

I. 서론

영상추적은 영상 기반 제어, 사람-컴퓨터 인터페이스, 감시장치, 농업자동화, 의료영상분석, 영상복원 등을 포함하는 일부 응용분야에서 시스템 가장 중요한 요소 중 하나로 부각되고 있다. 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 프레임에서 이동물체의 위치를 예측하는 움직임 예측 기법 등을 포함하게 된다. 영상 열로부터 움직임 파라미터의 예측 연구가 활발히 진행되고 있다. 이들 연구의 방향은 특징 기반 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 이동물체를 구하여 이동영역을 추출하는 방법을 이용하여야 한다. 이동물체를 추적방법으로는 모델을 기반으로 하는 방법, 영역을 기반으로 하는 방법, 능동 윤곽선을 기반으로 하는 방법, 특징을 기반으로 하는 방법 등이 있다. 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 이동물체 추적은 물체의 경계를 윤곽선으로 표현하고 그것을 동적으로 갱신하면서 추적하는 기법으로 영역기반에 비해 복잡한 계산이 줄어들지만 추적 물체가 부분적으로 가려질 경우 추적이 어렵다는 단점이 있다. 특징기반 방법은 물체의 전체를 추적하지 않고 이동물체의 특징을 추출한 다음 추적하는 기법으로 부분적인 가려짐이 발생해도 추적이 가

능하다. 이동물체 외각선 추출 알고리즘에 대한 연구로서 이동벡터나 차 영상 구한 다음 일정한 기준 값과 비교하여 물체영역과 배경영역을 분할하는 방법이 이용되고 있으며, 확률분포함수를 적용하여 최적의 기준 값을 정하는 연구도 꾸준히 진행되고 있다. 고속으로 이동물체를 추적하기 위한 방법으로 제한된 영역에서 추적 창을 이용하는 방법이나 시공간적 상관관계에 의한 적응 탐색 방법 확률 분포함수를 이용하여 계산된 예측치를 바탕으로 신뢰 구간을 이용하는 방법들이 발표되고 있다. 한편 이동물체를 추적하기 위해 이동물체의 분할이 선행되어야 한다. 현재 이동물체의 분할기법으로는 이동벡터 중 크기와 방향성분을 이용하여 분할하는 기법이나 입력의 칼라정보를 이용하여 분할하는 방법 등이 활발히 연구되고 있다. 이들 방법 중 이동벡터를 이용하는 경우 이동벡터의 추출기법에 따라 성능이 달라지는 문제점을 내포하고 있으며, 입력 영상의 칼라정보를 이용하는 경우 그레이 영상에 비하여 처리 데이터 량의 증가로 고속화에 어려움이 내포되어 있다.

본 논문에서는 이동물체를 고속으로 추적하기 위한 이동물체 분할방법과 추적 알고리즘을 제안하였다.

이동물체의 분할은 이동벡터와 정합 오차영상을 추출, 영역 분할 그리고 영역 병합으로 구성된다. 이동벡터와 정합 오차영상을 추출단계는 현재 영상과 이전 영상으로부터 고속이동벡터 추출 알고리즘을 이용하여 이동벡터와 정합 오차영상을 추출한다. 고속 이동벡터 추출 알고리즘은 전역검색 블록매칭 알고리즘에서 연산량을 줄이기 위하여 블록매칭 알고리즘을 수행하기 전에 에지를 구하여 에지값이 기준값보다 큰 경우에만 BMA를 수행하도록 하였다. 이동물체의 분할의 첫 번째 단계는 Watershed 알고리즘을 사용하여 정합 오차영상을 영역 분할한다. 두 번째 단계로 정합 오차영상의 에지를 구하고 에지 히스토그램 분포도를 구한 후 최적의 확장 레벨을 계산하여 Seed 영역을 확장하여 영상을 분할하였다. 세 번째 단계로 영역 병합단계는 각각의 Seed 경계선들의 이동벡터를 비교하여 같은 크기와 방향을 갖는 경계선을 제거하여 최종 이동물체의 경계선을 추출하도록 수행하였다.

다음으로 추적알고리즘은 이동물체의 특징벡터 추출과 이동물체의 추적 그리고 특징벡터의 갱신으로 이루어진다. 이동물체 초기 특징 벡터는 전체 외각선 영역 중 상하좌우의 외각선 영역 영역에 일치하는 영상데이터의 일부를 특징벡터로 저장한다. 다음은 추적단계로 이전 프레임으로부터 구한 특징벡터를 현재 프레임에서 최적의 매

칭 영역을 검색하여 이동벡터를 구한다. 특징벡터 갱신은 다음 프레임에서 추적을 수행하기 위해 현재 프레임의 영상데이터로 특징 벡터를 갱신한다.

제안된 알고리즘의 당성을 검토하기 위해 table tennis 영상과 실외에서 운행되는 차량 영상을 가지고 이동물체 분할과 이동물체 추적 실험을 수행한다.

