 장점이 있다. 그러나 스크린 디지털이징은 수동입력방법에 비해 입력시간의 절감효과가 거의 없고, 자동입력방법은 처리된 지적도면의 결과물인 벡터의 위치와 개수가 원시 데이터와 다를 뿐 아니라, 도면의 상태에 따라서 적용하기 어려운 경우가 있어 두 방법 모두 효과적인 지적도 전산화를 위한 도구라 할 수 없다(부산대학교, 1998). 특히 자동입력방법을 지적도 전산화에 사용하려는 시도가 있었지만, 지적도 전산화의 경우에는 국민의 재산권과 관련이 있는 부분인 만큼 일반 도면처럼 오차가 허용한계의 범위를 넘지 않으면 되는 성질의 것이 아니므로 그 적용이 불가능하였다.

1.2 연구범위 및 방법

본 연구에서는 오차가 허용되지 않는 지적도의 특성을 고려하여 정밀 벡터라이징 방법을 개발하였으며, 되도록 입력시간을 절감할 수 있는 방법을 고안하여 벡터라이징의 효율성을 증진시키고자 하였다. 따라서 정확도 확보를 위해 스크린 디지털이징을 원형(prototype)으로 하였으며, 작업자의 작업량을 줄이기 위해 선의 교점을 찾는 과정을 선추적 방식을 통해 자동화한 혼합형(hybrid) 벡터라이징 방식을 개발하였다. 이를 위해 도면내의 모든 선이 실제 측량결과를 바탕으로 정해진 점을 연결하는 직선이라는 지적도의 특성에 맞는 새로운 선추적 알고리즘을 개발하여 구현하였으며, 스크린 디지털이징에 있어 작업자에게 편리함을 줄 수 있는 여러 기능을 추가하였다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 벡터라이징 방법의 문제점에 대해 살펴보았

고, 3장에서는 혼합형 벡터라이징 방식을 개발하기 위해 사용된 연구개발 내용에 대해 살펴보았으며, 4장에서는 제시된 방법의 정확도와 효율성에 대해 검토하였다. 마지막으로 5장에서는 결론 및 향후연구방향을 제시하였다.

2. 기존 벡터라이징 방법의 문제점

지적도면의 전산화를 위한 방법은 좌표독취기를 이용한 수동입력방법, 스캐닝한 지적도를 컴퓨터화면 상에서 디지털라이징하는 스크린 디지털라이징방법, 그리고 스캐닝한 지적도를 선추적 알고리즘을 이용해 벡터화하는 자동입력방법으로 구분할 수 있다. 다음에서는 각 방법의 문제점에 대해 살펴보려고 한다.

2.1 수동입력 방법

좌표독취기를 이용한 수동입력방법은 비교적 정확한 벡터자료를 얻을 수 있으나, 지적도의 좌표독취에 있어서 작업자의 성격, 숙련도, 심리상태 등에 따라 입력되는 자료의 정확도에 차이가 있을 수 있으며, 작업장의 온도와 습도가 변하여 처음 작업을 시작할 때 정해놓은 기준점의 좌표가 틀려질 수 있는 단점이 있다(대한지적공사, 1987). 또한 앞에서 살펴본 것처럼 엄청난 시간을 요하는 단점이 있어 이를 이용하여 지적도를 전산화하는 것은 거의 불가능하다.

2.2 스크린 디지털라이징 방법

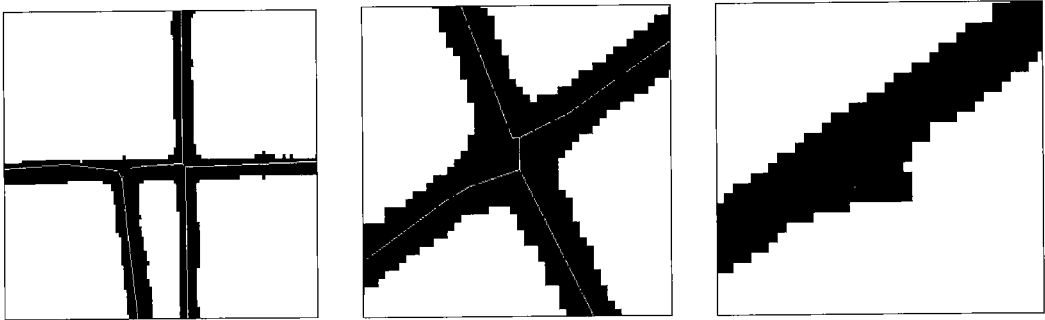
스크린 디지털라이징 방법은 원시 데이터가 변하지 않고, 벡터화에 사용할 기기가 컴퓨터이므로 기기의 제약은 받지 않으며, 입력된 데이터가 좌표독취기를 이용하는 방법만큼 정확하다는 장점이 있다. 스캐닝한 레스터파일을 AutoDesk사의 AutoCad 프로그램에서 화면상에 띄운 후 작업하는 것은 스크린 디지털라이징의 대표적인 방법이다. 그러나 이

방법은 작업자가 스크린 상에서 포인터를 직접 움직여야 하므로 좌표독취기를 이용한 수동입력방법과 마찬가지로 작업자가 수동으로 필지경계점을 찾아야한다. 즉, 스캐닝한 지적도면에서 필지의 크기가 크거나, 스캐닝 해상도가 큰 경우에는 작업자의 작업량이 더욱 증가하게 된다. 그러므로 이 방법도 지적도 전산화를 위한 효율적인 방법이라 할 수 없다.

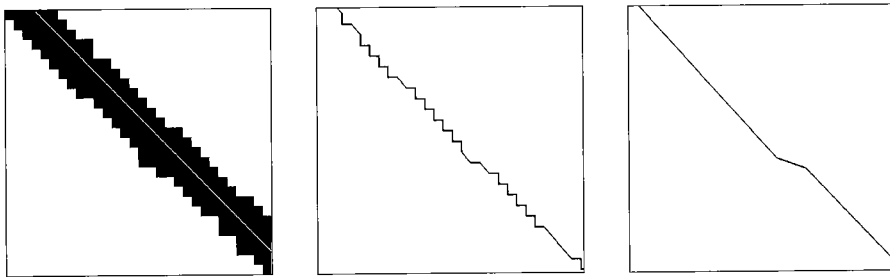
2.3 자동입력방법

도면관리 분야 등에서 사용하고 있는 선추적 방식에 의한 자동벡터라이징은 기존 종이도면의 입력을 효율적으로 하기 위해 개발한 것으로 도면의 상태가 양호한 것은 완전자동화가 가능하고, 벡터의 위치가 부정확하고 입력이 올바르게 되지 않은 부분에 대해서는 스크린 디지털라이징을 할 수 있도록 되어있다.

