

論文97-34C-10-11

선분들간의 관계를 이용한 지도 인식

(Map Recognition Using the Relation between Line Segments)

張 京 植 * , 金 在 熹 **

(Kyung Shik Jang and Jaihie Kim)

요 약

지도는 많은 선분들로 구성되기 때문에 지도에 나타나는 대상체들을 인식하는 것은 많은 직선 중에서 특정 대상체의 성질을 만족하는 것들을 그룹화하는 과정에 해당한다. 따라서 수많은 선분들 중에서 한 대상체를 구성하는 선분들을 찾는 방법에 대한 문제는 지도를 인식하는 과정에서 가장 중요한 문제이다. 이 논문에서는 지적도를 대상으로 하여 블록, 보도, 도로의 인식을 위한 새로운 접근 방법으로, 선들간의 관계들을 정의하고, 이를 이용하여 탐색공간을 줄이며 대상체를 구성하는 선을 찾는 방법과 대상체들간의 구조적 정보를 기반으로 지도를 인식하는 방법을 제안하였다. 실험 결과, 탐색공간을 효과적으로 줄일 수 있었으며, 선들이 겹치거나 블록을 구성하는 선들이 끊어지는 경우에도 대상체를 정확히 인식하는 결과를 얻을 수 있었다.

Abstract

Because a map consists of many lines, a process of recognizing an object in a map is one to group lines that are consistent with the structure of the object. Therefore, it is important to establish an efficient method for finding lines which constitute objects. In this paper, a novel method for recognizing roads and districts in a land register map is proposed. In this method, lines that constitute objects are searched in a reduced search space by defining some relations between lines, and objects are recognized by using the structural information of the objects. As a result, the search space is reduced effectively, and even objects that embody the broken lines and the crossed lines are recognized.

1. 서 론

각 대상체(object)가 뚜렷이 구분되는 일반적인 문서와는 달리 지도에 나타나는 대부분의 대상체는 서로

간에 겹쳐서(overlap) 표현되어 각 대상체를 구성하는 선분들은 교차(intersection)하게 된다. 또한, 기호와 문자 등이 선들 위에 표시되기 때문에 많은 선분들이 끊어지게 된다. 이 때문에 지도 인식 문제는 그 동안 많은 연구가 수행되었음에도 불구하고 해결되어야 할 기본적인 문제점을 지니고 있다. 즉, 지도에 나타나는 수많은 선분 중에서 한 대상체를 구성하는 선분들이나 선분들 상호간에 연관된 것들을 탐색공간을 줄이며 효율적으로 찾는 방법에 대한 문제이다. 지도에 나타나는 각 대상체를 인식하는 문제는 많은 직선 중에서 각 대상체의 성질을 만족하는 것들을 그룹화 하는 과정에 해당한다^[1, 2]. 그러나, 지도는 수많은 선분들로 구성

* 正會員, 大宇電子 戰略技術 第 1 研究所

(Advanced Technology Laboratory 1, Daewoo Electronics Co., Ltd)

** 正會員, 延世大學校 電子工學科

(Dept. of Electronic Engineering, Yonsei University)

※ 본 연구는 94년도 한국과학재단 연구비 지원에 의

한 결과임 (과제번호 : KOSEF 941-0900-072-2)

接受日:1997年6月28日, 수정완료일:1997年9月23日

되므로 이들간의 연관관계를 찾기 위한 탐색공간 (search space)은 인식하고자 하는 지도에 포함된 선분의 수가 많아짐에 따라 기하 급수적으로 커지게 된다.

이에 따라 Dominique Antoine^[2, 3]는 시가도(urban map)에 나타나는 모든 대상체는 다각형의 형태로 구성되는 성질을 이용하여 연결된 선들간의 각도, 추적한 횟수, 추적방향 및 이전에 추적여부 등과 같은 선들의 연결성에 대한 규칙을 설정하였다. 그는 이 규칙에 따라 다각형 형태의 구획에 해당하는 선을 추적하는 방법(line following)을 제안하고 구획을 인식하였다. Satoshi Suzuki^[4]는 한 영역을 둘러싸는 선들로서 경계(border)를 정의하고 한 영역을 둘러싸는 내부 경계(hole border)와 한 영역에 의하여 둘러싸인 외부 경계(outer border)로 구분하였다. 이를 이용하여 전체 선 성분을 추적하여 내부 경계를 갖는 선분들을 하나의 폐곡선으로 인식하는 경계 추적(border tracing) 방법을 제시하였다. 폐곡선을 찾은 후 폐곡선의 면적 및 구성 선분들의 길이에 대한 특성을 조사하여 건물을 인식한 후, 이 결과를 이용하여 도로를 인식하였다. Luca Boatto^[5]는 대상으로 정한 지도에서 건물은 빗금으로 표시된다는 특성을 이용하여 평행하게 연속적으로 배열된 선들을 빗금으로 인식한 후, 빗금 주변의 선을 추적하여 건물을 인식하였다. 그러나 이 방법들은 기본적으로 끊어진 선분이 없다는 가정을 기반으로 하기 때문에 대상체를 구성하는 선분들이 끊어지는 경우에는 올바른 인식을 하지 못하였다. 또한, 대상체를 구성하는 선을 찾는 방법이 연결된 선들간의 방향성과 같은 지엽적인 정보만을 사용하므로 선들간의 겹침 정도가 심한 지도에서는 인식오류가 많이 발생하기 때문에 다른 종류의 지도에 적용시 많은 문제점을 내포하고 있다.

