

Cross Mask와 에지 정보를 사용한 동영상 분할

(Dynamic Scene Segmentation Algorithm Using a Cross Mask and Edge Information)

姜 政 錫*, 朴 來 弘**, 李 商 郁***

(Jung Suk Kang, Rae Hong Park, and Sang Uk Lee)

要 約

본 논문에서는 cross mask와 에지를 이용한 동영상 분할 알고리즘을 제안하였다. 이 방법은 기존의 feature-based 방법과 pixel-based 방법을 조합한 것으로 에지를 특징으로 사용하여 각 에지 화소들의 이전 영상과의 유사성을 계산함으로써 움직인 물체를 검출해낸다. 이때 유사성을 계산하기 위해 cross mask를 사용하였다. 본 논문에서 제안한 방법은 기존의 방법과 비교할 때 계산이 간단하고, 배경이 복잡하거나 움직인 물체가 여러개 존재할 경우에도 적용이 가능하다. 또한 병진운동 뿐만 아니라 회전운동, 시선방향의 운동 등에도 좋은 성능을 보임을 확인하였다.

Abstract

In this paper, we propose the dynamic scene segmentation algorithm using a cross mask and edge information. This method, a combination of the conventional feature-based and pixel-based approaches, uses edges as features and determines moving pixels, with a cross mask centered on each edge pixel, by computing similarity measure between two consecutive image frames. With simple calculation the proposed method works well for images consisting of complex background or several moving objects. Also this method works satisfactorily in case of rotational motion, translational motion, and motion of optical axis.

*準會員, **正會員, 西江大學校 電子工學科
(Dept. of Elec. Eng., Sogang Univ.)

***正會員, 서울大學校 制御計測工學科
(Dept. of Control and Instrumentation Eng.,
Seoul Nat'l Univ.)

接受日字: 1989年 5月 31日

(※본 논문은 한국과학재단 목적기초 연구의 일부로 수행되었음.)

I. 서 론

동영상 분할(dynamic scene segmentation)이란 시변 영상으로부터 움직이는 물체를 분리해내는 것을 말하는데, 여러 응용분야에 적용될 수 있기 때문에 많은 연구가 진행되어 왔다. 시변 영상이란 움직이는 물체를 포함하는 영상을 말하는데 이러한 시변 영상으로부터 움직이는 물체를 분리하는 방법은 크게 feature-based 방법과 pixel-based 방법으로 나눌 수 있

다.¹¹⁾

Feature-based 방법은 관심있는 영역인 특징(feature) 영역에서 움직인 물체의 분리를 행하는 방법으로, 먼저 각 영상에서 특징을 추출한 후 이들의 대응 관계를 조사하여 대응되는 특징들의 속도를 구한다. 그 다음에 강제운동을 가정하는 경우 한 물체에 속하는 특징은 동일한 속도를 갖는다는 점을 이용하여 같은 속도를 갖는 특징들을 grouping 하여 움직이는 물체를 분리하는 것이다. 이때 특징은 물체의 꼭지점이나 모서리, 무게중심, 면적 등이 될 수 있다.¹²⁾¹³⁾

한편 pixel-based 방법은 영상의 각 화소들을 대상으로 하는 방법으로 시공간 경사 방법(spatio-temporal gradient method)과 template matching 방법, 차영상(difference picture)을 이용한 방법 등 여러가지 방법이 있다. 시공간 경사 방법은 움직인 물체에 속하는 화소가 이동함에 따라 일어나는 화소값의 시간적 변화율을 인접한 화소간의 공간적 변화율과 연관시킴으로써 각 화소들의 속도를 구하여 같은 속도를 갖는 화소들을 grouping 하므로써 움직인 물체를 분리해내는 방법이고^{14~16)} template matching 방법은 특정 형태를 갖는 틀(template)을 사용하여 영상 간의 유사성을 측정하므로써 유사성이 최대가 되는 위치를 찾아 움직인 물체를 분리해내는 방법이다.¹⁷⁾¹⁸⁾ 그리고 차영상을 이용한 방법은 두 영상의 차를 구하여 이를 이진영상화한 다음 연속된 영역사이의 관계를 이용하거나 차영상 내의 영역의 일부가 물체의 영역에 속해 있다는 사실을 근거로 하여 움직인 물체를 분리해내는 방법이다. 예로서 Jain 등은 연속된 차영상의 영역을 추적하여 움직이는 물체를 분리하였고¹⁹⁾ Jain 등은 차영상 영역의 형(type)을 결정하고 결정한 형에 따라 각기 다른 방식으로 region growing 하므로써 움직인 물체를 분리하였다.¹⁶⁾ 한편 Yalamanchili 등은 차영상의 영역으로 둘러싸인 영역으로부터 region growing과 여러가지 heuristic을 사용하여 움직이는 물체를 분리하였고¹³⁾ Jain은 차영상 영역의 경계를 해석하여 움직임 정보를 추출하였다.¹²⁾

위에서 언급한 방법들은 각각 단점을 가지고 있다. 예를 들면 feature-based 방법은 정영상에서 특징을 추출하는데 정영상이 복잡할 경우 특징의 추출이 어려우며 추출된 특징의 대응 관계를 구하는 것도 쉬운 문제가 아니다. 또한 시공간 경사 방법은 물체가 회전 운동을 하거나 움직임이 커지면 적용하기가 힘들다. 차영상을 이용한 방법 중에서 Jain 등의 방법¹⁹⁾ 여러 물체가 인접하여 움직이는 경우에는 적용이 불가능하다는 단점이 있으며 Yalamanchili 등의 region

growing 방법은¹³⁾ 물체의 내부가 균일한 경우에만 적용될 수 있고 물체가 자신의 한변의 방향으로 움직였을 경우에는 부가적으로 복잡한 과정이 필요하게 된다. 또한 Jain의 방법¹²⁾ 잡음이 많을 경우 경계 해석이 어렵다는 단점이 있다.

