

# 마이크로 컴퓨터를 이용한 2 차원 이동물체의 이동거리와 속도측정에 관한 연구

## (A Study on the Moving Distance and Velocity Measurement of 2-D Moving Object Using a Microcomputer)

李 周 信\*, 崔 甲 石\*\*

(Joo Shin Lee and Kap Seok Choi)

### 要 約

본 논문에서는 어떤 특정 배경내의 2 차원 연속화상으로 부터 얻은 휘도정보를 처리하여 단일 이동물체의 이동거리 및 이동속도를 측정하였다. 이동물체의 검출은 배경으로부터 얻은 기준휘도정보와 동일배경내에서 이동물체를 포함한 휘도 정보사이 및 동일물체가 이동한 후의 휘도정보 사이의 차화상 추출에 의하였으며, 두 차화상의 인식 파라미터의 비교에 의해서 동일물체 여부를 판별하였다.

차화상 추출 과정에서는 휘도변화로 인한 오차를 감소시키기 위하여 화소의 비트 데이터 재배열이라는 새로운 방법을 제안하여 사용하였고, 동일물체 인식 파라미터로서는 휘도치, 버텍스(Vertex) 수, 버텍스 간의 길이를 사용하였다. 이동거리를 구하기 위해서 컴퓨터내에 좌표계를 설정하여 이동거리를 구했으며, 이동속도는 샘플링 프레임과 프레임 사이의 시간 간격과 이동거리를 가지고 구하였다.

### Abstract

In this paper, the moving distance and velocity of a single moving object are measured by sampling three frames in a two-dimensional line sequence image. The brightness of each frame is analyzed, and the bit data of their pixel are rearranged so that the difference image may be extracted. The parameters for recognition of the object are the gray level of the object, the number of vertex points and the distance between the vertex points.

The moving distance obtained from the coordinate which is constructed by the bit processing of the data in the memory map of a microcomputer, and the moving velocity is obtained from the moving distance and the time interval between the first and second sampled frames.

### I. 序 論

동화상 처리에 의하여 이동물체의 이동속도 및 이동거리를 간단하게 측정할 수 있다면 차량감시나 운동선수의 인체동작, 작업자의 작업동작 분석등, 모든 동물의 동작상태를 감시, 관찰하는데 매우 유용하리라고 기대된다.

최근 2 차원 연속화상 처리에 의하여 특정지역에서

\*正會員, 淸洲大學校 電子工學科  
(Dept. of Elec. Eng., Choungju Univ.)

\*\*正會員, 明知大學校 電子工學科  
(Dept. of Elec. Eng., Myong Ji Univ.)

接受日字: 1985年 9月 2日

보행자들에 대한 이동상태 파악 및 차량의 이동상태 관측을 목적으로 한 연구가 활발히 행해지고 있으며<sup>11)</sup> 이동물체의 이동거리 및 이동속도 측정에 대해서는 Fennema와 Thompson<sup>9)</sup>등에 의한 많은 연구보고가 있다. 이와같은 방법들은 많은 인터프레임(inter-frame)을 처리하지 않으면 안되기 때문에 처리과정이 복잡하고 처리시간이 많이 걸린다.

본 연구에서는 고정된 배경내에서 단일 직선 이동물체의 이동거리 및 이동속도에 대한 측정을 시도하기 위해 2차원 연속화상을  $t_1$ 인 시간에 샘플링된 프레임 을 가지고 휘도를 분석하고, 분석된 각 프레임의 휘도치를 배경의 휘도치와 비교하여 차화상을 추출하고 차화상 추출시 배경의 휘도 변화에서 기인하는 차화상 오차를 줄이기 위해 각 화소의 비트데이터를 재배열하는 방법을 제안하고 이 방법을 이용하여 차화상을 추출한다.

이동물체의 판별은 다음과 같은 3가지 인식 파라미터를 가지고 시행한다.

- ① 물체의 버텍스 수
- ② 물체의 휘도치
- ③ 버텍스와 버텍스 사이의 길이

물체의 이동거리는 첫번째 차화상에서 결정된 초기 버텍스와 두번째 차화상에서 결정된 초기 버텍스 사이를 이동 거리로 놓고 속도는  $t_1$ 과  $t_2$ 사이의 샘플링 간격을 속도정보로 결정하고자 한다. 또한 컴퓨터 메모리에 좌표계를 설정하여 실제거리와 대응시킬 수 있기 때문에 이동물체의 이동거리와 속도측정을 시도하였다.

II. 차화상 추출

컴퓨터 메모리를  $M \times N$ 의 배열로 구성하여  $280 \times 192$ 의 비트를 그림과 같이  $3 \times 3$ 비트로만 단위 소영역으로 분할하였으며 분할된 이 소영역을 한 화소로 결정하였다. 한 화소의 휘도치는 9개의 비트가 취할 수 있는 값 0~9레벨이 된다.