이 논문에서는 지적도(cadastral map)를 대상으로 하여 구획(parcel), 블록(district), 도로(road)의 인식을 위한 새로운 접근 방법으로, 선들간의 관계들을 정의하고, 이를 이용하여 탐색공간을 줄이며 대상체와 연관된 선을 찾고 대상체들간의 구조적 정보를 기반으로 지도를 인식하는 방법을 제안하였다. 수많은 선분들 중에서 한 대상체를 구성하는 선들을 효과적으로 구별하기 위하여 각 선분들간에 오른쪽 이웃 관계, 왼쪽 이웃 관계, 평행 관계, 겹침 관계, 교차 관계, 일직선 관계 등을 정의하였다. 또한, 이들 관계들을 이용하여 블록의 특성을 정의하였으며, 도로 상호간의 관계

와 도로와 블록간의 관계를 이용하여 도로의 특성을 정의하였다. 대상체를 구성하는 선들은 제안한 선들간의 관계를 이용하여 쉽게 찾아지며, 제안한 도로와 블록의 특성을 이용하여 이들 대상체들을 효과적으로 인식하였다. 선을 찾고 블록을 인식하는 과정에서, 제안한 직선들간의 관계를 이용하면 선 추적 방법^[2, 3]에 비하여 탐색공간을 효과적으로 줄일 수 있다. $N*N$ 개의 구획으로 구성된 블록을 인식하는 경우, 선 추적 방법은 $6*N^2+2*N$ 개의 선을 추적하여야 하지만, 제안한 방법은 $2*N^2+2N$ 개의 선만을 조사하면 된다. 또한, 선 추적방법과는 달리 선들이 겹치거나 블록을 구성하는 선들이 끊어지는 경우에도 대상체를 정확히 인식하는 결과를 얻을 수 있었다.

2절에서는 전처리 과정과 선 성분 추출 과정을 서술하고 3절에서는 선들간의 관계와 대상체에 대한 구조적 정보를 이용하여 각 대상체를 인식하는 방법을 제시한다. 4절에서는 제안한 방법의 효용성을 분석하였다.

II. 벡터화(Vectorization)

스캐너(scanner)를 통하여 지도를 입력하여 256 계조도(grey level)의 디지털화된 영상(digital image)을 얻은 후 이진화(binanzation) 과정을 거친다. 영역 분리를 수행하여 각 영역의 크기에 대한 임계치(threshold)를 사용하여 고립 문자(isolated character)를 제거하고 문자가 분리된 지도 영상에 대한 골격선(skeleton)을 얻은 후 선 성분을 추출한다. 지도는 모든 정보가 선성분으로 나타나기 때문에 식(1)과 같이 표현될 수 있다. 이 식에서 V 는 중요점(important point)들의 집합, 원소 v , 는 각 점의 좌표값 (x_i, y_i) 이며, E 는 직선들의 집합, 원소 e_k 는 (v_k, v_m) 으로 표현되며, 중요점 v_k, v_m 을 잇는 직선을 뜻한다. 중요점은 선 성분을 조각화하는 과정에서 기준으로 사용되는 점으로 이 논문에서는 선분들이 만나는 특징점(feature point)과 한 선분 상에서의 곡률(curvature)이 급격히 변하는 굴곡점(corner point)을 정의하였다.

$$D = (V, E), V = \{v_1, v_2, \dots, v_M\}, E = \{e_1, e_2, \dots, e_K\} \quad (1)$$

특징점은 3 개의 선분이 만나는 연결점(junction point), 4 개의 선분이 만나는 교차점(crossing po-

int), 선분의 끝점(end point)들이며, 이 논문에서는 3*3 마스크 영역에 대하여 식(2)를 사용하여 각 마스크 중심점에서의 교차수(crossing number : CN)^[6]를 구하고 식(3)의 규칙을 사용하여 특징점을 추출하였다. 식(2)에서 Q_i 는 마스크 중심화소에 이웃한 8개의 화소를 시계방향으로 탐색할 때 각 화소에 대한 값으로서 흑화소(black pixel)이면 1, 아니면 0 값을 갖는다.

$$CN = \frac{1}{2} \sum |Q_i - Q_{i+1}| \quad (i=0, \dots, 7) \quad (2)$$

- if $CN = 1$, then 중심 화소는 끝점
- if $CN = 3$, then 중심 화소는 연결점
- if $CN = 4$, then 중심 화소는 교차점