본 논문에서는 에지를 특징으로 사용하고 이 에지 화소들에 대해 십자 형태의 틀(cross mask)을 씌워 이전 영상과 현재 영상의 유사성을 계산하므로써 움직인 물체를 분리해내는 새로운 방법을 제안하였다. 제안한 방법은 크게 세부분으로 구성되어 있는데 움직인 물체 주위에 윈도우(window)를 씌우는 부분과 이 윈도우 내에서 에지와 cross mask를 사용하여 움직인 물체의 경계를 추출하는 부분, 추출된 경계에서 잘못 찾아진 부분을 제거하는 후처리 부분 등이다. 첫번째 부분은 연산시간을 줄이기 위해 제안한 것으로 이 부분을 분리하여 기존의 다른 방법들과 결합하여 사용하더라도 상당량의 계산을 감축할 수 있다. 후처리 과정에 해당하는 region filling 부분은 배경이 매우 복잡할 경우 이전 과정에서 잘못 구해진 경계를 제거하기 위해 제안된 것이다.

이 방법을 사용하여 동영상을 분할하기 위해서 다음과 같은 가정이 필요하다.

1. 두 연속된 영상 사이의 물체의 형태는 일정하다.
2. 물체의 운동에 급격한 변화가 없다.
3. 영상에서 움직이는 물체의 갯수에는 제한이 없으나 움직인 물체의 일부분이 다른 물체에 의해 가려져서는 안된다.

여기서 1,2 항목은 대부분의 영상 분할 알고리즘에서 전제하는 가정으로서 1 항목은 물체가 강제되어야 한다는 의미이고 2 항목은 물체가 갑자기 급격히 움직여 자기 자신의 길이보다 큰 변위 만큼 움직이는 것을 방지하기 위해서이다. 영상에서 프레임(frame) 사이의 시간 간격이 매우 작다는 것을 고려하면 실제 움직이는 물체는 강제로 근사시킬 수 있으나 물체가 매우 급격히 움직이면 형태도 변할 가능성이 있으므로 1,2와 같은 가정이 필요하게 된다. 3 항목은 물체가 다른 물체에 의해 가려지므로써 자신의 형태에 관한 정보를 부분적으로 잃게 되는 경우를 피하기 위한 가정이다. 물론 여러 프레임을 연속적으로 고려한다면 바로 이전 영상에서의 물체의 형태를 참조하므로써 겹쳐진 부분의 형태도 복원이 가능하지만 본 논문에서는 이전 영상과 현재 영상의 두 프레임만을 가지고 움직인 물체를 복원해내는 경우만을 연구 대상으로 하였으므로 그와 같은 가정을 세우게 되었다.

II 절에서는 움직인 물체의 주위에 윈도우를 씌우는 방법에 대해 언급하고, III 절에서는 움직인 물체의 경계를 추출하는 방법에 대해 설명하며 IV 절에서는 후처리 과정에 대해 설명한다. V 절에서는 제안한 방법의 성능을 보이기 위해 시뮬레이션 결과에 대해 언급하였으며 VI 절에서는 결론을 맺는다.

II. 가산투영 (projection)을 이용한 연산량 감축

동영상 분할에서의 최종적인 목적은 움직인 물체만을 분리해내는 것이다. 이때 찾고자 하는 움직인 물체는 영상의 일부분만을 차지하고 있으므로 움직인 물체 주위에 윈도우를 씌우므로써 해당 윈도우 내에서만 모든 과정을 실행한다면 대단한 연산량 감축을 얻을 수 있을 것이다. 이점을 해결하기 위해 본 논문에서는 가산투영 방법¹⁾을 이용하여 움직인 물체 주위에 윈도우를 씌우는 방법을 제안하였다. 그 방법은 다음과 같다.

먼저 현재 영상과 이전 영상에서 row 방향과 column 방향에 대해 각각 가산투영을 행한다. $M \times N$ 크기의 영상의 경우 가산투영은 다음과 같이 표현된다.

$$P_{c, \text{row}}(i) = \sum_{j=0}^{M-1} I_c(i, j), \quad i=0, 1, \dots, N-1$$

$$P_{c, \text{col}}(j) = \sum_{i=0}^{N-1} I_c(i, j), \quad j=0, 1, \dots, M-1$$
(1)

여기서 $P_{c, \text{row}}$ 는 현재 영상에서의 row 방향에 대한 가산투영이고 $P_{c, \text{col}}$ 는 column 방향에 대한 가산투영이다. 마찬가지로 이전 영상에 대한 가산투영도 $P_{p, \text{row}}, P_{p, \text{col}}$ 로 표현할 수 있다. 이와 같이 현재 및 이전 영상에 대해 각각 가산투영을 행한 후 다음과 같이 현재 영상의 가산투영과 이전 영상의 가산투영의 차의 절대값을 구한다.

$$D_{\text{row}}(i) = |P_{c, \text{row}}(i) - P_{p, \text{row}}(i)|, \quad i=0, 1, \dots, N-1$$

$$D_{\text{col}}(j) = |P_{c, \text{col}}(j) - P_{p, \text{col}}(j)|, \quad j=0, 1, \dots, M-1$$
(2)

여기서 D_{row} 는 row 방향에 대한 현재 영상과 이전 영상의 가산투영의 차의 절대값이고 D_{col} 은 column 방향에 대한 차의 절대값이다. 실제 영상에서 한개의 물체가 움직였을 경우의 D_{row} 와 D_{col} 의 예를 그림 1에 보였다. 그림 1에서 보는 바와 같이 피이크 (peak)가 존재하는 부분이 물체가 움직인 영역이 되며 나머지 부분은 배경영역이다. 이상적인 경우 배경영역은 차가 0이어야 하나 실제의 경우 0이 아닌 값을 갖는다. 그러므로 움직인 물체 영역의 올바른 추

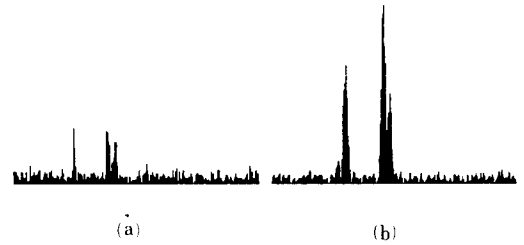


그림 1. 가산투영의 차

(a) row 방향 (b) column 방향

Fig. 1. Difference of projections.

