

## 도로 자동인식을 위한 연산자 및 알고리즘 개발 Developing Operator and Algorithm for Road Automated Recognition

임인섭\* · 최석근\*\* · 이재기\*\*\*

Lim, In Seop · Choi, Seok Keun · Lee, Jae Kee

### 要 旨

최근 들어, 수치항공영상을 이용하여 지형정보를 추출하고자 하는 많은 연구가 수행되어 왔다. 그러나 수치항공영상에서 기존의 경계선 검출기법을 이용하여 대상물을 자동으로 인식하고 추출하는 것은 매우 어려우므로, 수동이나 반자동의 형태로 이루어졌다. 따라서, 본 연구에서는 먼저 도로 영역을 자동으로 추출하기 위해 밝기값이 분할된 항공영상의 의미론적인 정보의 대역을 중첩한 영상을 이용하여 인식에 장애가 되는 요소를 제거한 다음, 도로정보를 자동으로 인식하고 추출할 수 있는 알고리즘을 개발하여 시스템개발시 적용하고자 한다. 이를 위해 먼저, 횡단보도 대역 영상으로부터 횡단보도영역을 자동으로 인식하기 위한 "템플릿 회전 이동 연산자"와 인식된 횡단보도의 장변의 길이를 바탕으로 도로영역을 추적할 수 있는 "윈도우 법선 탐색 추적 알고리즘"을 개발하므로써 항공영상으로부터 직접 도로정보를 자동으로 추출할 수 있는 기법을 제시하고자 한다.

### Abstract

Recently, many studies extracting the geography information using digital aerial image have been implemented. But it is very difficult that automatically recognizing objects using edge detection method on the aerial image, and so that work have practiced manually or semi-automatically. Therefore, in this study, we have removed impedimental elements for recognition using the image which overlapped the significant information bands of brightness-sliced aerial images, then have developed the algorithm which can automatically recognize and extract road information and we will try to apply that method when we develop a system. For this, first of all, we have developed the "template conformal-transformation moving operator" for automatically recognizing crosswalk area from crosswalk band image and the "window normal search algorithm" which is able to track road area based on long-side length of crosswalk, so that we have proposed the method that can extract directly the road information from the aerial image.

\* 정회원 · 충북대학교 토목공학과 · 공학박사 · 043-273-0485 (E-mail : iso3295@trut.chungbuk.ac.kr)

\*\* 정회원 · 충북대학교 토목공학과 · 조교수 · 043-261-3352 (E-mail : skchoi@chungbuk.ac.kr)

\*\*\* 정회원 · 충북대학교 토목공학과 · 교수 · 043-261-2403 (E-mail : leejk@chungbuk.ac.kr)

## 1. 서론

현대에 들어와서는 디지털 기술의 발달로 인하여 자동화의 기술이 발달되었으며, 지도제작 및 GIS의 정보 구축에 대해서도 디지털 시스템을 활용한 컴퓨터 비전 및 인공지능과 관련한 기술들을 접목하는 등 새로운 접근방식을 도입하고 있다.<sup>1)</sup>

이에 수차항공영상을 이용하여 영상처리기법 및 기타 다른 방법에 의해 대상물을 추출하고자 하지만, 수차항공영상이 실제 복잡한 지형·지물을 촬영한 것으로 빛의 밝기와 그림자의 영향 등에 따라 촬영된 영상의 명암도가 변하기 때문에 영상의 밝기값이 매우 복잡하게 이루어져 있다. 이런 특성으로 인해 항공영상에서의 대상물 인식은 경계선 검출기법을 이용하여 윤곽선을 찾은 후 수동으로 이루어지거나, 대상물의 몇몇 특징점들을 수동으로 선택하여 그 데이터를 바탕으로 다음 단계에서 자동으로 하는 반자동 형태로 이루어지고 있다.<sup>2)</sup>

수차항공영상에서 대상물을 자동으로 인식하고 추출하기 위해서는 기하학적 형상이나 특징 또한 자동으로 묘사할 수 있어야 한다. 어떤 영상처리 기법을 이용하더라도 자동으로 형상을 명확하게 나타낼 수 있도록 한다면 특징-기반의 지식들을 통해 관련 대상물을 보다 쉽게 추출할 수 있을 것이다.<sup>3,4)</sup>

주어진 영상으로부터 초기 조건을 자동으로 생성하여 수치지도를 제작하거나 지형정보를 추출할 때,



그림1. 수차항공영상(8bit gray-scale)

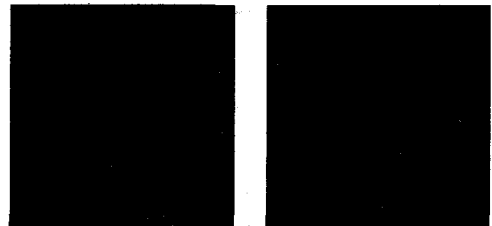
수작업으로 인해 발생하는 시간과 경제적인 손실을 줄이고, 전문가가 아니더라도 쉽게 활용할 수 있는 자동화 처리기법이 요구된다.

따라서, 본 연구에서는 먼저 도로 영역을 자동으로 추출하기 위한 처리 단계를 정립하고 그 가운데 횡단보도를 자동으로 인식하기 위한 “템플릿 회전 이동 연산자”와 인식된 횡단보도의 장변의 길이를 바탕으로 도로영역을 추적할 수 있는 “윈도우 법선 탐색 추적 알고리즘”을 개발함으로써 항공영상으로부터 직접 도로정보를 자동으로 추출할 수 있는 기법을 제시하고자 한다.

## 2. 자동화 조건 및 처리과정

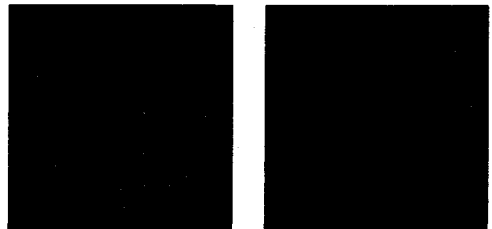
그림 1은 8bit 회색등급의 수차항공영상으로써, wild rc30 항공카메라에 의해 촬영된 사진을 Vexel vx 3000이라는 항공사진 스캐닝용 정밀 스캐너로 1600dpi(15.9 $\mu$ m)로 스캐닝한 것이다.