특징점을 구하여 전체 선 성분을 특징점이 양 끝점인 선분들로 분리하고, 한 선분을 구성하는 각 점의 좌표를 사용하여 식(4)에 의하여 곡률을 계산한다^[7]. 곡률 K 는 선분의 길이인 t 의 함수이며, $x(t)$ 와 $y(t)$ 는 각 점의 좌표 값을 나타낸다. 곡률 신호에서 지역적 최대치(local maxima)를 찾음으로써 굴곡점을 구하고 한 선분에서 양 끝점인 특징점과 그 사이에 존재하는 굴곡점들을 차례로 연결하는 직선들을 구함으로써 전체 선 성분을 직선들로 표현하였다. 식(4)에서의 미분 계산시 정확한 수학적 표현은 $m=1$ 인 경우이지만, 이 논문에서는 m 을 매개변수로 사용하여 m 값이 커질수록 더 중요한 곡률 변화만을 찾으려 하였으며 각 선분의 길이에 비례하는 m 값을 사용하였다. 그림 1(b)는 (a)의 지도 영상에 대하여 특징점과 굴곡점을 추출한 결과를, (c)는 이들 점들을 연결하는 직선을 표시한 것이다. 원에 해당하는 선분은 곡률이 일정하기 때문에 굴곡점을 찾지 못하고 양 끝점인 특징점을 연결하는 한 직선으로 나타난다.

$$K = \frac{x' \ddot{y} - \ddot{x} y'}{(x'^2 + y'^2)^{3/2}} \quad (4)$$

where,

$$x' = \frac{dx}{dt} \quad x'' = \frac{d^2x}{dt^2} \quad y' = \frac{dy}{dt} \quad y'' = \frac{d^2y}{dt^2}$$

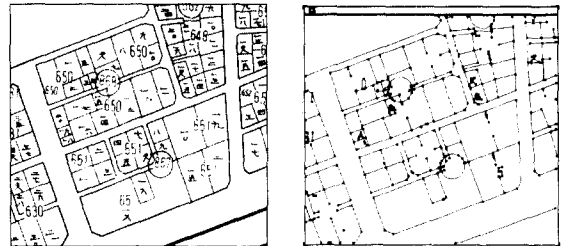
$$\frac{dx}{dt} \cong x(t+m) - x(t-m), \quad \frac{d^2x}{dt^2} \cong x(t+m) + x(t-m) - 2x(t)$$

III. 블록 및 도로망 인식

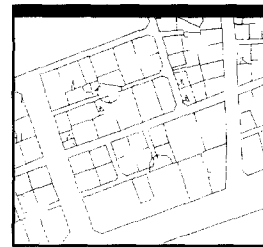
1. 원의 구조적 정보를 이용한 원 인식

이 논문에서 인식대상으로 선정한 지도에는 곡선의 전형적인 형태인 원이 존재하며, 이 원은 다른 대상체

의 인식을 어렵게 하기 때문에 우선적으로 제거하여야 한다. 벡터화 과정의 결과로 모든 선분은 직선으로 표현되기 때문에, 원을 찾기 위해서는 많은 직선 중에서 곡선에 해당하는 직선들을 탐색하여야 한다. 이 논문에서는 지도상의 실제적인 점들과 직선 벡터간의 차이, 즉 벡터 오류(vector error)가 큰 직선을 곡선으로 정의하였다. 그러나 원을 구성하는 선들이 이웃하는 선들과 연결되어 잘게 잘라지는 경우에는, 벡터 오류가 작아지게 되어 직선과 곡선을 구분할 수 없게 된다. 따라서 원의 부분을 이루는 몇 개의 곡선을 찾은 후, 이를 기반으로 원의 구조적 성질을 이용하여 나머지 선들을 탐색하였다. 원을 인식하는 과정은 다음과 같으며, 그림 2(b)는 원을 인식한 결과이다.



(a) 지도영상 (b) 중요점 추출



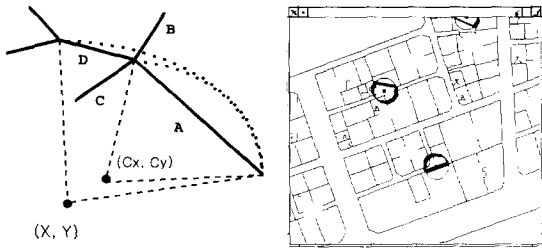
(c) 벡터화

그림 1. 직선 근사화
Fig. 1. Vectorization.

- 단계 1. 곡선으로 선택된 선(그림 2(a)에서 선분 A)을 기준 원호로 정하고 원 영상에서 이 선에 해당하는 실제 점들의 좌표를 이용하여 원의 중심점 (C_x, C_y)을 구한다.
- 단계 2. 기준 원호와 연결된 한 직선을 원의 일부분으로 가정하고, 이들로 구성되는 원호의 중심점 (X, Y)을 구한다. 이때 연결된 직선이 여러 개인 경우(선분 B, C, D)에는 각 직선에 대하여 원호를 구성하고 각 원호의 중심점을 얻는다.
- 단계 3. 구성한 원호들의 각 중심점과 기준 원호의 중심점간의 거리를 구하고, 이 값이 특정 값 이하

인 직선을 원의 일부분으로 선택한다. 이때 여러 선이 선택되면 가장 작은 거리 값을 갖는 선을 원의 일부분으로 정하며 해당하는 선이 없으면 멈춘다.

단계 4. 선택된 선(선분 D)을 기준 원호에 연결하여 이를 새로운 기준 원호로 정하고 위의 단계를 반복 수행한다.



(a) 원 추출방법 (b) 원 추출 결과

그림 2. 원 인식
Fig. 2. Recognition of a circle.