연속적인 화상의 해석은 연속적인 각 프레임들을 정지화상 프레임의 연속으로 처리하는 것을 기초로하고 있다<sup>10, 11)</sup> 따라서 배경내에 어떤 물체가 이동하고 있다면 그 배경의 프레임을 샘플링하면 샘플링 된 프레임은 정지화상이 되므로 그 휘도치는 시간적 변화에 관계없이 일정하다.<sup>5)</sup> 그러므로 2차원 연속화상의 프레임을 등간격으로 그림 2와 같이 샘플링하여 배경화상을 기준화상  $I_r$ 로 설정해 놓고, 동일배경내에서 이동하고 있는 물체를  $t_1$ 인 시간에 샘플링된 화상을  $I_r$

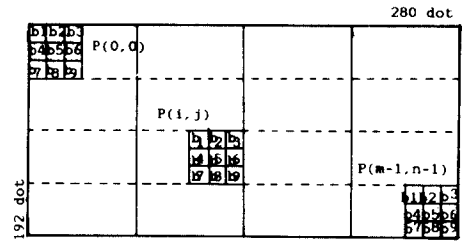


그림 1. 3 \* 3 영역분할  
Fig. 1. 3 \* 3 Domain .

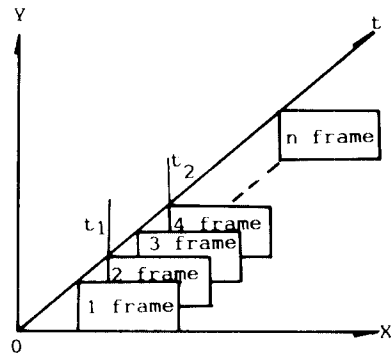


그림 2. 프레임 샘플링  
Fig. 2. Frame sampling .

라하고  $t_2$ 인 시간에 샘플링된 화상  $I_s$ 을 가지고 그림 3과 같은 방법으로 차화상  $I_b$ 을 얻는다.  $I_r$ 과  $I_s$ 의 차화상을  $I_{b1}$ 이라하고,  $I_r$ 과  $I_s$ 의 차화상을  $I_{b2}$ 라 한다.  $I_r$ 과  $I_s$ 의 화상이 각기 내포하고 있는 휘도분포를 행렬로 나타내면 다음과 같다.

$$\begin{aligned}
 I_r &= \begin{bmatrix} b_{11}, & b_{12}, & \dots, & b_{1n} \\ b_{21}, & b_{22}, & \dots, & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{m1}, & b_{m2}, & \dots, & b_{mn} \end{bmatrix}, \\
 I_s &= \begin{bmatrix} b_{11}, & b_{12}, & \dots, & b_{1n} \\ b_{21}, & b_{22}, & \dots, & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{m1}, & b_{m2}, & \dots, & b_{mn} \end{bmatrix}, \\
 I_b &= \begin{bmatrix} b_{11}, & b_{12}, & \dots, & b_{1n} \\ b_{21}, & b_{22}, & \dots, & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{m1}, & b_{m2}, & \dots, & b_{mn} \end{bmatrix} \quad (1)
 \end{aligned}$$

여기서 행렬 요소들은 각 화상의 비트데이터(1또는 0)이다. 따라서  $I_r$ 과  $I_s$ 의 차화상을  $I_{b1}$ 이라 한다면

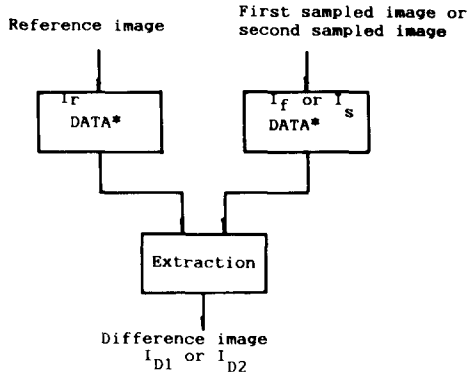


그림 3. 차화상 추출  
Fig. 3. Extraction of difference image.

$I_{D1}$ 은 다음에 의해 얻어진다.

$$I_{D1} = I_r \oplus I_f \quad (2)$$

또한  $I_r$ 과  $I_s$ 의 차화상  $I_{D2}$ 라 한다면

$$I_{D2} = I_r \oplus I_s \quad (3)$$

이 된다. 따라서  $I_{D1}$ 과  $I_{D2}$ 는 물체만 검출된 화상이 된다.