(a) row direction. (b) column direction.

출을 위해서 임계값을 고려할 필요가 있다. $D_{\text{row}}(i)$ 에 대해 임계값을 설정하여 임계값을 넘는 위치들만을 고려한다면 이 영역이 바로 물체가 움직인 영역이 된다. 마찬가지로 $D_{\text{col}}(j)$ 에 대해서도 임계값을 설정하여 임계값보다 큰 값들만을 고려하면 물체가 움직인 영역을 얻을 수 있다. 이것은 가산투영의 차가 0이 아닌 값들을 갖는 부분은 물체가 움직이므로써 생긴 영역이라는 사실을 근거로 한 것이다. 이 움직인 영역들의 양끝의 위치를 구하면 바로 윈도우의 좌표가 된다. 즉, i 를 0에서부터 하나씩 증가시켜 나가면서 $D_{\text{row}}(i)$ 값을 임계값과 비교하여 만일 임계값보다 큰 $D_{\text{row}}(i)$ 값이 발견되면 이 지점은 물체가 움직인 영역이 시작되는 부분이며 이때의 i 값이 윈도우 상단의 좌표가 된다. 또한 i 를 $N-1$ 에서부터 하나씩 감소시키면서 $D_{\text{row}}(i)$ 값과 임계값을 비교하여 임계값보다 큰 부분이 나타나면 이 지점을 움직인 영역이 끝나는 부분으로 판단하여 그때의 i 값을 윈도우 하단의 좌표로 한다. 마찬가지로 과정을 $D_{\text{col}}(j)$ 에 대해 수행하면 윈도우 좌측 및 우측의 좌표를 구할 수 있다. 하지만 이 과정에서 $D_{\text{row}}(i)$ 와 $D_{\text{col}}(j)$ 에 대해 적절한 임계값을 일일이 정해주는 것도 문제가 되므로 본 논문에서는 이들 값을 자동적으로 설정해주는 방법을 고려하였다.

그림 1에서 보는 바와 같이 배경영역의 길이는 상대적으로 움직인 물체의 영역보다 길다. 따라서 D_{row} 와 D_{col} 에 대해서 각각 평균을 취하면 그 평균들은 배경영역의 영향이 더 크므로 배경영역의 값에 더 가까울 것이다. 그러므로 각각의 평균값을 $D_{\text{row}}, D_{\text{col}}$ 에 대한 각각의 임계값으로 사용하면 된다. 그러나 움직인 영역이 매우 작을 경우 $D_{\text{row}}, D_{\text{col}}$ 의 평균은 배경영역의 평균과 거의 같아지므로 이때는 배경영

역과 움직인 영역을 올바르게 구별할 수 없게 되어
서 결과적으로 윈도우의 크기가 매우 커지는 현상을
가져온다. 따라서 평균에다 어떤 보정항을 더해준 값
을 임계값으로 사용할 필요가 생기는데 본 연구에서
는 보정항으로 표준편차를 사용하였다. 실험에 의하
면 이 값을 임계값으로 사용하였을 때 배경 부분이
물체가 움직인 영역으로 잘못 추출되는 경우를 대부
분 제거할 수 있었으며, 잘못 추출된 경우라도 윈도
우의 크기가 매우 커지지는 않는 것을 확인하였다.

이와 같은 방법으로 설정된 윈도우를 사용할 경우,
실험에 의하면 배경과 물체 크기에 따라 다르지만 최
소한 1/2이상의 연산량을 감축할 수 있었다.

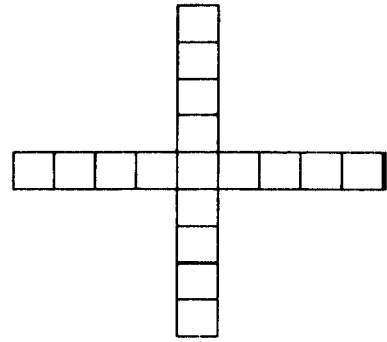


그림 2. Cross mask
Fig. 2. Cross mask.

Ⅲ. 움직인 물체의 경계추출

제안한 방법은 현재 영상에서 에지를 구하고 각 에
지 화소들에 대해 현재 영상과 이전 영상의 비유사
도 (dissimilarity)를 화소 단위로 계산한 후 이 값을
임계값과 비교하여 움직인 영역에 속하는가 여부를
판단하는 방법이다.

먼저 현재 영상에서 Prewitt 연산자를 사용하여 에
지를 구한다. Prewitt 연산자는¹⁴⁾ 덧셈과 뺄셈 연산
만으로 에지를 구할 수 있기 때문에 연산 속도가 매
우 빠르다는 장점이 있다. 구해진 에지 중에는 움
직인 물체의 경계가 포함되어 있으므로 에지로부터
움직인 물체의 경계를 추출하기 위해 다음과 같은
알고리즘을 사용한다.

```
IF current pixel ε edge point
THEN dissimilarity calculation
  IF dissimilarity > Th THEN the pixel ε moving object
  ELSE goto next pixel
ELSE goto next pixel
```

여기서 Th는 임계값으로서 잡음의 정도를 고려하
여 설정한다.

