

# 효율적인 이동물체 분할과 고속 추적 알고리즘에 관한 연구

조영석<sup>†</sup> · 이주신<sup>††</sup>

## 요약

본 논문에서는 매칭 에러 영상과 이동벡터를 이용한 효율적인 이동물체 외곽선 검출 알고리즘과 부분외곽선 정보를 이용한 이동물체 고속 추적 알고리즘을 제안하였다. 이동물체의 외곽선 검출은 watershed 알고리즘을 기반으로 확률분포합수를 적용하여 seed 영역을 생성하고 seed 영역을 확장하여 이동물체의 윤곽선을 검출한 다음 이동벡터를 이용하여 최종 외곽선을 추출한다. 외곽선 중 일부를 특징으로 하여 이동물체를 추적하는 알고리즘을 사용하였다. 이동물체 초기 특징 벡터는 이동물체의 외곽선 영역 중 상하좌우의 외곽선 일부분을 특징벡터로 정한다. 다음은 추적단계로 이전 프레임에서 얻은 특징벡터를 이용하여 현재 프레임에서 이동물체의 추적을 수행하였다. 실제영상에 대하여 제안된 알고리즘으로 이동물체추적 모의 실험을 수행한 결과 기존 능동 윤곽선 추적알고리즘은 물체 외곽선 전체를 추적하기 때문에 물체의 외곽선 길이에 따라 처리시간이 변화하지만 제안된 알고리즘은 이동물체의 외곽선 영역을 특징정보로 하여 추적하기 때문에 추적 연산이 간단하였다. 고속 이동벡터를 추출 BMA 연산은 기존 알고리즘 보다 연산량이 약 39% 감소였고, 이동 물체 외곽선 검출 알고리즘은 과분할 문제점이 발생하지 않았으며, 상하 좌우 외곽선 정보를 이용하여 이동물체를 추적한 결과 추적오차는 특징벡터의 크기가 [15×5]일 때 검색오차가 4 화소 이하로 양호하게 나타났다.

## A Study on Effective Moving Object Segmentation and Fast Tracking Algorithm

Youngseok Cho<sup>†</sup> · Ju-Shin Lee<sup>††</sup>

## ABSTRACT

In this paper, we propose effective boundary line extraction algorithm for moving objects by matching error image and moving vectors, and fast tracking algorithm for moving object by partial boundary lines. We extracted boundary line for moving object by generating seeds with probability distribution function based on Watershed algorithm, and by extracting boundary line for moving objects through extending seeds, and then by using moving vectors. We processed tracking algorithm for moving object by using a part of boundary lines as features. We set up a part of every-direction boundary line for moving object as the initial feature vectors for moving objects. Then, we tracked moving object within current frames by using feature vector for the previous frames. As the result of the simulation for tracking moving object on the real images, we found that tracking processing of the proposed algorithm was simple due to tracking boundary line only for moving object as a feature, in contrast to the traditional tracking algorithm for active contour line that have varying processing cost with the length of boundary line. The operations was reduced about 39% as contrasted with the full search BMA. Tracking error was less than 4 pixel when the feature vector was [15x5] through the information of every-direction boundary line. The proposed algorithm just needed 200 times of search operation.

**키워드 :** 이동물체 분할(Segmentation), 외곽선 검출(Boundary Line Detection), watershed, 블록매칭(BMA), 이동물체추적(Tracking)

### 1. 서론

영상추적은 영상 기반 제어[1, 2], 사람-컴퓨터 인터페이스 [3, 4], 감시장치[5, 6], 농업자동화[7, 8], 의료영상분석[9, 10], 영상복원[11, 12] 등을 포함하는 일부 응용분야에서 가장 중요한 요소 중 하나로 부각되고 있다[13, 14]. 영상추적의 주요 연구 방향은 인접된 두 영상으로부터 움직임 파라미터를 추출하여 이동물체를 추적하는 것으로 인접된 두 영상사이의 영상매칭 기법, 배경영상에서 이동물체를 분할하는 영상 분할기법, 그리고 과거의 움직임 파라미터를 기초로 다음 프

레이스에서 이동물체의 위치를 예측하는 움직임 예측 기법 등을 포함하게 된다. 영상 열로부터 움직임 파라미터의 예측 연구가 활발히 진행되고 있다[15, 16]. 이들 연구의 방향은 특징기반 방식과 optic flow 방식으로 크게 대별된다.

이동물체를 추적하기 위한 동영상처리의 첫 번째 단계는 영상 평면 내에 이동물체를 검출하고 정확한 이동물체의 영역을 배경으로부터 분리하는 것이다. 카메라가 고정되어 배경영상이 변하지 않는 경우 현재의 입력영상에서 배경영상으로 저장된 표준 영상의 차를 구하는 차 영상 기법이 많이 이용되지만 배경이 변화가 큰 경우 적용이 곤란하다. 따라서 이동물체 주위의 환경변화가 존재하는 연속 영상에서 이동벡터를 구하여 이동영역을 추출하는 방법을 이용하

<sup>†</sup> 정 회 원 : 극동정보대학 전산정보처리과 교수

<sup>††</sup> 정 회 원 : 청주대학교 첨단공학부 교수

논문접수 : 2001년 12월 28일, 심사완료 : 2002년 5월 13일

여야 한다. 이동물체를 추적방법으로는 모델을 기반으로 하는 방법, 영역을 기반으로 하는 방법, 능동 윤곽선을 기반으로 하는 방법, 특징을 기반으로 하는 방법 등이 있다[17, 18]. 모델을 기반으로 하는 방법은 정확한 기하학적 모델이 주어져야 하고 소수의 추적 모델에 한정된다는 단점이 있으며, 영역기반 방법은 연속영상에서 특정 영역을 구하고 영역의 상관 도를 측정하여 이동물체를 추적하는 방법이다. 능동윤곽선을 기반으로 하는 방법은 물체의 경계를 윤곽선으로 표현하고 그것을 동적으로 갱신하면서 추적하는 방법으로 영역기반에 비해 복잡한 계산이 줄어들지만 추적 물체가 부분적으로 가려질 경우 이동물체를 추적하기 어렵다는 단점이 있다. 특징기반 방법은 물체의 전체를 추적하지 않고 이동물체의 특징을 추출한 다음 추적하는 기법으로 부분적인 가려짐이 발생해도 추적이 가능하다. 이들 영역기반추적과 능동 윤곽선을 기반으로 하는 이동물체의 추적시 배경이 변화하는 경우 이동물체의 윤곽선을 검출하기 위해 많은 연구가 진행되고 있다. 이동물체 외곽선 추출 알고리즘에 대한 연구로서 이동벡터나 차영상을 구하여 일정한 기준값을 기준으로 분할하는 방법이 이용되고 있으며 최근에는 확률분포함수를 적용하여 최적의 기준값을 정하는 연구가 활발히 진행되고 있다[19].

한편 고속으로 이동물체를 추적하기 위한 방법으로 추적창을 이용하는 방법이나 확률 분포함수를 이용하여 계산된 예측치를 바탕으로 신뢰 구간을 이용하는 방법들이 발표되고 있다. 이와 같이 검색영역을 축소하여 이동물체를 빠르게 추적하는 기법으로, 이동물체의 외곽선을 검출하고 이동물체 외곽선 중 특정 영역만을 검색하여 이동물체의 검출하면 추적시간을 더욱 감소시킬 수 있다. 또한 검출된 외곽선의 정보를 특징정보로 이용하면 이동물체의 일부가 가릴 경우에도 추적이 가능하게 된다[20].