자동입력방법은 작업의 속도는 현저히 향상되지만 정확도가 확보되지 못하고, 도형인식결과에 대한 신뢰도가 미약할 뿐만 아니라 도면의 상태가 좋지 못할 경우 작업을 할 수 없는 문제가 있다. 대부분의 자동입력방법은 세선화를 먼저 실행한 후, 선추적 방식에 의해 벡터라이징을 실시하는데 이 과정에서 형성된 벡터는 선을 중심으로 한 것이다. 그러나 지적도면내의 선은 점을 연결한 직선일 뿐, 지적도 상의 선은 그 자체로 의미가 없으므로 이 방법 역시 지적도 전산화에 적합하지 않다. 즉, 자동입력방식의 문제점은 지적도내에서 의미를 지닌 점을 제대로 찾지 못하는 것이다. 그림1과 2는 자동 벡터라이징 소프트웨어를 사용한 결과를 나타내고 있다. 이 중 그림 1은 자동입력방식에 의해 생성된 벡터라이징의 결과물을 보여주고 있는데 과다하게 많은 벡터가 생성되고 있음을 보여준다. 일부 프로그램에서는 일반화과정을 통해 이런 문제점을 해결하기도 한다(그림 2 참조). 그러나 이 방법 역시 일반화과정에서 선의 위치가 올바



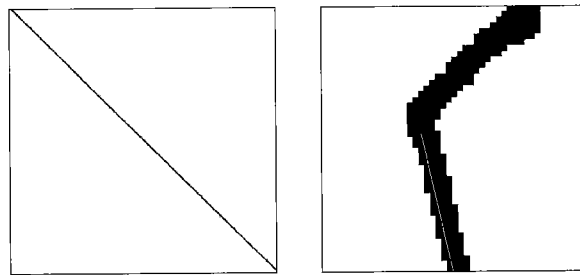
[그림 1] 자동입력방법을 적용한 벡터라이징 결과



(a)

(b)

(c)



(d)

(e)

[그림 2] 자동입력방법의 일반화과정의 문제점

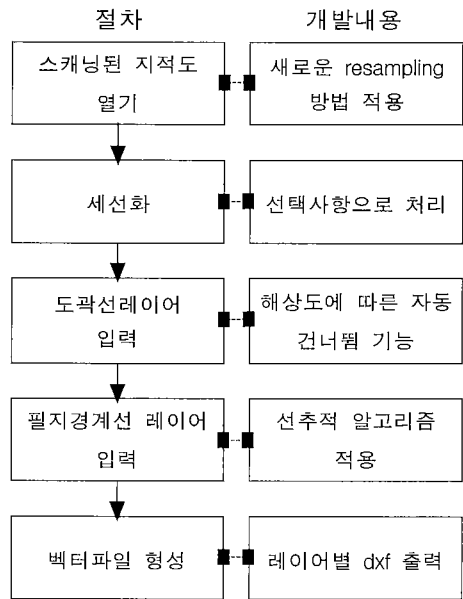
르지 않게 입력되는 단점이 있다. 그림 2(a)의 두꺼운 선은 지적도를 스캐닝한 후 얻은 래스터 파일이고, 가는 실선은 옳게 입력되어야 할 벡

터이다. 실제 프로그램을 실행한 결과 그림 2(b)와 같은 1차 벡터라이징 결과를 얻을 수 있었고, 일반화 과정을 통해 그림 2(c)와 그림 2(d)와 같

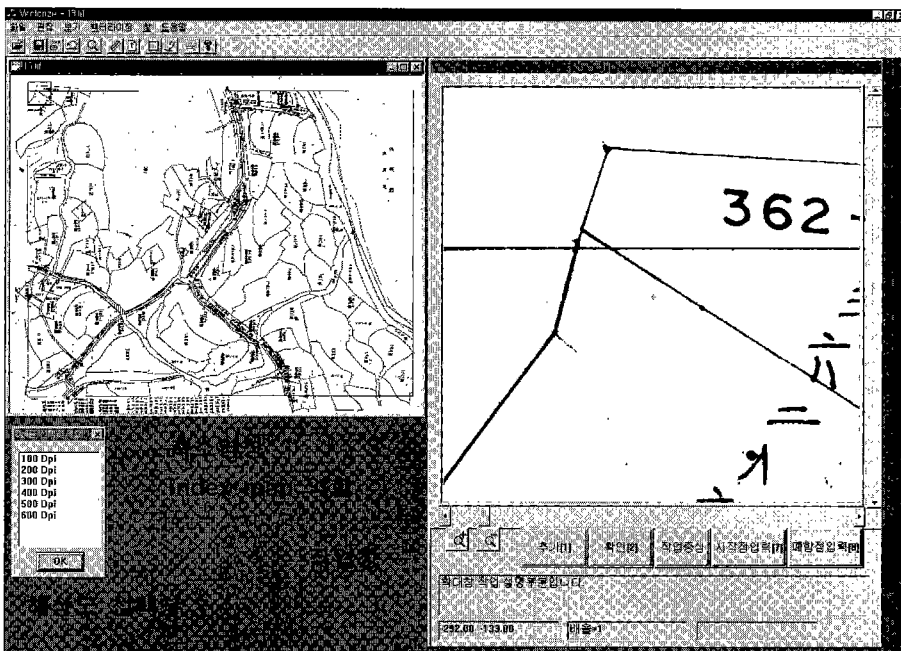
은 벡터를 얻을 수 있었으나 그림2(b)(c)(d)에서 보여지는 모든 벡터는 그림2(a)의 벡터와 개수, 위치 등이 달라 그 결과를 사용하기에 부적합하다. 그러한 문제점을 그림 2(e)에서 볼 수 있다.

