

점진적인 화소 확장에 의한 선분 추출

이재광[†], 박창준^{‡‡}

요 약

본 논문에서는 점진적인 화소 확장을 이용하여 영상 내에 존재하는 선분을 찾아내는 선분 추출 알고리즘을 제안한다. 본 논문에서 제안하는 선분 추출 방법에서는 기존의 선분 추출 방법인 허프 변환 기반 방법이나 선분의 그룹화 기반 방법과는 다른 접근법을 사용하였다. 영상이 입력되면 케니 테두리를 구하고, 테두리 화소 중 임의의 점을 선택하여 선분을 근사화 시킨 기본 직선을 만든 후, 선택된 점에서 임의의 반경 내에 있는 테두리 화소들을 선택한다. 직선과의 거리 오차와 기울기 각의 오차를 이용하여 선택된 화소에 가중치를 부여한다. 가중치 합 비교에 의해 선택된 화소들이 떨어져 있는지를 판별한 후, 가중치를 적용한 최소 자승법에 의해 선 맞춤을 하여 선분을 구하게 된다. 제안된 알고리즘은 기존에 제안된 방법들과 결과를 비교하였으며, 계산 속도가 빠르면서 실제 존재하는 선분 추출이 가능하다는 것을 실험 결과를 통해 제시한다.

Detecting Line Segment by Incremental Pixel Extension

Jaekwang Lee[†], Chang-Joon Park^{‡‡}

ABSTRACT

An algorithm for detecting a line segment in an image is presented using incremental pixel extension. We use a different approach from conventional algorithms, such as the Hough transform approach and the line segment grouping approach. The Canny edge is calculated and an arbitrary point is selected among the edge elements. After the arbitrary point is selected, a base line approximating the line segment is calculated and edge pixels within an arbitrary radius are selected. A weighted value is assigned to each edge pixel, which is selected by using the error of the distance and the direction between the pixel and the base line. A line segment is extracted by fitting a line using the weighted least square method after determining whether selected pixels are linked or delinked using the sum comparison of the weights. The proposed algorithm is compared with two other methods and results show that our algorithm is faster and can detect the real line segment.

Key words: line segment detection(선분 추출), Canny edge(케니 테두리), pixel extension(화소 확장), weight sum comparison(가중치 합 비교)

1. 서 론

선분은 영상에서 중요한 정보를 가지고 있으며, 물체 인식이나 물체 추적, 배경의 3차원 복원[1] 등에

중요한 기하 정보를 제공한다. 선분은 다른 특징점과 달리 영상 잡음의 영향을 덜 받으며, 여러 영상에서 선분을 정합시키면 매우 많은 3차원적인 정보를 얻을 수 있다. 이러한 이유로 선분 추출은 컴퓨터 비전

* 교신저자(Corresponding Author) : 이재광, 주소 : 대전 광역시 유성구 가정동 161번지 (305-700), 전화 : 042)860-5345, FAX : 042)860-1050, E-mail : ljk64386@etri.re.kr 접수일 : 2007년 8월 28일, 완료일 : 2008년 1월 3일

[†] 정회원, 한국전자통신연구원 (UST 연구연수생)
^{‡‡} 정회원, 한국전자통신연구원

(E-mail : chjpark@etri.re.kr)

* 본 논문은 정보통신부 및 정보통신연구진흥원의 IT신성장동력핵심기술개발사업의 일환으로 수행하였음.[2006-S-044-02, 멀티코어 CPU 및 MPU 기반 크로스플랫폼 게임 기술]

분야에서 중요한 문제로 많은 연구가 되어 왔으며, Hough 변환[2-4], 다각형 근사화[5], 연쇄 부호화[6], 기초 선분 그룹화[7-9] 등의 많은 방법들이 제안되었다. Duda와 Hart[2]는 공간 영상을 파라미터 공간으로 변환하여 직선을 검출하는 허프 변환(Hough Transform)을 제안하였다. 허프 변환은 직선 성분의 테두리를 검출하는데 효과적인 방법으로써, 영상에 잡음이 많아도 안정적으로 검출할 수 있다는 장점이 있다. 또한 검출한 모양의 크기가 변화되고 회전된 것에 대해서도 적용이 가능하기 때문에 많이 연구되고 있는 방법이다. 하지만 처리해야 할 영상의 크기가 커지면 사용하는 메모리의 크기와 처리시간이 급속하게 증가하게 되며, 검출 정확도가 파라미터 영역의 양자화 크기에 의존적이다. 또한 복잡한 영상에서의 선분 검출력이 떨어지는 단점이 있다. Guil. et al.[3]은 복잡성과 메모리 요구량을 줄임으로 일반화된 허프 변환 알고리즘을 개선시켰으며, Breuel[4]은 일반화된 허프 변환과 확률적인 허프 변환의 최대값을 추출하기 위한 여러 가지의 통계적 에러 모델을 사용하였다. Chung et al.[5]은 일반적인 다각형 근사화(Polygonal Approximation) 방법을 흡필드(Hopfield) 신경 회로망을 사용하여 개선시켰으며, Yuan과 Suen[6]은 $O(n)$ 시간으로 연쇄 부호화(Chain Coding)에 의해 아크의 직선성을 결정하는 알고리즘을 제안하였다.

Nacken[7]은 테두리 화소 연결에 의해 얻어지는 기초 선분들(Elementary Line Segments)을 그룹화하여 선분을 찾는 방법을 제안하였다. 이 방법은 미리 찾아진 기초 선분들에 그룹화 제약조건을 사용하여 그룹화하고, 그룹화 된 기초 선분들을 하나의 선분으로 간주하여 선분을 찾아내는 방법이다. 최근 Jang과 Hong[8]은 이 방법에 허프 변환의 장점을 추가하여 그룹화 하는 방법의 연구를 진행하고 있으며, Chaleshale et al.[9]은 테두리 지도(Edge Map)에 연쇄 부호화 방법을 사용하여 기초 선분을 추출한 후, 선분의 길이와 오차를 이용하여 더 긴 선분으로 그룹화하며 선분을 찾아내는 접근법도 연구되고 있다.

본 논문에서는 카메라 트래킹에 사용될 선분 정합용 직선을 점진적인 화소 확장에 의해 추출하는 방법을 제안한다. 카메라 트래킹에는 많은 선분이 필요하지 않으며, 연속된 여러 장의 영상에서도 개인하게 추출 될 수 있는 긴 선분이 필요하다. 본 논문이 제안하는 방법은 건물의 테두리와 같이 비교적 긴 선분

추출이 용이하며, 확률적으로 직선에 가까운 선분을 추출할 수 있는 특성을 갖는다. 본 논문에서 제안하는 점진적인 화소 확장에 의한 선분 추출 방법은, 먼저 캐니 테두리(Canny Edge) 화소 중 선분을 이루고 있는 화소들을 선택한 후, 각 화소들에 기본 직선(Base Line)과의 거리 오차, 방향 오차를 이용한 가중치를 부여한다. 이렇게 부여된 가중치 값들의 합을 이용하여 선분의 끊어지고 이어짐을 계산함으로써 선분을 추출할 수 있는 방법이다.