$I_{D1}$ 과  $I_{D2}$ 의 휘도분포가 그림 4와 같고  $I_{D1}$ 의 전체 휘도치를  $I_{DB1}$ 이라고 하면 다음과 같이 쓸 수 있다.

$$I_{DB1} = P_{b1} + P_{b2} + \dots + P_{bn} \\ = \sum_{i=1}^n P_{bi} \quad (4)$$

그림 4 (b)의 전체휘도치를  $I_{DB2}$ 이라하면 다음 식과 같다.

$$I_{DB2} = P'_{b1} + P'_{b2} + \dots + P'_{bn} \\ = \sum_{i=1}^n P'_{bi} \quad (5)$$

즉  $I_{DB1}$ 과  $I_{DB2}$ 는  $I_{D1}$  및  $I_{D2}$ 의 휘도치이다. 식 2와 식 3에서 차화상을 추출하면  $I_r$ 의 배경에 해당되는 각 행열 요소들이  $I_r$ 의 행열 요소와 같으므로 서로 상쇄된다. 그러나 실제  $I_r$ 과  $I_f$ 의 시간차에 의해서 휘도차가 존재하므로 차화상 추출시 그림 5와 같은 차화상 오차가 생기게 된다. 이와같이 차화상을 추출하는 과정에서 생기는 오차를 줄이기 위해서 화상의 각 화소

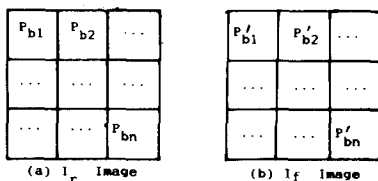


그림 4. 화상휘도치  
Fig. 4. Image brightness.

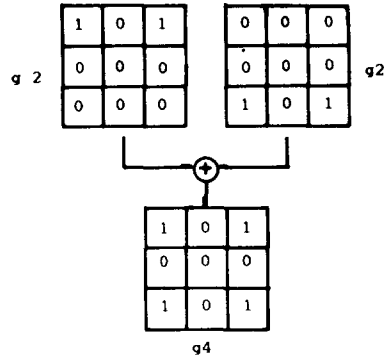


그림 5. 화상오차  
Fig. 5. Error image.

내의 비트데이터를 그림 6과 같이 화소의 휘도치에 따라 비트데이터를 재배열시켰다. 불규칙한 화소의 비트데이터를 규칙적으로 재배열 시키므로써 양질의 차화상을 얻을 수 있었다. 그림 7에서 (a)는 화상데이터 재배열을 하지 않고 처리된 차화상이며 (b)는 재배열 처리를 한 후 추출된 차화상이다.

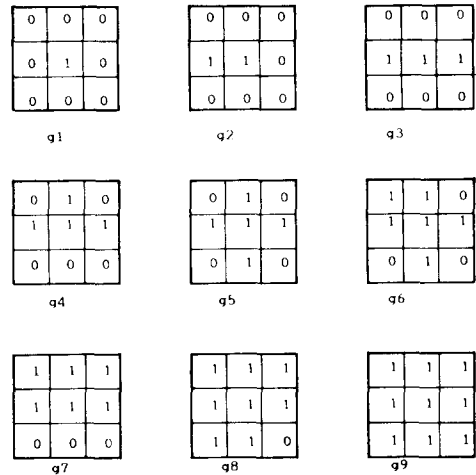


그림 6. 재배열 처리 과정  
Fig. 6. Processing of rearrangement.

### III. 이동물체 인식

#### 1) 윤곽선 검출

차화상의 화소배열  $i \times j$ 를 행열로 표현한다면 다음과 같다.

$$I_b = \{P(i, j)\} \quad (6)$$



(a) Difference image before rearrangement



(b) Difference image after rearrangement

그림 7. (a)와 (b) 비교

Fig. 7. Comparison of (a) with (b).

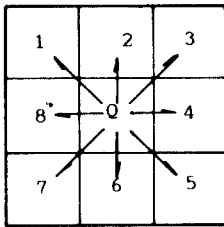


그림 8. 8 방향  
Fig. 8. 8 Curves.

여기서  $P(i, j)$ 는 화소이며  $i=0, 1, 2, \dots, 84, j=0, 1, 2, \dots, 63$ 이다. 윤곽화소  $P(i, j)=g$ 라 하면  $g$ 는  $I_{DB}$ 의 윤곽화소에 대한 임계치이다.

1. 윤곽화소 탐색

탐색할 방향벡터를 그림 8 과 같이 8 개로 지정하고, 탐색은 최상단 좌에서 우로 탐색해간다.

알고리즘

STEP 1.  $P(i, j)$ 의 화소가  $g=T$ 이면  $Q_1(i, j)$ 로 놓는다.

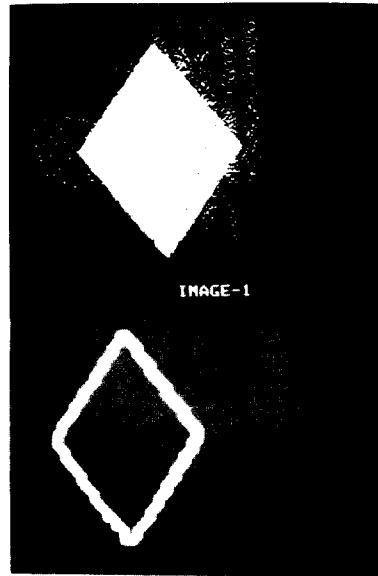
STEP 2.  $Q_1(i, j)$ 을 중심으로 8 방향 번호순으로  $g=T$ 인 화소를 탐색.

STEP 3.  $g=T$ 가 존재하면  $Q_2(i, j)$ 로 한다.