이것으로부터 대상물을 추출하기 위해 일반적으로 많이 활용하고 있는 경계선 검출 기법을 이용하여 영상을 처리한 결과 그림 2이다. 그림 2(a)는 라플라



(a) Laplacian

(b) Sobel



(c) Prewitt

(d) Canny

그림2. 경계선 검출기법에 의해 처리된 항공영상

표1. 도로 자동인식 처리단계 및 기법

| Step of process                         | Processing method   |
|---|---|
| Brightness slicing                      | • K-means clustering  |
| Composition of bands relative to object | • Overlapping and binary processing   |
| Recognizing crosswalk area              | • <u>template window</u><br>• <u>conformal transformation</u><br>• <u>moving operator</u><br>• differencing<br>• erosion/dilation |
| Measuring Crosswalk area feature        | • spatial moment  |
| Road area recognition and extraction    | • <u>window normal search</u><br>• <u>tracking</u><br>• erosion/dilation<br>• skeletonization                                     |

시안(laplacian) 연산자를 이용하여 경계선을 추출한 영상이고, (b)는 소벨(sobel), (c)는 프리위트(prewitt) 그리고 (d)는 캐니(canny) 연산자를 이용하여 항공영상 전체에 대해 경계선 추출을 실시한 것이다.<sup>5)</sup>

그림 2에서 보는 바와 같이 항공영상 전체에 대해 실시할 경우 매우 복잡한 형태를 나타내고 있어 초기 후보영역이나 인식조건을 찾을 수 없었다. 또한 이를 강화하거나 원 영상에 중첩하여 경계선을 보다 명확하게 표현할지라도 시각적으로 영상에서 대상물의 윤곽을 뚜렷하게 하는 효과는 얻을 수 있지만 컴퓨터에서 자동으로 인식할 수 있는 조건이 발생되지는 않았다. 이것은 경계선 검출기법이 항공영상과 같은 복잡한 영상으로부터 대상물 인식 및 추출을 자동화할 경우 부적절하다는 것을 의미한다.

따라서, 본 연구에서는 자동으로 인식할 수 있는 초기조건을 결정하기 위해 경계선 검출과 같은 형상 추출이 아닌 대상물별 밝기값 분포특성을 이용하여 대상물을 인식하도록 하였다.<sup>6,7)</sup>

수치항공영상으로부터 도로를 자동으로 인식하기 위한 전체적인 처리단계는 밝기값 분할, 대상물 관련 대역영상 조합, 횡단보도 영역 인식, 횡단보도 영역의 형상 측정, 도로 인식 및 추출 등 5단계로 이루어지며, 각 단계별 처리기법과 함께 표 1에 나타내었다.

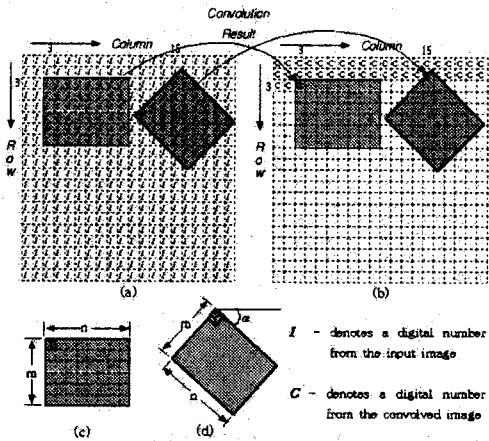
이중에서 3번째 단계 및 5번째 단계에서 기존의 영상처리기법이나 대상물 인식기법으로는 앞 단계에서 생성되는 조건이나 데이터를 처리하고 그 다음 단계에서 활용할 수 있도록 하는 데이터를 생성할 수 없었기 때문에, 3번째 단계에서 횡단보도를 자동으로 인식하기 위해 템플릿 회전 이동 연산자를 개발하였고, 5번째 단계에서는 횡단보도의 장변의 길이를 도로 폭으로 하여 도로 영역을 추적할 수 있는 윈도우 법선 탐색 추적 기법을 개발하여, 가능하면 모든 처리과정이 자동으로 연계될 수 있도록 하였다.

### 3. 템플릿 회전 이동 연산자

항공영상에서 대상물을 추출하기 위해 여러 가지 경계선 검출 기법을 이용하여 실시한 결과 자동으로 대상물을 인식할 수 있는 기본 조건을 찾을 수는 없었다. 또한, 특정 대상물을 인식하고 추출하기 위해 템플릿 정합 기법을 활용할 경우 항공영상에서 특정 대상물들의 크기 및 분포된 방향이 일정하지 않아 동일 대상물이라 할지라도 방향 및 크기에 따른 모든 템플릿을 만들어 놓아야 한다. 왜냐하면, 템플릿 정합이 대상물의 특징을 표현한 템플릿을 통해 영상에서 대상물과 템플릿과의 상관성을 통해 이루어지는 것이므로, 동일 대상물이라 할지라도 영상속에서 대상물의 형상에 변화가 있거나 대상물의 방향이 원 템플릿과 다른 경우에는 인식하지 못하기 때문이다.

따라서, 인식하고자 하는 대상물의 크기 및 분포된 방향이 다른 경우에도 대상물의 초기 후보 영역을 추출할 수 있도록 하기 위해 일반화할 수 있는 대상물을 선정해야 하며, 일반화된 템플릿을 적용하기 위해 본 연구에서는 '템플릿 회전 이동 연산자'라는 개념을 도입하였다. 템플릿 회전 이동 연산자는 특정대상을 인식하도록 하기 위해 대상물의 형상을 일반화된 템플릿으로 만들고 이것을 일정한 각도로 회전하면서 정합하도록 하는 것으로써 템플릿 정합과 이동 윈도우 연산자의 개념을 확장한 것이라고 할 수 있다.<sup>8)</sup>

그림 3은 템플릿 회전 이동 연산자에 대한 개념을



**그림 3. 템플릿 회전이동 연산자**  
**(a) 입력데이터 (b) 결과영상 (c) 템플릿 원도우 (d)  $\alpha$  만큼 회전된 템플릿 원도우**

나타내고 있다. 그림 3의 (a), (b)는 각각 입력영상과 연산후의 결과영상을 나타낸 것이며, (c)는 템플릿으로 만들어진 탐색윈도우이며, (d)는 템플릿 탐색 윈도우를  $\alpha$  각도 만큼 회전 변환한 템플릿형상을 나타낸 것이다.