본 논문의 구성은 2장에서 점진적인 화소 확장에 의한 선분 추출 알고리즘에 대해 설명하고, 3장에서는 기존 방법들과의 실험결과를 비교하였으며, 4장에서 결론 및 향후 과제를 제시하였다.

2. 점진적인 화소 확장에 의한 선분 추출

본 논문에서는 영상에 존재하는 선분을 점진적인 화소 확장 방법에 의해 추출하는 방법을 제안한다. 그림 1은 본 논문에서 제안하는 점진적인 화소 확장에 의한 선분 추출 알고리즘의 전체 흐름도이다.

영상이 입력되면 먼저 선분의 기본이 되는 캐니 테두리 영상[10]과 방향 영상을 생성한다. 제안된 알

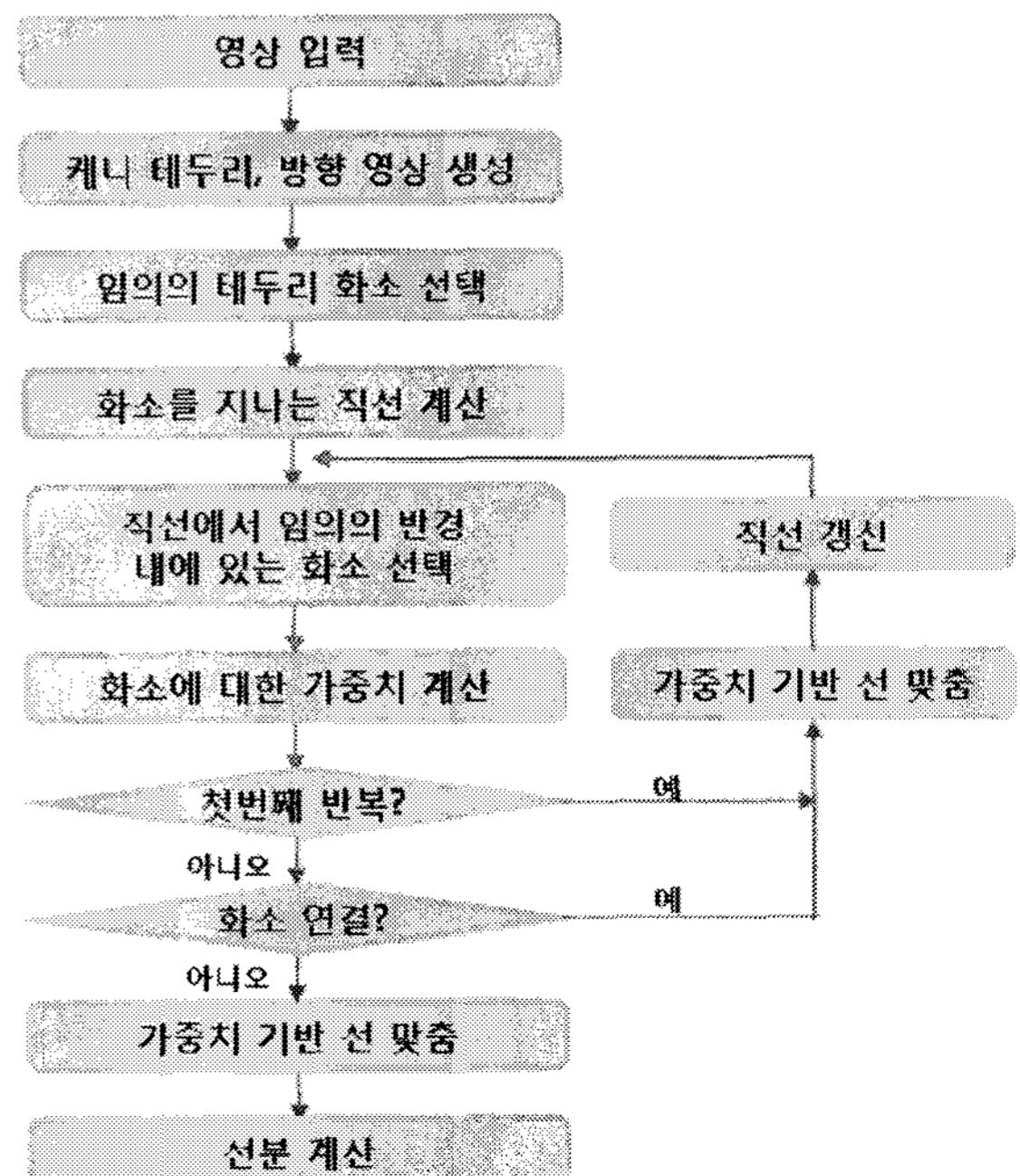


그림 1. 점진적인 화소 확장에 의한 선분 추출 흐름도

고리즘은 선분 추출 단계에서 각 테두리 화소에 가중치를 부여하게 되는데, 가중치 부여 시, 그 화소의 방향 값이 필요하기 때문에 계산의 효율성을 증가시키기 위해 방향 영상을 미리 계산해 놓는다. 캐니 테두리 영상과 방향 영상을 계산한 후, 캐니 테두리 화소 중 임의의 화소를 선택하고, 선택된 화소를 지나면서 그 화소의 방향 성분과 기울기가 같은 직선을 계산하여 직선을 초기화하게 된다.

직선을 초기화 한 후 직선에서 임의의 반경 내에 있는 화소들을 선택하고 각 화소에 대해 본 논문에서 제안하는 가중치 부여 방법을 통해 가중치를 부여한다. 부여된 가중치를 기반으로 가중치 기반 선 맞춤을 수행하여 초기화 한 직선을 갱신한다. 갱신된 직선을 따라 반경을 점진적으로 확장시켜가며 화소들을 선택하게 되고, 선택된 화소들의 가중치 비교에 의해 화소들의 연결 여부를 판별하여 선분을 추출한다.

2.1 방향 영상 생성

본 절에서는 선분 추출 전처리 단계로 방향 영상을 생성하는 방법에 대해 설명한다. 일단 영상이 입력되면 영상의 캐니 테두리를 계산하고, 캐니 테두리 영상에서 각 테두리 화소들에 대해 방향 영상을 계산한다.