II. 이동 벡터 추출과 영역 분할

배경이 변화하는 상황에서 이동벡터의 연산은 이동물체가 존재하는 이전 영상과 현재 영상사이에 이동물체의 유사도가 가장 높은 영역을 검색하는 방법이 이용된다. 유사도가 높은 영역을 탐색하는 블록 매칭 알고리즘은 이전영상에서 설정된 탐색영역과 현재 영상에서 대하여 가장 잘 일치되는 위치를 검색하는 방법을 이용하게 된다. 이때 사용되는 매칭의 척도는 상호 상관계수(cross-correlation)나 MAD(mean absolute difference)가 이용되고 있다. MAD는 상호상관 계수에 비해 계산이 간단하며 정의는 식(1)과 같다.

$$MAD(x, y) = \frac{1}{N^2} \sum_{i=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{j=-\frac{N-1}{2}}^{\frac{N-1}{2}} \text{식(1)}$$

$$| I_{t-1}(k+i, l+j) - I_t(k+x+i, l+y+j) |$$

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

MAD를 이용한 (k, l) 번째 화소의 움직임 벡터 $v(k, l)$ 는 MAD가 가장 작은 값을 가지는 영역까지의 변위 (x, y) 이며 식(2)는 이동벡터의 계산식이다.

$$v_{(k, l)} = [(x, y) : \min MAD(x, y)] \text{식(2)}$$

이동벡터 계산 시 이용물체의 경계영역에서 큰 블록매칭 오차를 보이며 배경영역에서는 작은 오차를 보이는 특성을 이용하면 이동물체의 영역분할이 가능하다. 따라서

정합오차 영상을 구하여 이동물체 분할에 이용하고자 한다. (k, l) 번째 화소의 정합 오차 $Imm_{(k, l)}$ 은 최소값 $MAD(x, y)$ 로 식(3)과 같다.

$$Imm_{(k, l)} = \min MAD(x, y) \text{식(3)}$$

BMA를 이용하여 움직임 벡터를 계산할 경우 검색 방법에 따라 전체 영역을 검색하는 전역 탐색 블록매칭 알고리즘과 검색영역 중 일부만을 검색하는 부분탐색 알고리즘이 있다.

이들 중 전역 탐색 블록 매칭 알고리즘은 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 탐색하므로 정확한 이동벡터를 구할 수 있으나 계산이 많은 문제점이 있다. 부분탐색 알고리즘으로는 3-스텝 검색, 로그검색 등이 있으며 이들 부분검색 알고리즘은 이동물체의 이동방향으로 매칭 에러가 단조감소 한다는 가정으로 연산량이 적은 대신 국부 최소(Local minimum)에 빠질 경우 오차가 크다는 단점을 가지고 있다. 따라서 오차를 최소화하며 연산량을 감소시키는 연구가 필요하다. 또한 그림 1과 그림 2는 입력 영상이고 그림 3은 전역 탐색 블록 매칭 알고리즘에 의해 계산된 이동벡터 중 크기성분을 보였다. 그림 3에서 밝기는 이동벡터의 크기를 나타내는 것으로 밝은 부분은 이동벡터의 크기가 크고 검은 부분은 이동벡터가 존재하지 않는다는 것을 의미한다. 그리고 이동 벡터연산 과정에서 정합 오차영상을 구하여 그림 4에 표시하였다.

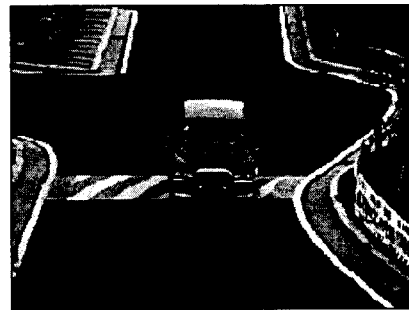


그림 2. 입력 프레임#1
Fig. 1. Input frame#1

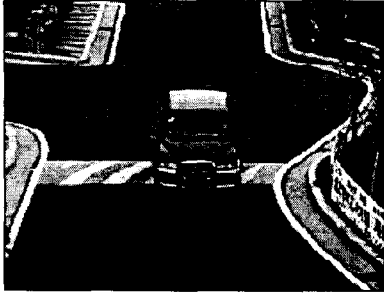


그림 3. 입력 프레임 #2
Fig. 2. Input frame #2

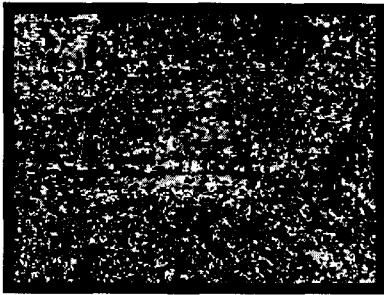


그림 4. 이동벡터영상
Fig. 3. Moving Vector Image

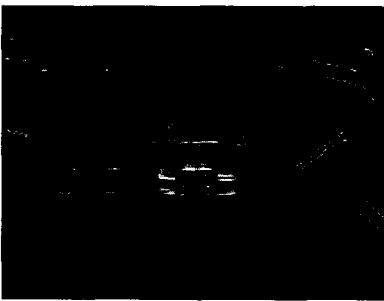


그림 5. 정합 오차영상
Fig. 4. Matching error Image

전역 탐색 블록 매칭 알고리즘의 계산 량을 줄이기 위한 방법으로 여러 가지 방법들이 제시되고 있으나 본 연구에서는 전역 탐색 기법의 장점을 그대로 유지하면서 연산량을 줄이기 위한 방법으로 탐색 점에서 공간 기울기인 에지를 구한 다음 에지 값이 기준 값보다 큰 경우 이동벡터를 계산하고자 한다. 표 1은 고속 이동벡터 계산 알고리즘(FBMA: Fast Block Matching Algorithm)이다.