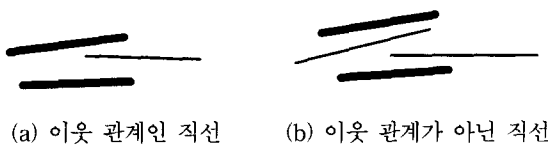
2. 선분간 관계

백터화 결과 얻어진 직선들은 아무런 의미 없는 단순한 자료(data)들이며, 이들중에서 각 대상체를 구성하는 선들만을 찾는 과정은 매우 큰 탐색공간을 필요로 하기 때문에 선들간의 관계를 규정하고 인식과정에서 이를 기반으로 탐색공간을 줄이는 것은 매우 효과적이다^[8, 9]. 이 논문에서는 선들간의 이웃관계^[8]를 이용한 오른쪽 이웃 관계, 왼쪽 이웃 관계와 겹침 관계, 교차관계, 평행관계 및 일직선 관계를 정의하였다.

이웃 관계

두 직선과 두 직선의 끝점을 연결하는 가상 직선으로 구성되는 사변형 내에 다른 직선을 거치지 않고 두 직선을 직접 연결하는 직선이 존재할 때 두 직선은 이웃 관계를 만족한다.

이웃 관계의 정의에 의하여 그림 3(a)에서 굵은 두 직선은 이웃 관계를 만족하며, (b)의 굵은 두 직선은 이웃 관계를 만족하지 않는다.



(a) 이웃 관계인 직선 (b) 이웃 관계가 아닌 직선

그림 3. 직선들간의 이웃 관계
Fig. 3. The neighborhood relation between lines.

오른쪽 이웃 관계

한 선분과 이웃 관계를 만족하는 선분 중에서 식(5)를 만족하는 선분들은 오른쪽 이웃 관계를 만족하는 선분들이다.

왼쪽 이웃 관계

한 선분과 이웃 관계를 만족하는 선분 중에서 식(6)을 만족하는 선분들은 왼쪽 이웃 관계를 만족하는 선분들이다.

$$Nei_R(l) = \{ l_i | l_i \in Nei(l), \theta_i < \theta \} \quad (5)$$

$$Nei_L(l) = \{ l_i | l_i \in Nei(l), \theta_i > \theta \} \quad (6)$$

식 (5)와 (6)에서 $Nei(l)$ 은 직선 l 과 이웃관계를 만족하는 직선들의 집합, θ 는 기준 직선 l 이 x 축과 이루는 각도, θ_i 는 좌표축의 원점을 직선 l 의 중점으로 평행이동 시킨 후 직선 l 과 이웃관계를 만족하는 직선 l_i 의 중점과 직선 l 의 중점을 잇는 직선이 x 축과 이루는 각도이다. 직선간의 오른쪽(왼쪽) 이웃 관계를 구하는 방법은 다음과 같다.

- 단계 1. 모든 직선의 끝점들 x 좌표에 대한 순서집합 (ordered set) X를 구한다. 집합 X는 식(7)과 같이 표현된다.
- 단계 2. 집합 X에서 두 개의 연속된 끝점들로 구성되는 구간들의 집합을 구한다. 이 집합은 식(8)과 같이 표현된다.
- 단계 3. 각 구간에 포함되는 직선들의 집합을 구한 후, 이들을 각 구간 중심에서의 x 좌표값에 대응하는 직선들의 y 좌표 값을 사용하여 정렬 (sorting)한다. 이 결과 각 구간에서 인접한 직선들은 y 축 방향으로 이웃 관계를 만족한다.
- 단계 4. 위와 동일한 방법으로 x 축 방향으로의 이웃 관계를 구한다.
- 단계 5. x 축과 y 축을 $30^\circ, 60^\circ, -30^\circ, -60^\circ$ 의 방향으로 각각 회전 이동시켜 각 방향에 대하여 위의 단계를 수행하고 각 방향에서 한 직선과 이웃관계를 만족하는 선들의 집합을 구한 후, 이들의 합집합을 구하면 그 직선과 이웃관계를 만족하는 직선들의 집합을 얻는다.
- 단계 6. 한 직선 (l)과 이웃관계를 갖는 직선들(l_i)만을 대상으로 하여, 직선 l 이 x 축과 이루는

각(θ)을 구하고 좌표축의 원점을 직선 l 의 중점으로 평행이동 시킨 후 직선 l 의 중점과 직선 l_i 의 중점을 잇는 직선이 x 축과 이루는 각(θ_i)을 구한다. 이때, θ_i 가 θ 보다 작은(큰) 직선들은 그 직선과 오른쪽(왼쪽) 이웃관계를 만족한다.

$$X = \{x_i | x_i < x_{i+1}, i = 1, 2, \dots, n\} \quad (7)$$

$$S = \{(x, y) | x_i < x < x_{i+1}, -\infty < y < \infty, i = 1, 2, \dots, n-1\} \quad (8)$$

겹침 관계

직선 l_a 의 양 끝점 좌표가 각각 $(x_{a1}, y_{a1}), (x_{a2}, y_{a2})$ 이고 직선 l_b 의 양 끝점 좌표가 각각 $(x_{b1}, y_{b1}), (x_{b2}, y_{b2})$ 일 때, 다음 조건중 하나 이상을 만족하면 두 직선은 겹침 관계를 만족한다.