템플릿 회전 이동 연산이 항공영상에서 특정 대상물의 형상을 일반화하여 영상속에서 부분적인 형상의 패턴을 인식하여 어떤 특정 대상물을 인식하기 위해 도입한 것이므로, 그림 3(c)와 같이  $m \times n$  크기의 일반화된 템플릿 탐색을 위한 커널(kernel) 집합으로 결정하였다.

이것은 대상물의 일부분을 패턴화하여 일반화한 템플릿의 크기가 이동 윈도우 연산자에서 컨벌루션을 하기 위한 커널값들의 집합과 같이  $3 \times 3, 5 \times 5, 7 \times 7$ 의 크기보다는 훨씬 더 큰 영역의 집합을 갖게 된다. 또한, 이동 윈도우 연산자에서 연산을 하기 위한 커널값들의 집합이 홀수개의 행·열로 이루어져 연산후의 결과값은 커널 집합의 중앙에 기록되도록 하였다. 하지만, 템플릿을 결정할 때 영상에서 대상물의 일부분을 샘플링하여 실시함으로, 탐색 윈도우의 커널 집합의 행·열의 값이 반드시 홀수가 되지 않을 뿐만 아니라, 템플릿을 일정한 각도로 회전하여 영상내의 대상물을 탐색하도록 하였기 때문에 탐색 윈도우의 중심을 명확하게 결정할 수 없다는 문제가

발생한다. 이를 해결하기 위해 템플릿을 전 영상에 대해 연산을 실시할 때, Fig. 2.18(c)와 (d)에서 템플릿 탐색 윈도우의 모서리 부분에  $\otimes$  표시한 곳에 기록하도록 한다. 따라서, 연산된 영역의 결과를 그림 3(b)와 같이 나타낼 수 있다.

본 연구에서는 템플릿을 패턴화된 영상으로 만들고 영상속에서 그것과 동일한 패턴을 찾기 때문에 윈도우 연산자와 같이 커널의 각 요소와 대응하는 영상의 화소값을 곱하여 그 합을 구하는 방식으로 결정하지 않고, 직접적으로 템플릿과 템플릿내에 포함된 화소값들간의 동일성 여부를 결정하도록 하였기 때문에 다음과 같이 계산하도록 한다.

영상에서 화소의 순서대로 비교해나갈 때 템플릿 윈도우의 첫 번째 화소 모서리의 위치가  $i$  번째 행과  $j$  번째 열에 위치할 때의 템플릿 탐색 윈도우 영역( $m \times n$ )내에 포함된 입력영상의 화소집합을 식(1)과 같이 정의한다. 또한 영상내에서 템플릿 윈도우 영역(커널 집합)의 화소 집합(T)을 식(2)와 같이 정의할 수 있다. 템플릿 윈도우 영역 내에 포함된 화소값이 특정 대상물의 템플릿과의 동일성을 판단하기 위해 식(3)과 같이 그 차이값을 계산하도록 한다.

탐색윈도우 내에 있는 화소집합과 템플릿 영역 내에 있는 화소집합간의 값이 일치한다면 오차가 0이 될 것이고 일치하는 부분이 없는 경우에는 오차가 255가 될 것이다.

화소값이 0 이나 255로 기록되고 그렇지 않더라도

$$I_{ij} = \begin{bmatrix} I_{11} & I_{12} & I_{13} & I_{14} & \dots & I_{1n} \\ I_{21} & I_{22} & I_{23} & I_{24} & \dots & I_{2n} \\ I_{31} & I_{32} & I_{33} & I_{34} & \dots & I_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ I_{m1} & I_{m2} & I_{m3} & I_{m4} & \dots & I_{mn} \end{bmatrix} \quad (1)$$

$$T_{ij} = \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} & \dots & T_{1n} \\ T_{21} & T_{22} & T_{23} & T_{24} & \dots & T_{2n} \\ T_{31} & T_{32} & T_{33} & T_{34} & \dots & T_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ T_{m1} & T_{m2} & T_{m3} & T_{m4} & \dots & T_{mn} \end{bmatrix} \quad (2)$$

$$C_{ij} = \frac{1}{m \times n} \{ T_{ij} - I_{ij} \} \quad (3)$$

8 bit 그레이 영상내에서의 모든 화소의 밝기값이 (0~255)사이에 있으므로, 패턴화된 템플릿의 크기가  $m \times n$  개의 영역에 대해 계산한 값이  $C_{ij}$ 에 저장할 전체 오차량은 (0~255)  $\times$  ( $m \times n$ )이 되어  $m \times n$ 의 크기에 따라서 영상으로 표현할 경우 8 bit 회색 영상으로 표현하지 못할 경우가 발생하게 된다. 따라서,  $C_{ij}$ 에 저장되는 오차량은 샘플링된 영상의 크기에 영향을 받지 않도록 하기 위해서 오차량을  $m \times n$ 으로 나눈 상태 즉 0~255의 값을 갖도록 하여  $C_{ij}$ 에 저장한 값을 그레이 영상으로 표현할 수 있도록 한 것이다.

템플릿 회전 이동 연산자는 도로를 추출하기 위한 기본 조건이 되는 횡단보도의 후보 영역을 자동으로 인식할 수 있도록 하기 위함이다.

그림 4는 횡단보도 대역중첩영상으로 항공영상의 밝기값을 20개의 band로 나누었을 때 15~17band를 중첩한 영상이며, 이 영상은 원영상에 비해 불필요한 정보들을 상당히 제거한 것이라 볼수 있다. 하지만 여전히 횡단보도와 동일한 대역의 밝기값을 가진 다른 정보들도 함께 표현되어 있어 횡단보도가 인식된 상태가 아니다. 따라서 이 영상위에서 템플릿 회전 이동 연산자를 통해 보다 명확하게 인식하도록 하면 그림 5와 같다.