본 연구에서는 watershed를 이용한 적응 이동물체 분할과 이동물체 고속 추적 알고리즘을 제안한다. 연속된 동영상 입력영상에 대하여 이동벡터와 매칭 에러 영상을 구한 후 매칭 에러에 대한 히스토그램을 구한다. 이동물체의 윤곽선은 watershed 기법을 이용하여 추출한다. 기존 watershed 기법의 단점인 과분할 문제는 확률 분포함수를 이용하여 seed 영역을 결정하고 seed 영역을 확장하는 알고리즘을 이용하여 해결한다. 이동물체의 외곽선은 이동물체 윤곽선의 이동벡터를 검색하여 최종 선정한다. 이들 외곽선 영역 중 상하 좌우의 외곽선 일부분을 특징벡터로 정한다. 이동물체 추적 단계로 이동물체의 외곽선 정보로부터 생성된 특징벡터를 이용해 이동물체를 추적한다. 이동물체의 일부가 가려질 경우 나머지 특징벡터들의 이동벡터를 기초로 이동물체 외곽선을 추정함으로써 이동물체를 추적하도록 구성하였다. 제안된 알고리즘을 실험한 결과 기존 알고리즘에 비해 약 39%의 연산량이 감소되었고 특징벡터의 크기가 [10×5]일

때 가장 양호결과를 얻었다. 제 2장은 이동벡터추출과 이동영역분할에 대하여 설명하고 3장은 부분 외곽선 정보를 이용한 이동물체 추적에 대하여 논하였으며 4장은 실험 및 고찰 그리고 결론으로 구성하였다.

## 2. 이동 벡터 추출과 이동영역 분할

배경이 변화하는 상황에서 이동물체 검출은 이전 영상과 현재 영상에서 유사도가 가장 높은 영역을 구하는 방법이며 평균자승오차를 이용하여 이동벡터를 구한다. 일반적인 이동벡터의 계산은 이전영상 중 검색하고자하는 탐색영역과 가장 잘 일치되는 현재 영상의 위치를 찾는 방법인 블록 매칭 알고리즘(BMA : Block Matching Algorithm)을 이용하게 된다. 이때 매칭의 척도는 MAD(mean absolute difference)를 이용하였고 MAD는 식 (1)과 같다.

$$MAD(x, y) = \frac{1}{N^2} \sum_{i=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{j=-\frac{N-1}{2}}^{\frac{N-1}{2}} |I_t(k+i, 1+j) - I_{t-1}(k+x+i, 1+y+j)| \quad (1)$$

식 (1)에서  $I_t$ 는 현재 영상이고  $I_{t-1}$ 는 이전 영상이며  $N$ 은 검색 영역의 크기이고  $K, 1$ 은 검색영역의 중심좌표이며,  $x, y$ 는 탐색영역에서 탐색 점의 좌표를 의미한다.

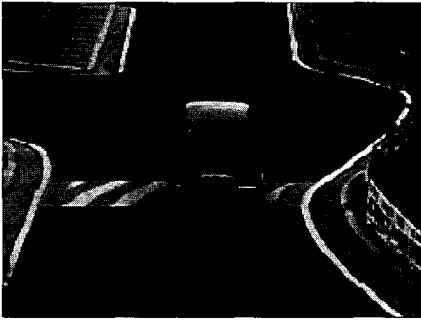
MAD를 이용한  $(k, 1)$ 번째 화소의 움직임 벡터는 식 (2)와 같다.

$$v_{(k, 1)} = \min MAD(x, y) \quad (2)$$

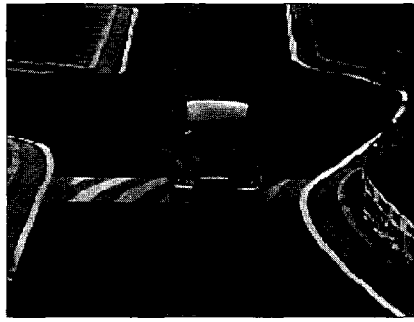
BMA를 이용하여 움직임 벡터를 추정할 경우 전역 탐색 블록매칭 알고리즘으로서 정합 가능한 모든 화소에 대하여 MAD를 구하고 가장 작은 MAD값을 가지는 탐색 점  $(x, y)$ 의 좌표 값을 움직임 벡터로 정하는 방법이다. 전역 탐색 블록매칭 알고리즘은 모든 탐색 점을 탐색하여야 하므로 계산이 많은 문제점이 있다. (그림 1)과 (그림 2)는 입력 영상이고 (그림 3)은 전역 탐색 블록 매칭알고리즘에 의해 계산된 이동벡터의 크기를 보였다. (그림 3)에서 밝기는 이동벡터 중 크기를 나타내는 것으로 밝은 부분은 이동벡터의 크기가 크고 검은 부분은 이동벡터가 존재하지 않는다는 것을 의미한다.

이와 같이 전역 탐색 블록 매칭 알고리즘의 계산량을 줄이기 위한 방법으로 여러 가지 방법들이 제시되고 있으나 본 연구에서는 탐색 점에서 공간 기울기인 에지를 구한 다음 에지의 값이 기준 값보다 큰 경우 이동벡터를 계산하고자 한다. <표 1>은 고속 이동벡터 계산알고리즘(FMVPA : Fast Moving Vector Processing Algorithm)이다.

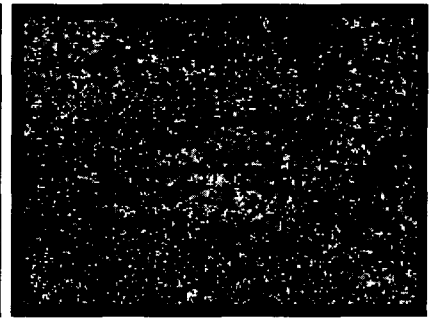
(그림 4)는 이동벡터를 계산할 후보영역이고, (그림 5)는 고속 이동벡터 알고리즘에 의해 계산된 이동벡터를 표시하였다. 고속 이동벡터 계산 알고리즘에서 후보 영역의 결정은 먼저 계산영역의 공간 기울기인 에지를 구한 다음 에지



(그림 1) 입력 프레임 #1



(그림 2) 입력 프레임 #2



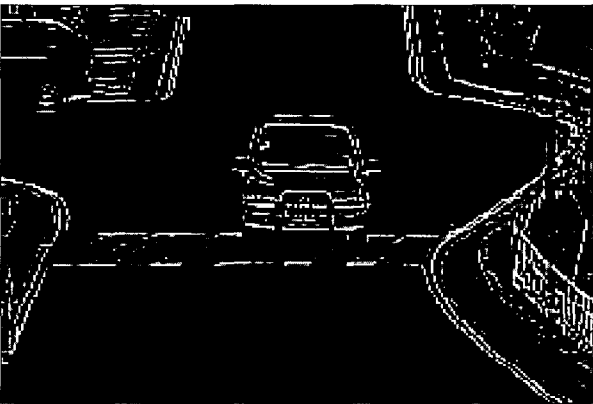
(그림 3) 이동벡터 검출

의 크기가 설정 값보다 큰 경우 고속 이동벡터를 구하도록 함으로서 연산량을 줄이도록 구성하였다.

