

論文2000-37SP-5-8

## Hausdorff 거리를 이용한 이동물체 추적

(Tracking Moving Object using Hausdorff Distance)

金太植\*, 李周信\*\*

(Tea-sik Kim and Ju-Shin Lee)

## 요 약

본 논문에서는 배경이 움직이는 자연환경에서 모델기반 이동물체 추적 알고리즘을 제안하였다. 제안한 알고리즘은 이동물체의 형태변화에 적응하기 위하여 Hausdorff 거리를 모델과 영상사이의 유사도로 사용하였으며, 이동물체의 위치탐색시간을 줄이기 위하여 2D-Logarithmic 탐색기법을 사용하였다. 이동물체 추적 실험은 도로에서 주행하는 차량을 대상으로 수행하였다. 그 결과 자동차 영상과 오토바이 영상에서 실제위치와 추적결과에 대한 평균지승오차는 각각 1.15와 1.845로 이동물체의 정확한 추적이 가능함을 알 수 있었다.

그리고 추적 정합 시도 회수는 제안한 알고리즘이 기존알고리즘 보다 자동차 영상에서는 평균 11.25회, 오토바이 영상에서는 평균 5.23회 적게 시도하여 추적시간을 단축할 수 있음을 알 수 있었다.

## Abstract

In this paper, we propose a model based moving object tracking algorithm in dynamic scenes. To adapt shape change of the moving object, the Hausdorff distance is applied as the measurement of similarity between model and image. To reduce processing time, 2D logarithmic search method is applied for locate the position of moving object. Experiments on a running vehicle and motorcycle, the result showed that the mean square error of real position and tracking result is 1.150 and 1.845; matching times are reduced average 11.25times and 5.23 times than existing algorithm for vehicle image and motorcycle image, respectively. It showed that the proposed algorithm could track the moving object accurately.

## 1. 서 론

최근 무인감시시스템, 보안 출입관리, 공장자동화, 지능형 교통시스템 등 다양한 산업분야에서 이동물체 검출 및 추적 시스템에 대한 필요성이 증가되고 있으며, 일부는 이미 실용화가 추진되고 있는 실정이다. 이동물

체 추적 시스템은 동영상의 각 프레임에 대한 분석을 통하여 추적대상을 검출하고 위치변화를 계산하여 이동물체의 이동방향, 속도 등과 같은 정보를 추출하거나, 카메라를 통하여 지속적으로 이동물체를 감시할 수 있도록 제어한다<sup>[1][4]</sup>. 이동물체를 추적하기 위해서는 입력된 영상에 대하여 분석, 예측 및 판단을 진행하여야 하는데, 적용하는 방법에 따라 차영상을 이용한 방법, 광류를 이용한 방법, 모델기반 이동물체 추적 방법 등으로 나눌 수 있다.

Jane은 연속적인 영상 프레임의 차영상을 이진화 한 뒤 CCL(Connected Component Labeling)에 의해 이동물체를 분리하는 방법을 제안하였고<sup>[2]</sup>, Leung는 연속적인 입력 영상에서 배경을 추정된 뒤 현재프레임과 추

\* 正會員, 晉州專門大學 電子計算 電子系列

(Dept. of Computer &amp; Electronic, Chinju College)

\*\* 正會員, 淸州大學校 電子工學科

(Dept. of Electronic Engineering, Chongju University)

接受日字: 1999年9月20日, 수정완료일: 2000年4月21日

정된 배경의 차이에 임계치 연산을 적용해서 움직이는 사람을 검출하는 방법을 제안했다<sup>[3]</sup>. 그러나 이러한 차영상을 이용한 이동물체 검출 및 추적 알고리즘은 비교적 알고리즘이 간단하고 처리속도가 빠르다는 장점은 있지만, 배경이 변하거나 카메라가 움직이는 경우에는 그 적용이 힘들며, 정확한 이동물체의 판별을 위하여 퍼지 추론이나 신경회로망기법을 결합하여 사용한다<sup>[8-9]</sup>. Horn과 Schunck는 광류기반 이동물체 추적 방법을 제안하였으며, Mae등은 광류기반 실시간 이동물체 추적 시스템을 구현하였다<sup>[4]</sup>. 그들은 GGM(Generalized Gradient Model)을 사용하여 광류를 계산하였으며, 실시간 추적을 위하여 8개의 DSP칩(TMS320C40)을 사용하였다. 이 방법은 비록 움직임 추정에 높은 정확도를 보이나 프레임간의 움직임이 작은 경우에 적용이 가능하며 또한 계산량이 방대하기 때문에 실시간 구현이 어려운 단점을 보인다.

모델기반 기법은 사전에 정의된 특정한 이동물체의 모델이나 추적 초기 단계에서 취득한 추적대상의 모델링된 특징들의 프레임간 정합에 의해 이루어지므로 프레임간의 움직임이 클 경우에도 적합하며, 휘도의 변화, 배경의 움직임 등의 영향을 적게 받으나 물체의 형태가 프레임간에 유지되어야 한다는 제약성이 있다. 따라서 모델기반 기법은 추적과정에서 이동물체의 형태변화에 적응하는 능력이 추적성능을 좌우하게 된다.