그림 5는 횡단보도의 형상의 일부분에 해당하는 횡단보도의 템플릿을 만들어 제작된 템플릿을 윈도우로 하여 15°각도로 변환하면서 탐색하여 횡단보도를 인식하도록 한 것이다.

이 때, 매칭 결과는 그 오차량(0~255)이 80미만인 정보를 다시 8 bit로 계산하여 표현한 영상이므로 원영상보다는 다소 정보량이 적게 표현되어 있다. 하지만 이것은 초기 후보 영역에서 인식된 2차 후보영역을 결정한 결과이다.

템플릿을 계산된 영상위에 부착하면 그림 6과 같고 이 영상과 원영상과의 차영상을 구하면 잡음이 제거되고 팽창/침식 기법을 통해 횡단보도를 명확하게 인식하도록 한다.

#### 4. 윈도우 법선 탐색 추적 알고리즘

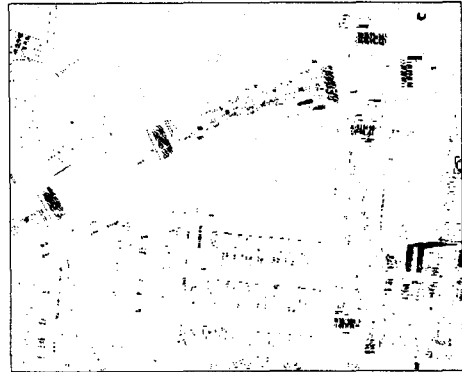


그림 4. 횡단보도대역영상 (15~17band 중첩영상)

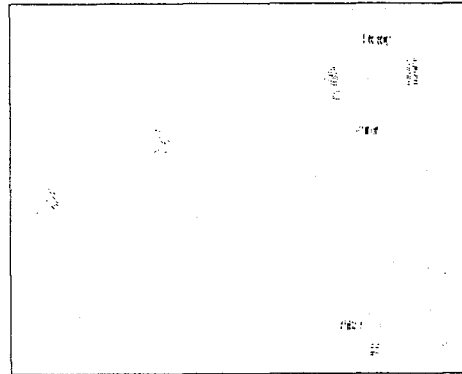


그림 5. 횡단보도 템플릿 매칭

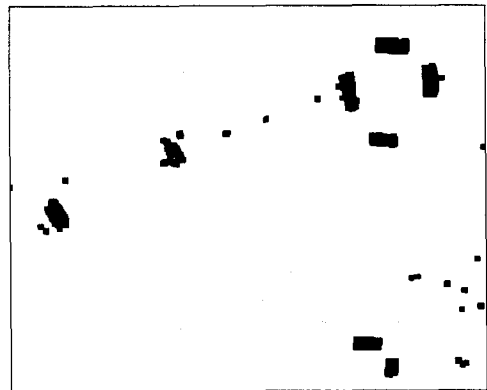


그림 6. 횡단보도 템플릿 부착

윈도우 법선 탐색 추적(Window Normal Search Tracking ; WNST) 알고리즘은 도로면을 인식하도

록 하기 위해 횡단보도 장변의 양단 좌표에 의한 초기 일차 선형방정식을 이용하여 그 선형의 법선방향으로 진행하면서 도로 차선을 추적하고 도로면을 인식할 수 있도록 하는 것이다. 본 연구에서 개발한 도로 자동추출 알고리즘은 다음과 같다.

4.1 탐색윈도우의 초기조건 결정

도로를 자동으로 인식하기 위한 초기 조건을 찾기 위해서, 형상 측정자에 의해 결정된 횡단보도의 장변의 길이를 이용하였으며, 이때 선형방정식은 장변의 두 끝점의 좌표를 이용하여 결정한다. 영상에서 인식된 횡단보도의 개수가 K개라고 한다면 입력되는 좌표는 식(4)와 같다.

여기서,  $[(X_1, Y_1), (X_2, Y_2)]_1^K$ 는 K번째 횡단보도의 1번째 장변의 두 좌표이고,  $[(X_1, Y_1), (X_2, Y_2)]_2^K$ 는 K번째 횡단보도의 2번째 장변의 두 좌표이다.

그림 7에서 보는 바와 같이 횡단보도의 장변의 직선 및 법선방정식은 주어진 각 횡단보도 장변의 좌표를 이용하여 장변에 대한 1차 선형방정식을 구하고 식 5와 같이 나타낼 수 있으며 이에 대한 법선의 방정식은 식(6)과 같이 나타낼 수 있다.

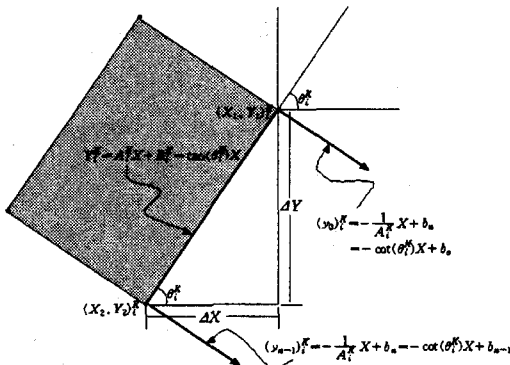


그림 7. 횡단보도 장변의 직선 및 법선방정식

$$\left\{ \begin{array}{ll} [(X_1, Y_1), (X_2, Y_2)]_1^1 & , [(X_1, Y_1), (X_2, Y_2)]_2^1 \\ [(X_1, Y_1), (X_2, Y_2)]_1^2 & , [(X_1, Y_1), (X_2, Y_2)]_2^2 \\ [(X_1, Y_1), (X_2, Y_2)]_1^3 & , [(X_1, Y_1), (X_2, Y_2)]_2^3 \\ \vdots & \vdots \\ [(X_1, Y_1), (X_2, Y_2)]_1^K & , [(X_1, Y_1), (X_2, Y_2)]_2^K \end{array} \right\} \quad (4)$$

$$\left\{ \begin{array}{l} Y_i^k \mid Y_i^k = A_i^k X + B_i^k = \tan(\theta_i^k) X + B_i^k \\ (X_i)_i^k \leq X \leq (X_2)_i^k \text{ and } (Y_1)_i^k \leq Y \leq (Y_2)_i^k, \\ i=1 \text{ and } 2, \quad K: \text{natural number} \end{array} \right\} \quad (5)$$

$$\left\{ \begin{array}{l} (y_N)_i^k \mid (y_N)_i^k = -\frac{1}{A_i^k} X + b_N = -\cot(\theta_i^k) X + b_N, \\ N=0, 1, 2, \dots, n-1 \end{array} \right\} \quad (6)$$