〈표 1〉 고속 이동벡터의 계산 알고리즘

```

Step 1: 검색좌표 loop (x, y)
Step 2: 탐색 블록 loop (i, j)
Step 3: 현재 검색점의 에지 계산
Step 4: 에지 값이 기준 값보다 작으면 step 8로 이동
Step 5: MAD 계산
Step 6: 이동 벡터 계산
Step 7: end loop (i, j)
step 8: end loop (x, y)
    
```



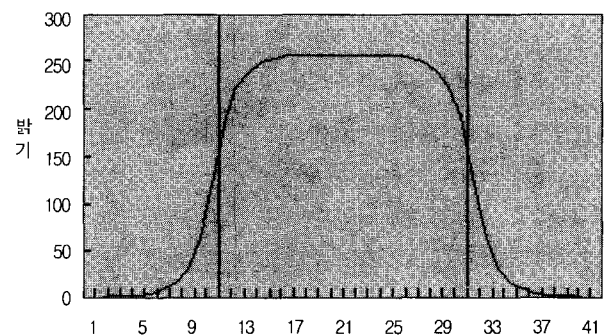
(그림 4) MAD 계산 후보 점(흰색영역)



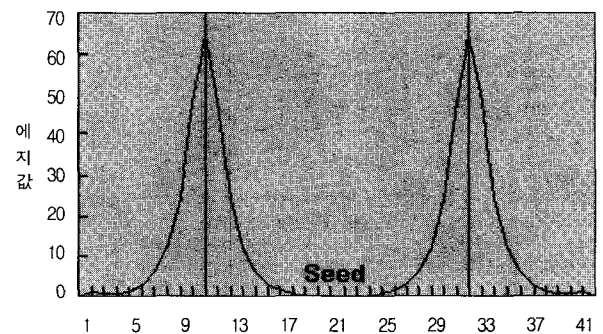
(그림 5) FMVPA에 의한 이동벡터

watershed 알고리즘은 지형학분야에서 연구된 것으로 영상에 대한 에지 값을 고도정보로 처리함으로서 영상처리에 응용되고 있다. 이 알고리즘은 담수지역 또는 seed 영역을 구분하는 분수령을 찾음으로 각각의 국부최소치에 관련된 영역을 분할하는 알고리즘이다. watershed 알고리즘은 여러 가지가 연구되고 있으며 모의 침수(immersion simulation)기법이 가장 효과적인 것으로 알려져 있다[21].

모의침수기법은 낮은 고도의 영역으로부터 물을 채워 나간다. 물이 다른 영역과 만나게되면 경계에 가상의 둑을 쌓는 기법으로 영역을 분할한다.



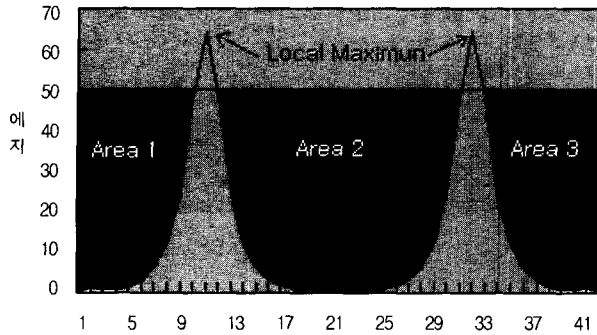
(그림 6) 영상의 1차원 단면도



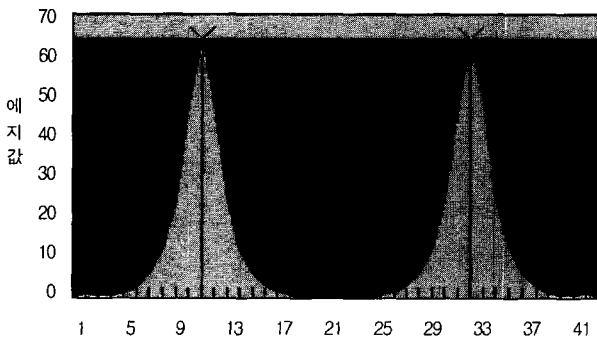
(그림 7) (그림 2)의 에지 단면

(그림 6)은 영상의 1차원 단면을 표시한 것이고 (그림 7)은 (그림 6)에 대한 에지 단면이며 (그림 8)은 seed 영역을 확장한 것으로 3개의 영역으로 나뉘어진다. 계속해서 seed 영역을 증가시키면 영역1과 영역2가 만나며 만나는 점을 경계선으로

표시하면 영역을 분할하게 된다. (그림 9)는 두 영역이 만나는 영역을 경계선으로 마크하는 것을 표시하고 있다.

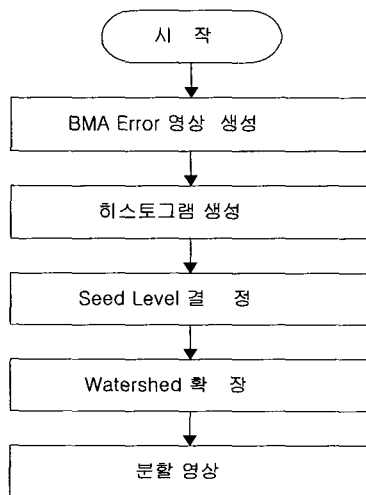


(그림 8) Seed 영역 확장



(그림 9) 경계 마스크(두 영역이 만나는 곳)

watershed 알고리즘을 이동물체의 이동영역 분할을 위해 첫 번째로 입력벡터 에러영상의 히스토그램을 구하고 이동벡터영상에 대한 seed 영역을 계산한다. seed 영역을 확장하여 이동물체의 윤곽선을 검출한다. seed 영역은 답수가 시작되는 곳으로 최적의 국부 최소 값을 결정하여야 한다. (그림 10)은 watershed 알고리즘에 의한 이동물체 분할 처리 순서도이다.



(그림 10) Watershed에 의한 이동물체 분할

이동벡터의 연산에서 발생하는 매칭 에러 영상을  $I_{mm}$ 이라고 하면 식 (3)과 같이 표현할 수 있다.

$$I_{mm} = MAD_{Min}(I_t - I_{t+1}) \quad (3)$$

또한 t 시간에 입력된 영상을  $I_t$ 라고 하면 고정된 배경부분( $I_{bt}$ ), 이동물체부분( $I_{mt}$ ), 그리고 외부에서 유입된 각종 잡음( $I_{nt}$ )으로 구성할 수 있으며 식 (4)와 같다.

$$I_t = I_{bt} + I_{mt} + I_{nt} \quad (4)$$

$$I_{(t+1)} = I_{b(t+1)} + I_{m(t+1)} + I_{n(t+1)}$$

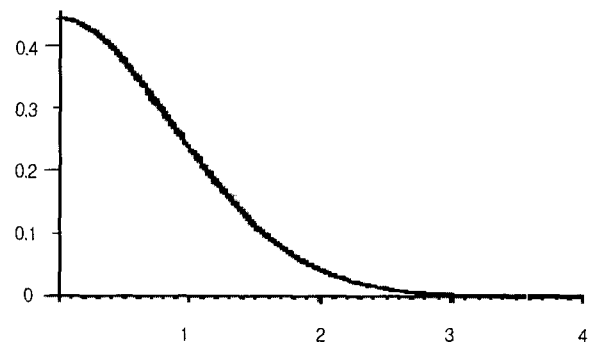
따라서 입력 프레임  $I_t$ 와  $I_{(t+1)}$ 이 최적으로 정합되었을 때 매칭 에러 영상  $I_{mm}$ 은 식 (4)를 이용하여 다시 정리하면 식 (5)와 같다.