표 1. 고속 이동벡터의 계산 알고리즘
Table 1. Algorithm of fast moving vector processing

Step 1: 검색좌표 초기화($x = 2, y = 2$)
Step 2: 탐색 블록 초기화($i=0, j=0$)
Step 3: 현재 탐색점의 에지 계산
Step 4: 에지 값이 기준 값보다 작으면 다음 검색좌표로 이동
Step 5: MAD 계산
Step 6: 이동 벡터 계산
Step 7: 정합 오차 계산
Step 8: i, j loop goto 3
Step 9: x, y loop step 2

고속 이동벡터 계산 알고리즘에서 후보 영역의 결정은 먼저 계산영역의 공간 기울기인 에지를 구한 다음 에지의 크기가 설정 값보다 큰 경우 고속 이동벡터를 구하도록 함으로서 연산량을 줄이도록 구성하였다. 그림 5와 그림 6은 Table Tennis 영상으로 입력이라면 그림 7은 전역 검색 BMA에 의한 이동벡터이고 그림 8은 제안된 알고리즘에서 이동벡터를 계산할 후보영역이며, 그림 9는 고속 이동벡터 알고리즘에 의해 계산된 이동벡터이다.

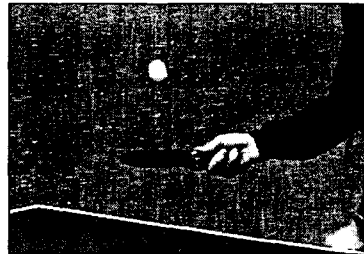


그림 6. 입력 영상(프레임 #00)
Fig. 5. Input image(Frame #00)

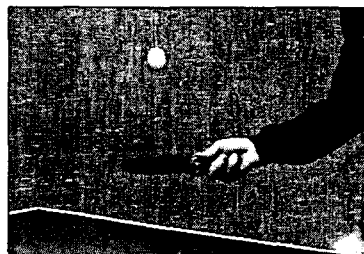


그림 7. 입력영상(프레임 #01)
Fig. 6. Input image(frame#01)

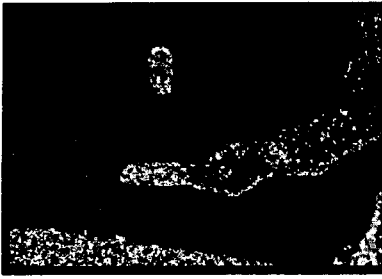


그림 8. 이동벡터 계산(BMA)
Fig. 7 Moving vector(BMA)

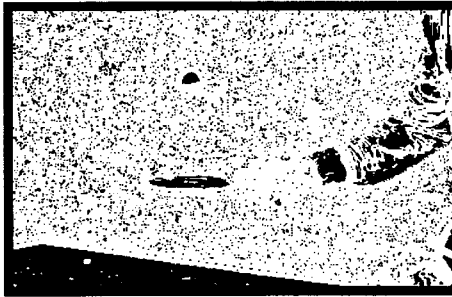


그림 9. BMA 후보화소(흰색영역)
Fig. 8. BMA candidate pixel(white)

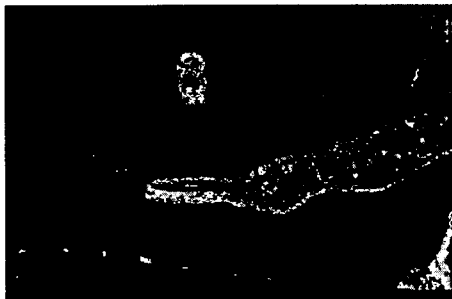


그림 10. FBMA에 의한 이동벡터 계산
Fig. 9. Moving vector image with FBMA

Watershed 알고리즘은 지형학분야에서 연구된 것으로 영상에 대한 에지 값을 고도정보로 처리함으로써 영상 처리에 응용되고 있다. 이 알고리즘은 담수지역 또는 Seed 영역을 구분하는 분수령을 찾음으로 각각의 국부최소치에 관련된 영역을 분할하는 알고리즘이다. 워터 슈드 알고리즘은 여러 가지가 연구되고 있으며 모의 침수(immersion simulation) 기법이 가장 효과적인 것으로 알려져 있다.

모의 침수 기법은 낮은 고도의 영역으로부터 물을 채워 나간다. 물이 다른 영역과 만나게되면 경계에 가상의 독을 쌓는 기법으로 영역을 분할한다.