본 논문에서는 배경이 움직이는 자연환경에서 모델기반 이동물체 추적 알고리즘을 제안하였다. 제안한 알고리즘은 인터프레임간의 차영상에 대한 가산투영<sup>[11]</sup>을 가지고 이동물체의 영역을 검출하여, 이 영역에 대한 윤곽선 영상을 이동물체의 모델로 하였다. 제안한 알고리즘은 이동물체의 형태변화에 적용하기 위하여 Hausdorff 거리를 모델과 영상사이의 유사도를 결정하는데 사용하였으며, 이동물체의 위치탐색시간을 줄이기 위하여 2D-Logarithmic 탐색기법을 사용하였다. 제안된 방법의 타당성을 검토하기 위하여 실제 도로에서 주행하는 자동차와 오토바이 영상을 대상으로 이동물체를 추적하고자 한다.

## II. 하우스돌프 거리와 2D-Logarithmic 탐색

### 1. 하우스돌프 거리에 의한 물체의 정합

하우스돌프 거리는 두 개 유한집합 내 점들의 유사

정도를 비교하기 위한 Min-Max거리이다<sup>[1]</sup>. 유한개의 점을 포함하는 두 개의 집합  $P$ 와  $Q$ 의 하우스돌프 거리는 식(1)과 같이 정의한다.

$$H(P, Q) = \max(h(P, Q), h(Q, P)) \quad (1)$$

여기서  $h(P, Q)$ 는  $P$ 에서  $Q$ 까지의 Min-Max거리이며 식 (2)와 같이 구할 수 있다.  $\|b - d\|$ 는 점  $b$ 와  $q$  사이의 거리를 나타내기 위한 척도로서 일반적으로 기하학적거리(Euclidean Distance)를 사용한다.

$$h(P, Q) = \max_{b \in P} \min_{q \in Q} \|b - d\| \quad (2)$$

$h(P, Q)$ 는 Max연산의 고유적인 특성으로 인하여 집합 내의 고립점(Isolated Clutter)들에 대해 매우 민감하다. 즉, 중심으로부터 멀리 떨어져있는 하나의 점이 하우스돌프 거리에 큰 영향을 줌으로 노이즈에 매우 약하다. 따라서 점들간의 최대거리를 선택하는 대신에  $k$ 번째로 큰 값을 취하는데 이것을 부분 하우스돌프 거리(Partial Hausdorff Distance)라고 하며, 식 (3)과 같이 정의한다.

$$h_k(P, Q) = \max_{b \in P} \min_{q \in Q} \|b - d\| \quad (3)$$

부분 하우스돌프 거리(Partial Hausdorff Distance)는  $P$ 의 부분집합의 매개 점으로부터  $Q$ 의 일부분 점까지의 거리가  $d$ 보다 작으면  $h_k(P, Q) = d$ 이다. 즉  $P$ 의 일부분의 점들은  $Q$ 의 어떤 점들과도 정합되지 않아도 된다. 이 특성은 본 연구에서 추적대상의 형태 변화를 허용하는 근거가 된다. 즉,  $P$ 에서는 최소한  $K$ 개의 점들이 식 (4)를 만족한다면  $P$ 와  $Q$ 는 동일 물체로 판별하게 된다.

$$h_k(P, Q) \leq \delta \quad (4)$$

식 (4)는 직접 계산하면 많은 계산량을 필요로 하기에 다음과 같이 변형하여 계산할 수 있다.

여기서  $\delta$ 는 이동물체의 형태변화의 허용정도를 정의하는 변수로써,  $\delta = 0$ 인 경우에는 형태의 변화를 허용하지 않으며,  $\delta = n$ 인 경우에는 디지털 영상에서 이동물체를 구성하는 점들이 전후 프레임 사이에서 최대로  $n$  픽셀까지 위치 변화를 가질 수 있음을 의미한다.

식 (5)에 보여준 바와 같이  $C_\delta$ 를 반경이  $\delta$ 인 원과

원내 모든 점들의 집합이라 하고,  $Q'$ 는  $Q$ 와  $C_\delta$ 의 Minkowski합이라 할 때,  $P$ 가  $Q'$ 의 부분집합이면 식 (4)를 만족하게 된다.

$$h(P, Q) \leq \delta \Leftrightarrow P \subseteq Q' \text{ where } Q' = Q \oplus C_\delta \quad (5)$$

2. 2D-Logarithmic 탐색

2D-Logarithmic 탐색 알고리즘은 DMD(Direction of Minimum Distortion) 방식으로도 불리는데 기본 개념은 표 1과 같이 세 단계에 걸쳐 탐색 영역을 효과적으로 줄여가면서 진행한다.

