

## 실시간 영상 처리를 위한 향상된 영역 경계 검출

김승희\*, 남시병\*\*, 임해진\*\*

# An Improved Area Edge Detection for Real-time Image Processing

Kim Seung Hee \*, Nam Si Byung \*\*, Lim Hae Jin \*\*

### 요약

경계 검출은 영상 인식의 성능을 좌우하는 중요한 단계로서 지금까지 많은 경계 검출 방법들이 연구되어 왔음에도 불구하고 여전히 어려운 문제로 남아있으며, 영상에서 객체를 인식하거나 특정 영역을 추적하는 등의 유일한 방법은 아니지만 영상 인식 응용 분야에서 중요한 요소들 중의 하나이다. 본 논문에서는 미분 연산자를 이용한 경계 검출 방법과는 달리 이진화 영상에서 2개의 이웃된 픽셀 정보를 참조하여 미리 정의된 4개의 경계 픽셀 패턴과 비교하여 경계 픽셀을 찾고, 다음 경계 검출 탐색 픽셀에 대한 방향을 결정하여 영역의 경계를 검출하며, 다음 영역의 경계 검출을 위하여 방문하지 않은 픽셀들을 탐색하면서 경계 검출 단계를 반복 수행하여 이진화 영상의 경계를 검출하는 방법을 제안하였다. 영상 인식에 있어 미분 연산자를 이용하여 경계 검출을 할 경우 경계 검출 다음 단계인 세션화 과정을 생략할 수 있었으며, 기존의 영역 경계 추적 방법에 비해 경계 검출 알고리즘 수행 시간을 단축하여 실시간으로 영상을 인식하는 시스템에 적용하여 전체적인 영상 인식 시간을 단축할 수 있었다.

### Abstract

Though edge detection, an important stage that significantly affecting the performance of image recognition, has been given numerous researches on its execution methods, it still remains as difficult problem and it is one of the components for image recognition applications while it is not the only way to identify an object or track a specific area. This paper, unlike gradient operator using edge detection method, found out edge pixel by referring to 2 neighboring pixels information in binary image and comparing them with pre-defined 4 edge pixels pattern, and detected binary image edge by determining the direction of the next edge detection exploring pixel and proposed method to detect binary image edge by repeating step of edge detection to detect another area edge. When recognizing image, if edge is detected with the use of gradient operator, thinning process, the stage next to edge detection, can be omitted, and with the edge detection algorithm executing time reduced compared with existing area edge tracing method, the entire image recognizing time can be reduced by applying real-time image recognizing system.

▶ Keyword : Area, Boundary, Edge

• 제1저자 : 김승희  
• 투고일 : 2008. 10. 29, 심사일 : 2008. 11. 11, 게재확정일 : 2008. 12. 17.  
\* 관동대학교 강사 \*\* 강원대학교 삼척캠퍼스 전자정보통신공학부 교수

## I. 서론

기술의 발달로 영상 미디어가 다양화 되어감에 따라 영상을 기반으로 하는 기계 사용은 과학적인 분야의 큰 변화 속에서 폭 넓게 응용되고 있다. 이에 따라 수많은 픽셀 데이터로 영상 정보를 표현하고, 이러한 영상들을 처리하기 위해 로봇이나 컴퓨터에 인간과 같은 시각적으로 물체를 인식할 수 있는 능력이 필요하게 되었다. 경계 검출은 영상 인식의 성능을 좌우하는 아주 중요한 단계로서 입력 영상에 대한 대상 물체들의 위치, 모양과 크기, 텍스처 등 많은 정보가 서로 다른 영역간의 경계선에 위치할 뿐만 아니라 영상 속의 물체 형태에 관한 정보를 그대로 유지하면서 영상 인식의 후처리 단계에서 처리해야 할 자료의 양을 줄일 수 있고, 많은 물체 인식 알고리즘과 쉽게 결합이 가능하다는 장점을 가지고 있다[1, 2].