방향 영상은 선택된 화소에 대응하는 방향 벡터를 쉽게 구하기 위한 룩업 테이블과 같은 개념이다. 즉, 각 테두리 화소 위치에 하나의 각도 값을 부여한 원본 영상과 같은 크기의 행렬이다. 이러한 방향 영상을 생성하기 위해서는 화소의 기울기를 계산해야 하며, 기울기를 계산하기 위해 가우시안 함수를 사용한다. 가우시안 함수를 x 축, y 축으로 컨벌루션하여 x 축, y 축에 대한 기울기 영상을 생성한다. x 축 기울기 영상을 $dxIm$ 라고 하고, y 축 기울기 영상을 $dyIm$ 라고 하면, (i, j) 번째의 화소의 방향 값은 식 (1)과 같이 나타낼 수 있다.

$$dirIm(i, j) = \tan^{-1} \left(\frac{dyIm(i, j)}{dxIm(i, j)} \right) \quad (1)$$

그림 2(a)는 본 논문에서 제안하는 알고리즘을 설명하기 위해 사용된 원본 영상이며, 그림 2(b)는 본 논문에서 사용하는 방향 값을 이해하기 위해 방향 값을 영상에 중첩시킨 그림이다. 방향 값은 식 (1)에 의해 각 테두리 화소에 각도 값으로 부여가 되며, 그

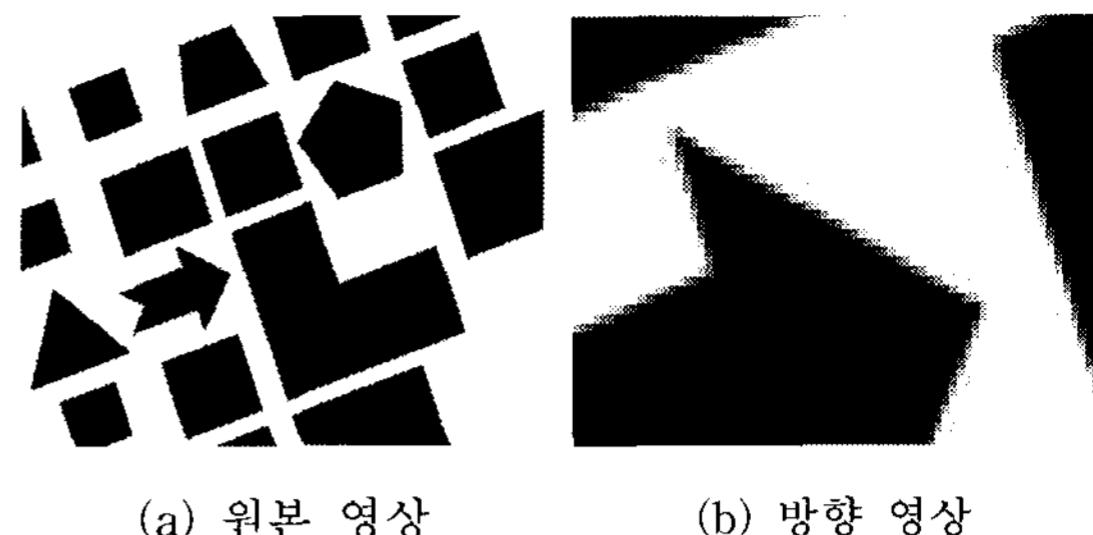


그림 2. 원본 영상과 방향 영상

림 2(b)의 화살표 머리 위 부분을 이루는 화소의 방향 값은 약 -45도 정도의 값이 된다.

2.2 화소에 가중치 부여 방법

그림 2(b)와 같이 계산된 방향 영상의 각도를 θ 라고 할 경우, 테두리의 방향 벡터(\vec{d})는 다음 식 (2)와 같이 나타낼 수 있다.

$$\vec{d} = (\sin(\theta), -\cos(\theta)) \quad (2)$$

여기서, 테두리 방향 벡터의 크기는 1인 단위 방향 벡터이다.

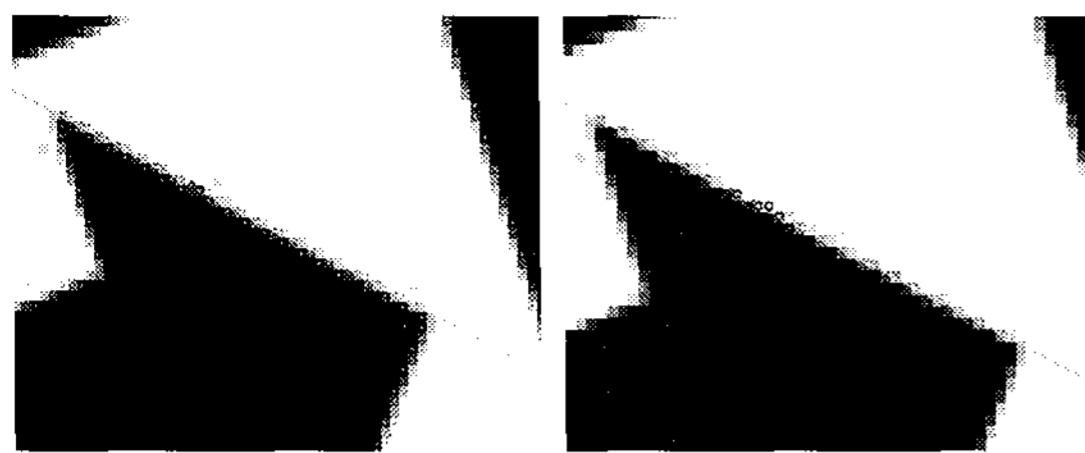
선분을 추출하기 위해 임의의 테두리 화소를 선택하게 되며, 현재 선택된 화소를 $p1$ 이라고 하면, 선택된 화소에 테두리 방향 벡터(\vec{d})를 더한 $p2$ 를 계산할 수 있고, $p1$ 과 $p2$ 를 지나는 직선의 식을 다음 식 (3)과 같이 표현할 수 있다.

$$\vec{n} \cdot \vec{x} + c = 0 \quad (3)$$

여기서, \vec{n} 은 현재 화소의 테두리 방향 벡터(\vec{d})의 법선(Normal) 벡터로써 $(\cos(\theta), \sin(\theta))$ 로 나타낼 수 있으며, \vec{x} 는 현재 선택된 화소($p1$)의 좌표 값이다. c 는 임의의 상수로써 $p1$ 과 $p2$ 를 대입하여 구할 수 있다.

그림 3(a)에서 현재 선택된 화소 $p1$ (○)과 테두리 방향 벡터를 더한 $p2$ (◇), Canny 테두리 화소(•) 그리고 식 (3)과 같이 표현되는 직선을 나타내었다. 이 과정은 선분을 구하기 위한 초기 단계로써, 현재 구하고자 하는 선분의 기본 직선을 구하는 단계이다.