$$\begin{aligned} I_{mm} &= MAD_{min}(I_t, I_{(t+1)}) \\ &= MAD_{min}[(I_{bt} + I_{mt} + I_{nt}), (I_{b(t+1)} + I_{m(t+1)} + I_{n(t+1)})] \end{aligned} \quad (5)$$

위의 식 (5)에서 최적으로 매칭되어 배경영역의 변화가 없다고 가정하면 식 (4)는 식 (6)과 같이 정리할 수 있다.

$$\begin{aligned} I_{mm} &= MAD(I_{bt}, I_{b(t+1)}) + MAD(I_{ot}, I_{o(t+1)}) + MAD(I_{nt}, I_{n(t+1)}) \\ &= 0 + MAD(I_{ot}, I_{o(t+1)}) + MAD(I_{nt}, I_{n(t+1)}) \\ &= A_o(\text{이동 물체 성분}) + A_n(\text{잡음 성분}) \end{aligned} \quad (6)$$

식 (6)에서 잡음성분  $A_n$ 은 ZMND(Zero Mean Normal Distribution)  $N(0, \sigma^2)$ 로 모델링이 가능하다[22, 23]. (그림 11)은 잡음의 정규 분포함수이고 (그림 12)는 매칭 에러 영상의 예이다. (그림 13)은 매칭 에러 영상의 히스토그램으로 물체영역과 잡음영역을 분할할 수 있다.

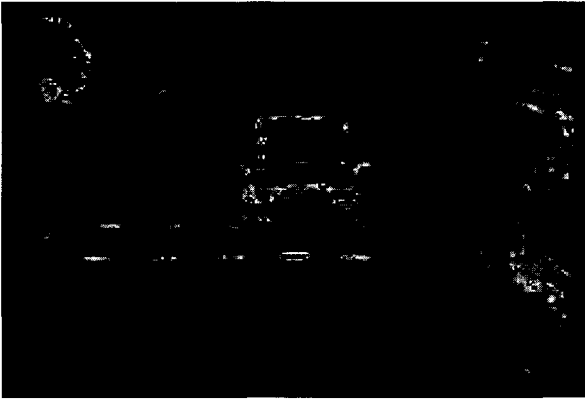


(그림 11) 잡음의 확률 분포함수

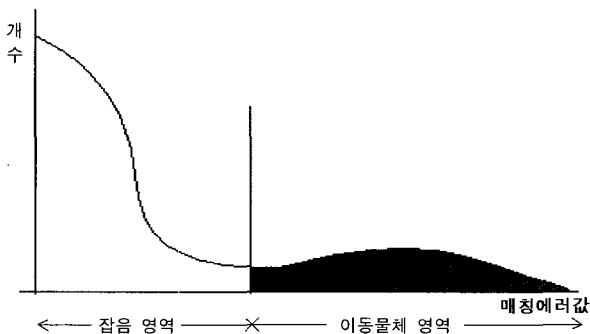
(그림 13)에서 seed 영역을 결정하기 위한 임계값( $Th$ )는 참고논문 [23, 24]을 참조하여 다음 식 (7)과 같이 정하였다.

$$Th = 1.65 \sigma \quad (7)$$

watershed 의 문제점인 과분할(over segmentation)을 해

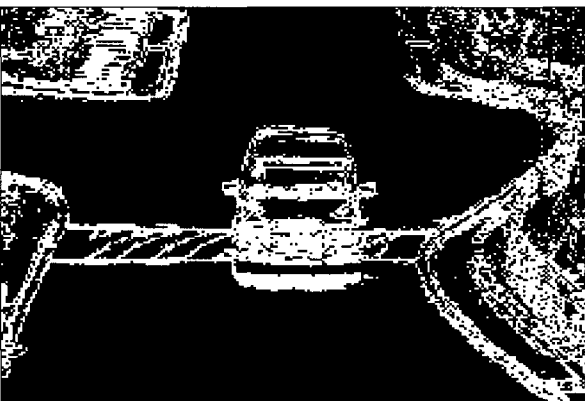


(그림 12) 매칭 에러 영상



(그림 13) 매칭 에러영상의 히스토그램 영역

결하기 위해 윤곽선 영역에 대하여 이동벡터의 크기와 방향이 같으면 이웃 영역을 병합하여 최종 이동물체 외곽선을 구한다. (그림 14)는 이동물체 분할 영상이다. 이동물체 추적방식 중 능동 윤곽선 모델은 전체 영상에 대하여 이동벡터를 구하지 않고 이동물체의 외곽선에 대하여 이동벡터를 구하기 때문에 처리속도가 빠른 특징이 있다.

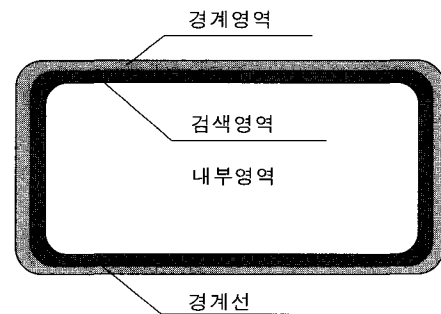


(그림 14) 이동물체 분할 영상

### 3. 부분 외곽선 정보를 이용한 이동물체 추적

초기 이동물체의 이동벡터에 의하여 이동물체가 검출되면 다음단계로 이동물체 추적을 위한 특징 벡터를 추출한

다. 첫 번째로 이동 영역내의 이동물체에 대한 외곽선을 추출한다. 이동물체의 외곽선은 이동영역부분에 대하여 에지를 구한 다음 에지를 watershed 기법을 이용하여 외곽선을 구한다. 두 번째 단계로 이동물체의 외곽선으로부터 특징벡터를 구한다. 3차원공간에서 이동하는 이동물체가 2차원 공간으로 투시 변환되면 시간에 따라 화소값이 변하기 때문에 이동물체의 경계선이 변하게 된다. 외곽선 정보를 이용하여 이동 물체를 추적하기 위하여 경계선영역을 추출한다. 이동물체의 경계영역을 3가지부분으로 나누면 이동물체 내부 영역과 외곽선 영역 그리고 이동물체 외부영역인 배경영역으로 나눌 수 있으며, 현재 프레임에서 이동물체의 경계선 정보를 검출하고 다음 프레임에서 경계를 검색하여 이동물체를 추적한다. (그림 15)는 이동물체의 검색영역과 경계영역 그리고 배경영역을 표시한다.