3. 연구개발 내용

본 연구에서는 오차가 허용될 수 없는 지적도면의 특성과 벡터라이징의 효율성을 감안하여 스크린 디지털라이징을 원형(prototype)으로 하고, 작업자의 작업량을 줄이기 위해 선의 교점을 찾는 과정을 선추적 방식을 통해 자동화하여 작업자가 직접 벡터라이징 할 점들을 옮겨다녀야 할 불편을 없앤 hybrid 방식의 벡터라이징 방식을 개발하였다(그림 3, 4 참조). 이를 위해 지적도의 특성에 맞는 새로운 선추적 알고리즘을 개발하여 구현하였으며, 스크린 디지털라이징에 있어



[그림 4] 연구개발 프로그램의 시행절차



[그림 3] 프로그램 구동화면

작업자에게 편리함을 줄 수 있는 여러 기능을 추가하였다.

구현 언어로는 Visual C++를 사용하였다.

3.1 새로운 resampling 방법의 적용

스캐닝은 벡터라이징 전체 과정의 기본 데이터를 제공하는 것이다. 스캐닝된 지적도는 그 파일의 형식에 관계없이 지적도 열기를 통하여 열려질 수 있다. 이 과정에서는 새로운 영상 resampling 방법이 사용되고 있다. 통상 400dpi로 스캐닝된 지적도라 할 지적도라 할 지라도 영상의 크기가 8000×7000 화소 이상이 되므로(표1 참조) 일반 상용 프로그램처럼 영상을 축소해서 볼 경우 지적도내의 선들이 거의 보이지 않게 된다. 그림3에서 좌측에 나타난 영상은 지적도를 최대 600화소 내로 resampling한 것이다. 좌측의 영상은 작업자가 지적도를 전산화하는 과정에서 index map의 역할을 하므로(본 프로그램에서는 작업하는 구역에 대한 표시 및 이미 작업을 한 지역에 대한 표시가 벡터로 나타난다) 지적도내의 선이 작업자에게 잘 보이도록 새로운 resampling 방법을 사용하였다. 그림 5는 11000×9000 영상을 resampling한 것으로 그림 5(a)는 본 프로그램 실행시 index map으로 나타나는 지적도의 형태이며, 그림 5(b)는 상용 영상 처리 소프트웨어를 실행 시켰을 경우 나타나는 축소영상인데 그림 5(a)의 경우가 훨씬 뚜렷한 축소영상을 나타내고 있음을 알 수 있다.

또한 지적도 열기를 한 후에는 지적도의 스캐닝된 dpi를 선택하는 모드가 있다. 이는 도곽선 레이어를 입력함에 있어 적절한 간격을 설정하는 역할을 하게 된다.

3.2 세선화

세선화는 문자인식, 회로기판의 결점 검사, 지문 인식 등 여러 가지 분야에서 광범위하게 사용되고 있는 중요한 기술로, 어떤 이미지 패

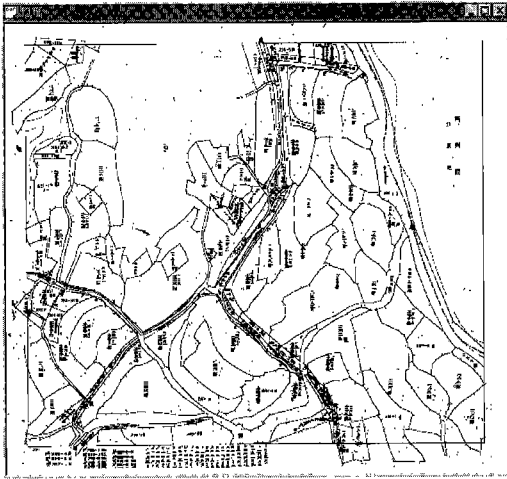
턴의 자동인식을 위해 필요한 최소의 양으로 정보의 양을 줄이는 것을 그 목적으로 한다. 즉, 세선화된 이미지는 끝점, 교차점, 그리고 성분들 사이의 연결성 등 중요한 특징들을 추출하는 것을 돕는 것으로 패턴인식에 있어서 중요한 요소 기술이다. 디지털 패턴의 세선화는 화소에 기반을 둔 반복적 외곽화소 제거 세선화 알고리즘과 화소에 기반을 두지 않는 세선화 알고리즘으로 크게 나눌 수 있다.

본 연구개발 프로그램에서 세선화는 반드시 거쳐야 할 과정은 아니지만, 세선화를 거친 영상을 이용한 선추적결과가 세선화를 거치지 않은 영상을 이용한 선추적결과보다 지적도내의 필지경계선을 결정하는 점의 위치에 보다 가깝게 위치할 가능성이 많아 사용자에게 편의를 제공할 수 있다. 그러나 영상의 크기가 표1에서처럼 매우 크므로 세선화에 소요되는 시간이 많아 세선화 과정을 거친 후 벡터라이징을 실시하기 위해서는 먼저 스캐닝된 데이터를 성능이 우수한 컴퓨터를 이용하여 배치모드로 세선화를 하는 과정이 필요하다.

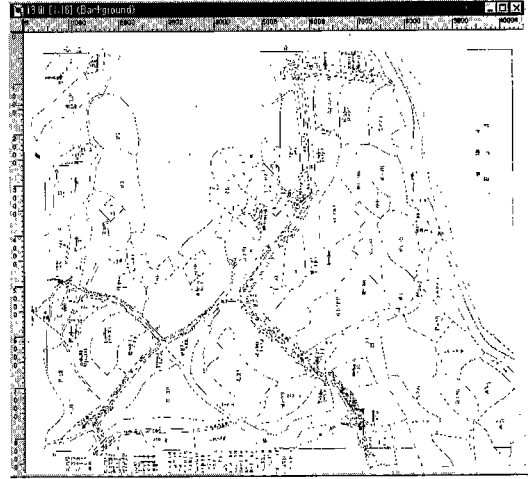
본 연구에서는 화소에 기반을 둔 반복적 외곽화소 제거 세선화 알고리즘 중 잡음에 강하면서도 우수한 결과를 나타내는 「가중치를 이용한 병렬 세선화 알고리즘」이다(한낙희와 이필규, 1996). 기존의 세선화 알고리즘이 교차점등에 있어 왜곡된 세선화 과정을 보이는데 반해, 이 알고리즘은 교차점이나 꺾인 점 등 지적도면에 주로 사용되는 선형에 대해 왜곡이 적은 세선화 결과를 보인다.