점진적인 화소 확장을 위해 $p1$ 과 $p2$ 의 중점을 잡고, 중점으로부터 임의의 반경 내에 있는 선분 후보 화소들(\vec{x}_{in})을 선택한다. 선택된 화소들은 그림 3(b)에 ○로 나타내었다. 임의의 반경 내에 있는 화소들



(a) 직선의 초기화 (b) 초기 선분 후보 화소들

그림 3. 직선의 초기화 및 후보 화소 선택

에 가중치를 부여하기 위해 다음과 같은 식 (4)와 식 (5)를 사용하였다.

$$err = \vec{n} \cdot \vec{x}_c + c \quad (4)$$

$$errOrient = \vec{n} \cdot \vec{d} \quad (5)$$

여기서, 식 (4)는 현재 선택된 화소 p_1 과 기본 직선과의 거리 오차를 나타내며, \vec{n} 과 c 는 식 (3)에서 구한 값이다. 그리고 식 (5)는 현재 선택된 화소 p_1 과 기본 직선의 방향각 오차를 나타내며, \vec{d} 는 식 (2)에서 구한 선택된 화소 p_1 의 테두리 방향 벡터이다.

그림 3(b)와 같이 선택된 선분 후보 화소에 대해 식 (6)과 같이 오차의 합을 정의하고, 각 후보 화소에 식 (7)과 같은 가중치(L)을 부여하여 이 화소가 직선과 어느 정도 밀접한 관계가 있는지 나타낸다.

$$errPixel = err^2 + 10 \cdot errOrient^2 \quad (6)$$

$$L = \exp(-errPixel) \quad (7)$$

여기서, 가중치를 부여하기 위해 \exp 을 사용하였으며, \exp 안의 값에 - 부호를 붙여 가중치 값이 0에서 1사이의 값이 되도록 하였다. 식 (6)은 직선과의 거리 보다는 그 선분을 이루는 화소의 방향 오차에 가중치를 두어 실제 같은 선분이 아니지만, 화소 확장에 의해 선택된 화소를 효과적으로 제거할 수 있도록 하였다.

선택된 각 화소에 대해 L 을 구하고, 선택된 후보 화소들 중 $errPixel$ 의 값이 일정 값 이상이 될 경우, 현재 구하고자 하는 선분이 아닌 것으로 간주할 수 있으며, 그 화소의 가중치를 0으로 한다. 본 논문에서는 $errPixel$ 의 값이 2보다 클 경우, 가중치를 0으로 하였다.

현재 선택된 화소들의 가중치 합을 $sumW$ 라 하고, 선분 추출을 위한 첫 번째 반복인지 여부를 검사한

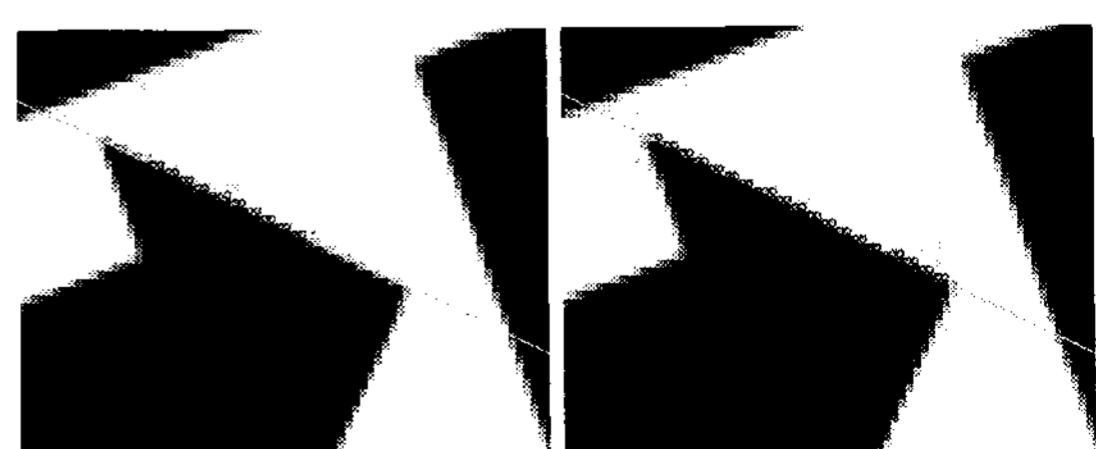
다. 만약 첫 번째 반복일 경우, 화소가 연결되는지의 여부를 결정할 필요가 없기 때문에 현재 가중치가 부여된(가중치가 0이 아닌) 선택된 화소들을 이용하여 가중치 기반 최소 자승법을 이용한 가중치 기반 선 맞춤(Weighted Line Fitting)을 수행한다. 가중치 기반 선 맞춤을 통해 새로운 직선을 계산하여 선분의 기본 직선을 생성한다.

2.3 점진적인 화소 확장법

본 절에서는 점진적으로 화소를 확장하는 방법에 대해 설명한다. 전 반복에서 선택된 화소들의 중심점을 계산 한 후, 화소 확장을 위한 경계선을 만들기 위해 다음과 같은 변수를 사용하였다. 새로 계산된 직선의 법선 벡터 $\overrightarrow{n_{\neq w}}$ 와 테두리 방향을 나타내는 방향 벡터 \vec{d} 를 이용하여, 직선의 법선 방향으로 화소 한 개의 거리만큼 떨어지고, 계산된 중심점에서 직선 방향으로 앞 서 정한 임의의 반경의 2배만큼 떨어진 경계선을 만들어, 경계선이 이루는 경계 내에 있는 화소들을 확장하여 선택하게 된다. 다음 그림 4(a)는 첫 번째 반복 시에 선택되었던 화소들과 화소 확장을 위한 경계선의 네 모서리 점(•)을 나타내었으며, 경계선 내에 있는 선택된 화소들도 ○로 표시하였다.

2.4 가중치 비교에 의한 선분 추출 방법

화소 확장에 의해 선택된 화소들에 대해 식 (7)을 이용하여 가중치를 계산하고, 새로 선택된 가중치의 합 $sumW$ 를 구한다. 본 논문에서는 전 반복에서 구했던 가중치의 합 $sumW$ 를 $prevSumW$ 라고 하여, 현재 선택된 화소들의 가중치 합과 비교하여 선분을 추출한다. 만약 두 가중치의 합이 같다면 더 이상 추



(a) 화소 확장에 의해 추가 선분을 이루는 화소들 (b) 가중치 부여된 선분을 이루는 화소들

그림 4. 원본 영상과 방향 영상

가된 화소들이 없거나, 선분과 관계없는 가중치가 0인 화소들만 추가가 되었다는 것을 의미하며, 더 이상 연결된 테두리 화소들이 없다고 가정할 수 있다.

이렇듯, 전 반복에서의 가중치 합과 현재 반복에서의 가중치 합의 변화가 없다면, 선분은 끊어진 것으로 간주할 수 있으며, 가중치 합의 변화가 있다면 선분은 계속 이어지는 것으로 간주할 수 있다. 이렇게 선택된 화소들을 이용하여 가중치 기반 선 맞춤을 수행하고, 선분을 계산하면 선분을 추출할 수 있는 것이다.