4.2 법선윈도우의 탐색영역 결정

본 알고리즘은 법선방향의 직진성에 대한 탐색을 기본 조건으로 한다. 그림 8에서 보는 바와 같이 장변의 좌표값에 의해 결정된 방정식의 길이가 n 이고, 법선상에서 탐색될 화소의 개수가 m개로 정의되면, 탐색 윈도우의 크기는  $n \times m$ 이 된다. 따라서, 각 법선윈도우에서 탐색된 화소들은  $P_0$ 에서부터  $P_{n-1}$ 까지 정의된 n 개의 법선영역( $1 \times m$ )에 대한 화소집합을 갖게 된다. 각 법선윈도우탐색 화소의 집합들을 모두 합쳐서 윈도우 전체에 대한 검색 화소 집합은 식(7)과 같이 나타낼 수 있다.

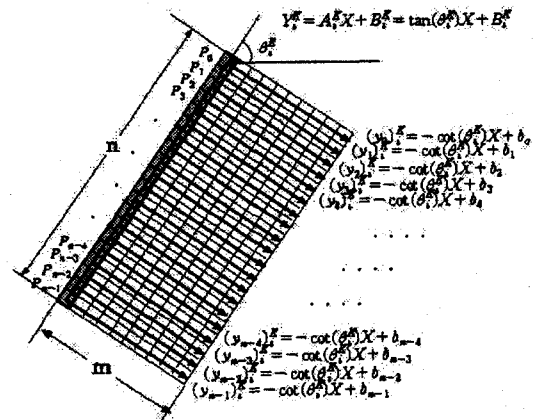


그림 8 생성된 법선방향 탐색영역 ( $n \times m$ )

$$(P^0)_i^K = \left\{ \begin{array}{l} (P_0^0)_i^K = \{(x_1, y_1)_0, (x_2, y_2)_0, \dots, (x_m, x_m)_0\} \\ (P_1^0)_i^K = \{(x_1, y_1)_1, (x_2, y_2)_1, \dots, (x_m, x_m)_1\} \\ (P_2^0)_i^K = \{(x_1, y_1)_2, (x_2, y_2)_2, \dots, (x_m, x_m)_2\} \\ \vdots \\ (P_{n-1}^0)_i^K = \{(x_1, y_1)_{n-1}, (x_2, y_2)_{n-1}, \dots, (x_m, x_m)_{n-1}\} \end{array} \right\} \quad (7)$$

### 4.3 도로 진행방향 결정 인자 및 지수

이렇게  $n \times m$  크기로 정의된 윈도우 영역에 대해 정보량을 검색하면 그림 9와 같이 법선방향과 일치하는 형태(a)와 법선방향과 일치하지 않는 형태(b)가 모두 계산된 정보량이 33개로 동일하다는 것을 알 수 있다. 이처럼 탐색윈도우 전체를 그대로 적용했을 경우, 컴퓨터가 인식하기에는 동일한 것으로 판단하여 도로의 진행방향과 같은 방향성을 전혀 고려할 수 없게 된다.

|    |   |   |   |   |   |   |   |   |    |
|----|---|---|---|---|---|---|---|---|----|
| 1  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1  |   |   |   |   |   |   |   |   |    |
| 2  |   |   |   |   |   |   | 1 | 1 | 1  |
| 3  |   |   |   |   |   |   |   |   |    |
| 4  |   |   |   |   |   |   |   |   |    |
| 5  |   |   |   |   |   |   |   |   |    |
| 6  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  |
| 7  |   |   |   |   |   |   |   |   |    |
| 8  |   |   |   |   |   |   |   |   |    |
| 9  |   |   |   |   |   |   |   |   |    |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  |
| 11 |   |   |   |   |   |   |   |   |    |
| 12 |   |   |   |   |   |   |   |   |    |
| 13 |   |   |   |   |   |   |   |   |    |
| 14 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  |
| 15 |   |   |   |   |   |   |   |   |    |

|    |   |   |   |   |   |   |   |   |    |
|----|---|---|---|---|---|---|---|---|----|
| 1  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1  |   |   |   |   |   |   |   |   |    |
| 2  |   |   |   |   |   |   |   |   |    |
| 3  |   |   |   |   |   |   |   |   |    |
| 4  |   |   |   |   |   |   |   |   |    |
| 5  |   |   |   |   |   |   |   |   |    |
| 6  | 1 |   |   |   |   |   |   |   |    |
| 7  |   |   |   |   |   |   |   |   |    |
| 8  |   |   |   |   |   |   |   |   |    |
| 9  |   |   |   |   |   |   |   |   |    |
| 10 | 1 |   |   |   |   |   |   |   |    |
| 11 |   |   |   |   |   |   |   |   |    |
| 12 |   |   |   |   |   |   |   |   |    |
| 13 | 1 |   |   |   |   |   |   |   |    |
| 14 |   |   |   |   |   |   |   |   |    |
| 15 |   |   |   |   |   |   |   |   |    |

(a) 법선방향과 일치 (b) 법선방향과 기울어짐

그림 9 탐색윈도우에 검색된 정보의 형태

|       |   |   |   |   |     |       |       |     |   |
|-------|---|---|---|---|-----|-------|-------|-----|---|
|       | 1 | 2 | 3 | 4 | ... | $m-2$ | $m-1$ | $m$ |   |
| 1     |   |   |   |   |     |       |       |     |   |
| 2     |   |   |   |   |     |       | 1     | 1   | 1 |
| 3     |   |   |   |   |     |       |       |     |   |
| 4     |   |   |   |   |     |       |       |     |   |
| 5     |   |   |   |   |     |       |       |     |   |
| ...   |   |   |   |   |     |       |       |     |   |
| $n-4$ | 1 | 1 | 1 | 1 | 1   | 1     | 1     | 1   | 1 |
| $n-3$ |   |   |   |   |     |       |       |     |   |
| $n-2$ |   |   |   |   |     |       |       |     |   |
| $n-1$ | 1 | 1 | 1 | 1 | 1   |       |       |     |   |
| 0     |   |   |   |   |     |       |       |     |   |

$f_0 = 0$

$\frac{l_1}{m}$   $f_1 = 1$

$f_2 = 0$

$f_3 = 0$

$\frac{l_4}{m}$   $f_4 = 1$

...