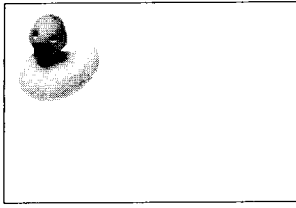


(그림 15) 이동물체의 영역정의

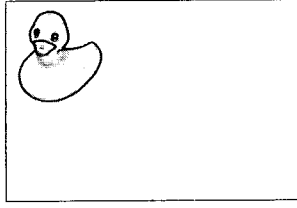
능동 윤곽선모델에서 경계선의 일부분이 다른 이동물체에 가려질 경우 추적이 곤란한 문제점을 해결하기 위한 방법으로 특징기반 영상추적모델에서 이용하는 곡물이나 분기점 교차점과 같은 특징요소를 검색하는 대신 상하좌우 경계선을 특징벡터로 하여 검색함으로써 일부 경계선이 가려질 경우에도 이동물체를 추적할 수 있도록 구성하고자 한다. 입력 영상이 (그림 16)과 같을 때 (그림 17)과 같이 이동물체의 경계선을 구하고 상하 좌우의 경계선으로부터 특징벡터를 구하였다. <표 2>는 이동물체의 특징벡터로 이동물체의 경계영역의 상하좌우의 영상으로부터 일부분을 저장하여 사용한다. <표 3>은 이동물체 추적을 위한 특징 벡터로서 이동물체가 존재하는 영상의 상하좌우에 영상데이터를 보인 것이다.

세 번째 단계로 특징벡터와 현재 프레임사이의 MAD를 이용하여 이동물체를 추적한다. <표 3>과 같이 이전 프레임으로부터 추출한 특징 벡터를 이용하여 현재 영상에서 이전프레임의 이동물체 좌표로부터 특징 벡터의 일정한 영역에 대한 MAD를 구하고 가장 큰 상관 도를 갖는 영역을 특징 벡터의 이동 좌표로 한다.

네 번째 단계로 특징 벡터를 갱신한다. 이동물체는 추적 과정에서 3차원 회전이나 이동에 의해 외곽선이 변하게 된다. 본 논문에서는 추적 대상의 형태변화에 에 적응하기 위

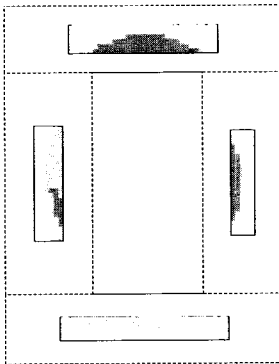


(그림 16) 입력영상의 예



(그림 17) 이동물체의 경계검출

<표 2> 이동물체 특징벡터

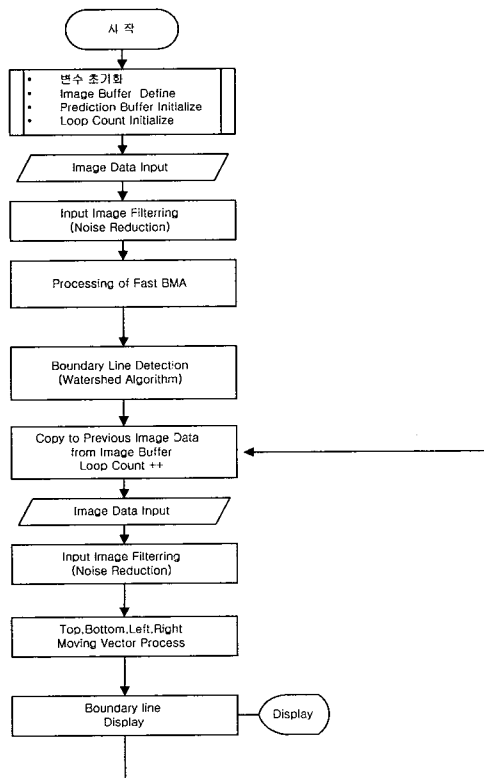


<표 3> 특징벡터의 예

39	63	63	63	63	63	63	63	39	39
63	63	63	63	63	63	63	63	63	63
63	63	63	63	63	63	63	63	63	63
00	39	63						63	63
39	39	39						63	63
39	39	39						63	63
39	39	39						63	63
39	39	39						63	63
39	39	39						63	63
39	39	39						63	63
39	63	63						63	63
39	39	63						63	63
00	39	63						63	63
39	39	39	39	39	39	39	39	39	39
39	39	39	39	39	39	39	39	39	39
39	39	39	39	39	39	39	39	39	39

하여 매 프레임마다 특징 벡터를 갱신한다. 좌표계산이 종료되면 이동물체에 대한 외곽선 정보를 취득하여 다시 외곽선 영역의 영상데이터를 특징 벡터로 저장한다.

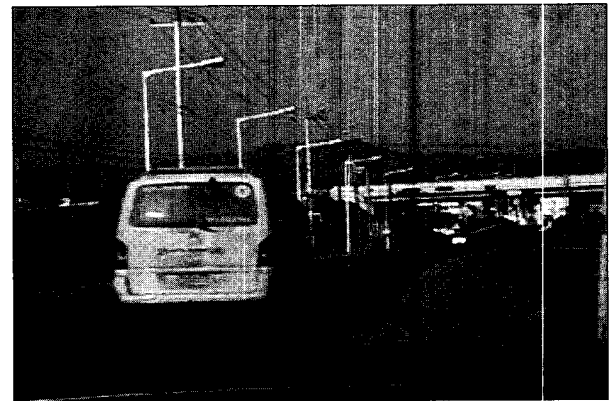
#### 4. 실험 및 고찰



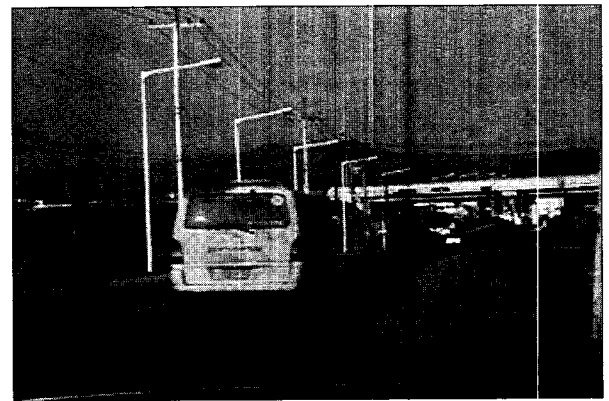
(그림 18) 이동물체 고속추적 알고리즘 처리 순서도

본 실험에서 고속이동물체 추적 알고리즘의 성능을 평가하기 위하여 외곽선 도로서 디지털 캠코더로 직접 얻은 [310×240×8Bit]크기의 영상을 이용하였고 PIII-500이 장착된 컴퓨터에서 C언어를 이용하여 프로그램을 수행하였다. (그림 18)은 처리 순서도이다.