그림 4(b)는 선분을 추출하기 바로 전 최종적으로 계산한 확장된 화소들에 가중치가 부여된 결과를 보여준다. ○표시는 화소 확장에 의해 선택된 화소들 중 식 (6)의 오차 값이 2이하가 되어 가중치가 부여된 것을 나타내며, △표시는 화소 확장에 의해 선택은 되었지만, 식 (6)의 오차 값이 2 이상이 되어 가중치가 0으로 부여된 경우이다. 그림 4(b)의 다음 반복에서 더 많은 테두리 화소가 선택되었지만, 추가된 화소들의 가중치가 식 (6)에 의해 모두 0이 되어 가중치 합의 변화가 없었고, 결과적으로 선분은 현재 선택된 화소를 이용하여 추출될 수 있다.

3. 실험결과

이번 장에서는 본 논문에서 제안하는 알고리즘의 선분 추출 과정을 살펴보고, 다양한 실험 영상에서 제안하는 알고리즘과 기존에 제안되었던 대표적인 알고리즘인 Nacken[7]과 J.H. Jang[8]의 결과를 비교한다.

그림 5는 그림 8(e)에서 사용되었던 영상으로 잡음이 포함된 일반 영상에서의 선분 추출 과정을 나타낸 것이다. 그림 5(a)는 실험 영상 일부분이며, 방향 영상과 테두리 화소들(•)을 나타내었다. ○표시로 나타낸 부분이 선택된 임의의 화소 p1이고 ◇ 표시는 p1에 단위 방향 성분 벡터를 더한 p2이다. 직선은 p1과 p2를 지나는 초기화된 직선이며, 본 영상은 알고리즘 설명 영상과 달리 초기화된 직선이 구하고자 하는 선분과 많이 어긋나는 모습을 볼 수 있다. 이는 그림 5에서 사용된 영상은 잡음이 포함된 영상이며, 잡음과 양자화 오차 등으로 인해 화소의 방향 값이 영상의 테두리 방향 값과 다른 경우이다. 그림 5(b)는 제안된 알고리즘의 첫 번째 반복 결과로써, 선택된 화소에서 반경이 3 이내인 화소들을 선택하여 식 (7)

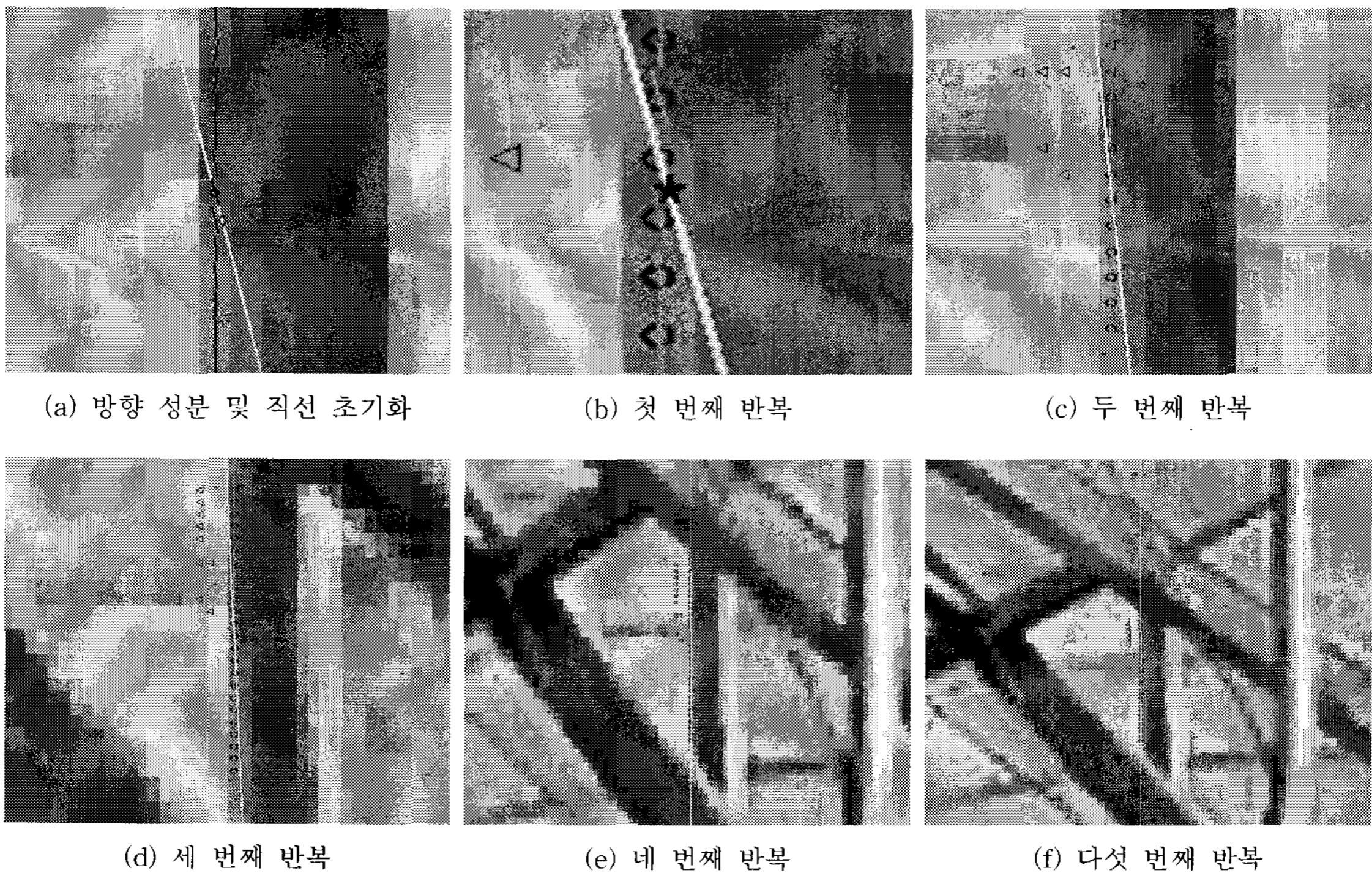


그림 5. 제안하는 알고리즘의 선분 추출 과정

에 의해 가중치가 부여된 결과를 나타낸 것이다. Δ 은 반경 내에 있어 선택이 되었지만 식 (6)에 의해 가중치 값이 0으로 부여된 화소이며, \circ 는 가중치가 식 (7)의 L 로 부여된 화소를 나타낸다. 가중치 값은 그림으로 표현할 수 없어 표현을 하지 못했지만 각 화소마다 0에서 1사이의 값이 부여되었다. 가중치가 부여된 화소들의 중심을 구하여 * 표시로 나타내었으며, \circ 로 표시된 각 화소마다 화소의 중심을 지나는 희미한 짙은 선들은 그 화소에 대한 방향 값을 나타낸 것이다. 그림 5(c)는 두 번째 반복 결과로써, 직선을 따라 화소의 선택 범위를 확장시켰으며, 부여된 가중치 기반 선 맞춤을 통해 직선을 개신한 결과를 보여준다.