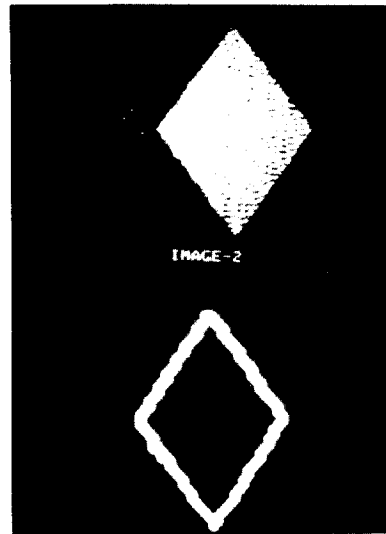
STEP 4.  $Q_2(i, j)$ 의 유입방향에 따라 지정된 방향벡터에 의해 다음 화소탐색(표 1)

STEP 5. 현재의 탐색화소가  $Q_m(i, j)=Q_1(i, j)$ 이면 탐색을 마친다.

그림 9은 윤곽화소의 임계치를  $g=3$  이상으로 결정하고,  $g=3$  이상인 경우에 윤곽을 검출한 것이다.



(a)  $I_f$  Edge Detection



(b)  $I_g$  Edge Detection

그림 9. 윤곽화소검출

Fig. 9. Detection of edge pixel.

2) 버텍스 결정

윤곽선에서 버텍스만을 결정하기 위해서 그림 7 (b)에서  $x$  및  $y$ 방향의 증분 파라미터를 표 2와 같이 정하고, 파라미터들에 의해서 버텍스를 결정한다.

1. 버텍스 검출 알고리즘

STEP 1.  $x, y$ 의 증분을  $+, 0, -$ 로 한다.

표 1. 8 방향 파라미터  
Table 1. 8 Curves parameter.

유입해 들어오는 방향벡터	유입해 나가는 방향벡터
1	6
2	7
3	8
4	1
5	2
6	3
7	4
8	5

- STEP 2. 증분변수는 X, Y로 한다.
  - STEP 3. 진행상태 변수는 OX, OY이며 초기값은 OX="+", OY="-"로 한다.
  - STEP 4. 탐색시작은 최상단 좌측에서 우측으로 탐색해 간다.
  - STEP 5. 윤곽화소  $Q_1(x, y)$ 에서  $Q_2(x, y)$ 의 순으로 탐색.
  - STEP 6. OX와X, OY와Y의 증분파라미터 값들의 동일여부 판단(단 "0"는 제외)
  - STEP 7. OX=X, OY=Y이면 탐색을 지속한다.
  - STEP 8. (OX=X, OY=Y), (OX≠X, OY=Y)인 화소의 바로 전 화소를 버텍스점으로 놓고 다음 STEP를 진행한다.
  - STEP 9. 버텍스가 결정되면 OX, OY에는 각각 X, Y의 내용을 기록한다.
  - STEP10. 다음 화소 탐색-GOTO STEP 6.
  - STEP11.  $Q_w(i, j) = Q_1(i, j)$ 이면 끝마친다.
- 그림 10은 이같은 방법으로 검출된 버텍스를 나타낸 것이다.

표 2. 버텍스 파라미터  
Table 2. Vertex parameter.

	증 가	감 소	변화없음
X	+	-	0
Y	+	-	0

2. 버텍스 선정.

① 초기 버텍스 선택은 좌에서 우로 위에서 아래로 탐색해가다가 X, Y 파라미터를 진행상태 초기 파라미터 (OX=+, OY=-)와 비교하여 변화가 생기면 초기 점으로 결정한다.

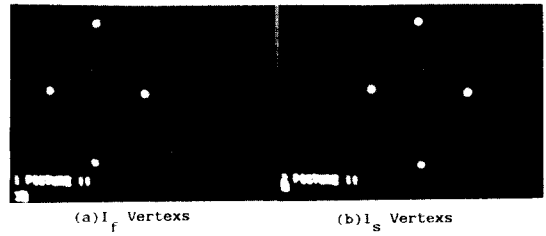


그림 10. 버텍스 검출  
Fig. 10. Detection of vertices.

② 초기 버텍스 외의 3개의 버텍스를 선정하는 과정은 검출된 버텍스들 중에서 X좌표값이 제일큰 버텍스와 X좌표값이 제일작은 버텍스, 그리고, Y값이 제일 큰 버텍스 3개를 선택한다.

③ X좌표값이 큰 버텍스 갯수가 복수면 그 중의 Y좌표값이 제일작은 점을 버텍스로 선정한다. X좌표값이 작은 버텍스 갯수가 복수이면 그 중의 Y좌표값이 제일 큰 버텍스를 택한다. Y좌표값이 큰 버텍스가 복수 개이면 그중 X좌표값이 큰 버텍스를 택한다.