표 1. 2D-Logarithmic 탐색 순서  
Table 1. 2D-Logarithmic search.

step 1	중앙과 중앙으로부터 두 화소만큼 떨어진 상하좌우에서 정합을 시도한다. 평가함수(Cost Function)의 값이 최소인 블록이 중앙지점이면 단계 3을 수행하고 그렇지 않으면 단계 2를 수행한다.
step 2	단계 1에서 최소의 오차를 갖는 지점을 중심으로 하여 단계 1과 같은 과정이 반복된다.
step 3	이전에 선택된 지점을 중앙으로 하여 그 중앙지점과 그로부터 한 픽셀 거리의 주변에서 9번의 정합을 시도한다. 이 때에 결정된 평가함수의 값이 최소인 위치가 탐색 결과가 된다.

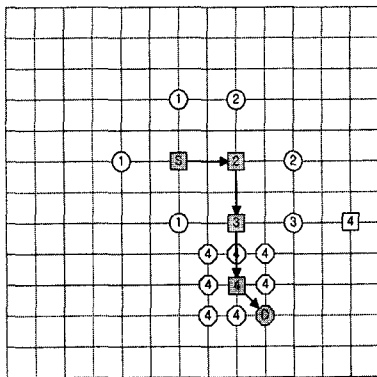


그림 1. 2D-Logarithmic 탐색 방법의 예  
Fig. 1. An example search by using 2D-Logarithmic algorithm.

본 논문에서는 2D-Logarithmic 탐색기법을 사용하여 이동물체의 위치를 탐색하고자 한다. 이 방법은 인접한 두 프레임의 영상사이에서 이동물체의 위치변화가 이

동물체의 지름보다 작아 겹치는 영역이 존재한다고 가정한다. 그림 1은 2D-Logarithmic 탐색의 한 예를 보여준다.

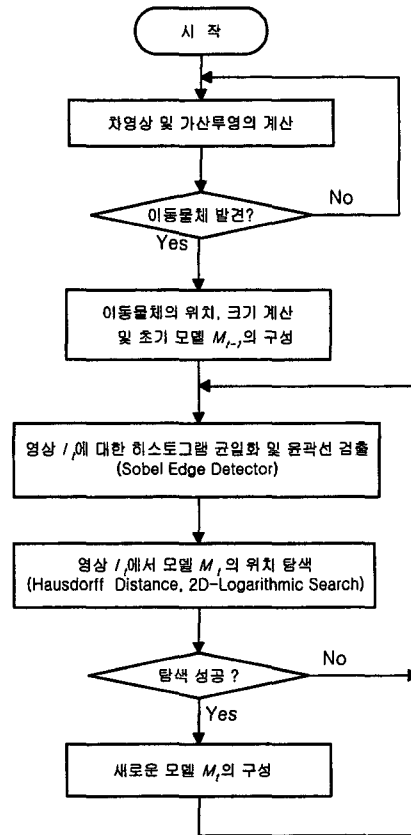


그림 2. 제안한 알고리즘의 흐름도  
Fig. 2. Flowchart for proposed algorithm.

III. 제안한 이동물체 추적 알고리즘

제안한 알고리즘은 검출 단계와 추적단계로 나눌 수 있다. 검출 단계에서는 연속된 두 프레임에 대한 차영상을 구하고 이 차영상에 가산투영 기법을 적용하여 이동하는 물체를 검출하여 추적마스크를 구성하였으며, 추적단계에서는 각 프레임에서 윤곽선 영상의 일부분 영역과 검출단계에서 구해진 추적마스크와의 하우스돌프 거리를 계산하여 추적 대상의 위치 및 크기를 판별하였다. 추적마스크 갱신은 현재 프레임에서 모델과의 하우스돌프 거리가 임계값보다 작은 화소들을 가지고 구성하였다. 추적마스크는 추적대상의 모델에 대한 윤곽선 정보를 포함한다. 그림 2는 제안한 알고리즘의 전

체 흐름도를 보여준다.

### 1. 이동물체의 검출 및 초기 모델의 생성

검출단계에서 이동물체를 검출하기 위하여 본 논문에서는 차영상에 대한 가산투영을 구하여 일정한 임계값을 초과하면 이동물체가 발견된 것으로 판별하였다. 영상에 대한 가산투영(Integral Projection)은 임의의 방향을 따라 존재하는 화소들의 적분으로 정의된다. 차영상  $f(x, y)$ 에 대한 수평 가산투영  $h(y)$ 와 수직 가산투영  $v(x)$ 는 식 (6)과 같다.

$$\begin{aligned} h(y) &= \int f(x, y) dx \\ v(x) &= \int f(x, y) dy \end{aligned} \quad (6)$$

연속하는 두 프레임의 차영상에 대한 가산투영이 일정한 임계값을 초과하면, 이동물체가 검출된 것으로 하며 임계값을 초과하는 영역에 대하여 윤곽선을 검출하여 이동물체에 대한 초기모델로 하였다.

