

# 잡영블럽 검출에 의한 잡영가지 제거 방법의 개선

김성옥\*, 임은경, 김민환  
부산대학교 컴퓨터공학과

## Enhancement of Removing Parasitic Branches by Detecting Noise Blobs

Seongok Kim\*, Eunkyung Lim, Minhwan Kim  
Dept. of Computer Engineering, Pusan National University  
E-mail : sokim@kiflt.re.kr, {eklim77,mhkim}@pusan.ac.kr

### 요 약

영상분할이나 연결요소 추출 등에 의해 구해진 물체 영역의 경계 부분에 잔가지 모양으로 나타나는 잡영 가지는 물체의 형상 특징 연산 및 인식 과정에서 오류를 일으키는 원인이 되므로 효과적으로 제거할 수 있는 방법이 필요하다. 기존의 연구에서는 한 픽셀 두께의 잡영가지만을 제거할 수 있는 방법이 제안되었다. 본 연구에서는, 부분적으로 두 픽셀 이상이 뭉쳐져 둥그스름한 덩어리(blob, 블럽)를 형성하고 있는 잡영가지도 4-8-방향 윤곽선 추적에 의해 제거할 수 있는 방법을 제안한다. 먼저 잔가지를 구성하고 있는 픽셀과 물체 영역을 구성하는 픽셀을 개략적으로 분리한 후, 윤곽선 추적을 하면서 사용자가 정의한 크기보다 작은 블럽을 포함한 잔가지를 추출해낸다. 이어서, 이러한 잔가지를 사용자가 정의한 잡영 가지의 크기와 비교하여 최종적으로 제거할 수 있도록 한다. 이와 같이 사용자 입력 변수에 따라 잡영 가지에 대한 개념을 보다 확장하여 자유롭게 정의할 수 있도록 명시화 함으로써, 영상의 특성에 따라 보다 유연하게 잡영가지를 정의하여 제거할 수 있도록 하였다.

### 1. 서론

패턴인식이나 영상인식과 같은 분야에서 잡영은 영상의 품질을 떨어뜨릴 뿐만 아니라 연산속도 저하 및 오인식을 야기시키는 원인이 되므로, 이것을 제거하기 위한 연구가 많이 이루어져 왔다. 잡영의 한 종류로서 물체 영역의 경계 부분에 잔가지 모양으로 나타나는 것을 잡영가지(parasitic branch)라고 한다[1]. 예를 들어, 칼라지도에서 지도 구성 요소로서 해안선을 추출해내고자 할 때, 유사한 칼라로 표

현된 바다 부분과 인접해 있어 많은 잡영가지가 포함된 형태로 추출된다[2]. 또한, 직물류 패턴을 스캔한 디지털 영상에서도 직물의 재료와 재단 공정의 특성으로 인해 많은 잡영가지가 발생하게 된다.

이러한 잡영가지들은 의미없는 정보로서 필요없이 기억장소를 차지할 뿐만 아니라 물체 영역의 형상 특징, 즉 둘레길이, 면적, 중심점 등과 같은 것을 계산하는데 오류를 발생시킨다. 또한 세션화 작업에서 오류를 일으키는 원인이 되므로, 세션화 알고리즘 자체에 가지치기 단계를 두거나[3] 잡영가지

발생이 최소가 되도록 세션화 알고리즘을 개발하기도 하였다[4].