$\frac{l_{n-4}}{m}$   $f_{n-4} = 1$

$f_{n-3} = 0$

$f_{n-2} = 0$

$\frac{l_{n-1}}{m}$   $f_{n-1} = 1$

$f_{n-2} = 1$

$f_{n-1} = 0$

그림 10 법선윈도우의 정보검색을 및 검색빈도 지수

따라서, 이런 방향성을 고려하기 위해 각 법선윈도우의 정보검색율( $\overline{ISR_N}$ )을 식(8)과 같이 정의 하고, 그림 10의 우측에 나타난 것과 같이 법선윈도우내에

정보량이 하나라도 검색되면 정보발생 지수( $f_N$ )를 1이라 하고 발생하지 않는 경우 0으로 규정하였으며 식 (9)와 같다.

$$\overline{ISR_N} = \frac{l_N}{m}, \quad N=0,1,\dots,n-1 \quad (8)$$

$$f_N = \begin{cases} 0, & \text{if } l_N=0 \\ 1, & \text{elsewhere} \end{cases} \quad (9)$$

여기서  $m$ 은 법선윈도우의 크기를 나타내고  $l_N$ 은  $N$  번째 법선윈도우에서 탐색된 정보의 개수이다. 그림 14에서 5번째 법선윈도우를 예로 든다면 법선윈도우에서 탐색된 정보의 개수는 5개가 된다.

또한 탐색윈도우의 진행방향 결정 인자(RCDF)를 식(10)과 같이 정의하여 임의의 한 방향에 대한 전체 윈도우의 방향성 인자를 결정하도록 하였다.

$$RCDF = \frac{ISR_N}{F} = \frac{\sum_{N=0}^n \frac{l_N}{m}}{\sum_{N=0}^n f_N} \times 100 \quad (10)$$

식(10)에서의  $ISR_N$ 은 식 4.5에서 정의된 각 법선윈도우의 정보검색률( $\overline{ISR_N}$ )을 모두 합한 것이며,  $F$ 는 식(9)와 같이 정의된 각 법선윈도우의 정보발생지수를 합한 값이다. 예를 들어, 그림 9의 (a)(b)의 경우, 정보검색율은 동일하지만 각 법선윈도우에 대한 정보발생 지수는 그림 9(a)와 같이 도로 차선정보가 법선방향과 일치하는 경우는  $F$ 가 4되며, 그림 9(b)와 같이 법선과 일치하지 않고 기울어진 경우는 모든 법선윈도우에서 1개 이상의 정보량을 가지고 있으므로 정보발생지수의 합  $F$ 는 15가 된다. 이 두 가지 경우를 비교해 보면 그림 9(a)가 그림 9(b)보다 작은 값을 가지게 되어 RCDF의 값이 큰 값을 갖게 되므로 그림 9(a)가 그림 9(b)보다 차선에 대해 직진성이 더 강한 경우라고 판단할 수 있다.

### 4.4 탐색윈도우의 추적방향 판별

실질적으로 도로의 진행방향을 추적하기 위해 그림 15와 같이 법선 윈도우의 방향을 초기의 법선방향에서  $\pm \alpha$ 만큼  $1^\circ$  간격으로 변화하여 각각의 변화

각에 대해 식(11)과 같이  $2\alpha+1$ 개의 진행방향 결정인자(RCDF)를 생성하도록 하였다.

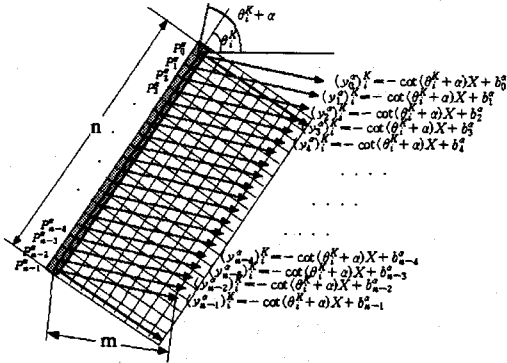


그림 11 도로차선의 진행방향을 검색하기 위해  $\pm \alpha$ 만큼 변화시키면서 탐색하는 각 법선원도우

$$\{ (RCDF^0)_i^k, (RCDF^1)_i^k, (RCDF^{-1})_i^k, \dots, (RCDF^{-\alpha})_i^k, (RCDF^\alpha)_i^k \} \quad (11)$$

이중에서 만약 임의의 각  $\alpha$ 에서 RCDF값이 가장 크다면  $\alpha$ 각에 대해 다시 초기 법선방정식을 식(12)와 같이 정의하여 새로운 윈도우를 생성하고 이들 중 가장 큰 값을 갖는 법선벡터집합의 윈도우를 결정하여 그림 12와 같이 labeling을 함으로써 도로면을 추적하여 인식하도록 하였다.