경계 검출이 신속하고 정확히 이루어진다면 가장 효율적인 방법이 될 수 있다. 경계를 검출하는 여러 가지 방법 중 하나로 미분 연산자를 이용한 다양한 경계 추출 방법이 있다. 가장 대표적인 기법이 소벨(Sobel)과 프리윗(Prewitt), 로버츠(Roberts), 라플라시안(Laplacian) 기법으로 국부적 미분연산자를 이용하여 불연속적인 두 영역간의 명암차를 구한다.[3, 4] 이러한 기법들은 응용 분야에 따라 영상 인식의 후처리 과정으로 세션화 단계를 수행하여야하며, 세션화 과정에서 경계선이 부정확해지는 경우가 있다[5, 6].

본 논문에서는 이진화 영상에서 2개의 이웃된 픽셀의 정보를 참조하여 미리 정의된 4개의 경계 픽셀 패턴과 비교하여 경계 픽셀을 찾고, 다음 경계 검출 탐색 픽셀에 대한 방향을 결정하여 경계를 검출하는 방법을 제안한다.

## II. 이론적 배경

### 2.1 소벨 마스크

경계 검출의 가장 대표적인 1차 미분 연산자 중의 하나인 소벨(Sobel) 마스크는 <그림 1>과 같다.

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

Gx 미분연산자 Gy 미분연산자

그림 1. 소벨 마스크  
Fig 1. Sobel mask

소벨 마스크는 x축, y축으로 각각 한번씩 미분하게 되어 있고, 기울기의 크기는  $G = \sqrt{Gx^2 + Gy^2}$ 이며, Gx는 x축 편미분이고 Gy는 y축 편미분이다. 연산 속도가 비교적 느리지만 모든 방향의 경계를 검출하고 돌출한 픽셀값을 평균화하므로 잡음에 대체로 강인한 편이다. 그리고 수직, 수평 방향의 경계보다 대각선 방향의 경계에 더 민감하게 반응한다.

### 2.2 프리윗 마스크

프리윗(Prewitt) 마스크는 소벨 마스크의 형태와 비슷하나 밝기의 경계에 대해 비중을 약간 다르게 주었으며 <그림 2>와 같다.

-1	0	1
-1	0	1
-1	0	1

1	1	1
0	0	0
-1	-1	-1

Gx 미분연산자 Gy 미분연산자

그림 2. 프리윗 마스크  
Fig 2. Prewitt mask

프리윗 마스크를 적용한 결과는 소벨 마스크를 적용한 결과와 비슷하며, 응답 시간이 다소 빠르지만 소벨 마스크에 비해 밝기 변화에 대하여 비중을 약간 적게 주므로 경계 검출 시 경계가 덜 부각되는 결과를 가져온다. 그리고 대각선 방향 경계보다 수직, 수평 방향 경계에 더 민감하게 반응한다[7, 8].

### 2.3 로버츠 마스크

로버츠(Roberts) 마스크는 경계 검출 마스크 중 기본이 되는 마스크이며 <그림 3>과 같다.

0	0	-1
0	1	0
0	0	0

-1	0	0
0	1	0
0	0	0

Gx 미분연산자 Gy 미분연산자

그림 3. 로버츠 마스크  
Fig 3. Roberts mask

로버츠 마스크는 소벨, 프리윗 마스크에 비해 매우 빠른 연산 속도를 나타내며, 주변과 관계없이 경계가 확실한 경계

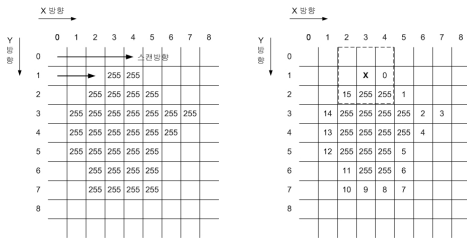
를 검출한다. 그러나 다른 마스크보다 크기가 작아 돌출한 픽셀값을 평균화 할 수 없는 관계로 잡음에 매우 민감하다.

### 2.4 Moore-Neighbor Tracing

영역의 경계 추적이란 이진화 영상 또는 라벨링 영상에서 일정한 밝기 값을 가지는 영역의 경계를 추적하여 경계 픽셀의 순서화된 정보를 얻어내는 것이다. 먼저 입력 영상을 이진화하고 이진화 영상에 대해 밝기값 255를 가지는 영역에 대해 경계를 추적한다. 경계를 추적하면 1 픽셀 두께를 가지는 픽셀의 순서화된 연속 체인(chain) 정보를 얻는 것이 가능하다[9].