- 조건 1. $x_{a1} < x_{b1} < x_{a2}$ 조건 2. $x_{a1} < x_{b2} < x_{a2}$
- 조건 3. $y_{a1} < y_{b1} < y_{a2}$ 조건 4. $y_{a1} < y_{b2} < y_{a2}$

교차관계

겹침 관계를 만족하는 두 개의 직선이 식(9)를 만족하는 경우, 두 직선은 교차관계가 성립한다. p 가 2차원상의 점을 나타낼 때, 식(9)에서 $f_a(p) = 0, f_b(p) = 0$ 는 각각 직선 l_a, l_b 의 방정식이며, p_{a1}, p_{a2} 는 직선 l_a 의 양 끝점을, p_{b1}, p_{b2} 는 직선 l_b 의 양 끝점을 나타낸다.

$$f_a(p_{b1}) \times f_a(p_{b2}) < 0 \text{ AND } f_b(p_{a1}) \times f_b(p_{a2}) < 0 \quad (9)$$

평행 관계

두 직선 방향의 차이가 임계치 이내이고 직선의 방향에 수직방향으로 두 직선이 겹침 관계가 성립하면 두 직선은 평행 관계를 만족한다.

일직선관계

두 직선의 방향이 동일하고 두 직선의 끝점을 연결하는 직선의 방향이 기존 두 직선의 방향과 동일하면, 두 직선은 일직선 관계를 만족한다.

3. 선분간 관계를 이용한 블록 및 도로망 인식

- (1) 오른쪽 이웃 관계와 왼쪽 이웃 관계를 이용한 블록 인식

선들간의 연결성을 이용하여 한 블록만을 고립시킨 경우에 블록의 외곽선을 구성하는 직선들의 일반적 특

성은, 선의 한쪽에만 직선들이 존재하고 다른 쪽에는 직선들이 존재하지 않는다는 것이다. 따라서 블록의 외곽선을 구성하는 직선들은 식(10)의 집합 C 에 의하여 나타낼 수 있다. 식(10)에서 l_i 는 한 직선을, CC 는 연결된 선들의 집합을, $ExistEle(\cdot)$ 는 집합의 공집합 유무를 나타내고 $n(A)$ 는 집합 A 의 원소 수를 나타낸다.

$$C = \{l_i | ExistEle(Nei_R(l_i)) \text{ XOR } ExistEle(Nei_L(l_i)) = 1, l_i \in CC\}$$

$$\text{Where, } ExistEle(A) = \begin{cases} 1 & \text{if } n(A) > 0 \\ 0 & \text{if } n(A) = 0 \end{cases} \quad (10)$$

이러한 방법은 블록의 외곽선이 볼록한(convex) 경우에는 모든 외곽선을 찾을 수 있으나 오목한(concave) 부분이나 돌출한 부분에서는 외곽선의 부분을 찾지 못한다. 이를 보완하기 위해서 추출한 외곽선을 기반으로 연결된 직선들을 추적하면서 최종적인 외곽선을 추출하였다. 그림 5(a)는 모든 블록들의 외곽선을 찾아낸 후 간략화한 결과이며, 보도는 블록을 구성하는 한 직선을 의미한다.

(2) 평행 관계와 일직선 관계를 이용한 도로 인식

도로는 평행한 두 보도에 의하여 구성되는 영역으로 이 논문에서는 다음과 같은 특성을 정의하였다.

도로의 특성

- 특성 1. 도로는 평행한 보도사이에 존재한다.
- 특성 2. 도로는 블록 내부에 고립되어 존재하지 않는다.
- 특성 3. 동일한 방향으로 연속적으로 존재하는 도로의 넓이는 동일하다.
- 특성 4. 도로는 홀로 존재하지 않고 다른 도로와 연결되어 있다.

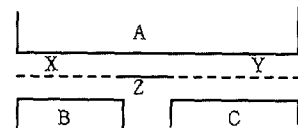
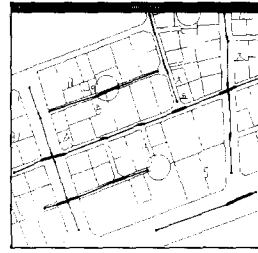


그림 4. 보도와 도로와의 관계
Fig. 4. The relation between roads and sidewalks.

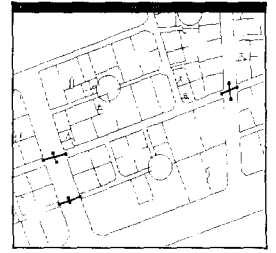
도로의 특성 1.을 이용하여, 평행 관계를 만족하는 두 개의 보도로 구성되는 부분을 도로로서 인식한다. 그림 4.에서 보도 A 와 B 는 도로 X 를, 보도 A 와 C 는 도로 Y 를 구성한다. 도로 Z 는 주변에 평행한 보도가

없기 때문에 보도에 의하여 인식할 수 없는 부분으로서 특성 3.을 기반으로 도로 X, Y의 일직선 관계를 이용하여 도로로 인식한다. 이 논문에서는 전자(X, Y)를 보도 연계 도로, 후자(Z)를 연결도로로 정의한다. 그림 5(b)는 한 보도와 평행 관계를 만족하는 보도들을 나타낸 것이다. 이 경우에 한 보도와 평행한 보도는 여러 개가 존재하기 때문에 그 중에서 넓이가 가장 작은 한 쌍을 도로로 인식하였다. 그림 5(c)는 이러한 과정으로 추출한 모든 도로들을 나타낸 것이며, 블록 내부에 많은 오 인식 도로들을 포함한다.