수직, 수평방향으로의 임계값을 각각  $TH_x, TH_y$ 로 표시할 때, 식 (7)과 같이 수직, 수평 두 방향에서 가산투영 평균값의  $\alpha$ 배에 해당하는 값을 임계값으로 정하였다.

$$\begin{aligned} TH_x &= \left[ \frac{1}{n} \sum_{y=1}^n h(y) \right] \times \alpha \\ TH_y &= \left[ \frac{1}{m} \sum_{x=1}^m v(x) \right] \times \alpha \end{aligned} \quad (7)$$

여기서  $h(y), v(x)$ 는 이전 프레임 영상  $I_{t-1}$ 과 현재 프레임 영상  $I_t$ 의 차영상  $f_t(x, y)$ 에 대한 수평 가산투영과 수직 가산투영을 나타낸 것이다. 또한  $n, m$ 은 영상의 수평, 수직 화소수로서 본 연구에서는  $n=176, m=144$ 이며,  $\alpha$ 는 실험에 의한 비례상수로서 2로 하였다.

### 2. 이동물체의 위치 탐색

초기 모델이 구성된 후, 알고리즘은 추적 단계로 진입하며, 그 다음 프레임들에서 이동물체의 위치를 탐색하기 위하여 매 프레임에 대하여 윤곽선을 검출하여 이동물체의 위치를 탐색한다. 이동물체의 이전 프레임에서의 모델의 위치를 중심으로 하여 2D-Logarithmic 탐색을 하며, 그 평가함수는 모델과 정합위치에서의 현재 영상의 하우스돌프 거리로 한다.

본 연구에서는 이동물체 위치 탐색 시간을 최소로

하기 위해 각 영상에 대하여 거리맵(Distance Map)을 구하였다. 거리 맵은 윤곽선 영상의 매개 점들에서 그와 가장 가까이 있는 윤곽선상의 점까지의 거리를 나타내는 2차원 배열이다. 그림 3은 영상에서 두 점 사이의 거리를 나타낸 것이고, 그림 4는 실험 영상의 한 프레임에서 일부 영역에 대한 거리 맵을 예로 보여준 것이다. 그림 4에서 회색으로 표시된 부분은 윤곽선을 표시한 것이며, 1, 2, 3 ...은 대응하는 위치에서 그 점과 가장 가까운 윤곽선과의 거리를 표시한다.

3	3	3	3	3	3	3
3	2	2	2	2	2	3
3	2	1	1	1	2	3
3	2	1	0	1	2	3
3	2	1	1	1	2	3
3	2	2	2	2	2	3
3	3	3	3	3	3	3

그림 3. 윤곽선영상에서 두 점사이의 거리

Fig. 3. The distance of two point in edge image.

5	4	3	2	1	0	1	2	3	4	5	6	6	6	6
4	4	3	2	1	0	1	2	3	4	5	5	5	5	5
4	3	3	2	1	0	1	2	3	4	4	4	4	4	4
4	3	2	2	1	0	1	2	3	3	3	3	3	3	3
4	3	2	1	1	0	1	2	3	2	2	2	2	2	2
4	3	2	1	0	1	1	2	3	2	1	1	1	1	1
3	3	2	1	0	1	2	2	2	1	1	0	0	0	0
3	2	2	1	0	1	2	2	2	1	0	1	1	1	1
3	2	1	1	0	1	2	2	1	1	0	1	2	2	2
2	2	1	0	1	1	2	2	1	0	1	1	2	2	2
2	1	1	0	1	2	2	1	1	0	1	2	2	1	1
2	1	0	1	1	2	2	1	0	1	1	2	1	1	0
2	1	0	1	2	2	2	1	0	1	2	2	1	0	1
2	1	0	1	2	3	2	1	0	1	2	2	1	0	1

그림 4. 실험영상에 대한 거리 맵의 한 예

Fig. 4. An example of distance map for experimental image.

2D-Logarithmic 탐색의 각 정합 위치에서 하우스돌프 거리의 구체적인 값을 계산하지 않고, 계산 과정에서 하우스돌프 거리를 기존 최소인 값과 비교하여 최소값을 초과하면 계산을 더 이상 진행하지 않고, 그 다음 정합위치로 넘어가기에 처리시간을 줄일 수 있다. 중복되는 각 정합위치에 대하여 한번만 값을 계산함

로써 계산시간을 더욱 줄일 수 있다. 그림 5는 한 프레임에서 이동물체 위치탐색 알고리즘의 흐름도를 나타낸 것이다.