이와같은 방법에 의해서 물체의 버텍스들이 각각 구해지면 그림 11같이 각 버텍스들을  $V_{p1}, V_{p2}, V_{p3}, V_{p4}$ 로 정하고,  $V_{p1}$ 를 기준버텍스로 놓고, 기준버텍스와 각 버텍스 사이의 길이들을  $a_1, a_2, a_3$ 로 놓고  $a_1, a_2, a_3$ 의 길이를 인식 파라미터로 적용했다.

3) 동일물체 인식 파라미터의 오차 설정.

측정하고자 하는 물체가 항상 임의의 물체가 되므로  $I_{D1}$ 의 물체와  $I_{D2}$ 의 물체를 서로 비교하여 동일물체를 판별하여 동일물체로 판정되면 이동거리와 속도를 측정한다. 따라서 동일물체 판별을 하기 위해서  $I_{D1}$ 과  $I_{D2}$ 의 파라미터 값들이 어떤 허용오차 범위내에 들면 동일물체로 판단하였다. 파라미터 설정은  $I_{D1}$ 과

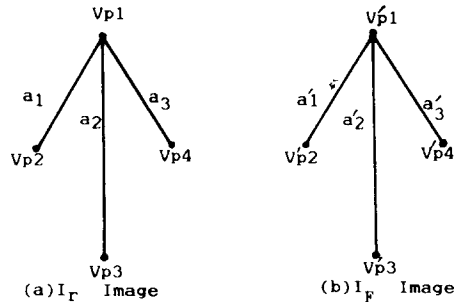


그림 11. 굴절점  
Fig. 11. Vertex point.

$I_{D2}$ 의 휘도치,  $I_{D1}$ 과  $I_{D2}$ 의 벡터스 수,  $I_{D1}$ 의 초기 벡터스와 벡터스 사이의 선분길이,  $I_{D2}$ 의 초기 벡터스와 벡터스 사이의 선분길이 들로 결정하였다.

1.  $I_{D1}$ 과  $I_{D2}$ 의 물체의 휘도치.

$I_{D1}$ 과  $I_{D2}$ 의 물체에 대한 휘도치 오차를  $\gamma$ 라 한다면 다음식과 같다.

$$\gamma = \frac{I_{D1} \text{의 물체의 총 휘도치} - I_{D2} \text{의 물체의 총 휘도치}}{I_{D1} \text{의 물체의 총 휘도치}} \times 100\% \quad (7)$$

식(7)에서  $\gamma \leq 20\%$  이면 동일물체로 일단 간주하고 벡터스 수를 비교한다.

2.  $I_{D1}$ 과  $I_{D2}$ 의 벡터스 수 비교

$I_{D1}$ 의 벡터스 수를  $V_1$ 이라하면

$$V_1 = V_{P1} + V_{P2} + \dots + V_{Pn} = \sum_{i=1}^n V_{P1} \quad (8)$$

$I_{D2}$ 의 벡터스 수를  $V_2$ 이라하면

$$V_2 = V'_{P1} + V'_{P2} + \dots + V'_{Pn} = \sum_{i=1}^n V'_{P1} \quad (9)$$

$I_{D1}$ 과  $I_{D2}$ 가 동일물체라면 두 물체의 벡터스 수의 차  $V_n$ 은 다음과 같다.

$$V_n = V_1 - V_2 = 0 \quad (10)$$

$I_{D1}$ 과  $I_{D2}$ 가 동일물체라면 식(10)을 만족해야 하나  $I_{D1}$ 과  $I_{D2}$ 의 휘도차에서 오는 오차를 감안하기 위해서  $\pm 2$ 개(벡터스수)의 오차를 두었다. 따라서 벡터스 수를 가지고 동일물체 판별은 다음과 같은 식에 만족하도록 하였다.

$$\left. \begin{aligned} V_n &= 0 \\ V_n &= \pm 2 \end{aligned} \right\} \quad (11)$$

식(11)에 만족하면 벡터스 사이의 각 선분길이를 비교한다.

3.  $I_{D1}$ 과  $I_{D2}$ 의 벡터스 선분길이 비교

그림 11 (a)에서 벡터스 선분길이는 다음과 같다.

( $V_{P1}$ 을 기준)

$$a_1 = \sqrt{x^2 + y^2}, \quad a_2 = \sqrt{x^2 + y^2}, \quad a_3 = \sqrt{x^2 + y^2} \quad (12)$$

그림 11 (b)에서 벡터스 선분길이는 다음과 같다( $V'_{P1}$ 을 기준).

$$a'_1 = \sqrt{x^2 + y^2}, \quad a'_2 = \sqrt{x^2 + y^2}, \quad a'_3 = \sqrt{x^2 + y^2} \quad (13)$$

$I_{D1}$ 과  $I_{D2}$ 의 각 선분길이 오차는 다음식과 같다.

$$\left. \begin{aligned} k_1 &= \frac{a_1 - a'_1}{a_1} \times 100\% \\ k_2 &= \frac{a_2 - a'_2}{a_2} \times 100 \\ k_3 &= \frac{a_3 - a'_3}{a_3} \times 100 \end{aligned} \right\} \quad (14)$$

따라서 각 선분길이의 평균오차를 K이라하면

$$K = \frac{k_1 + k_2 + k_3}{3} \times 100 \quad (15)$$

동일물체 판별은 평균오차  $K \leq 10\%$  이면 동일물체로 판단하였다.