오 인식 도로를 제거하기 위해 도로의 특성 2.를 이용하여 각 도로가 블록내부에 존재하는지를 검사한다. 이 논문에서는 블록과 도로간의 상대적 위치관계를 이용하여 도로가 블록 내부에 포함되는지를 판정하였다. 한 폐곡선을 시계방향으로 따라갈 때, 폐곡선 내부에 있는 것은 선분의 진행방향에 대하여 계속 오른쪽에 있게 된다. 따라서 각 블록의 외곽선을 시계방향으로 따라갈 때 계속 오른쪽에 있는 도로들은 오 인식 도로로 판단하여 제거하였다. 이 방법은 블록이 블록한 경우에는 대부분 블록 내부에 존재하는 도로를 찾아낼 수 있지만 오목한 부분을 갖는 블록에서는 오 인식 도로를 찾아낼 수 없는 경우도 발생한다. 그림 5(d)는 도로의 특성 2를 이용하여 오 인식 도로를 제거한 결과로서 다수의 오인식 도로를 포함하고 있다.



(e) 연결도로 추출



(f) 사교차점을 구성하는 도로들

그림 5. 도로 인식 과정
Fig. 5. Recognition of roads.

이들 오 인식 도로들은 이 후의 과정에서 도로의 특성 4인 도로의 연결성을 이용하여 찾게 된다. 도로의 특성 3.을 이용하여 일직선 관계를 만족하며 도로의 넓이가 동일한 보도 연계 도로들 사이에 연결도로를 생성한다. 그림 5(e)는 (d)에서 이 성질을 만족하는 연결 도로들을 생성한 결과이다.

(3) 교차관계를 이용한 도로 교차점 인식

이 논문에서는 도로가 만나는 교차점으로서 사교차점과 삼교차점을 정의하였다.

교차점

사교차점 : 네 도로가 만나는 점으로서 2 개의 연결 도로가 교차하는 점이다.

삼교차점 : 세 도로가 만나는 점으로서 한 연결도로와 한 보도 연계 도로가 만나는 점이다.

교차점은 보도가 존재하지 않는 부분에 존재하며, 이 논문에서는 직선간의 교차 관계를 이용하는 다음의 단계로 사교차점을 인식한다. 그림 5(f)는 연결 도로 중에서 교차 관계를 만족하는 도로들만을 나타낸 것이다.

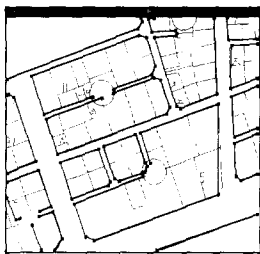
단계 1. 연결도로를 나타내는 선분들 간에 교차 관계를 만족하는 쌍을 구한다.

단계 2. 두 선분간의 교차점을 사교차 점으로 선정하고 교차점을 기준으로 2 개의 연결도로를 각각 2 등분한다.

삼교차점은 다음과 같은 과정을 통하여 인식한다.

단계 1. 연결도로와 한쪽 끝이 다른 도로와 연결되지 않은 보도 연계 도로의 쌍을 추출한다.

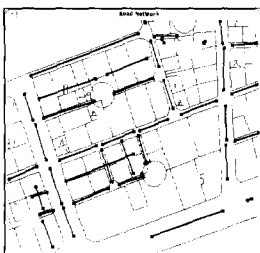
단계 2. 각 쌍을 구성하는 두 도로가 이루는 각도가



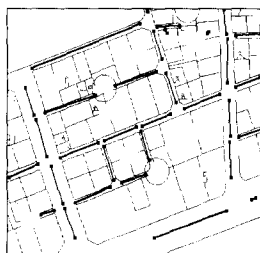
(a) 블록인식결과



(b) 한 보도와 평행관계 만족하는 보도들



(c) 보도연계 도로 추출



(d) 블록안의 도로 제거후

수직에 가까운 것을 선택하고 보도 연계도로의 연장선과 연결도로와의 교차점이 연결도로 위에 위치하는 쌍을 삼교차점을 구성하는 도로로서 선정한다.

단계 3. 교차점을 중심으로 연결 도로를 2등분하고 교차점과 보도 연계 도로간에 연결도로로서 새로운 도로를 생성한다.



(a) 삼교차점을 구성하는 도로들 (b) 최종 인식 도로망

그림 6. 도로 교차점과 도로망
Fig. 6. Road crossing points and road network.