잡영가지를 제거하는 방법으로서, 모폴로지(morphology)에서의 가지치기(pruning) 기법[5]과 윤곽선 추적 알고리즘을 이용한 잡영가지 제거 방법[6,7]이 있다. 모폴로지에서의 가지치기 기법은 주로 세션화된 결과에 대해 적용하는 것으로서, 가지의 끝점을  $n$ 번 순차적으로 제거하였다가 가지 길이가  $n$ 보다 긴 경우에는 여전히 잔여 끝점이 존재하므로 이것을 시작점으로 하여 원래의 가지를 복구하는 방법을 사용한다. 따라서, 가지 길이가  $n$ 보다 작은 것은 제거되는 효과를 얻을 수 있다. 그러나, 연산 시간이 많이 소요되며, 잔여 끝점 및 제거되었던 끝점 부근에 다른 잡영가지가 있을 경우에는 제대로 제거하지 못하는 문제점이 있다. 또한, 세션화되지 않은 물체 영역의 잡영가지를 제거하는데 확장하여 사용하기가 곤란하다. 한편,  $2 \times 2$  마스크를 이용한 윤곽선 추적에 의한 잡영가지 제거 방법[6,7]은 윤곽선 추적경로가 좌우 대칭이 되는 부분을 찾아, 일정 길이 이하인 경우에 제거하는 것이다. 이 방법은 보다 효과적으로 잡영가지를 제거할 수 있으나, 부분적으로 두 픽셀 이상 두께를 갖는 잡영가지를 제거할 수 없다.

본 논문에서는 8-방향 윤곽선 추적 알고리즘[8]을 수정한 4-8-방향 윤곽선 추적 알고리즘을 사용하여 부분적으로 동그스름한 작은 덩어리 형태를 갖는 잡영가지도 제거할 수 있는 방법을 제안한다. 이러한 덩어리를 불립(blob)이라고 부르며, 이것을 잡영가지 정의에 도입하여 명시적으로 기술한다. 한편, 잔가지를 구성하는 픽셀들을 먼저 선정한 후에, 윤곽선 추적 알고리즘에 의해 잡영가지 전체를 추출해내는 방법을 제안한다. 이때, 사용자에게 의해 불립의 크기와 제거할 잡영가지의 크기를 결정할 수 있게 함으로써, 사용자가 주어진 영상의 특성에 따라 효과적으로 잡영가지를 제거할 수 있도록 한다.

## 2. 잡영가지의 정의 및 기존 방법의 문제점

### 2.1 잡영가지의 정의

잡영가지는 물체 영역의 경계 부분에 그림 1에서와 같이 잔가지 형태를 갖는 것으로서, 이것을 정의하기 위해서는 픽셀간의 연결에 대한 정의뿐만 아니라 잔가지의 두께, 길이, 형태에 대한 것도 정의하여야 한다. 이러한 정의해야 할 요소들은 응용 상황에 따라 달리 정의될 수 있다. 예를 들어, 그림 1의 (a)는 길이에 대한 정의에 따라 잡영가지 여부가 달라지며, (b)는 복합 잔가지의 허용 여부에 따라 결정된다. 한편, 그림 1의 (c)와 (d)는 잔가지 일부분에 두 픽셀 이상의 두께를 갖는 경우를 나타낸 것으로서, 정의하기가 쉽지 않다.

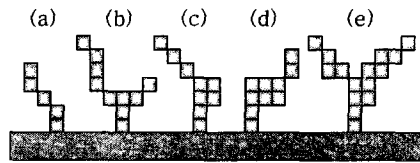


그림 1. 잡영가지의 정의를 위한 예제

본 연구에서는 그림 1의 (e)와 같은 잔가지도 제거할 수 있도록 다음과 같은 조건들을 만족하는 것을 잡영가지로 정의한다.

- (조건1) 물체 영역의 경계와 한 픽셀 두께로 연결된 4-연결 또는 8-연결에 의한 sub-연결요소(sub-connected component)
- (조건2) 두 픽셀 이하 두께의 (복합) 잔가지 형태
- (조건3) 사용자가 정의한 크기 이상의 불립이 없으며, 두 불립 사이에는 반드시 한 픽셀 두께로 연결된 잔가지
- (조건4) Sub-연결요소의 구성 픽셀 수가 사용자가 정의한 크기보다 작은 잔가지

위의 (조건1)은 잡영가지의 시작 부분을 정의하는 것으로서, 그림 1의 예제에서와 같이 물체 영역으로부터 반드시 한 픽셀 두께로 뺄어 나온 부분만으로 국한하기 위한 것이다. (조건2)는 복합 잔가지도 허용하기 위한 것이며, (조건3)은 그림 1의 (c),(d)에서와 같은 불립도 허용하기 위한 것이다. 그러나, 그림 2(b)와 같이 불립이 떨어져 있는 경우는 허용하지만 (a)와 같이 붙어 있는 경우는 배제한다. (조건4)는 잡영가지의 길이를 정의하기 어려우므로, 제거할 총 픽

셀 수로 크기를 제한하기 위한 것이다.