IV. 이동거리 및 속도

일정한 배경내에서 물체가 수평이동하고 있다고 가정한다면 그림 12에서  $t_1$ 인 시간에 샘플링된 화상  $I_r$ 과  $I_r$ 과의 차화상  $I_{D1}$ 의 초기 벡터스  $V_{P1}$ 의 화소가  $P(i,j)$  라면  $t_2$ 인 시간에 샘플링된  $I_s$ 와  $I_r$ 과의 차화상  $I_{D2}$ 의 초기 벡터스  $V'_{P1}$ 의 화소는  $P(i+m, j+n)$ 이다. 따라서  $I_{D1}$ 과  $I_{D2}$ 의 화소간격을 거리로 환산할 수 있다.

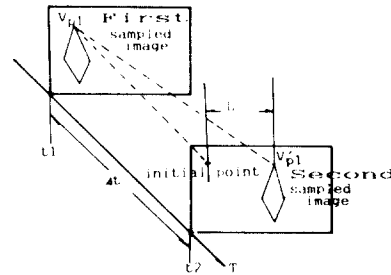


그림 12. 화상 샘플링  
Fig. 12. Image sampling.

1. 실제거리 환산

실제거리와 컴퓨터내에 메모리상의 좌표와의 관계는 카메라와 물체의 거리를 R로 고정하므로서 X축과 Y축의 거리는 고정된 상태가 된다. 따라서 컴퓨터 내부의 좌표계는  $85 \times 64$  ( $i=0, 1, 2 \dots 84, j=0, 1, 2 \dots 63$ )의 화소로 설정하였으므로 다음과 같은 식에 의해 산출할 수 있다. X축의 한 화소당 찾아하는 길이를  $\alpha$ 라 한다면

$$\alpha = \frac{X \text{축의 실제거리}}{X \text{축 1 행의 화소수}} \quad (16)$$

Y축의 한 화소당 찾아하는 길이를  $\beta$ 라 한다면

$$\beta = \frac{Y \text{축의 실제거리}}{Y \text{축 1 종의 화소수}} \quad (17)$$

이 된다. 따라서  $I_{D1}$ 의 초기 벡터스  $V_{P1}$ 과  $I_{D2}$ 의 초기 벡터스  $V'_{P1}$ 와의 실제이동거리는 다음과 같다.

$$L_x = X \text{축 이동화소수} \times \alpha \quad (18)$$

$$L_y = Y \text{축 이동화소수} \times \beta \quad (19)$$

이다. 그러므로 이동거리 L은 다음과 같다.

$$L = \sqrt{L_x^2 + L_y^2} \quad (20)$$

따라서 이동속도는 다음과 같다.

$$V = \frac{L}{\Delta t} \quad (21)$$

여기서  $\Delta t$ 는  $t_1$ 과  $t_2$ 사이의 시간 간격이다.

V. 실험 및 고찰

물체의 이동거리를 측정하기 위해서 그림13과 같이 카메라를 물체로부터 R=97cm 떨어진 장소에 설치해 놓고 물체의 수평이동거리 범위를 44cm로 설정하였다.

시스템구성은 그림14와 같이 카메라의 영상신호를 양자화하기 위해서 디지털이저를 통해 컴퓨터 메모리에 DMA 방식으로 저장시켰다. 디지털이저의 그레이 레벨(gray level)은 바이너리(binary)로 설계하였다.

양자화된 배경의 화상데이터를 컴퓨터에 기억시켜 놓고,  $t_1$ 인 시간에 샘플링된  $I_f$  화상데이터를 보조 기억장치에 저장시킨후  $t_2$ 인 시간에  $I_s$  화상데이터를 기억장치에 격납시킨다. 저장된 화상데이터를 가지고 그림15과 같은 순서로 처리하였다. 배경에 대한 화상을  $280 \times 192$  dots 크기로 양자화하여 양자화된 화상데이터를  $3 \times 3$ 비트로 영역분할을 하고난 후 배경에 대한 화상데이터 화일을 구성했다. 이와같은 방법으로  $I_f$ 와  $I_s$ 의 화상데이터 화일을 생성시킨후 차화상 추출시 생기는 오차를 줄이기 위해서 화상데이터를 재배열 시켰다. 재배열된 화상데이터를 가지고 배경과  $I_f$ 와의 화상을 EX-OR시켜서 차화상을 추출하고 휘도치를 해석한 후 윤곽검출과 벡터스들을 각각 검출하였다. 그림 16는  $I_{DB1}$ 과  $I_{DB2}$ 의 휘도를 나타낸 것이며 표 3은  $I_{DB1}$ 과  $I_{DB2}$ 의 휘도치를 비교한 것이다. 표 3 를 보면 휘도오차가 FLY모델에서 가장 많은것을 알 수 있다. 이 원인은 FLY·모델의 색상, 물체의 복잡성 때문이다.