현재 영상과 이전 영상의 비유사도를 측정하기 위
해서 그림 2와 같은 cross mask를 사용한다.¹⁵⁾ [15]
에서는 BMA(block matching algorithm)의 변형으로
서 cross mask를 사용하여 현재 영상과 이전 영상의
상관 관계를 조사하므로써 물체의 속도를 구한 예가
있으나 본 논문에서는 각 화소에 대한 현재 영상과
이전 영상의 비유사도를 계산하고자 하는 에지 화소
에 띄우고 마스크 내의 각 화소에 대한 현재 영상과
이전 영상의 화소 밝기의 차를 구한 다음 절대값을
취한다. 이 절대값들의 합을 해당 화소의 비유사도
값으로한다.

$$\text{dissimilarity} = \sum_{\substack{x,y \\ \epsilon \text{ cross mask}}} |I_c(x,y) - I_p(x,y)| \quad (3)$$

여기서 I_c 는 현재 영상의 화소 밝기 값이고 I_p 는
이전 영상의 화소 밝기 값이다. 이렇게 구한 값이
cross mask의 중심화소에 대한 비유사도 값이 된다.
이처럼 한 화소의 비유사도를 구하는데 단지 해당 화
소만을 고려하지 않고 cross mask를 쓴 이유는 주변
화소의 영향까지 함께 고려하기 위해서 이다. 즉 물
체가 움직일 경우 여러 화소가 함께 움직이므로 주
변 화소의 영향을 고려하면 더욱 좋은 결과를 기대
할 수 있다. 따라서 cross mask를 쓰면 그림 3과 같
은 경우에도 물체를 검출해낼 수 있다. 그림 3과 같
은 경우는 물체가 자신의 한변의 방향과 평행하게 움
직인 경우로서, 한 화소 단위로 차를 구하여 절대값
을 취하는 방법을 쓰면 중앙의 부분(C영역)은 이전
및 현재 영상에서 비슷한 화소 밝기를 갖기 때문에
움직임 영역으로 분리될 해낼 수 없다. 그러나 cross
mask를 사용하면 지금 계산하고 있는 화소가 C영역
에 속해 있다고 하더라도 주변 화소에서의 영향이
함께 고려되기 때문에 움직임 영역으로 분류될 수
있다.

Cross mask의 크기는 움직인 물체의 크기에 알맞
게 설정하여야 하는데 cross mask의 크기가 움직인
물체의 크기보다 작을 경우에는 자신의 한변과 평행
하게 움직인 물체를 정확히 검출해낼 수 없으며, 반
면 물체의 크기보다 너무 커지면 연산시간만 증가할
뿐 성능에는 차이가 없다. 실제로는 움직인 물체의
정확한 크기를 알 수 없으므로 물체의 크기를 포함
할 수 있는 충분한 크기로 설정한다. 만일 물체가 자

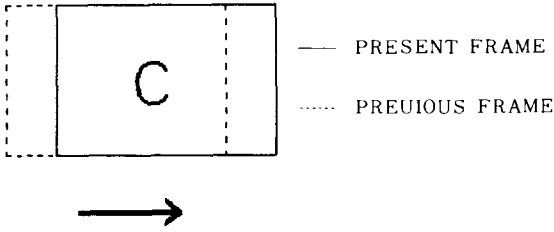


그림 3. 물체가 자신의 한변의 방향과 평행하게 움직인 예
 Fig. 3. Case of an object moving parallel to its side.

신의 한변과 평행하게 움직이는 경우가 없다면 그림 2와 같이 9개 화소의 크기를 갖는 마스크를 사용하여도 좋은 성능을 얻을 수 있다.

계산된 비유사도 값이 크면 클수록 이전 영상과 현재 영상이 차이점이 많다는 것으로서 이는 해당 화소가 현재 영상에서 움직인 물체에 속한다는 것을 의미한다. 그러나 움직인 물체 주위에 에지 성분이 있을 경우 이 부분에 대해 전술한 과정을 수행하면 주변의 화소의 영향도 함께 고려하는 cross mask의 특성 때문에 그 부분은 현재 영상에서 움직인 물체가 아닌데도 불구하고 검출이 된다. 이것은 해당 위치가 움직인 영역인가를 결정하는 척도로서 에지와 cross mask를 써서 구한 비유사도 값만을 사용하기 때문이다. 따라서 이 현상은 움직인 물체 주위에 에지 성분이 있으면 일어나게 되는 것으로 움직인 물체에 인접하여 다른 물체가 있을 경우에도 마찬가지로 현상을 볼 수 있다. 그러므로 이와같이 잘못 추출된 에지 성분들을 제거하여 움직인 물체의 경계만을 추출하는 과정이 필요하다. 이 방법을 다음 절에서 언급하겠다.

IV. 후처리 과정

전술한 바와 같이 cross mask를 사용하여 동영상 분할한 결과는 대부분의 영상에서 움직인 물체만을 잘 검출해내나 복잡한 영상의 경우, 예를 들면 배경의 밝기가 균일하지 않아서 에지 성분이 있다든지 혹은 움직인 물체와 인접하여 다른 정지 물체가 위치해 있는 경우는 움직인 물체의 경계뿐만 아니라 주변의 에지 성분도 함께 검출이 되는 단점이 있다. 따라서 cross mask를 사용하여 움직인 물체의 경계를 추출한 다음에 올바른 경계만을 얻기 위한 과정이

필요하게 된다.