그림 5(d)는 세 번째 반복 결과이며, 직선이 가중치가 부여된 화소들에 점점 가까워짐을 볼 수 있다. 그림 5(e)에서는 네 번째 반복에서 구하고자 하는 선분에 일치하는 직선을 구한 결과를 볼 수 있다. 그림 5(f)에서 화소들은 화소 확장에 의해 그림 5(e)보다

많이 선택되었지만, 추가된 화소의 가중치가 모두 0이 되었다. 다섯 번째 반복 결과, $sumW$ 의 변화가 없어 선분 추출 과정이 종료되었고, 선분을 이루는 화소들과 일치하는 최종 선택된 화소들을 나타낸다.

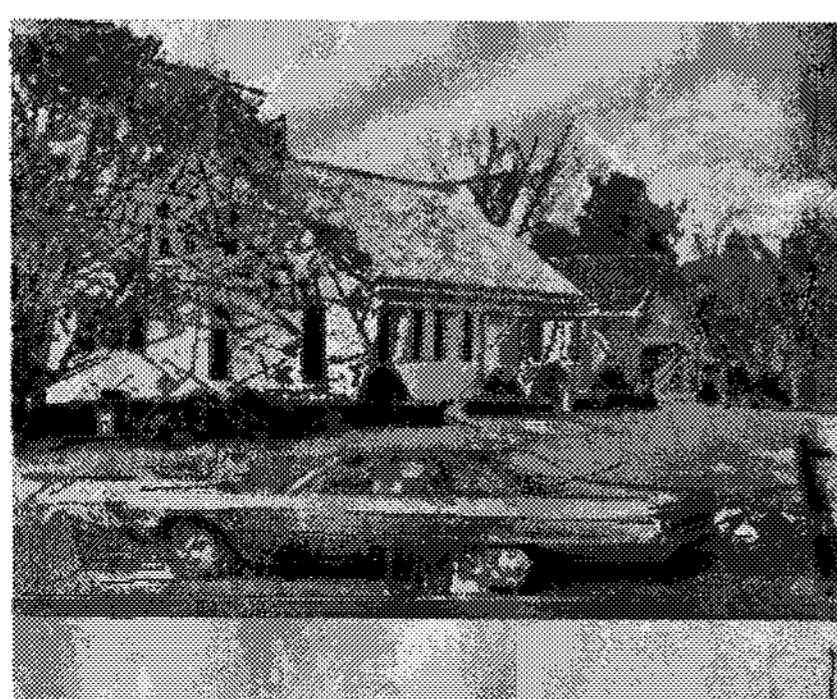
그림 6은 House 영상에 대한 선분 추출 결과이다. 그림 6(a)는 원본 영상이며, 그림 6(b)는 본 논문에서 제안하는 점진적인 화소 확장에 의한 선분 추출 결과이다. 그림 6(c)는 Nacken이 제안한 선분 추출 방법의 결과이며, 그림 6(d)는 J.H. Jang이 제안한 선분 추출 방법의 결과이다. 그림 6(c)와 (d)는 테두리 지도를 기본으로 기초 선분들을 추출한 후, 이들을 그룹화 함으로써 선분을 찾는 방법이다. 기초 선분을 그룹화 하기 때문에 실제 선분이 아니지만 비슷한 성질을 갖는 기초 선분들이 하나의 선분으로 간주되어 그룹화 되는 것을 볼 수 있다. 그림 6(c)와 (d)의 방법은 직선 성분들을 잘 찾을 수 있는 반면에 나무 숲과 같이 직선이 아닌 것까지 직선으로 간주하는 경우가 많기 때문에 잡음에 많은 영향을 받으며, 특