Moore-neighbor tracing 단계는 다음과 같다.

단계 1 : <그림 4>와 같이 영상의 왼쪽 맨 위에 있는 픽셀부터 차례로 스캔하면서 픽셀값이 255인 픽셀을 만나면 경계 픽셀로 표시하고 경계 추적을 시작한다.



(a) 이진화 영상 (b) 추적된 경계 위치

그림 4. 경계 추적의 예  
Fig 4. Example of edge tracing

단계 2 : 단계 1에서 찾은 픽셀을 중심으로 <그림 5>와 같은 마스크를 사용하여 마스크의 4번 위치에서 탐색을 시작하여 주위를 돌면서 픽셀값이 255인 주변 픽셀을 찾는다.



그림 5. 중심 픽셀의 주변 픽셀 번호  
Fig 5. The number of pixel surrounding center pixel

단계 3 : 단계 2에서 찾은 픽셀이 다시 중심 픽셀이 되어 주변 픽셀들의 픽셀값을 탐색한다. 이때 탐색 시작 위치는 아래의 <식 1>에 의해 결정된다.

$$n' = (n + 5) \& 7 \dots\dots\dots (2.1)$$

단계 4 : 단계 3을 반복 수행하다가 한바퀴 돌아서 처음 시작 픽셀에 도달하면 경계 추적이 종료된다.

Moore-neighbor tracing 방법의 경우 이진화 영상 또는 라벨링 영상에서 영역의 경계를 추적하는 방법으로 미분 연산자를 이용한 경계 검출 방법에 비해 세션화 된 경계를 획득할 수 있다. 그러나 경계 추적 과정에서 중심 픽셀의 주변 픽셀들을 중복하여 탐색하므로 불필요한 경계 검출 지연 시간을 가질 수 있으므로 실시간으로 영상을 인식해야하는 시스템의 경우 적용하기 어려울 수 있다.

### III. 제안된 영역 경계 검출

제안한 영역 경계 검출 방법은 이진화 영상에서 2개의 이웃된 픽셀의 정보를 참조하여 미리 정의된 경계 픽셀의 패턴과 비교하여 경계 픽셀을 찾고, 다음 경계 검출 탐색 픽셀에 대한 방향을 결정하여 이진화 영상의 경계를 검출하는 방법이다.

2개의 이웃된 픽셀값에 의한 경계 픽셀의 패턴은 <그림 6>과 같이 4가지로 정의하였으며, 제안한 방법의 흐름도는 <그림 7>과 같다.

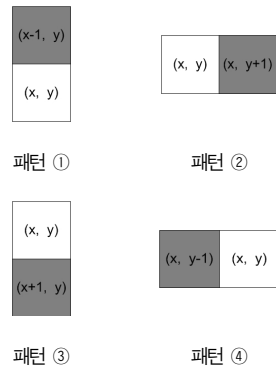


그림 6. 경계 픽셀의 패턴  
Fig 6. Edge pixel pattern

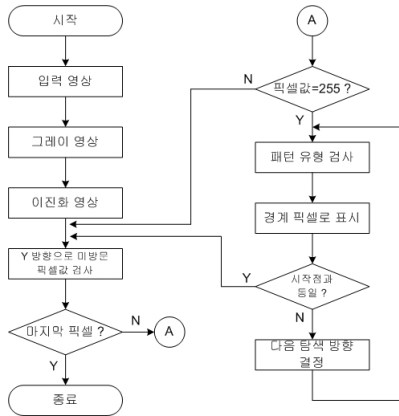


그림 7. 영역 경계 검출 방법의 흐름도  
Fig 7. Flowchart of area edge detection

이진화 영상의 왼쪽 맨 위의 픽셀부터 차례로 Y방향으로 탐색하다가 픽셀값이 255인 픽셀을 만나면 정의된 경계 픽셀의 패턴과 비교하여 일치하면 경계 픽셀로 표시하고, 경계 검출의 시작 픽셀로 정한다. 경계 픽셀과 이웃한 오른쪽 옆 픽셀값과 그 위의 픽셀값 정보를 이용하여 경계 픽셀의 여부와 다음 경계 검출 탐색 진행 방향을 결정하여 경계를 검출하면서 한 바퀴 돌아 시작 픽셀로 돌아오면 현재 영역의 경계 탐색을 종료한다. 다음 영역의 경계 검출을 위하여 방문하지 않은 픽셀들의 픽셀값을 탐색하면서 새로운 영역의 경계 검출 시작 픽셀을 탐색하여 경계 검출 단계를 반복 수행하면서 이진화 영상의 모든 픽셀들을 탐색하면 종료한다.