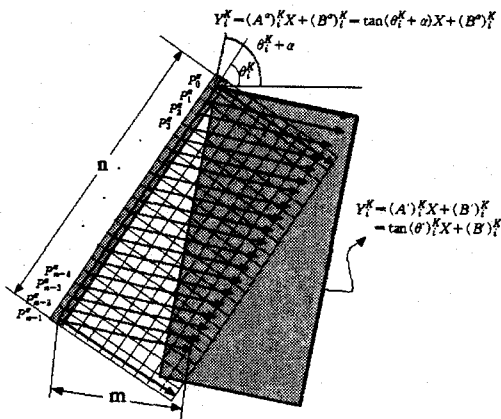
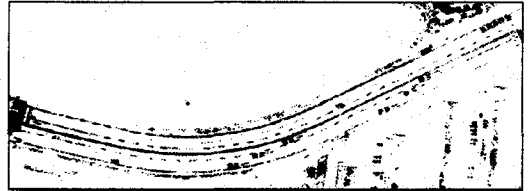


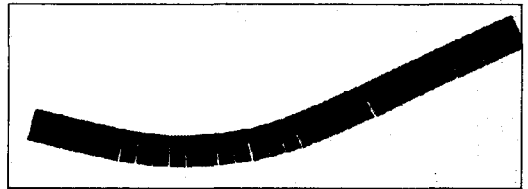
그림. 12 도로차선의 진행방향을 검색하기 위해  $\pm \alpha$ 만큼 변화시키면서 탐색하는 각 법선원도우

$$(y_0^k)_i^k = \cot(\theta_i^k + \alpha)X + b_i^k \quad (12)$$

새롭게 결정된 윈도우의 우측변에 대해 다시 일차 선형방정식을 결정하고 그에 대한 법선을 결정한 후 위의 모든 과정을 반복하면서 도로면을 자동으로 추적하도록 할 수 있다.



(a) 도로차선관련 대역 중첩 영상 (11~17band)



(b)윈도우 법선 추적방식에 의해 추적된 영상

그림 13 자동인식 처리 시스템에 의해 인식된 도로

#### 4.5 추적 종료 조건 결정

윈도우 법선탐색 기법에 의한 도로의 추적은 탐색영역내에 있는 차선정보를 바탕으로 이루어진다. 이때, 그림 14와 같이 횡단보도의 오른쪽 장변을 기준으로 검색하다가 검색되는 차선정보가 없는 경우, 검색을 중단하도록 하였다. 또한 그림의 왼쪽방향에서 화살표방향으로 검색하면 점선으로 표시된 탐색영역이 횡단보도 또는 이와 유사한 밝기값을 가지는 다른 정보와 접하게 되는데, 정도 이상으로 많이 탐색될 경우 중단하도록 하였다. 그 조건은 식(13)과 같다.

상기 조건은 법선 탐색 윈도우를 통해 도로영역을 검색하고 표시하면서 영상으로부터 도로를 분류하는 중에 탐색 윈도우내에 정보로 인식된 면적( $A_R$ )이 탐색윈도우 면적( $A_{SW}$ )에 대해 60%이상이거나 5%미



만일 경우에는 탐색을 멈추게 한다. 60% 이상일 경우에, 이것은 차선과 동일한 밝기값을 가진 장애물이 있거나 다른 횡단보도에 이르렀다고 판단할 수 있으며, 5% 이하일 경우에는 차선에 대한 정보가 매우 적어 방향성을 결정할 수 없거나 동일 밝기값의 잡영이라 판단할 수 있다.

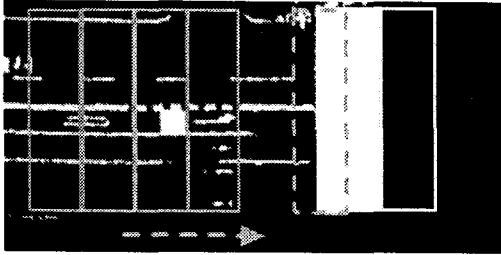


그림 14 도로 추적 멈춤

Tracking

$$= \begin{cases} \text{stop,} & \text{if } \frac{A_R}{A_{SW}} \geq 60\% \text{ or } \leq 5\% \\ \text{continue,} & \text{if } 5 < \frac{A_R}{A_{SW}} < 60 \end{cases}$$

(13)

#### 4.6 도로 자동 추적 알고리즘 설계

도로 추적 알고리즘의 흐름도는 그림 15와 같다. 특정대상물 인식에서도 설명한 바와 같이 횡단보도 영역을 자동 인식하고 그 영역을 기준으로 공간모멘트 기법에 의해 측정된 횡단보도 장변 길이를 입력값으로 사용하였다.

이처럼 윈도우 법선 탐색 알고리즘은 항공영상에서 횡단보도를 인식하고 그 장변의 길이를 초기 데이터로 활용하기 때문에, 횡단보도가 있는 도로에 대해서는 초기 입력 데이터에 의해 자동으로 처리되지만, 횡단보도가 없는 도로의 경우에는 수동으로 그 좌표를 입력함으로써 실행될 수 있도록 하였다. 다시 말해, 도로의 넓이 값을 결정해 주면 이를 바탕으로 도로를 추적할 수 있도록 한 것이다.

입력된 장변의 좌표를 이용하여 장변의 선형방정식을 결정하고 장축 위에 있는 N 개의 화소에 대해 장축의 선형방정식에 대한 N개의 법선방정식을 결정하였다. 그러나, 탐색 윈도우의 추적 방향을 결정하

기 위해서 N개의 법선방정식의 기울기 값을  $\pm \alpha^\circ$  만큼 변화시키면서  $2\alpha+1$ 개에 해당하는 각 법선 탐색 윈도우상에 있는 화소값들의 집합  $(P_N^0)^K, (P_N^{-1})^K, (P_N^1)^K, \dots, (P_N^{-\alpha})^K, (P_N^\alpha)^K$ 을 결정하고 이 값들에 대한 정보검색율(ISRN)과 발생빈도지수(F)를 통해 탐색윈도우의 도로방향 결정 인자들을 계산한다. 이렇게 계산된 값들 중에 가장 큰 값을 도로를 추적하기 위한 탐색윈도우의 추적 방향으로 결정하였다.

추적방향으로 결정되면 탐색 윈도우 내에 정보로 인식된 면적이 탐색윈도우 면적( $A_{SW}$ )에 대해 5~60% 사이에 있을 경우 그 영역을 표시하도록 하였다. 그런 다음, 결정된 윈도우에 대해 다시 선형방정식을 결정하고 앞에서 설명한 과정을 반복하도록 하였다.