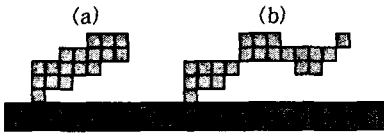


그림 2. 블럽 간의 분리 여부 예

### 2.2 기존 방법의 문제점 분석

일반적으로 잡영가지를 일괄적으로 정의하는 것은 곤란하다. 따라서, 사용자에게 의해 영상의 응용 환경 및 응용 목적에 따라 적절하게 잡영가지를 정의할 수 있어야 한다.

모폴로지에서의 가지치기 기법[5]은 세션화된 물체에서의 잡영가지를 제거하기 위한 것이다. 세션화에 의해 한 픽셀 두께로 표현된 대상물에서의 잡영가지는 단순히 잔가지의 길이를 제한함으로써 적절한 정의가 가능하다. 그림 3(a)에서 길이가 30이 아닌 잔가지를 제거하는 과정을 보여준다. 그러나, 이 기법은 (c)에서와 같이 잔여 끝점 부근의 잡영가지를 제거하지 못하는 문제점이 있다.

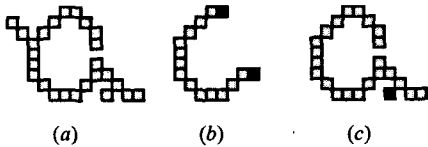


그림 3. 모폴로지 가지치기의 문제점

한편, 2x2 마스크를 이용한 윤곽선 추적 알고리즘에 의한 잡영가지 제거 방법[6,7]에서는, 그림 1의 (b)와 같은 복합 잔가지에 대해 정의된 잡영가지 길이에 따라 그림 4에서와 같이 세 가지 형태의 결과를 나타낸다. 또한, 그림 1의 (c),(d)에서와 같이 블럽이 포함된 경우에는 잡영가지로 정의하지 못한다.

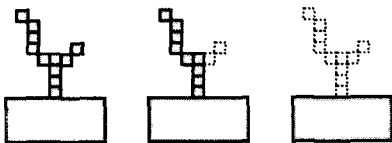


그림 4. 길이 정의에 따른 잡영가지 제거 예

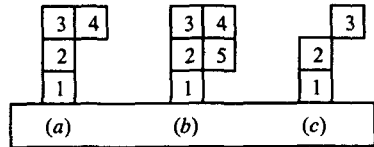
본 논문에서는, 그림 1의 (c),(d),(e)와 같이 잔가

지 일부분에 블럽이 포함된 경우도 잡영가지로 정의하여 제거할 수 있을 뿐만 아니라 그림 4에서와 같이 여러 가지의 제거 형태가 결과로 나오는 것을 방지하기 위해 잔가지 전체의 픽셀 수를 제한하여 제거가 가능하도록 한다.

### 3. 윤곽선 추적 및 잡영가지 제거 방법

#### 3.1 4-8-방향 윤곽선 추적 방법

물체 영역의 윤곽선 추적 방법으로는 3x3 마스크를 이용하여 8-방향의 추적이 가능한 알고리즘[8]을 수정하여 4-8-방향 추적이 가능한 것을 사용한다. 4-8-방향 추적은 그림 5에서와 같이 4-방향 추적을 우선적으로 선택하되, 이것이 불가능할 때는 8-방향 추적도 허용하는 것을 의미한다.