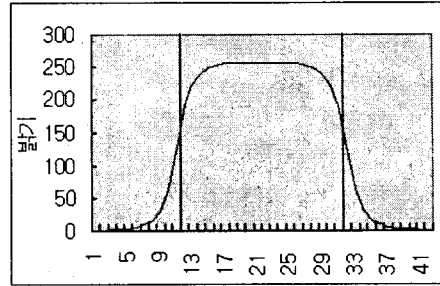


그림 11. 영상의 1차원 단면도
Fig. 10. Cross-section on Image

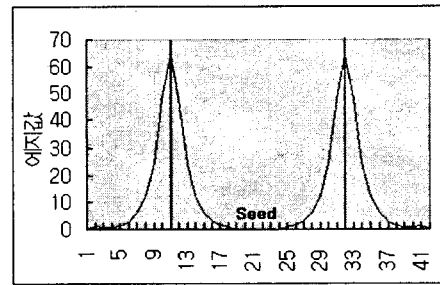


그림 12. 그림 2의 에지 단면
Fig. 11. Edge section of Fig. 2

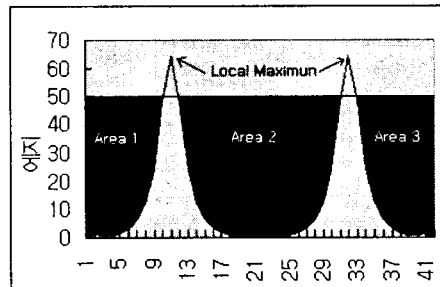


그림 13. Seed 영역 확장
Fig. 12. Growth of seed region

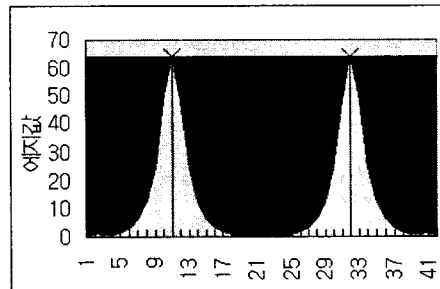


그림 14. 경계 마스크(두 영역이 만나는 곳)
Fig. 13. Boundary Mask(Contact Point)

그림 10은 영상의 1차원 단면을 표시한 것이고 그림 11은 그림 10에 대한 예지 단면이며 그림 12는 Seed 영역을 확장한 것으로 그림 10의 영상이 3개의 영역으로 나누어진다. 계속해서 Seed 영역을 증가시키면 영역1과 영역2가 만나며 만나는 점을 경계선으로 표시하면 영역을 분할하게 된다. 그림 13은 두 영역이 만나는 영역을 경계선으로 마크하는 것을 표시하고 있다.

워터 쉼(watershed) 알고리즘을 이동물체의 이동영역 분할을 위해 첫 번째로 입력벡터 에러영상의 히스토그램을 구하고 이동 벡터영상에 대한 Seed 영역을 계산한다. Seed 영역을 확장하여 이동물체의 윤곽선을 검출한다. Seed 영역은 답수가 시작되는 곳으로 최적의 국부 최소값을 결정하여야 한다. 그림 14는 워터셰드를 이용한 이동물체 분할 처리 순서도이다.

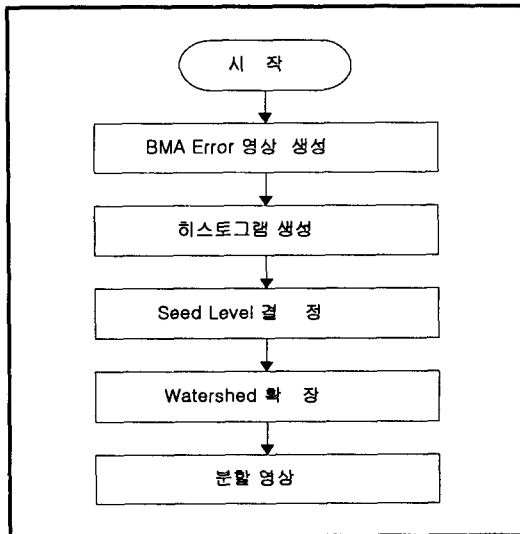


그림 15. 워터셰드에 의한 이동물체 분할
Fig. 14. Moving object segmentation with 워터셰드

이동벡터의 연산에서 발생하는 정합 오차 영상 I_{mm} 은 식(4)와 같이 표현할 수 있다.

$$I_{mm} = MAD_{Min}(I_t, I_{t+1}) \quad \text{식(4)}$$

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(5)와 같다.

$$\begin{aligned} I_t &= I_{bt} + I_{mt} + I_{nt} \\ I_{(t+1)} &= I_{b(t+1)} + I_{m(t+1)} + I_{n(t+1)} \end{aligned} \quad \text{식(5)}$$

따라서 입력 프레임 I_t 와 $I_{(t+1)}$ 이 최적으로 정합되었을 때 정합오차 영상 I_{mm} 은 식(5)를 이용하여 다시 정리하면 식(6)과 같다.

$$\begin{aligned} I_{mm} &= MAD(I_t, I_{(t+1)}) \\ &= MAD(I_{bt} + I_{mt} + I_{nt}) - [I_{b(t+1)} + I_{m(t+1)} + I_{n(t+1)}] \end{aligned} \quad \text{식(6)}$$

위의 식(5)에서 최적으로 정합되었고 배경영역의 변화가 없다고 가정하면 식(4)는 식(7)과 같이 정리할 수 있다.

$$\begin{aligned} I_{mm} &= MAD(I_{bt}, I_{b(t+1)}) + MAD(I_{ot}, I_{o(t+1)}) + MAD(I_{nt}, I_{n(t+1)}) \\ &= 0 + MAD(I_{ot}, I_{o(t+1)}) + MAD(I_{nt}, I_{n(t+1)}) \\ &= A_{\alpha(\text{이동물체 성분})} + A_n(\text{잡음 성분}) \end{aligned}$$

식(7)에서 잡음 A_n 은 ZMND(Zero Mean Norm Distribution) $N(0, \sigma^2)$ 로 모델링이 가능하다. 그림 15는 잡음의 정규 분포함수 이고 정합 오차 영상의 히스토그램을 구하면 그림 16과 같다.