여기서 언급하고 있는 가중치란 의미 있는 화소(흑화소)를 1로, 나머지 화소(백화소)를 0으로 두었을 때 3×3 창에서 중심화소를 제외한 나머지 8-이웃 화소들의 수를 세어 그 중심화소에 가중치로 부여하는 것이다. 따라서 가중치는 적게는 0이, 많게는 8이 나올 수 있다. 그러나 0은 의미 있는 화소를 제외한 나머지 화소들에 부여한 값과 같으므로 가중치가 0인 것은 9로 부여하기로 한다. 원영상으로부터 계산하여 얻은 가

지적도면 수치화를 위한 정밀 벡터라이징 도구 개발



(a) 새로운 resampling방법에 의한 축소영상



(b) 기존 resampling방법에 의한 축소영상

[그림 5] 새로운 영상 resampling 방법의 적용

<표 1> 세션화 소요시간 Pentium II -233Mhz 사용

DPI	경계선의 두께	영상크기	최소 RAM	세션화 시간
400	2-3-5	8000×6800	54M	1시간
500	2-4-6	10000×8500	85M	1시간 40분
600	3-5-8	12000×10200	122M	2시간 30분
800	4-6-10	16000×13600	218M	4시간

중치를 통해 새로운 영상을 생성하여 그 가중치의 수치와 배치상태에 따라 삭제여부를 결정하게 된다.

제안된 세션화 알고리즘의 개략적인 흐름은 다음과 같다.

Step 1. 각 흑화소의 가중치를 결정

Step 2. 외곽화소들은 각각의 가중치 K를 가지고 있으므로 가중치 K에 해당하는 삭제조건을 검색하여 만일 그 삭제조건을 만족하면 삭제한다.(가중치 9를 위한 삭제조건 포함)

Step 3. 각 흑화소의 가중치를 결정

Step 4. 외곽화소들은 각각의 가중치 K를 가지

고 있으므로 가중치 K에 해당하는 삭제조건을 검색하여 만일 그 삭제조건을 만족하면 삭제한다.(가중치 9를 위한 삭제조건 제외)

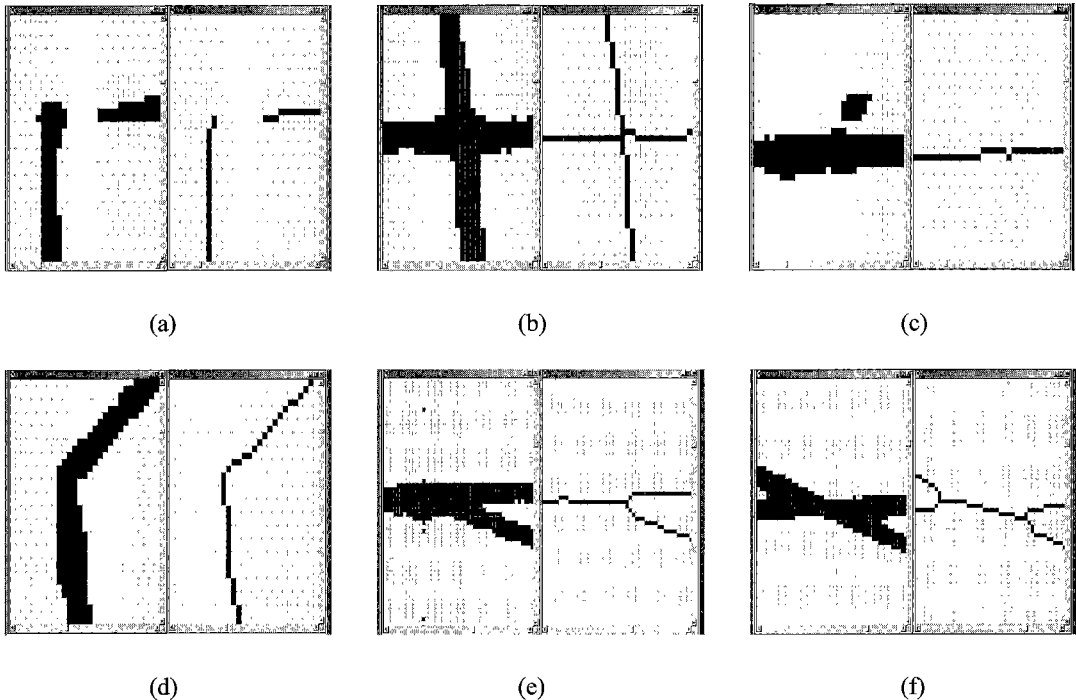
Step 5. 삭제조건을 만족하는 흑화소가 없을 때까지 Step 3에서 Step 4의 과정을 반복한다.

지적도면을 스캐닝하여 세선화한 결과를 원영상과 아래와 같이 여러 경우에 대하여 비교하여 보았다. 시작점, 세 방향 교차점, 네 방향 교차점, 꺾인 점, 잡음이 섞인 곳 등 다양한 경우에 대하여 비교해 보았을 때 대체적으로 좋은 결과를 보이고 있다. 그림 6부터 그림 6(d)까지는 특이할 만한 문제점을 발견하지 못하였고, 그림 6(e), 그림 6(f)와 같이 선들이 예각으로 만

나는 경우에는 그 교차하는 면적이 넓으므로 원하는 정도의 세선화 결과를 산출하지 못하였다. 그러나 이후 자동선추적을 통해 필지경계점을 추출하는 과정이 있으므로 이런 문제점은 해결될 수 있다.