(a) 1,2,3,4,3,2,1 (b) 1,2,3,4,5,2,1  
(c) 1,2,3,2,1

그림 5. 4-8-방향 추적 경로 예

4-8-방향 윤곽선 추적의 목적은 그림 5(a)와 같은 잔가지인 경우에도 추적 경로가 픽셀 4를 중심으로 대칭이 되도록 하기 위한 것이다. 이것에 대한 8-방향 추적 경로는 1,2,3,4,2,1이 되어 대칭이 되지 않는다. 한편, 그림 5(b)의 잔가지는 여전히 대칭을 이루지 않는다. 본 논문에서는, 픽셀 2,3,4,5를 묶어 하나의 블럽으로 간주함으로써, 그 블럽을 중심으로 대칭이 되는 것으로 해석한다.

4-8-방향 윤곽선 추적 알고리즘에서는 현재 추적하고 있는 픽셀  $c$ 를 3x3 마스크(그림 6)의 중심 픽셀로 하여 주변의 이웃 픽셀 8개를 시계 방향으로 탐색하면서 처음 만나는 물체 영역 픽셀을 검출하는 방법을 사용한다. 이때, 중심 픽셀의 4-이웃 픽셀들을 먼저 살펴봄으로써 4-방향 추적을 우선적으로 진행하도록 하고 있다. 이러한 4-이웃 픽셀들의 집합을  $N_4$ 라 하고, 나머지 대각선 방향의 8-이웃 픽셀들의 집

합을  $N_8$ 이라 한다. 만약  $N_4$ 의 픽셀들 중에서 물체 영역 픽셀을 찾지 못하면,  $N_8$ 의 픽셀들 중에서 처음 만나는 물체 영역 픽셀을 다음의 추적 픽셀로 결정한다.

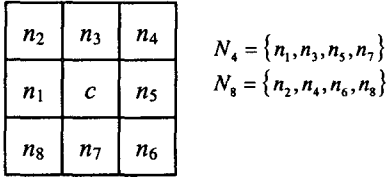


그림 6. 윤곽선 추적용 3×3 마스크

4-8-방향 윤곽선 추적 알고리즘은 다음과 같다.

- (단계1) 영상을 래스터 방식으로 스캔하면서, 물체 영역의 윤곽선 추적 시작점  $s$ 를 찾는다.
- (단계2)  $c$ 를  $s$ 로,  $s$ 의 왼쪽 픽셀을  $b$ 로 설정한다.
- (단계3)  $b$ 에서 시작하여  $c$ 의 이웃 픽셀들을 시계 방향으로  $n_1, n_2, \dots, n_8$  이라 할 때, 이들 중에서 물체 영역 픽셀에 해당하는 가장 처음의 것  $n_i$ 를 찾는다.
- (단계4) 만약  $n_i \in N_8$  &  $n_{i+1} \in N_4$ 이면,  $c = n_{i+1}$  와  $b = n_i$ 로 설정한다. 아니면,  $c = n_i$  와  $b = n_{i-1}$ 로 설정한다.
- (단계5)  $c$ 와  $s$ 가 같아질 때까지 (단계3)과 (단계4)를 반복한다.

### 3.2 불림의 유형 및 검출 방법

불림을 동그스름하게 덩어리진 픽셀들의 집합으로 정의하였다. 그림 7(a)는 하나의 픽셀을 나타낸 것으로서, 그림 5의 (a),(c)에서와 같이 제일 작은 크기의 불림으로 볼 수 있다. 그림 7의 (d)는 너무 긴 형태의 것이므로, 불림에서 제외한다. 또한, 그림 7의 (e)는 중심의 픽셀이 윤곽선 추적과정에서 제외될 뿐만 아니라 잔가지의 두께가 세 픽셀이 되므로 제외한다. 따라서, 본 논문에서 정의하는 불림은 그림 7의 (a),(b),(c) 유형만으로 국한된다.