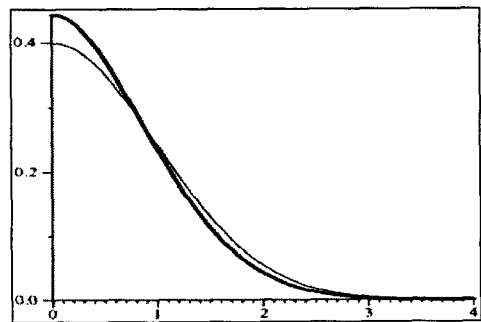


그림 16. 잡음의 확률 분포함수 모델링
Fig. 15. plot of Noise PDF

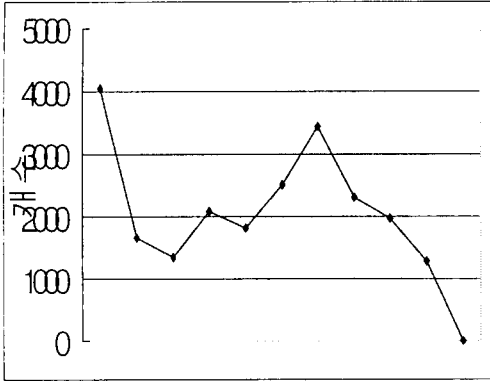


그림 17. 매칭 에러영상 히스토그램
Fig. 16. Histogram of matching error

그림 16에서 Seed 영역을 결정하기 위한 임계값(Th)은 다음 식(8)과 같이 정하였다.

$$Th = 1.65 \sigma \quad \text{식(8)}$$

이동물체 추적방식 중 능동 윤곽선 모델은 전체 영상에 대하여 이동벡터를 구하지 않고 이동물체의 외각선에 대하여 이동벡터를 구하기 때문에 처리속도가 빠른 특징이 있다. 워터셰드 의 문제점인 과분할(Over Segmentation)을 해결하기 위해 윤곽선 영역에 대하여 이동벡터의 크기와 방향이 같으면 이웃 영역을 병합하여 최종 이동물체 외각선을 구한다. 그림 17은 기존 알고리즘에 의한 이동물체의 영역분할에 대하여 보였고 그림 18은 워터 쉬이드 알고리즘과 이동벡터를 이용하여 영역 분할한 예이다.

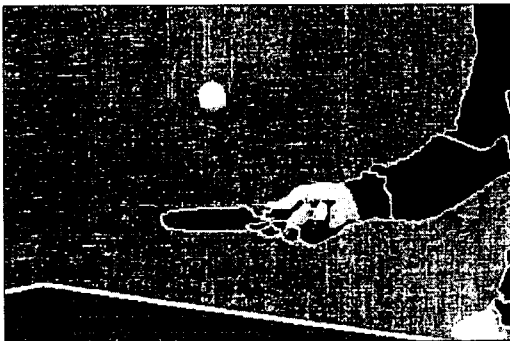


그림 18. 기존알고리즘에 의한 이동물체 분할
Fig. 17. Moving object segmentation with traditional algorithm

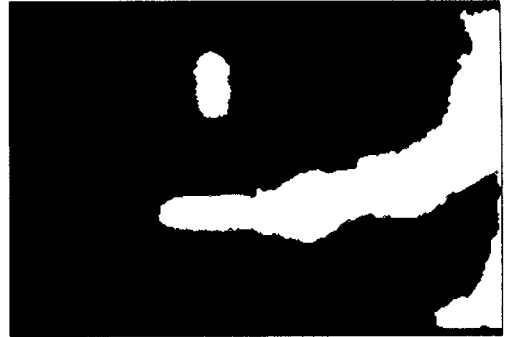


그림 19. 이동영역 분할
Fig. 18. Moving region segmentation

III. 이동물체 고속추적 알고리즘

초기 이동물체의 이동벡터에 의하여 이동물체가 검출되면 다음단계로 이동물체의 외각선으로부터 특징벡터를 구한다. 3차원공간에서 이동하는 이동물체가 2차원 공간으로 투시 변환되면 시간에 따라 변화하기 때문에 이동물체의 경계선이 변화하게 된다. 외각선 정보를 이용하여 이동 물체의 추적하기 위하여 경계선영역을 추출한다. 이동물체의 경계영역을 3가지부분으로 나누면 이동물체 내부영역과 외각선 영역 그리고 이동물체 외부영역인 배경영역으로 나눌 수 있으며, 현재 프레임에서 이동물체의 경계선을 정보를 검출하고 다음 프레임에서 경계를 검색하여 이동물체를 추적한다. 그림 19는 이동물체의 검색영역과 경계영역 그리고 배경영역에 대하여 표시하였다.

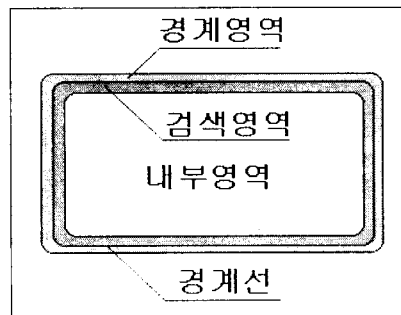


그림 20 이동물체의 영역
Fig. 19 Regions of moving object

능동 윤곽선모델에서 경계선의 일부분이 다른 이동물체에 가려질 경우 추적이 곤란한 문제점을 해결하기 위한 방법으로 특징기반 영상추적모델에서 이용하는 곡률이나 분기점 교차점과 같은 특징요소를 검색하는 대신 상하좌우 경계선을 특징벡터로 하여 검색함으로써 일부 경계선이 가려질 경우에도 이동물체를 추적할 수 있도록 구성하고자 한다. 입력 영상이 그림 20과 같을 때 그림 21과 같이 이동물체의 경계선을 구하고 상하 좌우의 경계선으로부터 특징벡터를 구하였다. 표 3은 이동물체의 경계영역의 상하좌우를 $(11 \times 3), (3 \times 11)$ 크기의 이동물체 영상 데이터로부터 특징 벡터로 사용하기 위해 저장한다. 그림 22는 특징벡터 영역이고 표 2는 이동물체 추적을 위한 특징벡터의 예이다.