경계 픽셀과 이웃한 2개의 픽셀 정보를 이용하여 다음 경계 검출 탐색 진행 방향을 결정하게 되며 경계 검출 탐색 진행 방향의 종류는 4가지로 정의하였으며 <그림 8>과 같다.

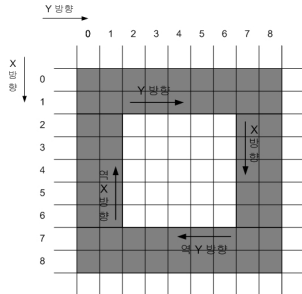


그림 8. 경계 검출 탐색 진행 방향의 종류  
Fig 8. Type of edge detection exploring progress direction

제안한 영역 경계 검출의 단계는 다음과 같다.

단계 1 : <그림 9>와 같이 이진화 영상의 왼쪽 맨 위에 있는 좌표 (0, 0)의 픽셀부터 Y 방향으로 차례로 탐색하면서 (x, y)의 픽셀값이 255이고, (x - 1, y)의 픽셀값이 0인 픽셀을 만나면 경계 픽셀로 표시하고 경계 검출의 시작 픽셀 (x, y)로 정한다.

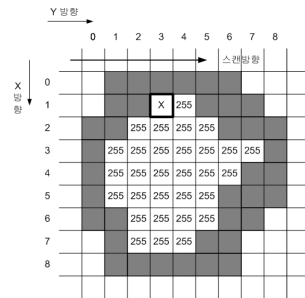
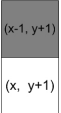
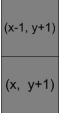
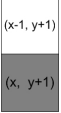
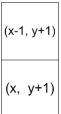


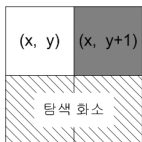
그림 9. 경계 추적 시작 픽셀  
Fig 9. Edge tracing starting pixel

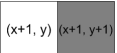
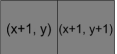
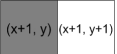
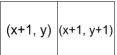
단계 2 : (x, y) 픽셀과 Y 방향으로 이웃한 (x, y + 1)과 (x - 1, y + 1)의 픽셀값을 검사하여 경계 픽셀의 여부와 다음 탐색 방향을 설정한다. Y 방향으로 탐색할 경우 경계 픽셀과 이웃한 2개의 픽셀값에 따른 다음 탐색 방향은 아래와 같다.



- ①   $(x, y + 1)$ 을 경계 픽셀로 표시하고,  $(x, y + 1)$ 을  $(x, y)$ 로 정하고 단계 2의 과정을 수행한다.
- ②   $(x, y)$  픽셀에 대하여 단계 3의 과정을 수행한다.
- ③   $(x, y)$  픽셀에 대하여 단계 3의 과정을 수행한다.
- ④   $(x - 1, y)$ 와  $(x - 1, y + 1)$  픽셀에 대하여 단계 5의 과정을 수행한다.

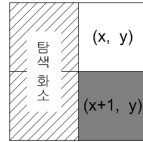
단계 3 :  $(x, y)$  픽셀과  $X$  방향으로 이웃한  $(x + 1, y)$ 와  $(x + 1, y + 1)$ 의 픽셀값을 검사하여 경계 픽셀의 여부와 다음 탐색 방향을 설정한다.  $X$  방향으로 탐색할 경우 경계 픽셀과 이웃한 2개의 픽셀값에 따른 다음 탐색 방향은 아래와 같다.