본 논문에서는 후처리 과정으로서 적합한 region filling 방법을 제안하였다. 기존의 region growing 방법이 그레이 레벨(gray level)을 갖는 영상에서 평균과 표준편차를 이용하여 비슷한 밝기를 갖는 화소들을 grouping 하는데 비해 이 방법은 이전영상에 대해 폐곡선을 이루는 경계만을 추출하기 위한 것으로서 region filling에 해당된다. 이때 이전영상은 cross mask와 에지를 사용한 동영상 분할의 결과이다. cross mask와 에지를 사용한 동영상 분할 결과는 움직인 물체의 완전한 경계를 포함하고 있고 그 밖에 인접한 부분의 에지 성분도 포함하고 있다. 이때 움직인 물체 주변의 잘못 검출된 에지 성분들은 움직인 물체에 비해 크기도 작으며 완전한 형태를 이루지 못하고 대부분 불규칙한 직선 혹은 곡선으로 이루어져 있다. 따라서 cross mask를 사용한 결과로 부터 일정크기 이상의 폐곡선만을 추출하면 움직인 물체만을 검출해낼 수 있다.

그림 4와 같은 물체의 에지를 고려하자. 그림에서 볼 수 있는 바와 같이 에지 추출을 위해 사용한 Prewitt 연산자의 특성상 에지는 두껍게 검출된다. 그러므로 이전영상에서 한 라인(line)에 움직인 물체의 에지가 존재할 때 바로 아래 라인에 대해 생각해 보면 현재 라인에서의 에지 점들의 양끝의 위치보다 좌우로 각각 한 화소씩 떨어진 화소들의 내부에는 반드시 에지 점들이 존재하게 되며 이 에지 점들의 내부는 움직인 물체의 내부가 된다. 이런 성질을 이용하면 폐곡선으로 둘러싸인 움직인 물체의 경계만을 추출할 수가 있다. 알고리즘은 다음과 같다.

먼저 이전영상에서 raster scan 방향으로 각 라인에 대해 scan하면서 임계값 이하의 갯수를 갖는 에지 점들은 잡음으로 판단하여 이전영상으로부터 제

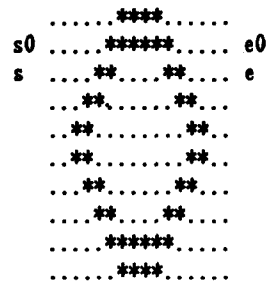


그림 4. 움직인 물체의 에지
 Fig. 4. Edge of a moving object.

거하고 임계값 이상인 에지 점들을 발견하면 그 에지 점들의 양끝점을 각각 s_0, e_0 라고 한다. 다음에 s_0 로부터 왼쪽으로 1 화소, e_0 로부터 오른쪽으로 1 화소 만큼 떨어진 위치의 다음 라인에서의 화소를 각각 s, e 라 한다. 이들의 위치를 그림 4에 표시하였다. 다음 과정으로서 s 에서 오른쪽으로, e 에서 왼쪽으로 각각 scan해 나가면서 첫 에지 성분이 나타나면 해당되는 화소를 각각 s_1, e_1 이라 하고 $e_1 > s_1$ 이면 s_1 부터 e_1 까지를 "1" 값으로 채운다. 이 과정을 마지막 라인까지 행하면 직선 성분을 갖는 잡음 에지는 제거되고 움직인 물체만을 얻을 수 있다. 이때 각 라인의 s, e 화소는 움직인 물체의 에지에는 포함되지만 실제 물체의 영역에는 속하지 않는다. 이것은 Prewitt 연산자를 사용하여 얻은 에지의 특성에 기인하는 것으로, 에지가 두껍게 추출되어 물체가 아닌 부분도 포함하고 있기 때문이다. 따라서 각 라인의 s, e 화소는 움직인 물체에 포함시키지 않으므로써 실제 움직인 물체의 크기보다 더 크게 검출되는 현상을 방지할 수 있다.

그러나 만일 그림 5(a)와 같이 아래로 뿔려있는 형태의 잡음 에지가 존재한다면 위에서 설명한 방법만으로는 그림 5(b)와 같은 결과를 얻게된다. 이 에지는 잘못 구해진 에지인데도 불구하고 마치 물체인 것처럼 인식된 이유는 region filling과정을 raster scan 방향으로 행하기 때문에 아래부분이 뿔려 있다는 것과는 관계없이 위 라인부터 순차적으로 채워지기 때문이다. 따라서 regionfilling 과정을 이진영상의 아래쪽에서 위로 행한다면 그림 5(a)의 형태는 처리방향에 대해서 위쪽이 뿔려 있는 것이 되므로 이 부분을 제거할 수 있다. 마찬가지로 이진영상의 좌측으로부터 또한 우측으로 부터도 이 과정을 수행하여 총 4개 방향에 대한 region filling 결과 중에서 공통된 영역만을 추출하면 움직인 물체만을 얻을 수 있다. 이 방법은 이진영상에서 행하기 때문에 실행속도가 매우 빠르다.

그림 6에 제안한 방법의 순서도를 보였다.

V. 시뮬레이션 결과

본 절에서는 제안한 동영상 분할 방법에 대한 시뮬레이션 결과를 보인다. 시뮬레이션은 IBM/AT를 사용하여 행하였으며 사용한 영상은 256그레이 레벨을 갖는 256×256 크기의 실제영상으로서 영상 취득 장치로부터 움직이는 물체를 직접 받아들여 그중 2프레임을 사용하였다. 제안한 방법과의 비교를 위해 기존의 방법중 성능이 우수하다고 알려진 region

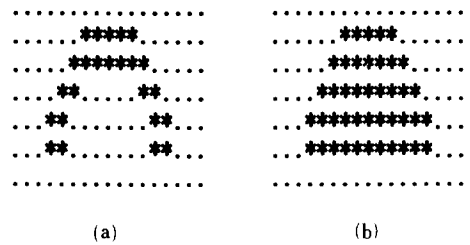


그림 5. 한 방향 scan때 제거되지 않는 부분
(a) 원 형태 (b) region growing 후
Fig. 5. Error of only one directional scanning.
(b) Original. (b) After region growing.