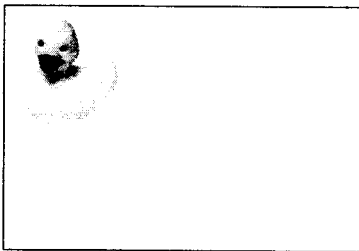


그림 21 입력영상
Fig. 20. Input image

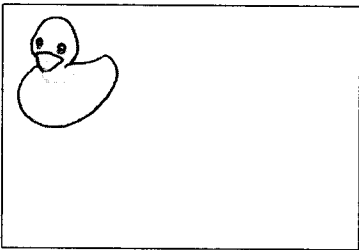


그림 22 이동물체 경계검출
Fig. 21. boundary detect

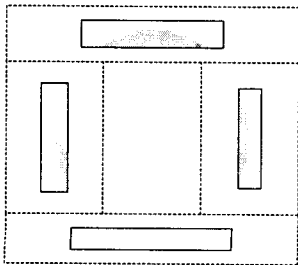


그림 22 특징벡터 영역
Fig. 22 Region of feature vector

표 2 특징벡터의 예
Table. 2 Example of feature vector

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

두 번째 단계로 특징벡터와 현재 프레임사이의 MAD를 이용하여 이동물체를 추적한다. 표 3과 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.

세 번째 단계로 특징 벡터를 갱신한다. 이동물체는 추적 과정에서 3차원 회전이나 이동에 의해 외각선이 변하게 된다. 본 논문에서는 추적 대상의 형태변화에 적응하기 위하여 매 프레임마다 특징 벡터를 갱신한다. 좌표 계산이 종료되면 이동물체에 대한 외각선 정보를 취득하여 다시 특징 벡터로 저장한다.

IV. 실험 및 고찰

본 실험은 고속이동물체 추적 알고리즘의 성능 평가기 위하여 352×240 크기의 stennis 영상과 용인근처의 각 도로에서 디지털 캠코더로 직접 얻은 $[310 \times 240 \times 8B]$ 크기의 영상을 이용하였고 PIII-500 이 장착된 컴퓨터에서 C언어를 이용하여 프로그램을 작성하여 수행하였다. 그림 23은 처리 순서도이다.

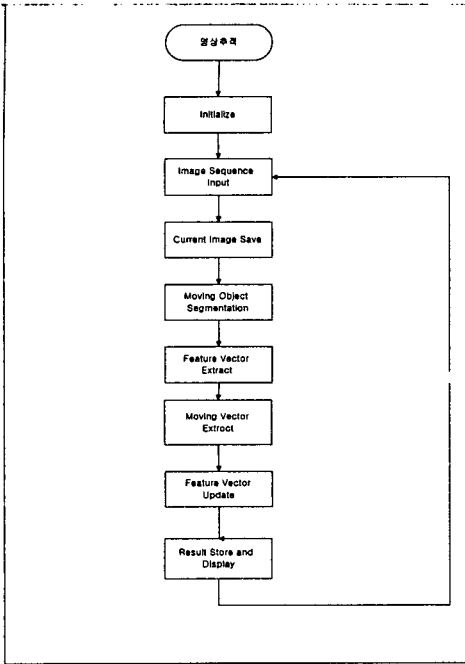


그림 23 이동물체 추적 알고리즘에 대한 순서도
 Fig. 23 Flowchart of moving object tracking al

그림 24는 입력 실험 영상이다. 입력 영상에 대한 이동벡터 계산은 고속 이동벡터 추출 알고리즘을 이용하였으며 검색영역의 크기가 3×3 이고 탐색영역의 크기가 15×15 로 정하여 수행하였다. 전역 탐색 알고리즘과 제안된 고속 이동벡터 계산 알고리즘의 연산 량을 비교하면 다음 표 3과 같다.



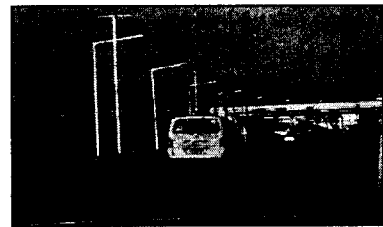
(a) Frame #01



(b) Frame #03



(c) Frame #05



(d) Frame #07



(e) Frame #09



(f) Frame #11

그림 24 실험 영상
 Fig. 24. Experience Image

표 3. 전역탐색알고리즘과 제안된 알고리즘의 단위 화소당 연산량 비교

Table 3. Calculation compare with full search BMA an algorithm

항목	기존전역탐색 BMA	제안된 알고리즘의 계산량	
		조건 판별 시	이동벡터 계산 시
가산/감산	2025((3×3)×15×15)	11	2025
승산/제산	0	2	0
비교	2250	1	2250

표 3에서 전역검색 BMA의 경우 66,896번 BMA 계산하였으며 제안된 알고리즘은 평균 40500회 BMA 계산을 수행함으로써 39%의 BMA 연산을 감소됨을 보였다. 표 4는 실험 영상에 대한 기존 BMA와 제안된 알고리즘의 연산량 비교한 결과를 보였다.