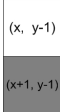
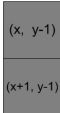
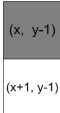
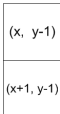


- ①   $(x + 1, y)$ 를 경계 픽셀로 표시하고,  $(x + 1, y)$ 을  $(x, y)$ 로 정하고 단계 3의 과정을 수행한다.
- ②   $(x, y)$  픽셀에 대하여 단계 4의 과정을 수행한다.
- ③   $(x, y)$  픽셀에 대하여 단계 4의 과정을 수행한다.
- ④   $(x, y + 1)$ 와  $(x + 1, y + 1)$  픽셀에 대하여 단계 2의 과정을 수행한다.

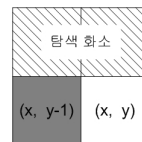
단계 4 :  $(x, y)$  픽셀과 역  $Y$  방향으로 이웃한  $(x, y - 1)$ 과  $(x + 1, y - 1)$ 의 픽셀값을 검사하

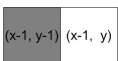
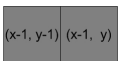
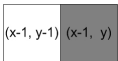
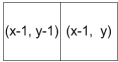
여 경계 픽셀의 여부와 다음 탐색 방향을 설정한다. 역  $Y$  방향으로 탐색할 경우 경계 픽셀과 이웃한 2개의 픽셀값에 따른 다음 탐색 방향은 아래와 같다.



- ①   $(x, y - 1)$ 을 경계 픽셀로 표시하고,  $(x, y - 1)$ 을  $(x, y)$ 로 정하고 단계 4의 과정을 수행한다.
- ②   $(x, y)$  픽셀에 대하여 단계 5의 과정을 수행한다.
- ③   $(x, y)$  픽셀에 대하여 단계 5의 과정을 수행한다.
- ④   $(x + 1, y - 1)$ 과  $(x + 1, y)$  픽셀에 대하여 단계 3의 과정을 수행한다.

단계 5 :  $(x, y)$  픽셀과 역  $X$  방향으로 이웃한  $(x - 1, y)$ 와  $(x - 1, y - 1)$ 의 픽셀값을 검사하여 경계 픽셀의 여부와 다음 탐색 방향을 설정한다. 역  $X$  방향으로 탐색할 경우 경계 픽셀과 이웃한 2개의 픽셀값에 따른 다음 탐색 방향은 아래와 같다.



- ①   $(x-1, y)$ 를 경계 픽셀로 표시하고,  $(x-1, y)$ 을  $(x, y)$ 로 정하고 단계 5의 과정을 수행한다.  $(x, y)$ 로 정하고 단계 3의 과정을 수행한다.
- ②   $(x, y)$  픽셀에 대하여 단계 2의 과정을 수행한다.
- ③   $(x, y)$  픽셀에 대하여 단계 2의 과정을 수행한다.
- ④   $(x-1, y-1)$ 과  $(x, y-1)$  픽셀에 대하여 단계 4의 과정을 수행한다.

단계 6 : 단계 2 ~ 단계 5의 과정을 반복 수행하면서 하나의 영역을 경계 검출하다가 한바퀴 돌아 처음 시작 픽셀로 도달하면 현재 영역의 경계 검출을 종료한다.

단계 7 : 다음 영역의 경계 검출을 위하여 Y 방향으로 미방문 픽셀들을 탐색하면서  $(x, y)$ 의 픽셀값이 255이고,  $(x-1, y)$ 의 픽셀값이 0인 픽셀을 만나면 단계 1 ~ 단계 6의 과정을 반복 수행하고, 영상의 마지막 픽셀까지 탐색이 끝나면 경계 검출을 종료한다.

#### IV. 실험결과

제안한 경계 검출 방법의 성능을 분석하기 위하여 실험 영상으로 256×256 크기의 이상적인 계단 경계를 가지는 원, 삼각형, 사각형 혼합 형태의 이진화 영상 30개와 배경이 존재하는 이진화 영상 70개를 사용하였으며, 경계 검출 결과 영상은 <그림 10>과 같다.