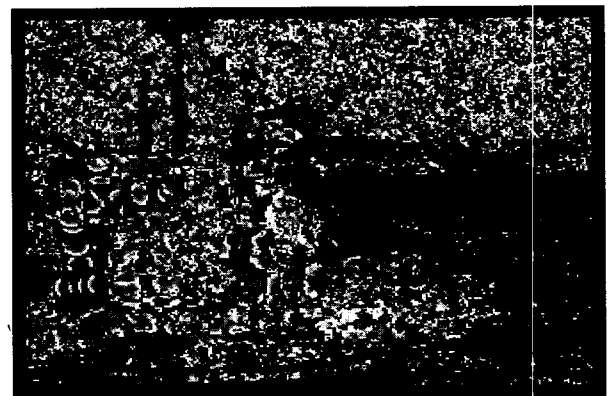
(그림 19)와 (그림 20)은 입력 영상이고 이동벡터 계산은 검색영역의 크기를 [3×3]으로 정하였고, 탐색영역의 크기는 [15×15]로 정하여 수행하였다. (그림 21)은 기존의 전역 검색 BMA에 의한 이동벡터를 추출한 것이다. (그림 22)는



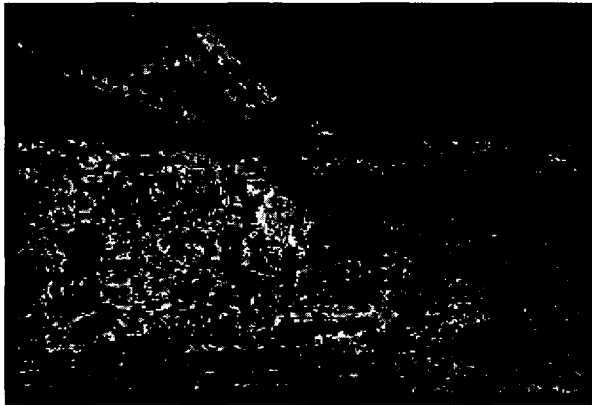
(그림 19) 입력 프레임(#01)



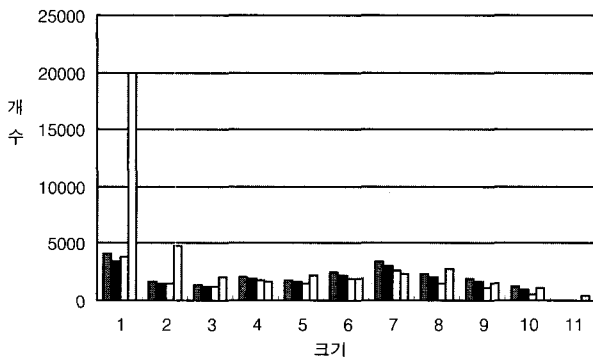
(그림 20) 입력 프레임(#03)



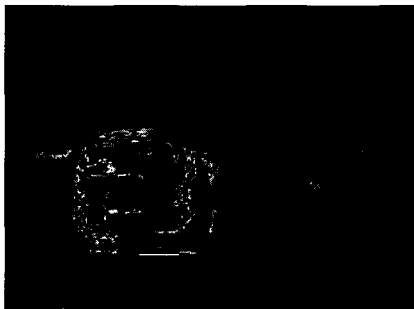
(그림 21) 전역 검색 BMA에 의한 이동벡터 추출



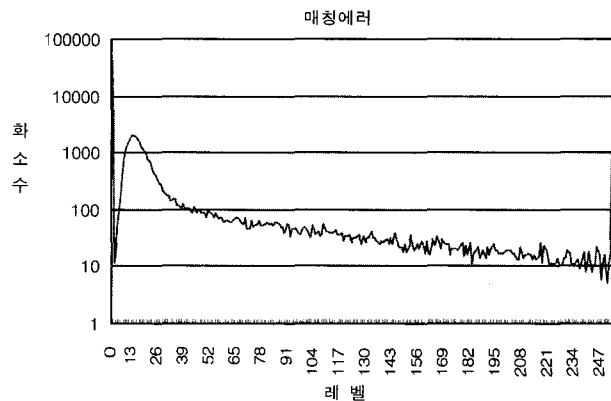
(그림 22) FMVPA에 의한 이동벡터추출



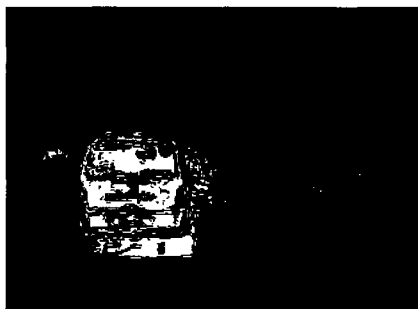
(그림 23) 이동벡터 크기 히스토그램



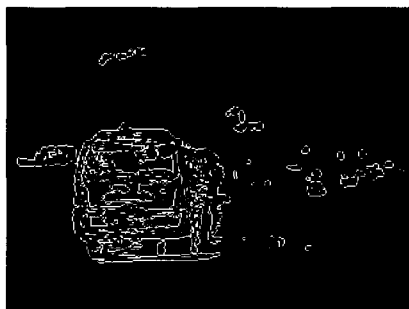
(그림 24) 매칭 에러 영상



(그림 25) (그림 22)의 히스토그램



(그림 26) Seed 영역



(그림 27) 이동영역 분할



(그림 28) 최종외곽선 검출

고속 이동벡터 추출 알고리즘에 의하여 이동벡터를 보였다. 또한 (그림 23)은 입력 프레임들에 대한 이동벡터 크기 히스토그램을 보였다.

전역 탐색 알고리즘과 제안된 고속 이동벡터 계산 알고리즘의 연산량을 비교하면 검색영역의 크기가  $3 \times 3$ 이고 탐색영역의 크기가  $15 \times 15$ 일 경우 다음 <표 4>와 같다.

<표 4> 전역 탐색알고리즘과 제안된 알고리즘의 단위 화소당 연산량 비교

항 목	기존전역탐색 BMA	제안된 알고리즘의 계산량	
		조건 판별 시	이동벡터 계산 시
가산/감산	2025( $(3 \times 3) \times 15 \times 15$ )	11	2025
승산/제산	0	2	0
비교	2250	1	2250

<표 5> 입력 영상에 대한 기존 BMA와 제안된 알고리즘의 연산량 비교

항 목	전역탐색 BMA	제안된 알고리즘	연산량 비교
BMA연산수	77,970	63,730	18.26% 감소
가산/감산	157,889,250	129,910,920	17.72% 감소
승산/제산	0	155,940	155,940증가
비교연산	175,432,500	143,470,470	18.21% 감소
전체 연산량	333,321,750	273,537,330	17.94% 감소



(a) Frame #01



(b) Frame #03



(c) Frame #05



(d) Frame #07



(e) Frame #09



(f) Frame #11

(그림 29) 실험 영상

<표 5>는 기존 BMA와 제안된 고속 이동벡터 추출알고리즘에 대한 연산량을 비교하였다. 이때 전역검색 BMA의 경우 66,896번 BMA 계산을 하였으며 제안된 알고리즘은 평균 40500회 BMA 계산을 수행함으로써 39%의 BMA 연산이 감소되었다.

(그림 24)는 매칭 에러 영상이고 (그림 25)는 (그림 24)의 히스토그램이며 (그림 26)은 입력 영상에서 seed 영역을 표시한 것이다. (그림 27)은 입력영상을 이동벡터의 watershed 알고리즘을 이용하였고  $Th = 1.65\sigma$ 를 기반으로 하여  $Th = 35$  기준으로 seed 영역을 확장하여 분할한 이동물체 분할영상이다. 또한 (그림 28)은 이동물체에서 최종 외곽선을 표

시한 것이다

초기 이동물체가 검출되면 다음단계로 이동물체를 추적하는 단계이다. 다음의 (그림 29)는 디지털 캠코더로 취득된 실험영상이다.

이동물체 추적을 위한 특징벡터는 이동물체의 경계영역 중 상하좌우 데이터를 이용하였다. 다음의 <표 6>부터 <표 9>까지는 특징 벡터의 크기별 추적 결과를 표시하였다.