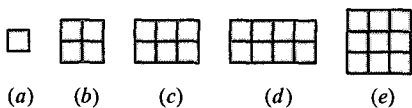


그림 7. 불림의 유형 예

윤곽선 추적에 의한 잡영가지 제거 방법의 기본 아이디어는 추적 경로중에서 좌우 대칭이 되는 일부 추적 경로를 추출해내는 것이다. 예를 들어, 그림 5의 (a)에서 픽셀 4의 위치에서 연이어 추적하는 픽셀들이 이전의 추적 픽셀 순서의 역순이 되면 잡영가지로 결정한다. 그림 7의 (b)와 (c)의 경우에는, 이전의 추적 픽셀들을 4개 또는 6개까지 각각 조사하면서 불림이 형성되는가를 먼저 결정한다. 만약 불림이 형성되면, 그 이전의 대칭 경로를 계속 추적하여 잡영가지로 추출한다.

### 3.3 잔가지 구성 픽셀 분리 방법

4-8-방향 윤곽선 추적 알고리즘으로 물체 영역의 윤곽선을 추적하면서 잔가지를 구성하는 픽셀을 분리해 낼 수 있다. 윤곽선 추적용 3×3 마스크를 현재 추적 픽셀에 적용하여,  $n_1, n_2, \dots, n_8$  순서로 픽셀을 살펴보다 0에서 1로 값이 변하는 횟수가 2회 이상이면 잔가지 구성 픽셀(그림 8의 흰색 픽셀)이라고 할 수 있다. 상대적으로, 그림 8의 검은 픽셀은 물체 구성 픽셀이거나 불림을 구성하는 픽셀이다. 그림 8의 흰색 및 검은 색 픽셀을 본 논문에서는 각각 잔가지 픽셀, 물체 픽셀이라고 명명한다.

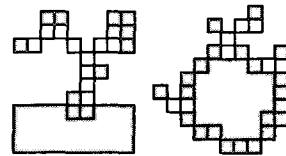


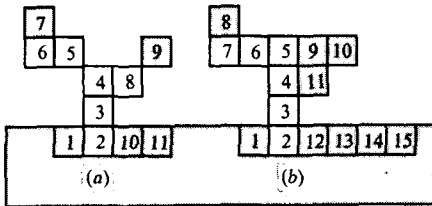
그림 8. 잔가지 구성 픽셀 분리 결과

### 3.4 잡영가지 추출 및 제거 방법

먼저 복합 잔가지를 구성하는 전체 픽셀들을 묶어내기 위해, 잔가지 추적 모드 플래그  $p$ 를 사용한다. 플래그가 0인 상태에서의 추적 과정에서 새로운 불림을 검출하게 되면  $p = 1$ 로 한다. 이 플래그를 0으로 리셋(reset)하는 것은 물체 픽셀이 불림의 크기만큼 연이어 나타날 때에 한다. 불림의 크기  $m$ 은 그림 7의 (a), (b), (c)의 경우에 대해 역추적하는 횟수를 감안하여 편의상 각각 2, 4, 6으로 정의한다.

예를 들어, 불림의 크기가 2로 주어진 경우에,

그림 9(a)에 대한 추적 경로에서 동그라미로 표시된 위치에서 플래그  $p$ 가 1로 설정되고, 사각형으로 표시된 위치에서 다시 0으로 된다. 괄호로 묶인 일부 경로는 대칭 경로로 검출된 것을 의미한다. 대칭 경로에서 처음의 픽셀을 제외하는 것은 과도한 제거를 방지하기 위한 것이다. 즉 픽셀 2는 제거하지 않음으로써 물체 영역이 침식당하는 것을 방지한다. 그림 9(b)의 경우는 불림의 크기가 4로 주어진 경우로서, 굵은 숫자로 표시된 것과 같은 불림을 역추적 과정에서 검출할 수 있다.



- (a) 1, 2, (3, 4, (5, 6, 7, 6, 5, 4), 8, 9, 8, 4, 3, 2,) 10, 11
- (b) 1, 2, (3, 4, 5, (6, 7, 8, 7, 6, 5,) 9, (10, 9,) 11, 4, 3, 2,) 12, 13, 14, 15

그림 9. 잡영가지 전체의 추출 예

한편, 플래그  $p$ 를 0으로 리셋(reset)되기 전에 다른 잔가지가 검출되어 여러 개의 잔가지가 한꺼번에 검출될 수 있다. 이러한 경우는 사용자에게 의해 불림 크기가 6으로 주어지고, 그림 10에서와 같이 바로 이웃한 위치에 잔가지가 있을 경우에 발생한다. 이러한 두 개 이상의 잔가지들은 플래그가 리셋된(그림 10의 검은 점에서) 후, 추적 경로의 역추적 과정에서 각 잔가지에 해당하는 대칭 경로 사이에 대칭이 되지 않는 픽셀(빗금친 점)이 존재함을 조사하여 분리할 수 있다.