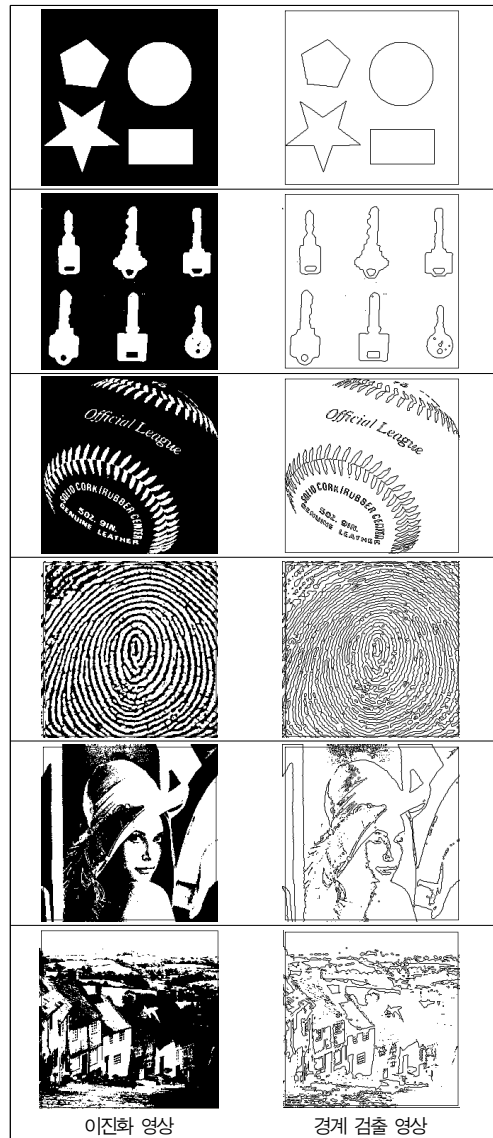


그림 10. 영역 경계 검출 결과 영상  
Fig 10. Area edge detection result image

기존의 Moore-Neighbor Tracing 영역 경계 추적 방법과 제안한 영역 경계 검출 방법의 경계 검출 수행 시간과 두 경계 검출 결과 영상에 대한 매칭 정도를 비교하였으며, 성능 분석을 위한 비교 화면은 <그림 11>과 같고, 두 경계 검출 알고리즘의 수행 시간 비교 및 매칭률은 <표 1>과 같다.

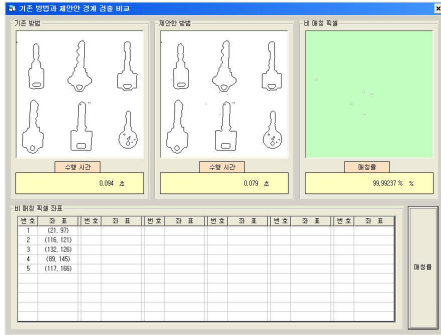


그림 11. 경계 검출 성능 비교 화면  
Fig 11. Edge detection performance comparison screen

표 1. 수행 시간 및 매칭률 비교  
Table 1. Execution time and matching rate comparison

입력 영상	Moore-Neighbor Tracing	제안한 방법	매칭률
영상 1	0.125초	0.047초	100%
영상 2	0.125초	0.062초	100%
영상 3	0.109초	0.047초	100%
영상 4	0.125초	0.063초	100%
영상 5	0.141초	0.047초	100%
영상 6	0.109초	0.079초	99.99%
영상 7	0.11초	0.063초	99.98%
영상 8	0.14초	0.094초	99.87%
영상 9	0.125초	0.063초	99.97%
영상 10	0.219초	0.125초	99.95%
평균	0.13초	0.07초	99.98%

기존의 Moore-Neighbor Tracing 영역 경계 추적의 방법과 제안한 방법을 <표 1>에 나타난 10개의 영상에 대하여 수행 시간을 비교한 결과 매칭률 평균 99.98%에서 0.06초 단축된 결과를 보였으며, 전체 영상 100개에 대한 기존 방법의 평균 수행 시간은 0.124초이며 제안한 방법의 평균 수행 시간은 0.068초로 기존 방법에 비해 약 45%의 수행 시간 단축을 보였으며 두 방법의 경계 픽셀 매칭률은 99.96%를 나타내었다. 두 경계 검출 결과에 대한 매칭률의 차이는 독립된 하나의 픽셀에 대하여 기존 방법의 경우에는 경계선으로 나타내지 않았으며 제안한 방법에서는 경계선으로 나타냈기 때문이다.