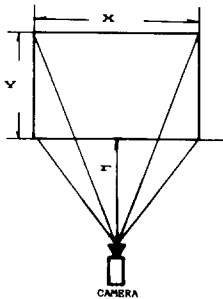


그림13. 카메라 설치 거리  
Fig. 13. Camera setting distance.

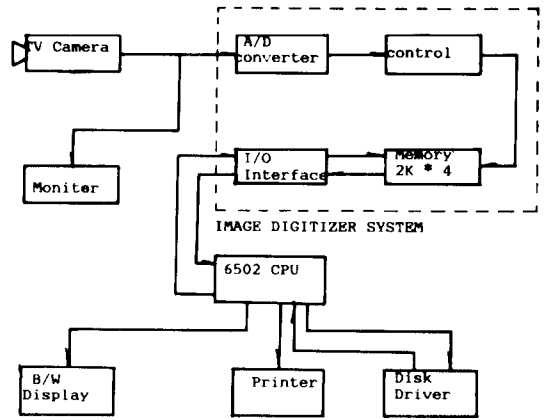


그림14. 시스템 구성도  
Fig. 14. System block-diagram.

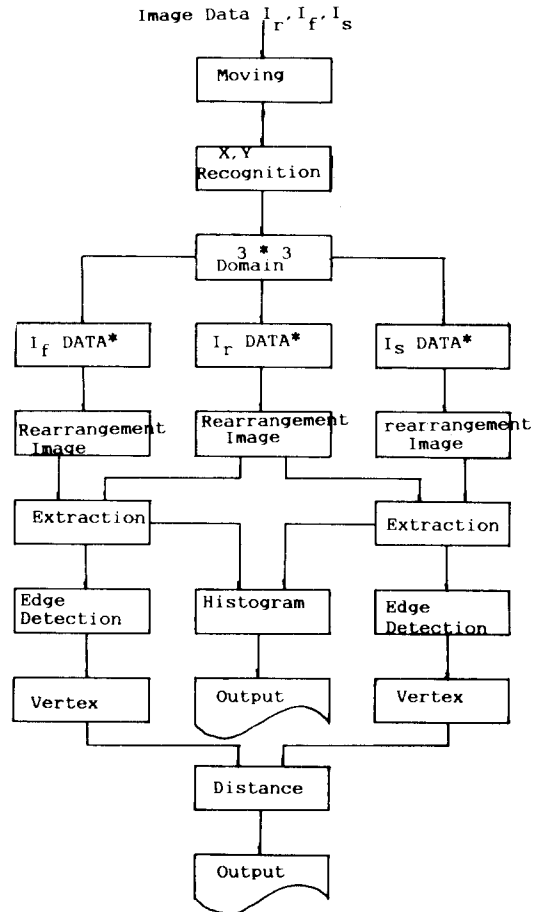
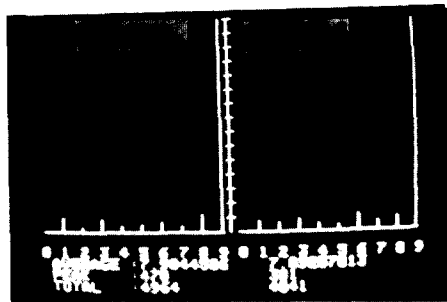
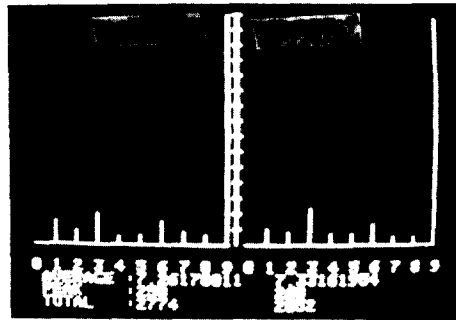


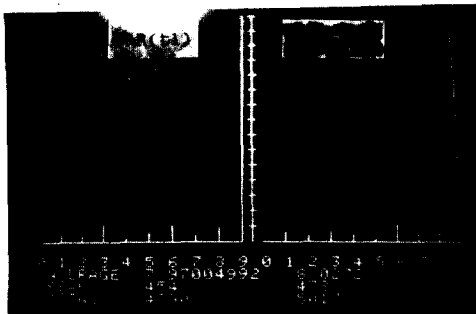
그림15. 처리순서도  
Fig. 15. Processing sequence.



(a) Histogram of FLY Model



(b) Histogram of JEEP Model



(c) Histogram of CAR Model

그림 16. 화상 휘도치

Fig. 16. Brightness of image.