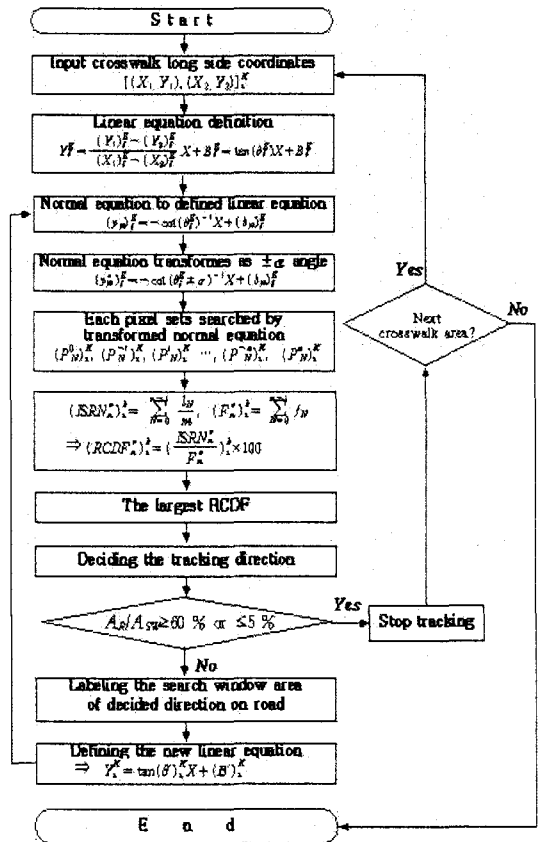


그림 15 윈도우 법선 탐색 추적 알고리즘

그러나 법선 탐색 윈도우를 통해 도로영역을 검색하고 표시하면서 영상으로부터 도로를 분류하는 도중에 탐색 윈도우 내에서 정보로 인식된 면적( $A_R$ )이 탐색윈도우 면적( $A_{SW}$ )에 대해 60%이상일 경우나 5%미만일 경우에는 탐색을 멈추게 된다. 추적을 종료하고, 다시 다른 횡단보도 영역이 있는지 확인한 후 처리해야 할 다른 영역이 있으면, 그 영역에 대해 다시 선형방정식을 결정하고 법선 방정식을 세우는 등 앞에서의 모든 과정을 반복하도록 하였다. 그리고 다른 횡단보도 영역이 더 이상 없을 경우 전체 영상에 대해 도로 추적을 종료하도록 하였다. 영상 전체에 대해 알고리즘을 적용하여, 도로 영역을 추출한 후 추출된 영역을 원 영상 위에 중첩하여 나타낸 것이 그림 16이다.



그림 16 항공영상에서 자동으로 인식된 도로 영역

## 5. 결론

본 연구는 수치지도제작 자동화를 위해 수치항공영상의 특성을 파악하여 수치항공영상으로부터 직접 대상물을 자동으로 인식하고 추출할 수 있는 자동화 처리기법 및 알고리즘을 개발하여 기존의 영상처리 및 인식 기법을 함께 적용한 결과 다음과 같은 결론을 얻을 수 있었다.

1. 수치항공영상에서 관련된 대상물의 밝기값 특징을 요약하고 대역중첩영상을 조건으로 특정대상물을 인식할 수 있는 템플릿회전 이동 연산자 및 윈도우 법선 탐색 추적 알고리즘을 개발하였다. 따라서, 처

리단계 중 일부 대상물 인식을 수동이나 반자동으로 처리하지 않고도 영상으로부터 지형정보를 얻을 때까지 모든 과정이 연계될 수 있도록 함으로써 일련의 자동화 처리가 가능하도록 하였고, 수치항공영상으로부터 지형정보추출 및 지도제작을 자동화 할 수 있다는 가능성을 제시할 수 있다.

2. 템플릿 회전 이동 연산자는 횡단보도 형상을 일반화하고 이것을 템플릿으로 제작하여 전체 영상에 대해 이를 일정한 각도로 변환하면서 정합하도록 하였으며, 템플릿 회전 이동 연산자를 실행하여 계산된 결과값에 대해 일정한 수준의 정합 오차(80 of 256, 약 30% 오차율)를 허용하고 이것을 8 bit 회색 등급 영상으로 표현하도록 함으로써 대상물을 인식하도록 하였기 때문에, 템플릿 정합의 1:1 정합성에 대한 문제점을 보완할 수 있었다.

3. 도로 차선관련 대역 중첩 영상에서 표현된 정보는 도로 영역을 자동으로 인식하기 어렵기 때문에, 윈도우 법선 탐색기법을 개발함으로써 원영상 중첩, 밝기값 강화, 판단, 수동추출 등의 처리과정을 거치지 않고도 도로의 방향성을 고려하여 도로 영역을 자동으로 추적할 수 있었다.

## 참고문헌

1. Arcot, S., and John, T., "Modelling and Representation Issues in Automated Feature Extraction from Aerial and Satellite Images", ISPRS Journal of Photogrammetry & Remote Sensing, Vol. 55, 2000, pp34~47.
2. 이재기, 박경식, 임인섭(2001) 수치정사투영영상으로부터 지형공간정보 추출기법, 대한토목학회논문집, 제 21권 제 6-D호, pp 915~924.
3. Hildreth, E. C., "Edge Detection", Artificial Intelligence Laboratory of the Massachusetts Institute of Technology, 1985, pp. 3~19.
4. Accame, M, and De Natale, F. B. B., "Edge Detection by Point Classification of Canny Filtered Images", Signal Processing, Vol. 60, 1997, pp. 11~22.

5. Rafael C. Gonzalez and Richard E. Woods (1993)  
Digital Image Processing, Addison- Wesley  
publishing com.
6. 이승환, “패턴인식의 원리 1권”, 홍릉과학출판사,  
1997, pp. 96~100.
7. Peter D., Peggy A., Anthony, S., and Mohamad  
M., (2001) Self-Organised clustering for road  
extraction in classified imagery” ISPRS Journal  
of Photogrammetry & Remote Sensing, Vol 55,  
pp347~358.
8. Wolf, P. R.(2000) Elements of Photogrammetry :  
with application in GIS, 3rd ed. McGraw Hill  
International Book Company, pp. 315~322.

---

(2002년 6월 10일 원고접수)