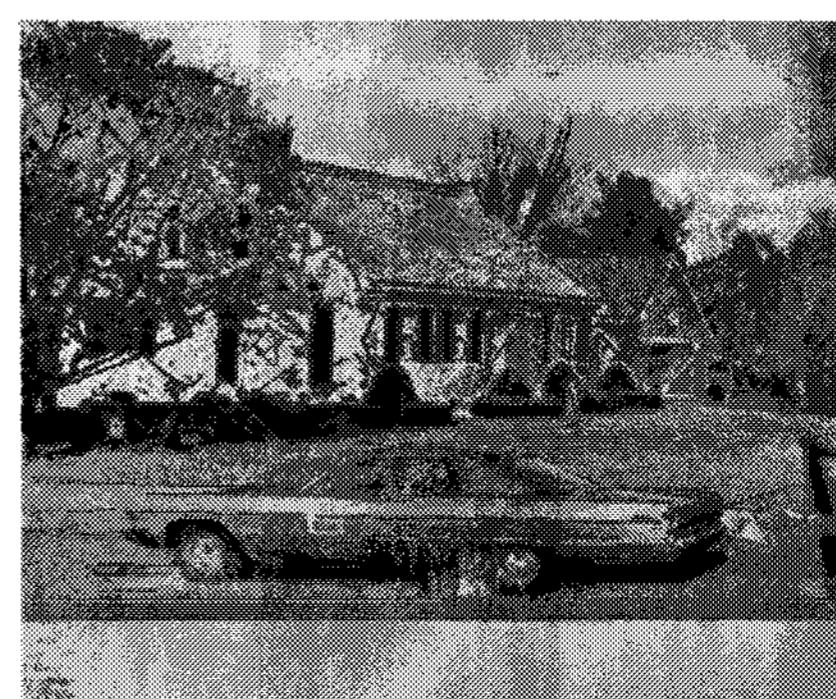
(a) 원본 영상



(b) 제안된 알고리즘



(c) Nacken의 방법



(d) J.H. Jang의 방법

그림 6. House 영상 선분 추출 결과

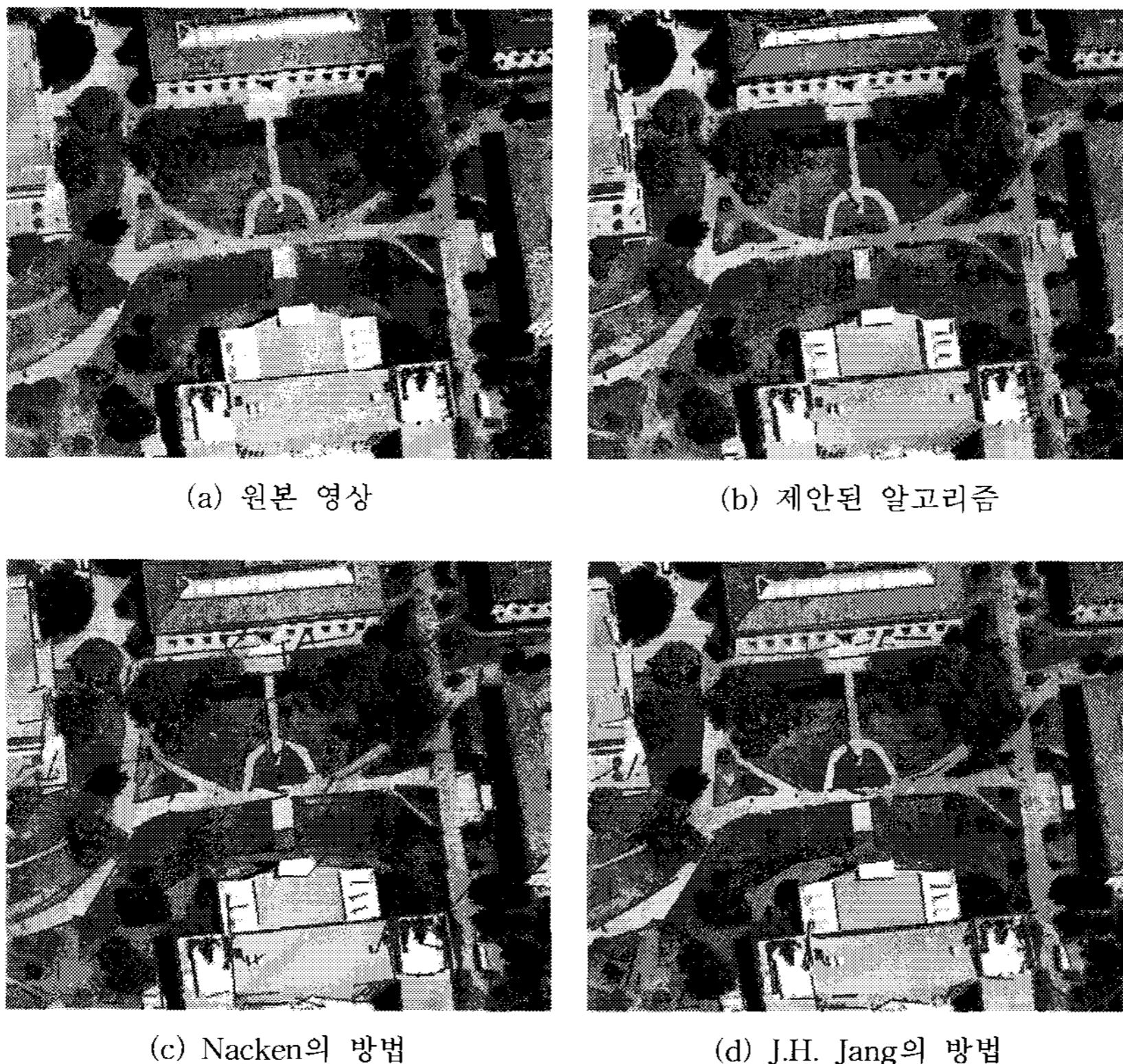


그림 7. UC Berkley 캠퍼스 영상 선분 추출 영상

히 본 논문과 같이 연속된 영상에서 선분을 추출하여 정합한 결과를 사용하는 분야에는 적용이 힘들다.

그림 7은 UC Berkley 캠퍼스 영상에 대한 각 알고리즘의 선분 추출 결과이며, 그림 6의 설명과 같다. 본 논문에서 비교하는 방법들은 영상 중간부분의 곡선 길과 같은 부분까지도 직선 선분으로 추출하는 경우가 있지만 본 논문에서 제안하는 방법은 곡선을 직선으로 간주하여 잘못된 직선 선분을 추출하는 경우는 거의 없다. 또한 표 1과 표 2에서는 제안된 알고리즘이 다른 알고리즘에 비해 선분을 빠르게 추출할 수 있음을 보여준다.

그림 8은 여러 영상에 대한 선분 추출 결과를 보여준다. 그림 8(a)는 임의로 다각형을 그린 후 영상을 만들어 그 영상에 대한 선분을 추출한 결과이다. 그림 8(b)는 그림 8(a)의 선분 추출 결과 영상으로써, 영상에 잡음이 첨가되지 않았기 때문에 매우 깨끗한 결과를 보여준다.

그림 8(c)는 한국전자통신연구원의 한 연구동을 촬영한 사진이다. 주로 직선 성분이 많은 영상으로

표 1. House 영상에 대한 선분 추출 결과

방법	제안된 방법	Nacken	J.H. Jang
추출된 선분 개수	256	324	343
계산 시간 (sec)	2.2	1414.8	3.4

표 2. UC Berkley 캠퍼스 영상에 대한 선분 추출 결과

방법	제안된 방법	Nacken	J.H. Jang
추출된 선분 개수	231	442	460
계산 시간 (sec)	2.1	2895.4	5.3

카메라 자기 보정 등에 유용하게 사용될 수 있으며, 선분 추출 결과는 그림 8(d)와 같다. 한눈에 봐도 어떤 영상인지 알 수 있을 정도로 선분이 안정적이며 깨끗이 추출된 것을 볼 수 있다. 그림 8(e)는 라이트 형제의 비행기의 날개 부분을 촬영한 영상으로 직선과 곡선이 존재하는 영상이며, 선분 추출 결과는 그림 8(f)와 같다.