그림 6(a)는 이 과정을 통하여 생성된 연결 도로를 나타낸 것이다. 이후, 오 인식 도로를 제거하기 위하여 도로의 특성 4.를 이용한다. 즉, 도로는 홀로 존재하지 못하고 다른 도로와 연결되어 존재해야 하기 때문에 교차점을 찾은 후의 과정까지 다른 도로에 연결되지 못한 도로들은 오 인식 도로로 판정하여 제거하였다. 그림 6(b)는 최종적인 도로망(road network)을 나타낸 것이다.

IV. 실험 및 고찰

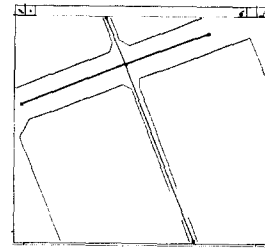
실험을 위하여 사용한 지도는 지적도이며 축적은 1:3000 이다. 스캐너를 사용하여 300 DPI(dot per inch)의 해상도(resolution), 256 계조도로 지적도를 입력하였으며, Sun Sparc-10 워크스테이션(workstation)에서 실험하였다. 그림 7과 그림 8은 각 실험 영상에서 대상체들을 인식한 결과이다.

지도를 인식하기 위한 기존의 연구들은 구획이나 블록을 구성하는 선들을 찾기 위하여 일반적으로 선 추적 방법이나 경계 추적 방법을 사용하지만, 이들 방법은 많은 탐색 공간을 필요로 하며 블록의 내부에 있는 구획을 구성하는 선들이 끊어지지 않은 상태에서만 블록의 올바른 인식결과를 얻을 수 있다. 그러나 이 논

문에서 제안한 방법은 선들의 탐색공간 측면에서도 많은 개선을 얻을 수 있었으며, 구획을 구성하는 선들이 끊어진 상태에서도 정확하게 블록을 인식하였다.

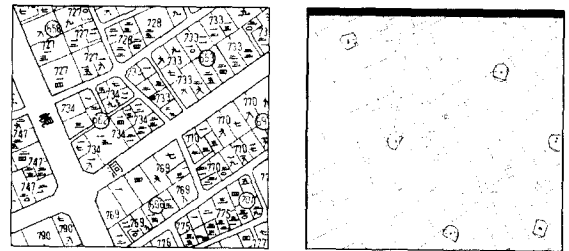


(a) 실험영상 (b) 블록 인식결과

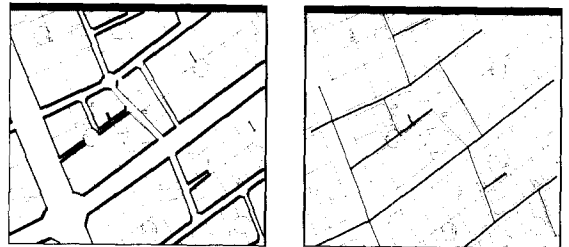


(c) 도로 인식결과

그림 7. 인식 결과(실험영상 1)
Fig. 7. Recognition results(test image 1).



(a) 실험영상 (b) 원 인식결과



(c) 블록 인식결과 (d) 도로 인식결과

그림 8. 인식 결과(실험영상 2)
Fig. 8. Recognition results(test image 2).

기존의 방법과 이 논문에서 제안한 방법을 탐색공간 측면에서 비교하기 위하여 그림 9(a)의 $N \times N$ 개 구획

으로 구성된 블록을 인식하는 과정에서 필요한 탐색공간을 비교하였다. $N \times N$ 개인 내부의 사각형들은 구획에 해당하고 외곽선들은 블록에 해당한다. 선 추적 방법^[2, 3]을 이용하여 블록의 외곽선을 추출하기 위해서는, 내부의 구획들을 모두 추적하여 인식한 후, 구획들간에 공유되지 않은 선들을 추적하여 블록을 인식한다. 그림 9(a)에서 한 구획은 4개의 선으로 구성되고 N^2 개의 구획이 존재하며, 전체 $2 \times N^2 + 2N$ 개의 선이 존재하며, 바깥쪽의 선은 $4 \times N$ 개이다. 따라서 블록 내부의 구획들을 인식하기 위하여서는 $4 \times N^2$ 개의 선을 추적하여 모든 구획을 인식하고, 블록을 인식하기 위하여서는 모든 선을 조사하여 구획간에 공유되지 않은 선들을 찾아야 하기 때문에 $2 \times N^2 + 2N$ 선을 추적하여야 한다. 결과적으로 $6 \times N^2 + 2 \times N$ 개의 선을 추적하여야 블록을 인식하게 된다. 그러나 제안한 방법은 각 선에 대하여 3 절의 식(10) 만을 조사하기 때문에 블록을 구성하는 전체 선들의 개수인 $2 \times N^2 + 2N$ 개의 선만을 탐색하면 된다. 그림 9(b)는 (a)의 블록에 대하여 선 추적 방법을 사용하여 블록을 인식하는 방법과 이 논문에서 제안한 선들간의 관계를 이용하여 블록을 인식하는 방법을 탐색공간 관점에서 비교한 것이다.