3.3 자동 선추적 알고리즘

앞에서 이미 제시된 것과 같이, 수동입력방법은 효율성에 문제가 있고 자동 벡터라이징은 정확도가 확보되지 않을 뿐만 아니라 도면의 상태에 따라 결과가 판이하다. 지적도면은 특히 개인의 재산권 행사라는 측면이 있기 때문에 지형도의 벡터라이징이나 도면의 벡터라이징처럼 허용오차를 감안한 선추적 방식의 자동 벡터라이징 과정을 통해 전산화할 수 있는 성격의 것



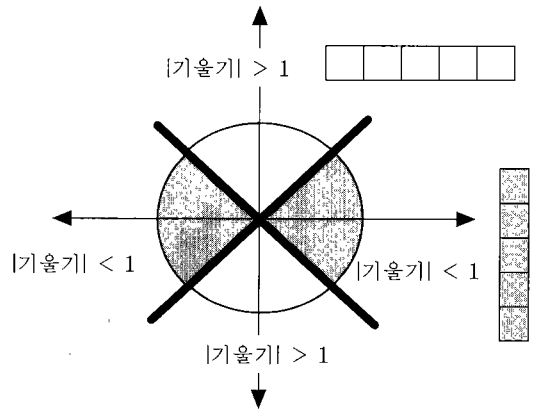
[그림 6] 원영상과 세선화 결과

이 아니다. 또한 지적도면내의 선은 모두 점을 연결한 직선이므로 지적도 상의 선은 그 자체로 의미가 없으며, 오직 선의 교점인 점만이 필지를 구성하는 요소이다. 그래서 허용오차를 최대한 줄일 수 있도록, 점을 정밀하게 입력하는 부분을 수동 작업으로 만들고 그 외 점이 존재하는 곳인 굴곡점, 교차점 등의 위치를 찾아주는 작업등은 자동으로 하는 자동 선추적방식을 채택하였다.

자동 선추적 알고리즘은 다음과 같다.

- Step 1. 굴곡점 또는 교차점을 수작업으로 입력한다.
- Step 2. 지적도상의 선 위에 방향 지시점(벡터라이징을 진행시킬 방향)을 입력한다.
- Step 3. 굴곡점 또는 교차점과 방향 지시점 사이의 기울기를 계산하고 사용할 템플릿의 모양을 결정한다.
- Step 4. 세선화를 하지 않은 영상인 경우, 템플릿을 지적도 선에 씌운 다음 선의 중심과 템플릿 중심이 일치하도록 한다. 그리고 템플릿이 진행될 기울기를 다시 계산한다.
- Step 5. Step 2에서 결정된 방향, Step 3에서 결정된 템플릿 모양, Step 4에서 계산된 기울기 등을 사용하여 선을 따라 템플릿을 이동시킨다. 그리고 템플릿의 양극단에 선의 화소값인 0이 들어오면 굴곡점 또는 교차점으로 인식하고 템플릿의 진행을 정지시킨다. 그리고 사용자의 점 입력을 기다린다.

Step 3에서 이전 굴곡점 또는 교차점과 방향 지시점 사이의 기울기에 따라 템플릿의 모양을 결정하는 규칙은 다음과 같다. 그림 7에서 좌표의 중심이 작업자가 입력했던 점(굴곡점 또는 교차점)이고 방향 지시점과의 관계에서 계산되는 기울기가 -1에서 1사이일 경우에는 세로 막

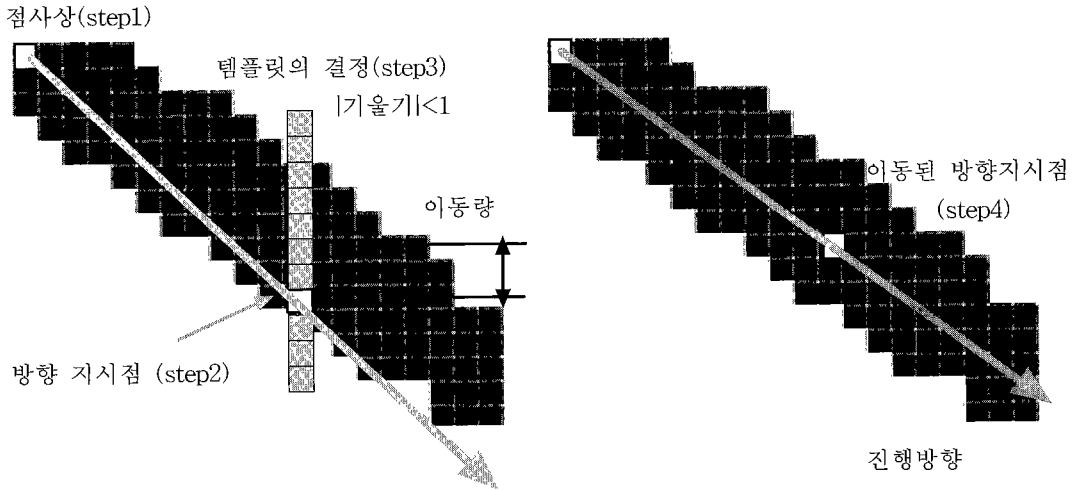


[그림 7] 자동 선추적 벡터라이징에 사용되는 템플릿의 결정

대 모양의 템플릿을, 1보다 크거나 -1보다 작을 경우에는 가로 모양의 템플릿을 사용하게 된다. 이 때, 템플릿의 길이는 지적도가 스캐닝되는 해상도(dpi)에 따라 결정되게 된다. 즉, 해상도가 높아질수록, dpi가 높아질수록(선의 굵기가 증가할수록) 템플릿의 길이는 길어져야 한다.

Step 4는 템플릿을 지적도 선 위에 정확히 올리고, 선을 추적할 방향을 다시 계산하는 단계이다(그림 8 참조). 템플릿을 임의로 선 위에 올린 다음, 템플릿 중심과 선의 중심간의 편차를 계산하여 그 양만큼 템플릿을 이동시킨다. 그러면 템플릿의 중심과 선의 중심은 일치하게 되며, 그 중심과 전 작업자가 입력했던 점과의 기울기가 곧 템플릿이 진행할 방향이 된다. 세선화를 한 경우에는 선의 굵기가 1이므로, Step 4는요성 필요성이 없어지게 된다.

Step 5는 결정된 규칙에 따라 템플릿을 이동시키며 굴곡점과 교차점을 자동적으로 찾아 화면에 띄워 주는 부분이다. 템플릿을 이동하는 동안 템플릿 양 극단에 선의 화소값인 0이 들어오는지는 모니터링을 하게 된다. 화소값 0이 들어오면 템플릿의 이동은 정지하고 굴곡점 또는



[그림 8] 자동 선추적 벡터라이징의 절차

교차점이 찾아졌음을 알리는 메시지를 내보낸다. 그림 9는 템플릿이 정지하는 예를 나타낸다.