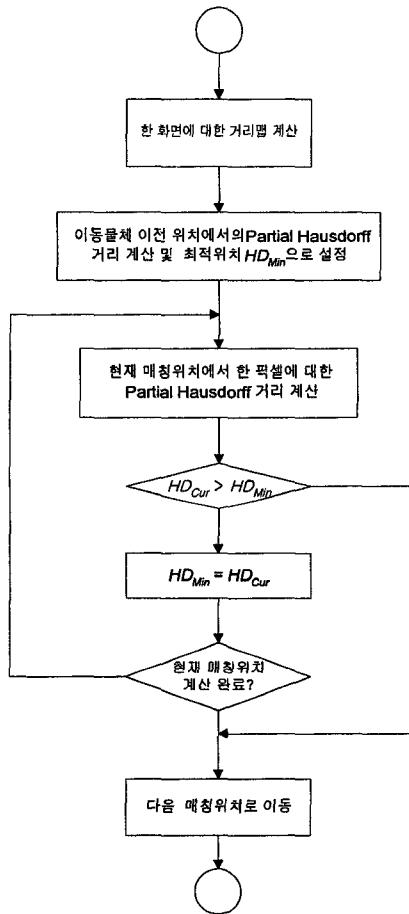


그림 5. 이동물체 위치탐색 알고리즘의 흐름도  
Fig. 5. Flowchart for allocating moving object.

3. 모델의 갱신

이동물체는 추적과정에서 3차원 이동이나 회전에 의해 2차원 형태가 변하게된다. 이런 2차원 형태의 변화에 적응해야만 강인한 추적이 가능하게 된다. 따라서 본 논문에서는 추적대상의 형태변화에 적응하기 위하여 매 프레임마다 이동물체의 위치를 탐색한 후 모델을 갱신하여 그 다음 프레임에 사용하였다. 새로운 모델은 추적 마스크 내에서 식(8)을 만족하는 점들을 가지고 구성하였다.

$$M_t = \left\{ q \in I_t \mid \min_{p \in M_t} \|g(p) - q\| \leq \delta \right\} \quad (8)$$

식 (8)에서  $I_t$ 는 현재 프레임의 윤곽선 영상이고,  $g(p)$ 는 현재 프레임에서 탐색한 이동물체의 추적 마스크 내 점들의 집합이며,  $q$ 는  $I_t$ 의 점이다. 본 연구에서는  $\delta = 2$ 로 하였을 때, 가장 최적의 결과를 보여주었다.

IV. 실험 결과 및 고찰

제안한 알고리즘의 타당성을 검증하기 위하여 자연 환경에서 비디오 카메라로 취득한 영상을 Pentium III 450 컴퓨터에서 DT3155 이미지 보드로 취득한 후, 단일 고속 이동물체 추적을 위한 모의실험을 수행하였다. 실험영상은 도로에서 평균시속 50Km로 주행하는 자동차와 오토바이를 추적대상으로 하였으며, 취득 영상의 해상도는 176×144 픽셀 크기의 256 그레이 레벨을 갖는 영상이다.

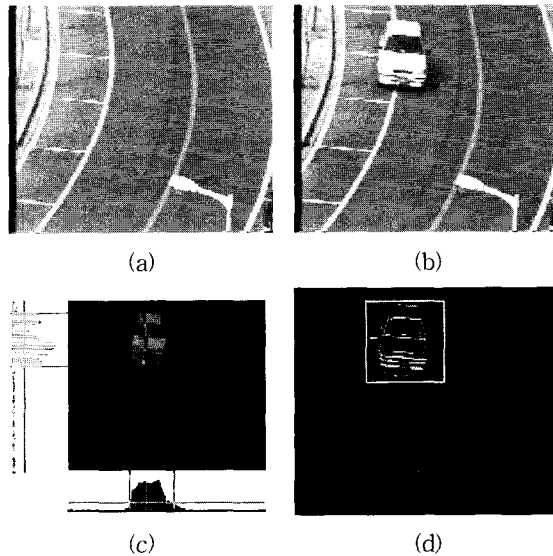


그림 6. (a) 배경영상 (b) 이동물체가 나타난 영상  
(c) 차영상에 대한 가산투영  
(d) 검출된 이동물체 영역

Fig. 6. (a) Background image  
(b) The image moving object appeared  
(c) Integral projection of difference image  
(d) The area of detected moving object

그림 6(a)는 이동물체가 나타나기 전의 배경영상을 보여주고, 그림 6(b)는 이동물체가 화면에 완전히 진입하였을 때의 영상을 보여준다. 그림 6(c)는 (a), (b) 두

영상의 차영상에 대한 가산투영을 보여주고, 그림 6(d)는 가산투영에 임계값을 적용하여 산출된 이동물체 위치를 보여준다.

가산투영에 대한 임계값은  $\alpha=2$ 로 하였을 때 자동차 영상에 대해서는 각각  $TH_x=987$ ,  $TH_y=1021$ 이었고, 오토바이 영상에 대해서는 각각  $TH_x=588$ ,  $TH_y=634$ 이었다.

표 2. 자동차 영상에 대한 위치 비교  
Table 2. Coordinate comparison for vehicle image.