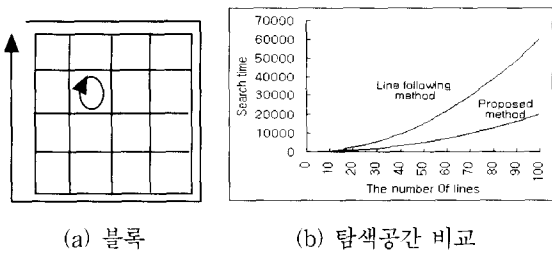


그림 9. 탐색공간 비교
Fig. 9. Comparison of search time.

V. 결 론

최근 들어 스캐닝된 지도 영상에서 지도 정보를 인식하기 위한 많은 연구들이 진행되고 있다. 그러나 지도를 인식하기 위한 기존의 연구들은 선들로 구성되는 각 대상체를 인식하기 위하여, 선들간의 연결성과 연결 방향 등과 같은 지엽적인 정보만을 사용하여 대상체의 형태에 부합되는 선들을 추출함으로써 대상체를 인식하기 때문에 선들이 끊어진 지도에서의 대상체 인식에는 많은 어려움을 내포하고 있다. 또한 이와 같은 지엽적인 정보만을 사용하기 때문에 많은 선들의 특성

을 하나씩 조사하여 필요한 선을 탐색해야 하는 문제점이 있다.

이 논문에서는 지적도를 대상으로 블록과 도로의 인식을 위한 새로운 접근 방법으로, 특정 문제 영역과 무관한 선들간의 관계들을 정의하고, 이를 이용하여 대상체를 인식하는 과정에서 탐색공간을 줄이며 대상체와 연관된 선을 찾는 방법과 대상체들간의 구조적 정보를 기반으로 지도를 인식하는 방법을 제안하였다. 수많은 선분들 중에서 한 대상체를 구성하는 선들을 효과적으로 구별하기 위하여 각 선분들간에 오른쪽 이웃 관계, 왼쪽 이웃 관계, 평행 관계, 겹침 관계, 교차 관계 등을 정의하였다. 직선들간의 관계를 정의하고 대상체 인식에 이용한 실험 결과, 선 추적 방법에 비하여 탐색공간을 효과적으로 줄이는 효과를 가져왔으며, 연관된 다른 기호가 올바르게 인식되지 못하는 경우에도 최종 인식 결과는 영향을 받지 않았다. 결과적으로 선들간의 관계 및 대상체의 구조적 정보를 기반으로 하는 인식 방법이 지도 인식 문제에 유용하였으며, 제안된 지도 인식 방법은 여타 방법에 비해 보다 다양한 응용 분야에서 활용이 가능하다.

참 고 문 헌

[1] Rangachar Kasturi, Rajesh Raman, Chakravarthy Chennubhotla, and Lawrence O’Gorman, “An Overview of Techniques for Graphics Recognition”, Structured Document Image Analysis, Springer-Verlag Berlin Heidelberg, pp. 285-324., 1992.

[2] Dominique Antoine, Suzanne Collin, and Karl Tombre, “Analysis of Technical Documents: The REDRAW System,” Structured Document Image Analysis, Springer-Verlag Berlin Heidelberg, pp. 385-402, 1992.

[3] Dominique Antoine, “CIPLAN: A Model-Based System with Original Features for Understanding French Plats,” Proc. of International Conference on Document Analysis and Recognition, pp. 647-655, 1991.

[4] Satoshi Suzuki and Toyomichi Yamada, “MARIS : Map Recognition Input System”, Pattern Recognition, vol. 23, no. 8, pp. 919-933, 1990.

- [5] Luca Boatto, Vincenzo Consorti, Monica Del Buono, Silvano Di Zenzo, Vincenzo Eramo, Alessandra Esposito, Francesco Melcarne, Marco Meucci, Andrea Morelli, Marco Mosciatti, Stefano Scarci, and Marco Tucci, "An Interpretation System for Land Register Maps," *COMPUTER*, pp. 25-32, July, 1992.
- [6] M. T. Musavi, M. V. Shirvaikar, E. Ramanathan and A. R. Nekovei, "A Vision Based Method to Automate Map Processing," *Pattern Recognition*, vol. 21, no. 4, pp. 319-326, 1988.
- [7] Nirwan Ansari and Edward J. Delp, "On Detecting Dominant Points", *Pattern Recognition*, vol. 24, no. 5, pp. 441-451, 1991.
- [8] Takashi Matsuyama, Hidekazu Arita, and Makoto Nagao, "Structural Matching of Line Drawings Using the Geometric Relationship between Line Segments," *Computer Vision, Graphics, and Image Processing*, vol. 27, pp. 177-194, 1984.
- [9] S. Sblameyko, V. Bereishik, N. Paramonova, "Vectorization and Representation of Large-Size 2-D Line-Drawing Images," *Journal of Visual Communication and Image Representation*, vol. 5, no. 3, pp. 245-254, September 1994.

 저 자 소 개



張 京 植(正會員)

1989년 연세대학교 전자공학과 학사.
 1991년 연세대학교 전자공학과 석사.
 1996년 연세대학교 전자공학과 박사.
 1994년 ~ 현재 대우전자 전략기술
 제 1연구소 선임연구원. 주관심분야
 는 컴퓨터 비전, 인공지능, 전문가 시
 스템

金 在 喜(正會員) 第 34卷 C編 第 5號 參照

현재 연세대학교 전자공학부 전자공
 학전공 교수