위와 같은 방법을 사용하여 자동적으로 찾아진 굴곡점, 교차점에 사용자는 정확한 점을 입력해주면 된다. 좀 더 정확한 굴곡점 교차점 검색을 위해서는 옵션인 세션화 과정을 거쳐야 한다. 하지만 지적도와 같이 크기가 큰 경우 세션화 시간이 오래 걸릴 수 있으므로 주의해야 한다.

이 알고리즘의 장점은 계산이 빠르고, 자동적으로는 찾아내기가 힘들었던 예각으로 꺾인 굴곡점을 찾아낼 수 있다는 것이다.

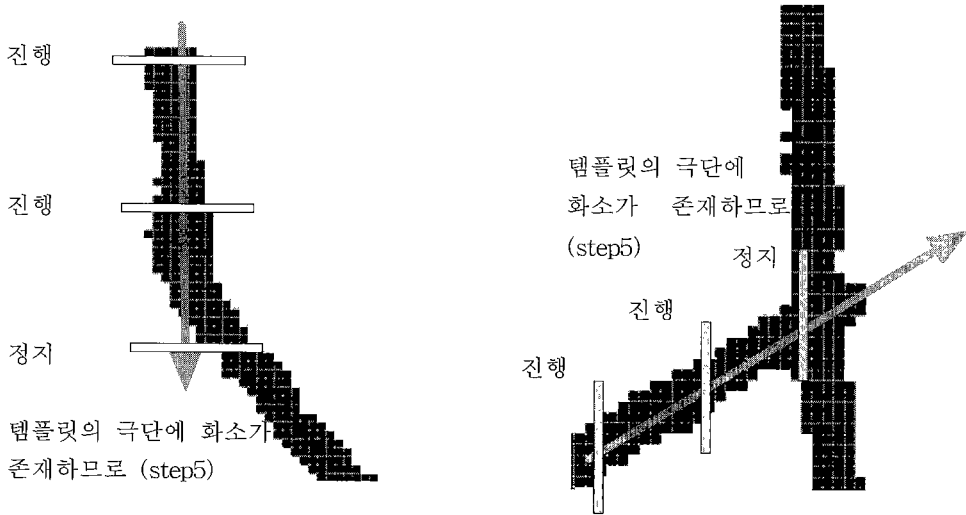
더욱이, 다음 그림 10과 같이 보존 상태가 좋지 않아 필지경계선 등이 끊어져 있는 경우에도 영향을 받지 않고 벡터라이징을 할 수 있는 장점이 있다.

3.4 레이어 입력 및 저장

지적도 전산화를 위해 입력해야할 레이어는

모두 7가지이다(Gr:격자점, Jm:지목, Jp:지번, Li:경계선, Po:매칭기준점, Qu:도곽선). 그러나 실제 벡터라이징 프로그램에서 입력할 레이어는 경계선레이어와 도곽선레이어이다.

도곽선레이어는 지적도의 도곽선을 입력한 것으로, 이는 벡터라이징이 끝난 후 지적도의 변형보정 등에 사용되는 중요한 레이어이다. 지적도의 도곽선은 4개의 직선으로 구성되어 있지만, 지적도의 변형 등으로 인해 실제 직선이 아닌 경우가 많다. 이를 위해 실제 지적도에서 약 3-4cm마다 도곽선을 입력해야 하는데 본 프로그램에서는 스캐닝된 지적도의 해상도에 따라 지적도의 약 3-4cm에 해당하는 화소에서 작업자의 입력을 기다리도록 하였고, 또한 선추적 알고리즘을 사용하여 도곽선이 직선에서 벗어났다고 판단될 때도 작업자가 벡터를 추가할 수 있도록 하게 하였다. 따라서 실제 도곽선 1번당 입력되는 도곽선레이어의 벡터 수는 10개 정도가 된다.

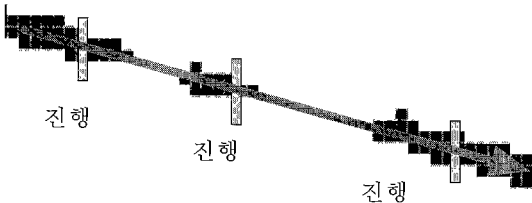


[그림 9] 자동 선추적 방식에 의하여 교점을 찾는 경우

도곽선레이어 입력이 끝나면 자동으로 도곽선레이어의 좌우/상하단의 4점을 연결한 경계선레이어가 창에 자동으로 화면에 출력되게 하고, 이 때부터 경계선레이어를 본격적으로 입력한다. 경계선레이어를 입력할 때 작업자의 편의를 위하여 확대창(실제 작업창으로 그림3의 오른쪽 창)에서 축소/확대가 자유롭게 하였다. 작업자의

편의를 위해 벡터레이어를 입력할 때 포인터를 옮길 필요가 없도록 단축키 기능을 추가하였다. 또한 경계선 폐합시 같은 점을 두 번 입력했을 때 발생하는 위치오차를 없앨 수 있도록 스냅 기능을 구현하였고, 다른 곳에서 벡터를 입력하기 시작할 때 기존의 벡터의 시작점이나 끝점에서 입력을 시작할 수 있도록 시작점입력 역시 스냅기능을 사용할 수 있도록 하였다.

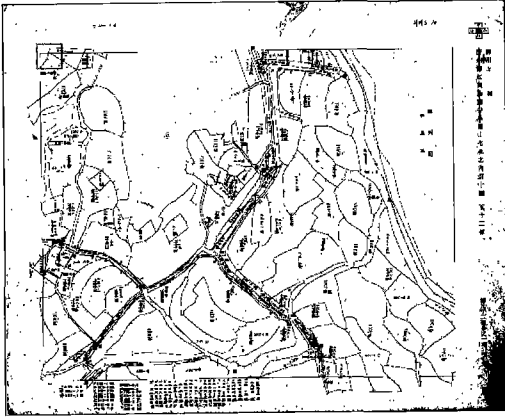
입력된 벡터데이터는 dxf 형식으로 저장되도록 하였으며, 저장시 레이어에 대한 정보를 가질 수 있도록 하였다.