Frame No.	추적 결과		실제 위치		절대 차	
	$\hat{X}$	$\hat{Y}$	$X$	$Y$	$X_e$	$Y_e$
35	37	12	37	13	0	1
45	65	40	65	40	0	0
55	84	41	83	41	1	0
65	94	36	94	37	0	1
75	74	40	75	38	1	2
85	70	38	72	38	2	0
95	82	43	82	43	0	0
105	44	46	44	47	0	1
115	77	28	78	28	1	0
125	76	21	75	20	1	1
135	92	36	92	37	0	1
145	85	31	85	29	0	2

표 3. 오토바이 영상에 대한 위치 비교  
Table 3. Coordinate comparison for motorcycle image.

Frame No.	추적 결과		실제 위치		절대 차	
	$\hat{X}$	$\hat{Y}$	$X$	$Y$	$X_e$	$Y_e$
35	30	78	32	77	2	1
45	85	75	85	75	0	0
55	83	74	83	74	0	0
65	90	72	90	70	0	2
75	91	82	89	82	2	0
85	78	69	78	69	0	0
95	77	70	76	72	1	2
105	88	65	88	65	0	0
115	92	71	91	70	1	1
125	91	84	94	81	3	3
135	92	75	92	77	0	2
145	87	88	87	88	0	0

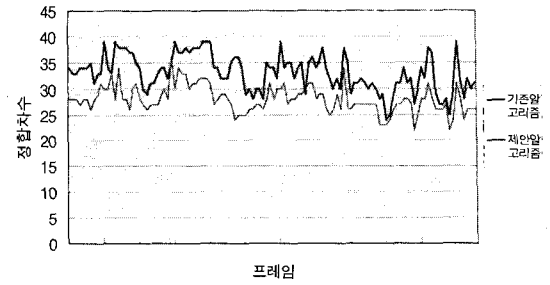


그림 7. 자동차 영상에 대한 정합 회수  
Fig. 7. matching times for vehicle image.

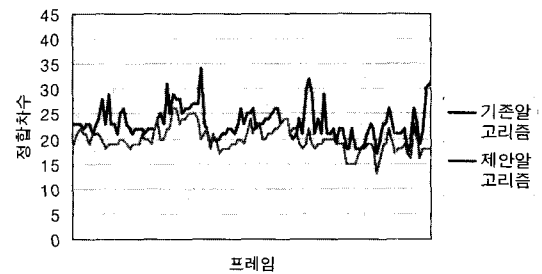


그림 8. 오토바이 영상에 대한 정합 회수  
Fig. 8. matching times for motorcycle image.

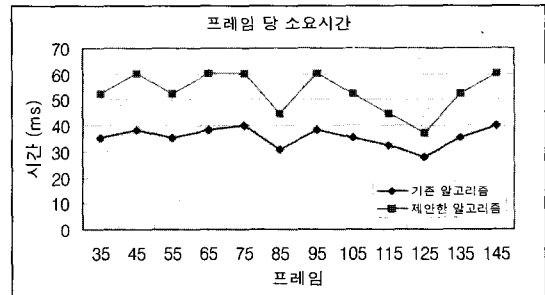
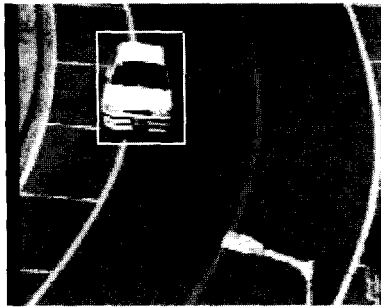


그림 9. 프레임 당 처리시간의 비교  
Fig. 9. comparison of processing time per frame.

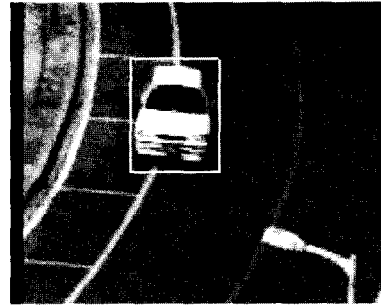
표 2와 표 3은 자동차 영상과 오토바이 영상의 일부 프레임에 대한 추적결과 좌표값  $\hat{X}$ ,  $\hat{Y}$ 와 실제 좌표값  $X$ ,  $Y$  및 그 오차  $X_e$ ,  $Y_e$ 를 보여준다. 추적 결과의 좌표와 실제좌표와의 오차  $e$ 는 두 좌표값 사이의 기하학적 거리로 표시할 수 있으며 그 수식은 식 (9)과 같다.

$$e = \sqrt{(\hat{X} - X)^2 + (\hat{Y} - Y)^2} = \sqrt{X_e^2 + Y_e^2} \quad (9)$$

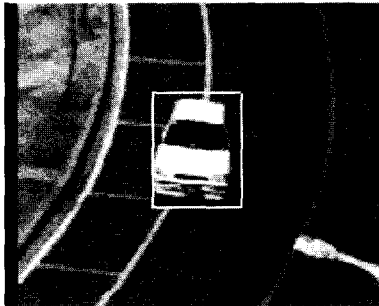
평균오차는 각 프레임에서 오차의 평균으로서 식 (10)을 이용하여 계산할 수 있다.