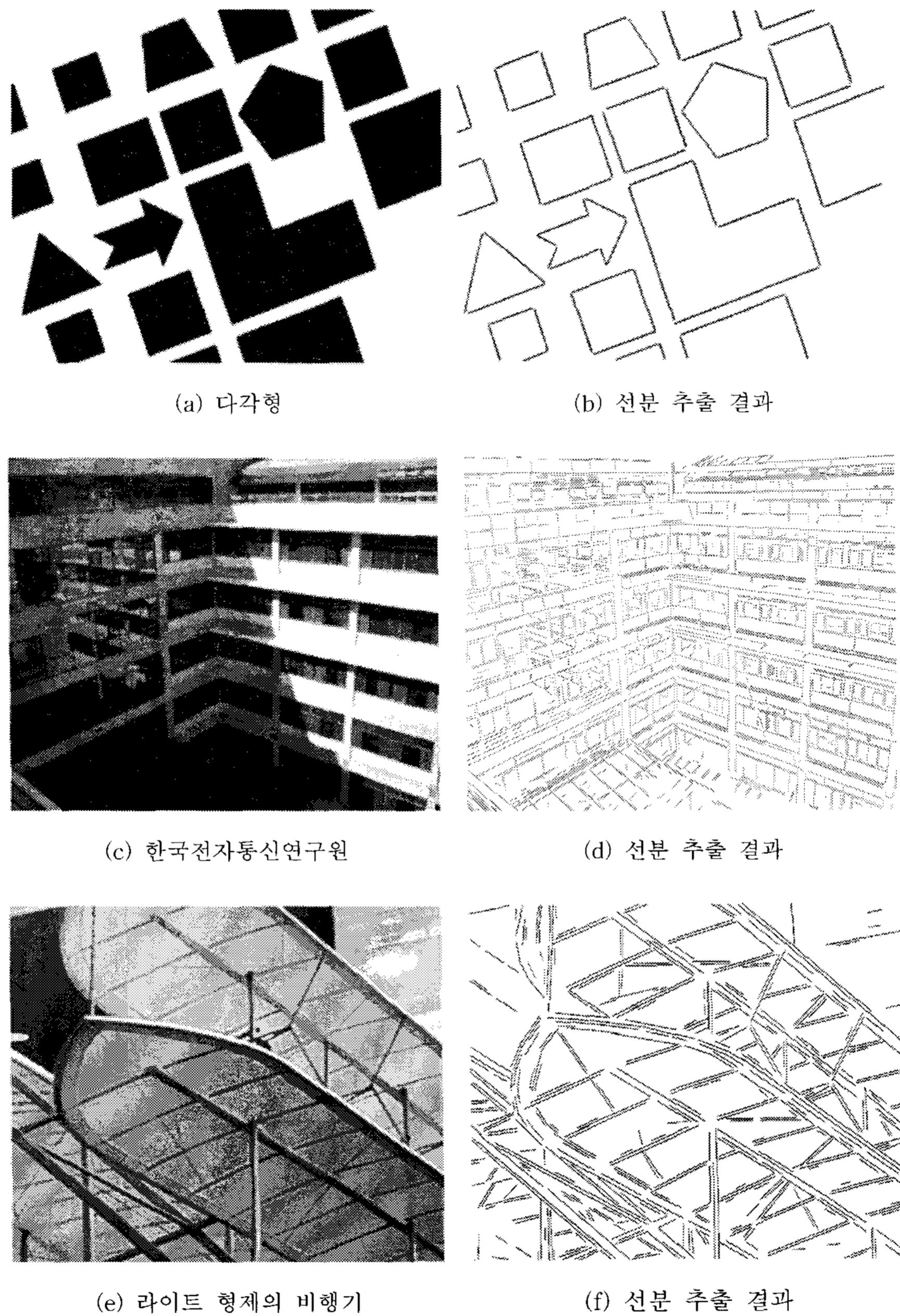


그림 8. 제안된 알고리즘 선분 추출 결과

5. 결 론

본 논문에서는 일반 영상에서 점진적인 화소 확장에 의한 선분 추출 방법을 제안하였다. 입력 영상으로부터 케니 테두리 영상과 방향 영상을 생성한 후, 테두리 화소 중 임의의 화소를 선택하여 그 화소를

중심으로 선분을 이루는 화소를 점진적으로 찾아가는 방법이다. 선택된 화소를 중심으로 본 논문에서 제안된 거리와 방향 오차를 이용한 가중치 부여 방법에 의하여 찾고자 하는 선분을 이루는 화소를 찾을 수 있다. 또한 가중치 합 비교에 의해 선분의 이어지고 끊어짐 여부를 판단하여 영상 내에 존재하는 선분

을 안정되고 빠르게 추출할 수 있었다. 그림 6과 그림 7에서 보는 바와 같이 영상의 크기에 따른 양자화 오차 등에 의해 간혹 곡선을 직선으로 간주하기도 하나, 이렇게 잘못 찾아진 선분은 비교적 길이가 짧기 때문에 선분 길이에 제약 조건을 주면 제거하기 쉽다. 잘못 찾아진 선분을 제거하게 되면, 불필요한 선분이 거의 없기 때문에 카메라 트래킹 등 연속된 영상의 선분 정합 시 유용하게 사용될 수 있는 알고리즘이다. 향후에는 본 알고리즘을 이용해 추출된 결과를 파라미터 공간에서 재분석하여 잘못된 선분을 제거하는 방법에 대해 연구하여, 카메라 트래킹 등 선분을 특징으로 사용하는 분야에 유용한 알고리즘 개발이 필요할 것이다.

참 고 문 헌

- [1] H. Bay, A. Ess, A. Neubeck, and L.V. Gool, "3D from line segments in two poorly-textured, uncalibrated images," *3D Data Processing, Visualization, and Transmission*, pp. 496–503, 2006.
- [2] R.O. Duda and P.E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, Vol.15, No.1, pp. 11–15, 1972.
- [3] N.Guil, J. Villaba, and E.L. Zapata, "A Fast Hough Transform for Segment Detection," *IEEE Trans. on Image Processing*, Vol.4, No. 11, pp. 1541–1548, 1995.
- [4] T.M. Breuel, "Finding Lines under Bounder Error," *Pattern Recognition*, Vol.29, No.1, pp. 167–178, 1996.
- [5] P.C. Chung, C.T. Tsai, E.L. Chen, and Y.N. Sun, "Polygonal Approximation using Competitive Hopfield Neural Network," *Pattern Recognition*, Vol.27, No.11, pp. 1505–1512, 1994.
- [6] J.X. Yuan and C.Y. Suen, "An Optimal O(n) Algorithm for Identifying Line Segments from a Sequence of Chain Codes," *Pattern Recognition*, Vol.28, No.5, pp. 635–646, 1995.

- [7] P.F.M. Nacken, "A metric for line segments," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol.15, No.12, pp. 1312–1318, 1993.
- [8] J.S. Jang and K.S. Hong, "Fast line segment grouping method for finding globally more favorable line segments," *Pattern Recognition*, Vol.35, No.10, pp. 2235–2247, 2002.
- [9] A. Chalechale, G. Naghdy, P. Premaratne, and H. Moghaddasi, "Chain-based extraction of line segments to describe images," *IEEE Int. Conf. on Multimedia and Expo*, pp. 355–358, 2004.
- [10] E. Trucco and A. Verri, *Introductory Techniques for 3D Computer Vision*, Prentice Hall, Inc. 1998.



이재광

- | | |
|-------------------------------------|------------------------|
| 2003년 | 홍익대학교 전자전기공학부 졸업(공학사) |
| 2005년 | 인하대학교 전기공학과 졸업(공학석사) |
| 2005년 ~ 현재 | 과학기술연합대학원대학교 박사과정 재학 중 |
| 2005년 ~ 현재 한국전자통신연구원 UST 연구연수생 | |
| 관심분야: 영상처리, 컴퓨터 비전, 3차원 복원, 카메라 트래킹 | |



박창준

- | | |
|--------------------------------------|----------------------|
| 1994년 | 경북대학교 전자공학과 졸업(공학사) |
| 1996년 | 경북대학교 전자공학과 졸업(공학석사) |
| 2000년 | 경북대학교 전자공학과 졸업(공학박사) |
| 1998년 ~ 현재 | 한국전자통신연구원 재직 |
| 2005년 ~ 현재 과학기술연합대학원대학교 겸임교수 | |
| 현재 한국전자통신연구원 HD게임연구팀 팀장 | |
| 관심분야: 컴퓨터 비전, 3차원 복원, 카메라 트래킹, 게임 엔진 | |