[그림 10] 선이 끊어진 경우의 자동 선추적 벡터라이징

4. 정확성과 효율성 비교

본 연구의 정확성과 효율성을 비교하기 위해 임의의 지적도 1매를 선택하여 AutoCad를 이용한 스크린 디지털라이징과 본 연구에서 개발한 프로그램을 이용한 벡터라이징을 실시하여 양



(a) 스캐닝한 지적도



(b) 벡터라이징 후 dxf 파일

[그림 11] 스캐닝한 지적도와 벡터라이징으로 생성된 dxf 파일

자를 비교하였다. 자동입력 방법과의 비교는 자동벡터라이징의 정확성과 생성된 벡터의 수가 달라 비교하지 않았다.

실험을 실시하기 위한 작업절차 및 결과는 다음과 같다. PentiumⅡ-233Mhz의 CPU와 128MB의 RAM을 가진 컴퓨터를 이용하여 실험을 실시하였으며, 실험대상 원도는 400dpi로 스캐닝한 지적도로 영상의 화소 크기는 8000×6700이었다. 대체적으로 필지의 크기는 컸지만 하천을 따라 복잡한 형상을 한 필지가 상당부분 있어 입력시간이 많이 소요되었다.

4.1 AutoCad를 사용한 스크린 디지털라이징

스크린 디지털라이징을 위해 먼저, 스캐닝된 지적도 래스터 파일을 AutoCad에서 배경화면으로 스크린에 출력하였다. 그런 후 디지털라이징을 실시하였는데 먼저 도곽선레이어를 입력하였다. 도곽선은 종방향으로 20개 횡방향으로 25개를 디지털라이징 하여 총 90개의 벡터를 가진 레이어를 형성하였는데 입력소요시간은 15분이었다.

도곽선 입력을 작업자가 직접 포인터를 움직여서 시행하므로 입력된 도곽선 벡터의 간격이 일정하지 않았다. 다음으로 경계선레이어를 입력하였으며 필지경계점의 중심부에 벡터의 시작점과 끝점이 위치하도록 충분히 확대한 후 입력하였으며 입력소요시간은 150분이었다.

4.2 연구개발 프로그램을 이용한 벡터라이징

본 연구에서 개발한 프로그램을 이용하여 벡터라이징을 실시함에 있어 먼저 실행하는 작업은 세선화이다. 세선화를 하면 레이어 입력에 걸리는 시간을 줄일 수 있는 장점이 있기는 하지만 세선화에 상당히 많은 시간이 소요된다. 실험대상 지적도를 세선화하는데 소요된 시간은 약 35분이었다. 그러나 세선화는 배치작업으로 할 수 있어 소요시간의 산정에서는 제외하였다. 그 후 작업순서는 스크린 디지털라이징과 같다. 먼저 도곽선레이어를 입력하였는데 이 과정에서 소요되는 시간은 10분 정도였다. 또한 일

<표 2> 스크린 디지털라이징 방법과 연구 개발프로그램의 벡터라이징 방법의 비교

레이어명	스크린 디지털라이징		연구개발 프로그램	
	입력시간	비 고	입력시간	비 고
도곽선	15분	일정하지 않은 도곽선벡터 형성	10분	일정한 간격의 도곽선 벡터 형성
경계선	150분	축소, 확대를 많이 해야 하는 번거로움 발생 작업자가 필지경계점 부근까지 수동으로 포인터를 옮김	120분	필지경계점 부근에 포인터가 자동으로 위치하므로 작업편의성 증대
세선화	-		(35분)	배치작업이 가능하므로 소요시간 산정에서 제외
계	165분		130분	

정한 간격으로 도곽선을 디지털라이징할 수 있었다. 그런 후 경계선레이어를 입력하였으며 입력 시 스크린 디지털라이징과 마찬가지로 영상을 충분히 확대해 필지경계점을 입력하였으며 입력에 소요된 시간은 120분이었다.

그림 11은 스캐닝한 지적도와 본 연구에서 개발된 프로그램에 의해 벡터라이징한 결과를 나타내고 있다.

4.3 방법간 비교

위의 두 방법은 필지경계점을 입력하는 방식이 모두 스크린 디지털라이징을 근간으로 한 것이기 때문에, 두 방법의 정확성은 작업자의 숙련도나 정성과 관련된 문제이지 방법자체의 차이 때문에 발생하는 것이라 할 수 없다. 따라서 방법간의 비교는 효율성 측면에서 실시하였다.

위의 양 방법을 비교한 결과는 표 2와 같다. 도곽선레이어를 입력하는데 있어서는 두 방법이 시간적으로 크게 차이가 나지는 않았으나, 본 연구에서 개발된 프로그램을 사용할 때, 일정한 간격의 도곽선을 입력할 수 있었고 또한 도곽선 변형된 경우에도 효과적으로 입력할 수 있는 장점이 있었다. 경계선레이어를 입력하는

데 있어서는 본 연구에서 개발한 프로그램을 이용하는 것이 30분 정도 시간절약 효과가 있었다. 뿐만 아니라, 포인터가 필지경계점의 중심부근에 거의 1-2 화소 내에 위치하므로 작업자가 작업과정에서 느끼는 부담감이 줄어들었다.

결과적으로 정확성에 있어서는 문제가 없었으며 효율성에 있어서는 30분 정도의 단축을 꾀할 수 있었다. 실험을 실시하기 위한 작업절차 및 결과들을 비교하였다.

4.4 벡터라이징의 정확도

지적도면 전산화 과정에서 가장 중요한 문제는 원시데이터인 지적도와 벡터라이징 작업의 결과물은 수치데이터의 오차이다. 벡터라이징 방식에 따라 정확도의 표현방법이 상이하다. 현행 사용되고 있는 수동입력방식의 경우 원시데이터인 지적도와 수동입력방식에 의해 산출된 벡터데이터를 플로터로 출력한 도면을 중첩하여 유관으로 검사하는 방법을 사용하고 있다. 반면, 스크린 디지털라이징이나 자동입력방법과 같이 지적도를 스캐닝한 래스터파일을 벡터라이징의 원도로 사용하는 경우 정확도는 다음과 같이 세측면에서 검토되어야 한다.