(a) Frame No. 35



(b) Frame No. 40



(c) Frame No. 45



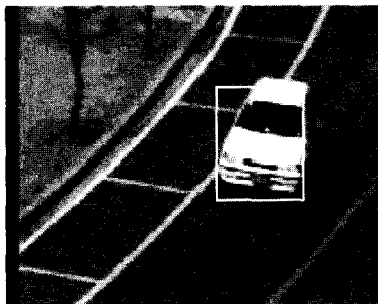
(d) Frame No. 50



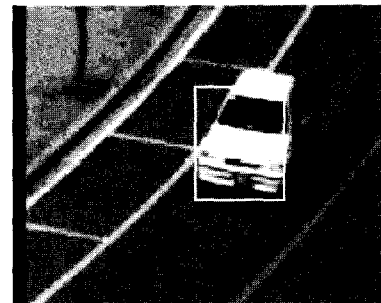
(e) Frame No. 55



(f) Frame No. 60



(g) Frame No. 65



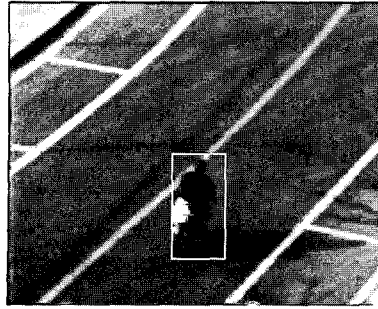
(h) Frame No. 70

그림 10. 자동차 영상에 대한 추적결과

Fig. 10. Tracking results for vehicle image.

$$E = \frac{1}{n} \sum_{i=0}^{n-1} e_i$$

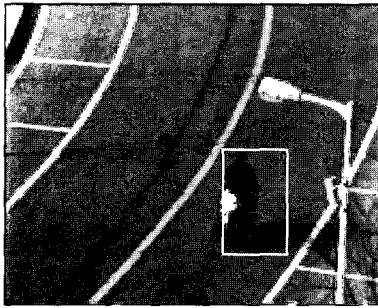
여기서  $n$ 은 전체 프레임 수를 의미하며,  $e_i$ 는  $i$ 프  
(10) 레임에서 실험에서 얻은 값과 실제 값과의 오차를 의



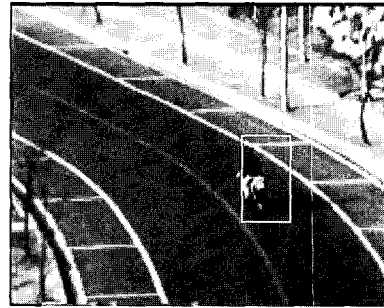
(a) Frame No. 40



(b) Frame No. 50



(c) Frame No. 60



(d) Frame No. 70

그림 11. 오토바이 영상에 대한 추적결과

Fig. 11. Tracking results for motorcycle image.

미한다. 이와 같은 방법으로 전체 150프레임의 영상에서 추적대상의 좌표를 실제값과 비교하여 그 평균자승오차를 계산한 결과 자동차 영상에서는  $E=1.15$ , 오토바이 영상에서는  $E=1.845$ 임을 알 수 있었다. 제안한 알고리즘의 이동물체 추적시간을 평가하기 위하여 매 프레임에서 정합의 시도회수와 처리시간을 기존 알고리즘<sup>[10]</sup>과 비교 평가하였다. 그림 7과 그림 8은 자동차 영상과 오토바이 영상에 대한 제안한 알고리즘과 기존 알고리즘의 정합회수를 보여준다. 또한 두 영상에 대한 기존 알고리즘과 제안한 알고리즘의 정합차수 평균을 구한 결과 자동차 영상에서 각각 34.41, 23.16으로 제안한 알고리즘이 평균 11.25회 적음을 알 수 있었고, 오토바이 영상에서는 각각 25.19, 19.96으로 제안한 알고리즘이 평균 5.23회 적음을 알 수 있었다.

그리고 그림 9는 매 프레임에서 이동물체의 탐색에 소요되는 시간을 그래프로 보여준 것이다. Pentium III 450 컴퓨터에서 실험영상에 대하여 처리속도가 프레임당 평균 35.7ms 즉, 28frame/sec로 실시간 처리가 가능함을 보였다.

그림 10과 그림 11은 제안한 방법으로 자동차 영상과 오토바이 영상을 추적하였을 때의 결과 영상을 보

여준 것이다.

## V. 결 론

본 논문에서는 배경이 움직이는 자연환경에서 모델 기반 이동물체 추적 알고리즘을 제안하였다. 제안한 알고리즘은 인터프레임간의 차영상에 대한 가산투영을 가지고 이동물체의 영역을 검출하여, 이 영역에 대한 윤곽선 영상을 이동물체의 모델로 하였다. 제안한 알고리즘은 이동물체의 형태변화에 적응하기 위하여 Hausdorff 거리를 모델과 영상사이의 유사도로 사용하였으며, 이동물체의 위치탐색시간을 줄이기 위하여 2D-Logarithmic 탐색기법을 사용하였다.