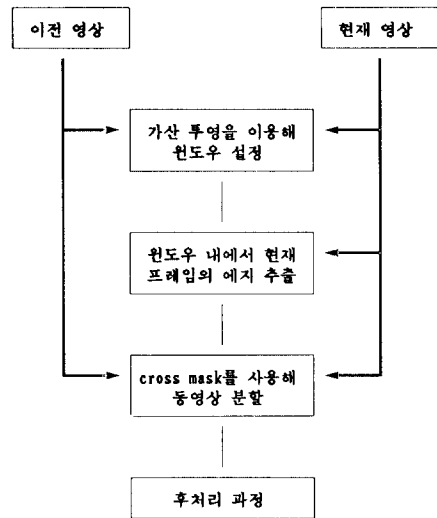


그림 6. 제안한 동영상 분할 방법의 순서도
Fig. 6. Block diagram of the proposed method.

growing 방법도^[11] 함께 시뮬레이션하였다. 이 방법은 이전 영상과 현재 영상과의 차를 구해 차영상을 만든 후 차영상 내의 각 영역들로 둘러 쌓인 부분을 초기 영역으로 하여 영역을 확장시켜 나가는 방법으로 여러개의 움직인 물체가 존재하거나 배경이 복잡할 경우에도 적용가능한 방법이다. 그러나 물체의 내부가 균일한 밝기를 가질 경우에만 적용할 수 있다는 단점이 있으며 물체가 자신의 한 변과 평행한 방향으로 움직였을 경우에는 한 물체의 차영상 영역이 두개 이상의 영역으로 분리되므로 차영상 영역으로 둘러

쌓인 부분을 구하기 위해서는 떨어져 있는 차영상 영역들을 이어주는 과정이 필요하게 된다. 또한 초기 영역을 구한 후 이 영역의 평균과 표준편차를 구해 region growing을 하게 되는데 주변화소가 같은 영역에 속하는가를 판단하는 기준의 설정을 어떻게 하는가에 따라 성능이 좌우된다.

그림 7은 배경이 단순하고 움직인 물체가 여러개인 경우로서 (a)가 이전영상이고 (b)가 현재 영상이다. (c)가 제안한 방법의 결과이고 (d)가 기존의 방법의 결과이다. 제안한 방법과 기존의 방법 모두 움직인

물체의 위치와 형태를 잘 추출해 내었다. 제안한 방법이 움직인 물체가 여러개일 경우에도 잘 적용된다는 것을 볼 수 있으며 물체의 내부가 균일하기 때문에 기존의 방법도 움직인 물체를 잘 검출해 내었다. 이 경우 우측의 물체가 자신의 한 변과 평행하게 움직였는데, 제안한 방법은 모든 움직인 물체를 동일한 과정으로 처리하는데 비해 region growing 방법은 이 물체에 대해 부가적인 과정을 수행할 필요가 있었다. 즉, 제안한 방법은 물체가 어떤 움직임으로 갖는가에 상관없이 움직인 물체를 검출해낼 수 있으나

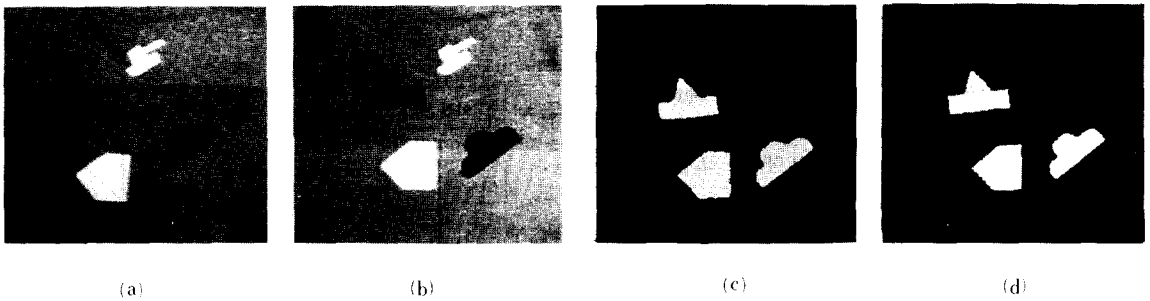


그림 7. 움직인 물체가 여러개일 경우

(a) 이전영상 (b) 현재 영상 (c) 제안한 방법의 결과 (d) 기존의 방법의 결과

Fig. 7. Case of several moving objects.

(a) Previous frame. (b) Current frame. (c) Result of the proposed method. (d) Result of the conventional method.

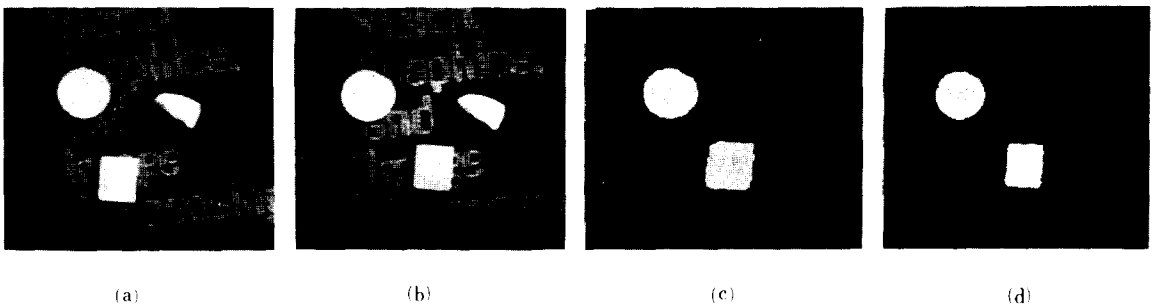


그림 8. 배경이 복잡한 경우

(a) 이전 영상 (b) 현재 영상 (c) 제안한 방법의 결과
(d) 기존의 방법의 결과

Fig. 8. Case of complex background.