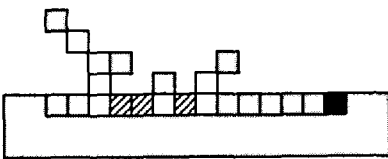


그림 10. 여러 개의 잔가지가 묶여 추출된 경우

자세한 잡영가지 제거 알고리즘은 다음과 같다.

(단계1) 윤곽선 추적을 시작할 점  $s$ , 불림의 크기  $m$ , 잡영가지의 크기  $l$ 을 입력받는다. 현재의 추적점  $i$ 를  $s$ 로 할당한다. 잔가지 추적 모드 플래그  $p$ 와 대칭 경로 추적 모드 플래그  $r$ 을 각각 0으로 초기화한다.

(단계2)  $i$ 의 다음 추적점  $i'$ 를 결정하여  $i = i'$ 으로 한다. 만약  $i' = s$ 이면, (단계4)로 간다.

(단계3) 다음의 항목 중에서 조건에 맞는 것을 수행한 후, (단계2)로 간다.

- (1) 만약  $r = 1$ 이면,  $i$ 가 대칭 경로의 점이 되는지 조사한다. 만약 대칭 경로의 점이 아닐 경우에는,  $r = 0$ 으로 하고 대칭 경로를 찾아 설정한다.
- (2) 만약  $p = 1$  &  $r = 0$ 이면서 경로를 역추적하여 새로운 불림이 형성되면,  $r = 1$ 로 한다.
- (3) 만약  $p = 1$  &  $r = 0$ 이면서 물체 픽셀이 연이어 불림의 크기  $m$ 만큼 존재하면,  $p = 0$ 으로 하고 잔가지들을 분리한다. 각 잔가지의 크기를  $l$ 과 비교하여 잡영가지를 추출하고 대칭 경로로 표기한다.
- (4) 만약  $p = 0$ 이면, 경로를 역추적한다. 만약 새로운 불림이 형성되면,  $p = 1$  &  $r = 1$ 로 한다.

(단계4) 시작점  $s$ 가 잡영가지의 일부가 되는지 여부를 조사한다. 만약 잡영가지를 이루면, 해당 잡영가지를 추출한다. 알고리즘을 종료한다.

4-8-방향 윤곽선 추적 알고리즘에 의해 주어진 어떤 물체 영역에서 추적되는 경계 픽셀을  $n$ 개라 할 때, 위의 잡영가지 제거 알고리즘의 시간 복잡도(time complexity)는  $O(n)$ 이다. 이것은 각 윤곽선 구성 픽셀이 방문되는 횟수는 최대  $\max\{m, 4\}$ 이기 때문이다.

#### 4. 실험 및 토의

그림 11은 여러 가지 잡영가지 크기와 불림 크기를 갖는 인위적인 실험영상에 대해 알고리즘을 적용한 예이다. 그림 11의 (a), (b), (c), (d)는 각각 9, 10, 10, 13 개의 픽셀로 구성된 잔가지이며, (b)와 (c)에는 각각 불림의 크기가 4, 6인 것이 포함되어 있다. 먼저 잡영가지의 크기를 10으로 하고 불림의 크기를 4로 준 경우에는, (a)와 (b)만 제거되었다. 여기에서, 불림

의 크기를 6으로 변경하면, (c)도 제거됨을 알 수 있었다. (d)의 잔가지는 잡영가지의 크기를 13 이상의 크기로 설정함으로써 제거할 수 있었다.