- 지적도와 래스터파일 간의 정확도
- 래스터파일과 이를 이용하여 입력된 벡터파일 간의 정확도
- 지적도와 입력된 벡터파일 간의 정확도

이 중 첫 번째 정확도는 결국 스캐너의 정확도를 의미하는 것으로, 일반적으로 드럼식 스캐너의 정확도가 평판스캐너의 정확도보다 우수하다고 하나, 지적도는 그 소재가 알루미늄 쉘트지 또는 한지로 되어 있는 경우가 많아 드럼식 스캐너를 사용할 수 없다. 또한 평판 스캐너는 정확도의 문제 뿐 만 아니라 평판의 크기가 지적도의 크기에 맞지 않아 때문에 어려움이 있었다. 그러나 최근 들어 평판스캐너의 정확도가 향상되고 평판의 크기 역시 증가하여 이러한 문제가 거의 해결되었다(과학기술부,1998).

두 번째 정확도는 개인의 성향, 경력 등과 좌우되는 것이다. 따라서 이것에 의하여 입력방법 자체의 정확도를 평가하는 것이 불가능하다. 실제적으로 본 연구에서 개발된 프로그램을 사용하여 지적도를 전산화하는 과정에서 지적도를 400dpi로 스캐닝하였을 경우 지적도 내의 선은 평균 3화소 정도이다. 작업시 중심점을 잘못 입력하는 경우 x,y 방향으로 최대 1 화소 정도의 오차가 나타나는데 이를 도면상에서의 거리로 계산하면,

$$\sqrt{2} \times 25.4/400 = 0.0898 (mm)$$

이다. 이는 통상사용되는 도면상에서의 허용 제오오차의 한계인 0.2mm나 수치지도에서 자동 제도기의 해상도인 0.1mm보다 작으므로 작업자 개인의 과실만 없다면 문제가 없을 것으로 판단된다.

마지막으로 지적도와 입력된 벡터파일 간의 정확도는 결과적으로 검수해야 할 정확도이지만 첫 번째와 두 번째의 정확도에 의해 결정되는 것이므로 첫 번째와 두 번째 정확도에 문제가 없다면 세 번째 정확도 역시 별 문제가 없을

것으로 판단된다.

5. 결론 및 향후연구과제

지적도 전산화를 위해서 기술적으로 가장 문제가 되는 것은 지적도면을 정확하고 효율적으로 입력하는 것이다.

수동입력방법의 단점을 보완할 수 있는 방법으로는 스크린 디지털라이징방법과 자동입력방법이 있다. 이 두 방법은 지적도면을 먼저 스캐닝한 후 작업을 실행하므로 일단 지적도면이 스캐닝된 후에서 원시데이터가 변형되지 않는 장점이 있다. 그러나 스크린 디지털라이징은 수동입력방법에 비해 입력시간의 절감효과가 거의 없고, 자동입력방법은 처리된 지적도면의 결과물인 벡터의 위치와 개수가 원시데이터와 다를 뿐 아니라, 도면의 상태에 따라서 적용하기 어려운 경우가 있어 두 방법 모두 효과적인 지적도 전산화를 위한 도구라 할 수 없다. 특히 자동입력방법을 지적도 전산화에 사용하려는 시도가 있었지만, 지적도 전산화의 경우에는 국민의 재산권과 관련이 있는 부분인 만큼 일반 도면처럼 오차가 허용한계의 범위를 넘지 않으면 되는 성질의 것이 아니므로 그 적용이 불가능하였다.

본 연구에서는 지적도 전산화를 위한 벡터라이징 도구를 개발하기 위해 오차가 허용될 수 없는 지적도면의 특성과 벡터라이징의 효율성을 감안하여 스크린 디지털라이징을 원형(prototype)으로 하고, 작업자의 작업량을 줄이기 위해 선의 교점을 찾는 과정을 선추적 방식을 통해 자동화한 hybrid 방식의 벡터라이징 방식을 개발하였다. 이를 위해 지적도의 특성에 맞는 새로운 선추적 알고리즘을 개발하여 구현하였으며, 스크린 디지털라이징에 있어 작업자에게 편리함을 줄 수 있는 여러 기능을 추가하였다.

개발된 프로그램을 구동한 결과 벡터라이징의 정확도에 있어서는 유관 검사시 스캐닝된 지적경계선을 벗어나는 벡터화된 선은 없어 스크

린 디지털라이징 방법과 동일하였다. 또한 효율성을 테스트하기 위해 AutoCad를 이용한 스크린 디지털라이징과 본 프로그램을 이용한 디지털라이징을 비교한 결과 본 프로그램에 의한 방법이 35분 정도 시간을 단축할 수 있었다.

향후 본 연구의 최종 결과물이 형태는 토폴로지가 형성되어있지 않은 dxf 포맷이라는 점을 고려할 때, 지적도를 효과적으로 이용하고, 지적도를 기반으로 한 통합적인 정보시스템을 구축하기 위해서 토폴로지를 형성에 관한 연구가 진행되어야 할 것이다. 이 과정이 이루어진다면 GIS의 기본도로 지적도를 효율적으로 이용할 수 있을 것이다.

감사의 글

본 연구는 과학기술부의 “지적도 전산화를 위한 도곽보정, 접합보정 및 품질검사 전문가시스템 개발”과제에 의해 수행된 연구로서 과학기술부에 감사드립니다.

참고 문헌

- 1) 과학기술부, 1998, 지적도 전산화를 위한 도곽보정, 접합보정 및 품질검사 전문가시스템 개발, pp. 79-81.
- 2) 대한지적공사, 1987, 지적(임야)도 정비사업에 관한 연구, p. 46.
- 3) 백승철, 1994, 스캐닝방법에 의한 지적도면의 재작성에 관한 연구, 청주대학교 석사학위논문.
- 4) 부산대학교, 1998, 매핑기술개발 -제1세부과제 자료획득 및 수정 S/W 개발-위탁과제 기존도면의 입력 및 수정 소프트웨어 개발, 과학기술부.
- 5) 서울시정개발연구원, 1993, 서울시 지리정보시스템 구축에 관한 연구(I), 시정연 93-R-25.
- 6) 한낙희, 이필규, 1996, 가중치를 이용한 병렬 세션화 알고리즘, Korea Journal of Cognitive Science, Vol.7, No.1, pp. 5-35.