(a) Previous frame. (b) Current frame. (c) Result of the proposed method. (d) Result of the conventional method.

기존의 region growing 방법은 물체의 움직임에 따라 처리하는 과정이 달라진다.

그림 8은 배경이 복잡하고 움직일 물체가 여러개인 경우에 대한 시물레이션이다. (a)가 이전 영상이고 (b)가 현재 영상이며 (c), (d)가 각각 제안한 방법과 기존의 방법의 결과이다. 제안한 방법의 결과에서 볼 수 있듯이 아래쪽의 움직인 물체의 좌측 부분이 실제보다 약간 두껍게 검출이 되었는데 이것은 그림자 부분도 함께 검출되었기 때문이다. 기존의 방법은 같은 밝기를 갖는 영역만을 검출하기 때문에 이 부분은 나타나지 않았다. 그러나 그림자 부분도 움직인 물체와 함께 움직였기 때문에 오히려 검출이 되는 것이 타당하다. 기존의 방법이 움직인 영역의 일부인 그림자를 검출하지 못했다는 것은 물체의 내부가 균일하지 않은 경우에는 적용이 어렵다는 증거가 된다.

그림 9는 물체의 내부가 균일하지 않은 경우에 대한 시물레이션이다. (c)가 제안한 방법의 결과이고 (d)가 기존의 방법의 결과이다. 제안한 방법은 움직인 물체를 잘 검출한 것을 볼 수 있으나 기존의 방법은 일부분 밖에는 검출하지 못하였다. 또한 임계값을 크게 하면 주변의 배경영역까지 확장되어 움직인 물체가 매우 크게 분리된다. 따라서 기존의 region growing 방법은 일반적인 물체의 움직임을 검출해 내는데는 적합하지 못하며 물체의 내부가 균일하다는 특수한 경우에만 적용 가능하다. 하지만 제안한 방법은 에지만을 특징으로 사용하기 때문에 물체의 내부

상태와는 관계없이 움직인 물체를 추출할 수 있다. 그림10은 물체가 회전운동을 하였을 경우에 대한 시물레이션이다. 역시 좋은 결과를 보이고 있으며 제안한 방법이 그림자 부분도 찾는데 비해 기존의 방법은 이 부분을 검출하지 못하였다. 또한 기존의 방법은 실제 물체의 크기보다 조금 작게 검출되었다.

연산 시간을 비교하자면 기존의 region growing 방법은 더 이상 합칠 영역이 없을 때까지 반복을 행하기 때문에 제안한 방법의 약 3 배 이상의 시간을 필요로 하는 것을 확인하였다. 또한 물체가 자신의 한 변의 방향과 평행하게 움직였을 경우에 기존의 region growing 방법은 부가적인 과정을 수행하여야 하므로 연산시간이 추가되나 제안한 방법의 연산 시간은 물체의 움직임 형태와는 관계가 없다.

이상의 시물레이션 결과를 종합하여 볼 때 제안한 방법은 배경이 복잡하거나 움직인 물체가 여러개인 경우에도 잘 동작함을 알 수 있으며 연산 시간 면에서도 만족할 만한 성능을 보였다.

VI. 결 론

본 논문에서는 새로운 동영상 분할 방법을 제안하고 기존의 방법과의 성능평가를 통해서 그 타당성을 검증하였다. 제안한 동영상 분할 방법은 전처리 과정과 움직인 물체의 경계를 추출하는 과정, 후처리 과정 등의 세부분으로 구성되어 있다. 전처리 과정은 연산 시간을 감축하기 위해 움직인 물체 주위에

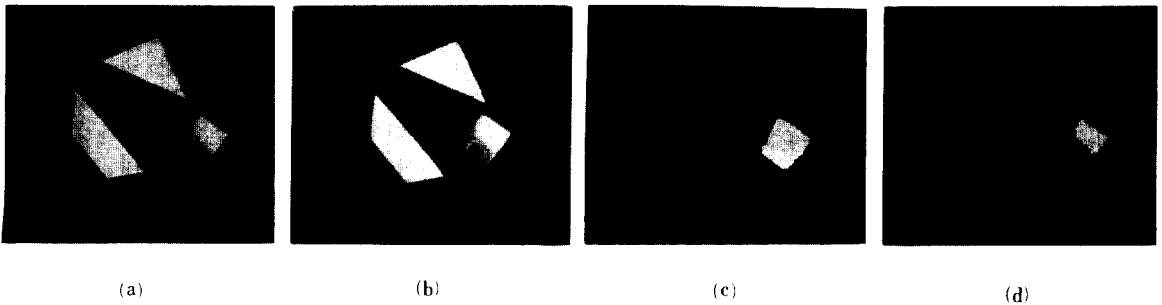


그림 9. 물체의 내부가 균일하지 않은 경우
 (a) 이전 영상 (b) 현재 영상 (c) 제안한 방법의 결과 (d) 기존의 방법의 결과
 Fig. 9. Case of nonhomogeneous object.
 (a) Previous frame. (b) Current frame. (c) Result of the proposed method.
 (d) Result of the conventional method.