특징 벡터 크기별로 실제 좌표와 검색좌표의 오차를 표시한 것으로  $[15 \times 5]$ 크기의 특징 벡터를 이용하여 이동물체를 추적한 결과 가장 양호한 결과를 얻었으며, 오차가 4화

<표 6>  $[5 \times 3]$  특징 벡터에 의한 추적 오차

추적 좌표								추적 오차							
상		하		좌		우		상		하		좌		우	
x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y
63	91	42	204	21	71	205	188	8	0	-1	-2	-1	-1	-2	0
76	96	54	200	37	68	204	184	8	-1	0	-3	-1	0	0	2
78	98	69	196	54	68	205	184	18	0	-1	-3	-2	-1	1	0
87	103	81	192	67	116	208	182	21	0	1	-3	0	-2	0	0
93	108	94	187	84	162	210	180	26	-1	1	-2	-1	-1	0	-1
96	109	105	184	98	160	213	176	32	1	1	-2	-2	-1	-1	0
104	114	117	179	109	158	214	175	33	-2	0	0	0	-1	0	-1
105	115	123	178	122	156	215	173	41	0	-1	-2	-2	-1	0	0
112	116	137	172	131	154	216	170	43	1	-1	0	-1	-1	1	0
122	117	142	170	135	154	216	178	36	1	0	1	0	-1	1	0
137	119	146	171	141	153	218	189	25	1	1	-1	0	0	0	-2

<표 7>  $[5 \times 5]$  특징 벡터에 의한 추적 오차

추적 좌표								추적 오차							
상		하		좌		우		상		하		좌		우	
x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y
63	91	42	204	21	71	205	188	8	0	-1	-2	-1	-1	-2	0
76	96	54	200	37	68	204	185	8	-1	0	-3	-1	0	0	1
78	97	67	196	54	68	205	184	18	1	1	-3	-2	-1	1	0
87	103	81	192	67	116	208	184	21	0	1	-3	0	-2	0	-2
93	108	94	187	84	162	210	180	26	-1	1	-2	-1	-1	0	-1
96	111	106	184	98	160	212	176	32	-1	0	-2	-2	-1	0	0
104	114	117	179	109	158	214	175	33	-2	0	0	0	-1	0	-1
105	114	123	178	122	156	213	173	41	1	-1	-2	-2	-1	2	0
112	116	137	172	131	154	216	170	43	1	-1	0	-1	-1	1	0
122	117	142	170	135	154	217	176	36	1	0	1	0	-1	0	2
137	119	146	171	141	153	218	187	25	1	1	-1	0	0	0	0

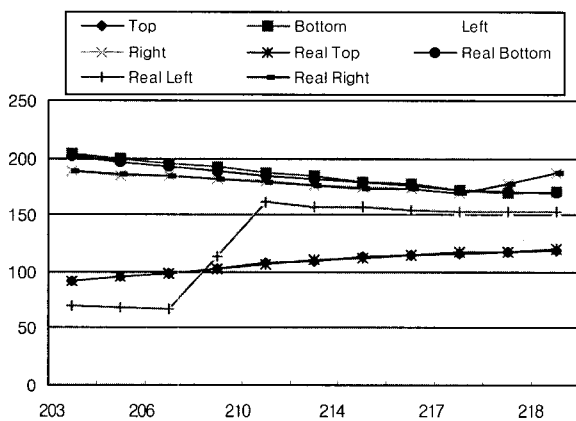


〈표 8〉 [10×3] 특징 벡터에 의한 추적 오차

추적 좌표								추적 오차							
상		하		좌		우		상		하		좌		우	
x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y
68	91	42	204	21	71	205	188	3	0	-1	-2	-1	-1	-2	0
80	96	54	200	37	68	204	185	4	-1	0	-3	-1	0	0	1
93	98	69	196	54	68	205	184	3	0	-1	-3	-2	-1	1	0
96	103	81	192	67	116	208	182	12	0	1	-3	0	-2	0	0
107	108	94	187	84	162	210	180	12	-1	1	-2	-1	-1	0	-1
118	109	105	184	98	160	210	176	10	1	1	-2	-2	-1	2	0
126	114	117	179	139	158	214	175	11	-2	0	0	0	-1	0	-1
134	115	123	178	122	156	215	173	12	0	-1	-2	-2	-1	0	0
145	116	137	172	131	154	216	170	10	1	-1	0	-1	-1	1	0
150	117	142	170	135	154	216	178	8	1	0	1	0	-1	1	0
156	119	146	171	141	153	218	187	6	1	1	-1	0	0	0	0

〈표 9〉 [15×5] 특징벡터에 의한 추적오차

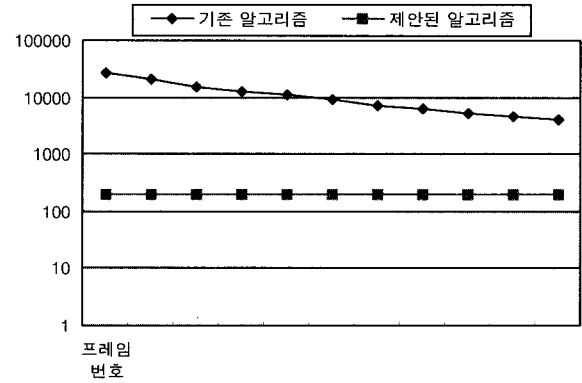
추적 좌표								추적 오차							
상		하		좌		우		상		하		좌		우	
x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y
69	91	41	204	20	71	203	188	2	0	0	-2	0	-1	0	0
86	96	52	200	36	68	204	185	-2	-1	2	-3	0	0	0	1
99	98	68	196	53	68	205	184	-4	0	0	-3	-1	-1	1	0
111	103	82	192	67	115	207	182	-3	0	0	-3	0	-1	1	0
117	108	94	187	84	162	210	180	2	-1	1	-2	-1	-1	0	-1
130	109	106	184	99	160	212	176	-2	1	0	-2	-2	-3	0	0
136	114	117	179	109	158	214	175	1	-2	0	0	0	-1	0	-1
145	115	121	178	121	156	215	174	1	0	1	-2	-1	-1	0	-1
157	116	136	172	131	154	216	170	-2	1	0	0	-1	-1	1	0
159	117	112	170	135	154	217	178	-1	1	1	1	0	-1	0	0
162	119	147	171	141	153	217	187	0	1	0	-1	0	0	1	0



(그림 30) 이동물체의 예측좌표와 실제좌표(15×5일 때)

소 미만으로 나타났다. (그림 30)에 이동물체의 예측좌표와 실제좌표를 보였다. (그림 31)은 전체 외곽선을 검색하는 기존의 알고리즘과 제안된 알고리즘과의 연산량을 비교한 것으로 초기 23760회의 검색 연산이 필요하였고 이동물체가 이동함에 따라 크기가 작아져 20820회, 15075회, 10752

회로 감소하였으며 제안된 알고리즘의 경우 물체의 크기에 관계없이 특징벡터의 크기에 따른 검색 연산이 필요하였다.



(그림 31) 기존 알고리즘과 제안된 알고리즘의 연산량 비교

## 5. 결 론

본 연구에서는 watershed 알고리즘에 의한 이동물체 분할과 부분 외곽선 정보를 이용하여 이동물체 추적 알고리즘을 제시하였고 제시된 알고리즘과 기존의 추적 알고리즘을 비교 평가결과 다음과 같은 결과를 얻었다.