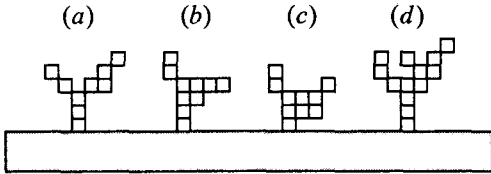


그림 11. 잡영가지 제거를 위한 인위적인 잔가지 예

그림 12는 지도의 해안선 영상으로 배경의 색상과 객체의 윤곽선의 색상이 유사하여 잡영가지가 많이 나타남을 보여준다[2]. 그림 13은 이 해안선에 대해 여러 가지 방법으로 잡영가지를 제거한 결과를 보여준다. 그림 13(a)는 2x2 마스크를 이용하여 잡영가지를 제거하는 방법[6]의 결과로서, 본 논문에서 제안한 방법에서 블러프의 크기를 2로 준 경우와 같다. 그림 13(b),(c)는 각각 블러프의 크기를 4, 6으로 주었을 때 잡영가지가 제거된 결과이다.

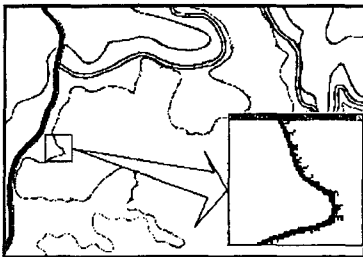
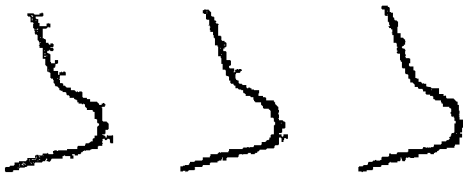


그림 12. 실제 잡영가지가 발생하는 예제



(a) 블러프 크기=1, (b) 블러프 크기=4, (c) 블러프 크기=6  
그림 13. 해안선에 대한 잡영가지 제거 결과

를 갖는 잡영가지도 제거할 수 있는 방법을 제안하였다. 특히, 블러프의 개념을 제안하고 유형을 정의함으로써 보다 다양한 방법으로 잡영가지를 제거할 수 있도록 하였다. 실험을 통해 제안한 알고리즘이 제대로 동작하는 것을 확인하였으며, 또한 사용자가 블러프의 크기 및 잡영가지의 크기를 제한함으로써 보다 효과적으로 잡영가지를 제거할 수 있었다.

패턴 인식이나 컴퓨터 시각 분야에서 형상에 대한 전처리 작업으로 활용될 수 있으며, 특히 세션화 연산의 전처리 작업으로 매우 유용하게 활용될 수 있을 것으로 기대된다.

### [참고문헌]

- [1] 이성환, 문자인식, 흥릉과학출판사, 1993
- [2] 김성영, 김종민, 김민환, "추적방식에 의한 칼라지도 영상에서의 직선 및 곡선 영역 추적에 관한 연구", 제 9회 신호처리학회합동학술대회, pp.641-644, 1996
- [3] Gabriela Sanniti Di Baja and Edouard Thiel, "(3,4)-Weighted Skeleton Decomposition for Pattern Representation and Description", Pattern Recognition, vol.27, No.8 pp.1039-1049, 1994
- [4] Nikolaos G. Bourbakis, "A Parallel-Symmetric Thinning Algorithm", Pattern Recognition, Vol.22, No.4, pp.387-396, 1989
- [5] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Addison Wesley, 1993
- [6] 권태균, 김종민, 김성영, 김민환, "2x2 마스크를 이용한 윤곽선 추적 알고리즘의 응용에 관한 연구", 한국정보처리학회 97추계 학술발표논문집, 제4권2호, pp.1163-1168, 1997
- [7] 김성영, 권태균, 김민환, "추적에 의한 단순화된 윤곽선 추출", 춘계한국멀티미디어학회, pp356-361, 1999
- [8] Ramesh Jain, Rangachar Kasturi, and Brain G. Schunck, Machine Vision, Publication Services, 1995

### 5. 결론

본 논문에서는 일부분에서 두 픽셀 이상의 두께