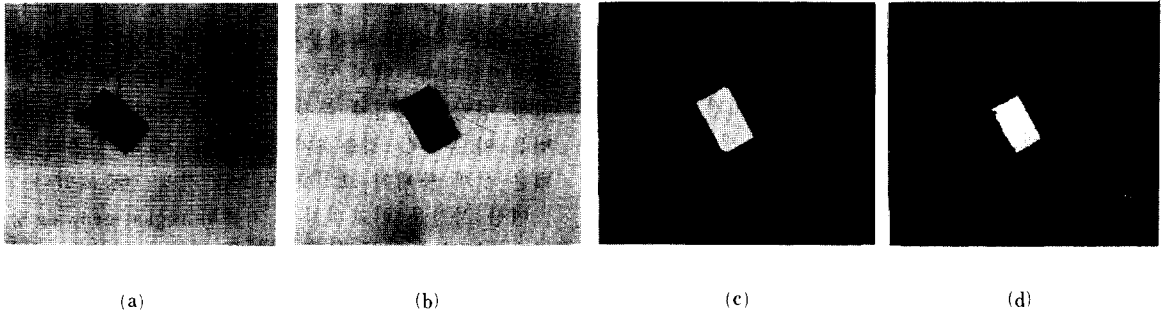


그림 10. 물체가 회전운동을 한 경우

(a) 이전 영상 (b) 현재 영상 (c) 제안한 방법의 결과 (d) 기존의 방법의 결과

Fig. 10. Case of rotational motion.

(a) Previous frame. (b) Current frame. (c) Result of the proposed method.
(d) Result of the conventional method.

윈도우를 띄우는 과정으로서 가산투영을 이용하여 물체의 대략적인 위치를 파악해낸다. 두번째 부분은 움직인 물체의 경계를 추출하는 과정으로서 전처리 과정으로부터 얻은 윈도우 내에서만 연산을 수행한다. 이 과정에서는 현재 영상으로부터 에지를 구한 다음 각 에지 화소들에 대해 cross mask를 띄워서 유사성을 계산한 후 이 유사성에 따라 해당 화소가 움직인 물체에 속하는가를 판단하여 물체의 경계만을 추출한다. 후처리 과정에서는 이전 과정에서 구한 경계로부터 잘못 찾아진 부분을 제거하고 움직인 물체의 경계는 내부를 동일한 화소값으로 채우므로써 움직인 물체만을 분리해낸다.

제안한 방법은 기존의 방법에 비해 계산이 간단하고 물체의 여러가지 형태의 움직임, 예를 들면 병진운동, 회전운동, 시선 방향의 운동 등에도 적용할 수 있다는 장점이 있다. 또한 배경이 복잡하고 움직인 물체의 내부가 균일하지 않을 경우에도 좋은 성능을 보이며 움직인 물체가 여러개일 경우에도 적용가능함을 시뮬레이션을 통해 보였다.

參 考 文 獻

- [1] J.K. Aggarwal and W.N. Martin, "Dynamic scene analysis," NATO ASI Series, vol. 2, *Image Sequence Processing and Dynamic Scene Analysis*, T.S. Huang (ed.), pp. 40-73, Springer-Verlag, 1983.
- [2] J.K. Aggarwal and R.O. Duda, "Computer analysis of moving polygonal images," *IEEE Trans. Computer*, vol. C-24, no. 10, pp. 966-975, 1975.
- [3] W.N. Martin and J.K. Aggarwal, "Computer analysis of dynamic scenes containing curvilinear figures," *Pattern Recognition*, vol. 11, pp. 169-178, 1979.
- [4] J.O. Limb and J.A. Murphy, "Estimating the velocity of moving edges in television signals," *CVGIP*, vol. 4, pp. 311-327, 1975.
- [5] C.L. Fennema and W.B. Thompson, "Velocity determination in scenes containing several moving objects," *CVGIP*, vol. 9, pp. 301-315, 1979.
- [6] W.B. Thompson, "Combining motion and contrast information for segmentation," *IEEE Trans. PAMI*, vol. PAMI-2, no. 6, pp. 543-549, 1980.
- [7] J.L. Potter, "Scene segmentation using motion information," *CVGIP*, vol. 6, pp. 558-581, 1977.
- [8] J.A. Leese, C.S. Novak, and V.R. Taylor, "The determination of cloud pattern motion from geosynchronous satellite image data," *Pattern Recognition*, vol. 2, pp. 279-292, 1972.
- [9] R. Jain and H.H. Nagel, "On the analysis of accumulative pictures from image sequences of real world scenes," *IEEE Trans. PAMI*, vol. PAMI-1, no. 2, pp. 206-214, 1979.
- [10] R. Jain, W.N. Martin, and J.K. Aggarwal, "Segmentation through the detection of changes due to motion," *CVGIP*, vol. 11,

pp. 13-14, 1979.

[11] S. Yalamanchili, W.N. Martin, and J.K. Aggarwal, "Extraction of moving object descriptions vis differencing," *CVGIP*, vol. 8, pp. 188-201, 1982.

[12] R. Jain, "Extraction of motion information from peripheral process," *IEEE Trans. PAMI*, vol. PAMI-3, no. 5, pp. 489-503, 1977.

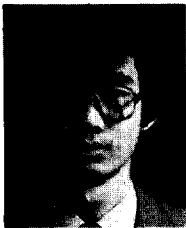
[13] 김근형, 박래홍, "가산 투영법을 사용한 two-stage template matching," 전자공학회 논문지, 제24권, 제 2 호, pp. 142-148, 1987.

[14] J.M. Prewitt, *Picture Processing and Psychopictorics*, pp. 75-149, Academic Press, New York, 1970.

[15] 황신한, 이상욱, "움직임 정보를 이용한 영상 영역화 기법," 한국통신학회 추계학술 발표회 논문집, pp. 7-10, 1987.

[16] S.W. Zucker, "Region growing: Childhood and adolescence," *CVGIP*, vol. 5, pp. 382-399, 1976. *

著 者 紹 介



姜 政 錫(準會員)
 1988年 2月 서강대학교 전자공학과 졸업. 1988年~현재 동대학원 전자공학과 석사과정. 주관심분야는 영상처리 및 컴퓨터 아키텍처 등임.

李 商 郁 (正會員) 第25卷 第7號 參照
 현재 서울대학교 제어계측 공학과 부교수.



朴 來 弘 (正會員) 第25卷 第12號 參照
 현재 서강대학교 전자공학과 부교수.