실험은 실제 도로에서 주행하는 자동차와 오토바이를 대상으로 모의 실험을 수행하였으며 추적 결과 제안한 알고리즘은 이동물체의 실제위치와 추적결과와의 평균자승오차가 자동차 영상에서는 1.15, 오토바이 영상에서는 1.845로 이동물체의 정확한 추적이 가능함을 알 수 있었으며, 추적 정합 시도 회수는 자동차 영상에서는 평균 11.25회, 오토바이 영상에서는 평균 5.23회 적게 시도하여 추적시간을 단축할 수 있었음을 알 수 있었다.



향후 연구 과제로는 특정 이동물체에 대한 추적 알고리즘과 형태의 변화가 심하거나 화면에 여러 개의 형태가 비슷한 이동물체가 존재할 때 특정 물체에 대한 효율적인 추적 알고리즘에 대한 연구가 계속되어야 한다고 사료된다.

참 고 문 헌

[1] D. P. Huttenlocher, J. J. Noh, W. J. Rucklidge, "Tracking Non-Rigid Objects in Complex Scenes," Proceedings of 4th ICCV, pp. 93-101, May 1993.

[2] R. C. Jane, "Segmentation of Frame Sequences Obtained by A Moving Observer," IEEE Trans. PAMI, vol. 6, no.5. pp. 624-629, 1984.

[3] M. K. Leung, "Human Body Motion Segmentation in A Complex Scene," Pattern Recognition, vol. 20. no 1, pp. 55-64, 1987.

[4] Y. Mae, S. Yamamoto, Y. Shirai, and J. Miura, "Optical Flow Based Realtime Object Tracking by Active Vision System," Proc. 2nd Japan-France Congress on Mechatronics, vol. 2, pp. 545-548, 1994.

[5] J. C. Clarke, A. Zisserman, "Detection and Tracking of Independent Motion," Image and Vision Computing, pp. 565-572, 1996.

[6] G. L. Foresti, "A Real-Time System for Video Surveillance of Unattended Outdoor Environments," IEEE Trans. on Circuit and

Systems for Video Tech., vol. 8 no. 6, pp. 142-145, Oct. 1998.

[7] F. J. Hampson, RE. Franich, J. C. Pesquet, J. Biemond, "Pel-recursive Motion Estimation in the Presence of Illumination Variations," ICIP 96, vol. 1, pp. 101-104, 1996.

[8] 이주신, 김윤호, "퍼지 추론에 의한 2-D이동물체의 동일성 판별," 한국통신학회 논문지, vol. 17, no. 4, pp. 346-360, 1992

[9] 이주신, 이명길, "DCT와 신경 회로망을 이용한 패턴인식에 관한 연구," 한국통신학회 논문지, vol. 22, no. 3, pp. 481-492. 1997

[10] 전춘, 김태식, 이명길, 이주신, "동적 배경에서 Hausdorff 거리를 이용한 이동물체의 추적," 한국통신학회 하계종합학술대회 논문집, vol. 19, no. 1, pp. 537-540, 1999

[11] 김준식, 박래홍, 이병욱, "가산투영을 이용한 2단계 고속 블록 정합 알고리즘," 전자공학회 논문지, 제30권 B편 제1호, pp. 45-54, 1993

[12] 이규원, 김영호, 이재규, 박규태, "무인감시장치 구현을 위한 단일 이동물체 추적 알고리즘," 전자공학회 논문지, 제31권, B편, 제11호, pp. 11-52, 1995

[13] 장석환, 김인행 등, "실시간 다중 물체 추적 시스템," 신호처리합동학술대회 논문집, 제11권, 1호, pp. 499-502, 1998

[14] 강재열, 김성대, "가산투영을 이용한 시공간 경사 추적기법," 신호처리합동학술대회 논문집, 제7권, 1호, pp. 545-548, 1994

저 자 소 개



金太植(正會員)  
1956년 11월 13일 생. 1883년 2월 명지대학교 전자공학과 졸업(공학사). 1989년 2월 명지대학교 대학원 전자공학과 졸업(공학석사). 1992년 9월~현재 청주대학교 대학원 전자공학과 박사과정. 1992년 3월~현재 진주전문대학 컴퓨터전자계열 부교수. 주관심분야는 영상처리, 신경 회로망, 컴퓨터 그래픽 등.

李周信(正會員)  
1975년 2월 명지대학교 전자공학과 졸업(공학사). 1977년 2월 홍익대학교 대학원 전자공학과 졸업(공학석사). 1986년 2월 명지대학교 대학원 전자공학과 졸업(공학박사). 1979년 2월~현재 청주대학교 전자공학과 교수. 주관심분야는 영상처리, 영상통신, 컴퓨터 비전 등