표 4 실험 영상에 대한 기존 BMA와 제안된 알고리즘의 연산량 비교

Table 4. Calculation compare with full search BMA and proposed algorithm on experience image

항목	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

그림 25는 기존의 전역검색 BMA에 의한 이동벡터를 추출한 것이고 그림 26은 고속 이동벡터 추출알고리즘에 의한 벡터추출 결과 영상이다.



그림 25. 이동벡터 추출(BMA)

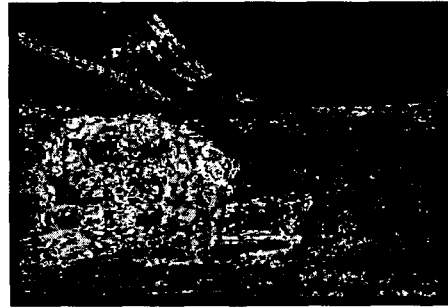


그림 26. FBMA에 의한 이동벡터 계산

연산결과 이동벡터의 크기 히스토그램은 그림 27과 같고 이동영역 분할결과 기존 알고리즘에 의한 이동영역 분할은 그림 28과 같다. 그림 29는 입력영상을 이동벡터의 워터셰드 알고리즘에 의해 분할한 것이고, 그림 30은 이동영역에서 이동물체의 윤곽선 정보의 예이다. 제안된 알고리즘에 의한 영역분할은 기존알고리즘에 비하여 그림자 부분이 제거되어 정확한 영역분할 됨을 보였다.

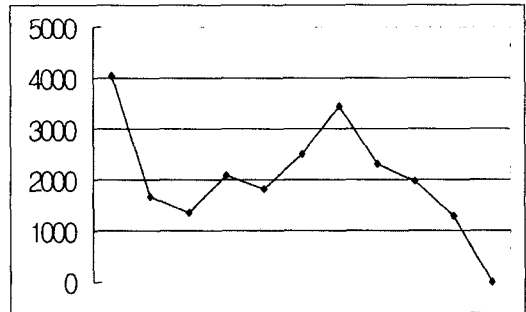


그림 27 이동벡터 크기 히스토그램

Fig. 27. Histogram of moving vector's amplitud



그림 28 기존알고리즘에 의한 영역분할
Fig. 28. Segmentation of traditional algorithm

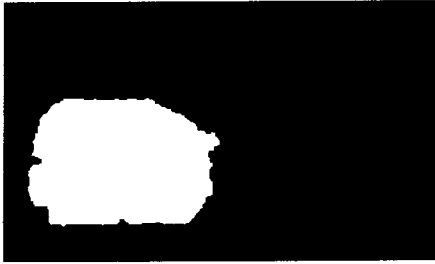


그림 29 제안된 알고리즘에 의한 영역분할
Fig. 29. Segmentation of proposed algorithm

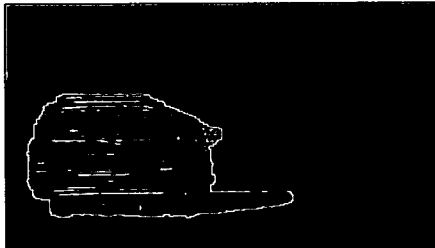


그림 30 이동영역에서 외각선 검출
Fig. 30 Boundary line on moving region.

이동물체 추적을 위한 특징벡터는 이동물체의 경계영역 중 상하좌우 데이터를 이용하였다. 다음의 표 5부터 표 8은 특징 벡터의 크기별 추적 결과를 표시하였다.

표 5 (5X3) 특징 벡터에 의한 추적 오차
Table 5. Tracking error with feature vector of (5X3)

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 6 (5X5) 특징 벡터에 의한 추적 오차
Table 6. Tracking error with feature vector of (5X5)

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 7 (10X3) 특징 벡터에 의한 추적 오차
Table 7. Tracking error with feature vector of (10X3)

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	109	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 8 (15X5) 특징벡터에 의한 추적오차
Table 8. Tracking error with feature vector of (15X5)

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	142	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

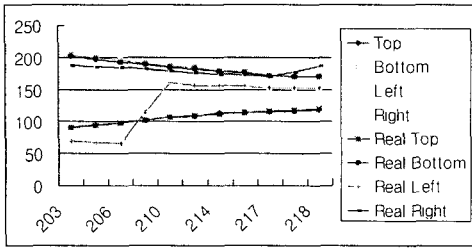


그림 31. 이동물체의 검색좌표와 실제좌표
Fig. 31. Search point and Real point of moving

그림 31은 특징 벡터 크기별로 실제 좌표와 검색좌표의 오차를 표시한 것으로 $[10 \times 5]$ 크기의 특징 벡터를 이용하여 이동물체를 추적한 결과 가장 양호한 결과를 얻었으며, 오차가 2화소 미만으로 나타났다. 전체 외각선을 검색하는 기존의 알고리즘과 제안된 알고리즘과의 연산량을 비교하면 초기 23760회의 검색 연산이 필요하였고 이동물체가 이동함에 따라 크기가 작아져 20820회, 15075회, 10752회로 감소하였으며 제안된 알고리즘의 경우 물체의 크기에 관계없이 특징벡터의 크기에 따른 검색 연산이 필요하였다. 그림 32는 기존 알고리즘과 제안된 알고리즘의 연산량을 비교한 것이다.