1. 초기 이동벡터 계산 시 고속 이동벡터 알고리즘을 실제 영상에 적용한 결과 39%의 BMA 연산량을 줄일 수 있음을 보였다.
2. 이동영역의 분할은 매칭 어려영상을 가지고 Watershed 알고리즘을 이용하였으며 Th값은 분포함수에 의해 계산한 다음 Seed 영역을 생성하고 확장하여 윤곽선을 구한 다음 이동벡터의 각도와 크기를 이용하여 최종 분할하였다.
3. 이동물체의 상하좌우 경계영역의 데이터를 이용하여 특징기반 이동물체추적을 위한 특징벡터로 정하고 특징벡터의 크기가 [15×5]일 때 4화소 이하로 정확한 이동물체 추적이 가능함을 보였다.
4. 실제 영상에 대하여 기존의 전체 외곽선을 검색하는 기존의 경우 20820회의 연산이 필요하였으나 제안된 알고리즘을 적용하여 이동물체를 추적에는 200회의 검색 연산이 필요하였다.
5. 경계선 중 상하좌우의 경계영역을 이용하여 이동물체를 추적함으로써 기존 알고리즘에 비해 연산량이 적은 이동물체의 추적이 가능함을 보였다.

향후 연구과제로는 이동물체의 크기에 외곽선과 따라 다양한 크기의 모델에 대한 특징벡터 추출에 대한 연구와 가려진 이동물체추적을 위한 추적알고리즘 연구라 하겠다.

## 참 고 문 헌

- [1] P. Allen, B. Yoshimi, and A. Timcenko, Hand-Eye Coordi-

- nation for Robotics Tracking and Grasping, K. Hashimoto, ed., *Visual Servicing*, World Scientific, pp.33-70, 1994.
- [2] S. Hutchinson, G. D. Hager, and P. Corke, A Tutorial Introduction to Visual Servo Control, *IEEE Trans. Robotics and Automation*, Vol.12, No.5, pp.651-670, 1996.
- [3] N. Papanikolopoulos, P. Khosla, and T. Kanade, Visual Tracking of a Moving Target by a Camera Mounted on a Robot : A Combination of Control and Vision, *IEEE Trans. Robotics and Automation*, Vol.9, No.1, pp.14-35, 1993.
- [4] E. Dickmanns and V. Graefe, Dynamic Monocular Machine Vision, *Machine Vision and Applications*, Vol.1, pp.223-240, 1988.
- [5] A. F. Bobick and A. D. Wilson, A State-Based Technique for the Summarization of Recognition of Gesture, *Proc. Intl Conf. Computer Vision*, pp.382-388, 1995.
- [6] T. Darrell, B. Moghaddam, and A. Pentland, Active Face Tracking and Pose Estimation in an Interactive Room, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp.67-72, 1996.
- [7] D. Gavrilu and L. Davis, Tracking Humans in Action : A 3D Model-Based Approach, *Proc. Image Understanding Workshop*, pp.737-746, 1996.
- [8] R. Howarth and H. Buxton, Visual Surveillance Monitoring and Watching, *Proc. European Conf. Computer Vision*, Vol.2, pp.321-334, 1996.
- [9] T. Frank, M. Haag, H. Kollnig, and H.-H. Nagel, Tracking of Occluded Vehicles in Traffic Scenes, *Proc. European Conf. Computer Vision*, Vol.2, pp.485-494, 1996.
- [10] R. C. Harrell, D. C. Slaughter, and P. D. Adsit, A Fruit-Tracking System for Robotic Harvesting, *Machine Vision and Applications*, Vol.2, pp.69-80, 1989.
- [11] D. Reynard, A. Wildenberg, A. Blake, and J. Marchant, Learning Dynamics of Complex Motions From Image Sequences, *Proc. European Conf. Computer Vision*, Vol.1, pp.357-368, 1996.
- [12] E. Bardinat, L. Cohen, and N. Ayache, Tracking Medical 3D Data With a Deformable Parametric Model, *Proc. European Conf. Computer Vision*, Vol.1, pp.317-328, 1996.
- [13] P. Shi, G. Robinson, T. Constable, A. Sinusas, and J. Duncan, A Model-Based Integrated Approach to Track Myocardial Deformation Using Displacement and Velocity Constraints, *Proc. Intl Conf. Computer Vision*, pp.687-692, 1995.
- [14] E. Boyer, Object Models From Contour Sequences, *Proc. European Conf. Computer Vision*, Vol.2, pp.109-118, 1996.
- [15] L. Shapiro, *Affine Analysis of Image Sequences*. Cambridge, England : Cambridge Univ. Press, 1995.
- [16] C. Tomasi and T. Kanade, Shape and Motion From Image Streams Under Orthography : A Factorization Method, *Intl J. Computer Vision*, Vol.9, No.2, pp.137-154, 1992.
- [17] Gregory D. Hager and Peter N. Belhumeur, "Efficient Region Tracking with Parametric Models of Geometry and Illumination," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, Vol.20, No.10, pp.1025-1039, 1998.
- [18] M. Jayakumar and Ravi N. Banavar, "Risk-Sensitive Filters for Recursive Estimation of Motion from Images," *IEEE TPAMI*, Vol.20, No.6, pp.659-666, June, 1998.
- [19] Paul L. Rosin, "Thresholding for Change Detection," technical report ISTR-97-01, pp.1-20, 1997.
- [20] 조영석, 이주신, "부분 외곽선 정보를 이용한 이동물체 추적 알고리즘", 정보처리학회논문집, 제8-B권 제5호, pp.539-548, 2001.
- [21] 고선영, "영역확장 기반추적을 이용한 효율적인 3차원 의료 영상 분할기법", 한국과학기술원 석사학위논문, 2000.
- [22] Paul L. Rosin, "Thresholding for Change Detection," Brunel University.
- [23] 채옥삼, 경태원, "이동물체 분리를 위한 Seed 선정 및 영역확장 알고리즘에 관한 연구", 신호처리학회논문집, 2001.



### 조영석

e-mail : yscho@cs.kdc.ac.kr  
 1991년 청주대학교 전자공학과 졸업(학사)  
 1993년 청주대학교 대학원 전자공학과 (공학석사)  
 1999년 청주대학교 대학원 전자공학과 박사과정(수료)  
 1998년~현재 극동정보대학 전산정보처리과 조교수  
 관심분야 : 디지털 영상처리, 영상 추적, 영상 감시시스템 등



### 이주신

e-mail : jushin2@chongju.ac.kr  
 1975년 명지대학교 전자공학과 공학사  
 1977년 홍익대학교 대학원 전자공학과 전자재료 전공 공학석사  
 1986년 명지대학교 대학원 전자공학과 영상처리 전공 공학박사  
 1980년~1982년 청주대학교 이공대학 전자공학과 학과장  
 1988년~1992년 청주대학교 이공대학 교학과장  
 1992년~1994년 청주대학교 산업과학연구소 연구소장  
 1995년~1998년 청주대학교 이공대학 학장  
 1998년~1999년 청주대학교 기획관리처 기획관리처장  
 1992년~현재 청주대학교 이공대학 첨단공학부 교수  
 관심분야 : 영상처리, 패턴인식, 이동물체추적, 퍼지/신경망 회로 영상통신 등