## V. 결론

영상에 포함된 물체를 인식하는데 있어서 영상 해석의 하위 단계에서 얼마만큼 정확하고도 유용한 정보를 찾아 주느냐에 따라 크게 달라질 수 있으며, 이때 얻어지는 중요한 정보가 바로 경계이다. 이 경계 검출은 인간의 능력 중에서 영상을 인식하는 접근 방법으로 영상의 많은 부분들은 경계를 주체로 표현되며, 경계 검출의 중요한 특성은 목적 윤곽에 일치하는 얇은 선을 생산하는 능력이라 할 수 있다.

본 논문에서는 영상 처리의 전처리 과정으로 사용되는 경계 검출을 이진화 영상에서 2개의 이웃된 픽셀의 정보를 이용하여 미리 정의된 경계 픽셀의 패턴과 비교하여 경계 픽셀을 찾고, 다음 경계 검출 탐색 픽셀에 대한 방향을 결정하여 경계를 검출하는 방법으로 물체의 윤곽에 일치하는 세선화 된 경계를 검출하여 기존 경계 검출 방법의 후처리 과정으로 사용되는 세선화 과정을 생략할 수 있었다. 또한 이진화 영상을 이용한 기존 영역 경계 추적 방법에 비해 수행 속도의 향상을 얻을 수 있었다. 그러나 제안한 알고리즘은 이진화 영상에서 경계를 검출하는 방법으로 입력 영상의 명암도에 따라 이진화 영상에 차이가 있으므로 경계 검출의 정확도가 달라질 수 있다. 따라서 영상의 이진화 기법에 대한 좀더 발전된 연구가 이루어져야 할 것이다.

## 참고문헌

- [1] (국외 논문) K. S. Fu and J. Mui, "A Survey on Image Segmentation," Pattern Recognition, Vol. 13, No. 1, pp.3-16, January 1981.
- [2] (국내 논문) 김수겸, 장유정, "에지의 구조적 정보를 이용한 에지 검출," 정보처리학회 논문지, 제 3권, 제 5호, 1337-1345쪽, 1996년 9월.
- [3] (국내 논문) 윤현주, 이상욱, "경계선 기반의대화형 영상 분할 시스템," 그래픽스학회 논문지, 제 8권, 제 2호, 15-22쪽, 2002년 6월.
- [4] 김동현, 이소행, 정진용, 양현호, 최우진, "컬러 에지 검출에 관한 연구," 컴퓨터정보학회논문지, 제 4권, 제 3호, 8-12쪽, 1999년 9월.
- [5] D. F. Rogers, "Procedural elements for computer graphics", McGraw-Hill, pp.176-192, 1985.
- [6] R. Crane, "A simplified approach to Image Processing",

Prentice-Hall, pp.108-117, 1997.

- [7] 한상훈, 조형제, "그림자 정보를 이용한 HSV 컬러 모델 기반의 전방 차량 검출 및 차선 정보 검출", 멀티미디어학회 논문지, 제 5권, 제 2호, 176-190쪽, 2002년 4월.
- [8] J. Scharcanski, A.N. Venetsanopoulos, "*Edge Detection of Color Images Using Directional Operators*" IEEE Trans. Circuits and System For Video Technology, Vol. 7 No. 2, pp.397-401, April 1997.
- [9] 강동중, 하중은, "Visual C++을 이용한 디지털 영상처리", 사이텍미디어, 261-269쪽, 2003년.



### 저 자 소 개



**김 승 희**  
2000년 2월 : 관동대학교 전자계산  
학과 석사  
2008년 2월 : 관동대학교 전자계산  
학과 박사  
현재 : 관동대학교 강사  
관심분야 : 패턴인식, 영상처리, 보안



**남 시 병**  
1994년 2월 : 단국대학교 전자공학  
과 박사  
1983년 ~ 2006년 2월 : 삼척대학  
교 전자공학과 교수  
현재 : 강원대학교 전자정보통신공학  
부 교수  
관심분야 : 임베디드시스템, USN,  
패턴인식



**임 해 진**  
1993년 8월 : 성균관대학교 전자공  
학과 박사  
1983년 ~ 2006년 2월 : 삼척대학  
교 전자공학과 교수  
현재 : 강원대학교 전자정보통신공학  
부 교수  
관심분야 : 정보통신일반, 통신, TMN,  
멀티미디어응용, BISDN