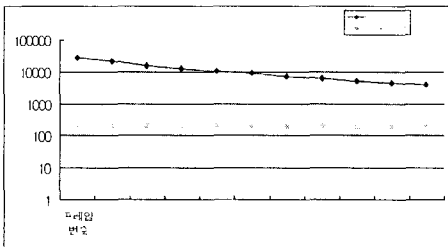


그림 32 기존 알고리즘과 제안된 알고리즘의 연산량 비교
Fig. 32 Calculation compare with traditionally al proposed algorithm

제안한 이동물체 분할과 이동물체 추적 알고리즘의 타당성을 검증하기 위해 table tennis 영상과 실외에서 운행되는 자동차영상을 가지고 실험한 결과 다음과 같은 결론을 얻었다.

1. 이동벡터 계산 시 고속 이동벡터 알고리즘을 이용한 결과 stennis 영상에서는 약 17%, 실제로 촬영된 영상의 경우 39%의 BMA 연산량이 감소됨을 보였다.
2. 이동물체 분할은 정합 오차영상에 위트체드 알고리즘을 적용하였고 국부 최소 영역을 검색하여 Seed 영역을 생성한 다음 최적 확장 레벨을 계산하여 Seed 영역을 확장 후 이동벡터를 이용하여 영역을 병합하도록 수행하였다.
3. 기존 분할 알고리즘과 제안 알고리즘을 비교한 결과 제안된 알고리즘이 우수한 이동물체 분할 성능을 보였다.
3. 이동물체의 특징벡터는 이동물체 상하좌우 외각선 영역에 해당하는 영상데이터의 일부로서 정하였다.
4. 특징벡터의 크기는 $[5 \times 5]$, $[10 \times 3]$, $[10 \times 5]$, $[15 \times 5]$ 로 변화시키며 실외에서 운행되는 자동차영상의 이동물체를 추적한 결과 $[10 \times 5]$ 일 때 2화소 이하로 정확한 이동물체 추적이 가능하였다.
5. 실제 영상에 대하여 기존의 전체 외각선을 검색하는 기존의 경우 20820회의 연산이 필요하였으나 제안된 알고리즘을 적용하여 이동물체를 추적에는 200회의 검색 연산이 필요하였다.

향후 연구과제로는 다양한 모델에 대한 이동 물체 추적과 고속 이동 물체 추적을 위한 연구라 하겠다.

V. 결론

본 논문에서는 이동물체를 고속으로 추적하기 위한 이동물체 분할 방법과 추적 알고리즘을 제안하였다.

참고문헌

- [1] P. Allen, B. Yoshimi, and A. Timcenko, Hand-Eye Coordination for Robotics Tracking and Grasping, K. Hashimoto, ed., Visual Servoing, pp. 33-70. World Scientific, 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. Int'l 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 Computer Vision and Pattern Recognition, pp. 67-72, 1996.
- [7] D. Gavrila 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. Bardinet, 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. Int'l 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, Int'l 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, JUNE 1998, pp659-666.
- [19] Paul L. Rosin, "Thresholding for Change Detection", technical report ISTR-97-01, pp1-20, 1997
- [20] 고선영, "영역확장 기반추적을 이용한 효율적인 3차원 의료영상 분할기법" 한국과학기술원 석사 학위논문, 2000
- [21] Paul L. Rosin, "Thresholding for Change Detection", Brunel University.
- [22] 채옥삼, 경태원, "이동물체 분리를 위한 Seed 선정 및 영역확장 알고리즘에 관한 연구", 신호처리학회 논문집, 2001년 12월9일.
- [23] Demin Wang, "Unsupervised Video Segmentation Based on 워터셰드s and Temporal Tracking", IEEE TCSVT, Vol. 8, No.5, Sep. 1998, pp539-546

저 자 소개



조 영 석

1991년 청주대학교 전자공학과 졸업(학사)
 1993년 청주대학교 대학원 전자공학과(공학석사)
 1999년 청주대학교 대학원 전자공학과 박사과정(수료)
 1998-현재 극동정보대학 전산정보처리과 조교수
 2002-현재 충북 전산인 협회 이사
 관심분야: 디지털 영상처리, 영상 추적, 영상 감시시스템 등



이 주 신

1975.5. 명지대학교 전자공학과 공학사
 1977.2. 홍익대학교 대학원 전자공학과 전자재료 전공 공학석사
 1986.2. 명지대학교 대학원 전자공학과 영상처리 전공 공학박사
 1980. 03 - 1982. 3 청주대학교 이공대학 전자공학과 학과장
 1988. 03 - 1992. 2 청주대학교 이공대학 교학과장
 1992. 12 - 1994. 2 청주대학교 산업과학연구소 연구소장
 1995. 08 - 1998. 2 청주대학교 이공대학 학과장
 1998. 02 - 1999. 1 청주대학교 기획관리처 기획관리처장
 1992. 02 - 현재 청주대학교 이공대학 첨단공학부 교수
 주요연구분야 : 영상처리, 패턴인식, 이동물체추적, 퍼지/신경망 회로 영상통신 등