표 3에서 오차가 모두 20%이내 임을 알 수 있다. 그림 17는  $t_1$  및  $t_2$ 인 시간에 각 물체의 윤곽화소 및 물체의 버텍스들을 검출한 것이다. 표 4은 물체의 버텍스들을 비교한 것이다.  $I_{b1}$ 의 버텍스와  $I_{b2}$ 의 버텍스 수가 JEEP 모델에서는 서로 차이가 있지만 나머지 모델은 서로 같다. 버텍스 수가 차이가 나는 원인은 모델구조의 복잡성 때문으로 본다. 따라서 인식과정에서  $I_{b1}$ 과  $I_{b2}$ 의 버텍스 오차를  $\pm 2$ 개 이내로 하였다. 또한 표 5는 버텍스와 버텍스사이의 길이를 나타낸 것이다. 표 5에서  $I_{b1}$ 과  $I_{b2}$ 의 버텍스와 버텍스 사이의 길이는 모두 10%이내인 것을 알 수 있었다. 표 6는 물체의 실제이동거리와 컴퓨터에 의한 측정결과이다. 측정

표 3.  $I_f$  화상과  $I_s$  화상비교

Table 3. Comparison of  $I_f$  image with  $I_s$  image.

MODEL.		물체의 총 휘도치	오 차
FLY.	$I_f$ Image	4964	18.59%
	$I_s$ Image	4041	
JEEP	$I_f$ Image	2774	2.18%
	$I_s$ Image	2852	
CAR	$I_f$ Image	4790	4.74%
	$I_s$ Image	5017	

표 4. 버텍스 수

Table 4. Number of vertex point.

Model	$I_f$ Image Vertex point	$I_s$ Image Vertex point
FLY	12	12
JEEP	14	12
CAR	12	12

표 5.  $I_{b1}$  선분길이라  $I_{b2}$  선분길이의 비교

Table 5. Comparison of  $I_{b1}$  line length with  $I_{b2}$  line length.

Model	$I_{b1}$			$I_{b2}$		
	$a_1$	$a_2$	$a_3$	$a_1$	$a_2$	$a_3$
FLY.	26	25	38.59	25.85	25.20	37
JEEP	30.68	17.89	30.68	30.68	19.70	30.68
CAR	22.47	24.35	27.73	23.02	24.04	28.64

표 6. 실험결과

Table 6. Experimental result.

Model	실정거리	실제이동거리	측정거리	오 차	속 도
FLY.	44cm	7cm	8cm	14.3%	8 cm/10 <sub>s</sub>
JEEP	44cm	11cm	12cm	9.09%	12cm/10 <sub>s</sub>
CAR	44cm	11cm	12cm	9.09%	12cm/10 <sub>s</sub>

한 거리는 3 가지 모델에 대한 평균오차가 약 10.8% 이었다. 이동물체의 속도는  $t_1$ 에서  $t_2$ 의 시간간격을 가지고 이동속도 시간으로 보았다. 본 실험에서는 시간 간격을 10초로 하였으며 처리하는데 걸리는 시간은 1시간 20분이 소요되었다.

### VI. 結 論

2 차원 연속화상을 등간격으로 샘플링한 3 쪽의 프레임용 가지고 이동물체의 이동거리와 속도측정을 한 결과 다음과 같은 결론을 얻었다.



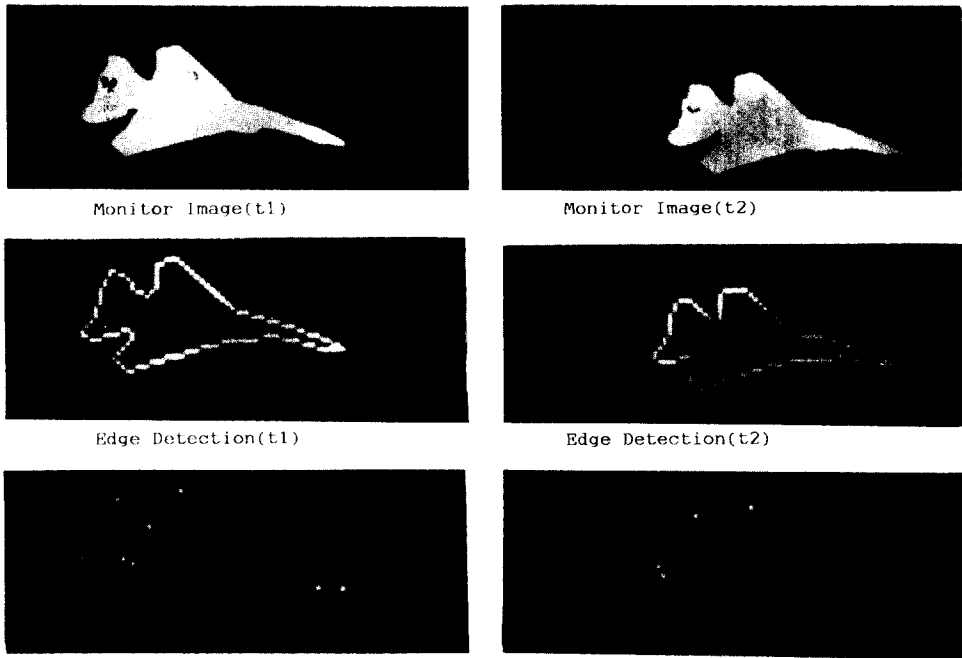


그림17. (a) 비행기모델  
Fig. 17. (a) FLY. model.

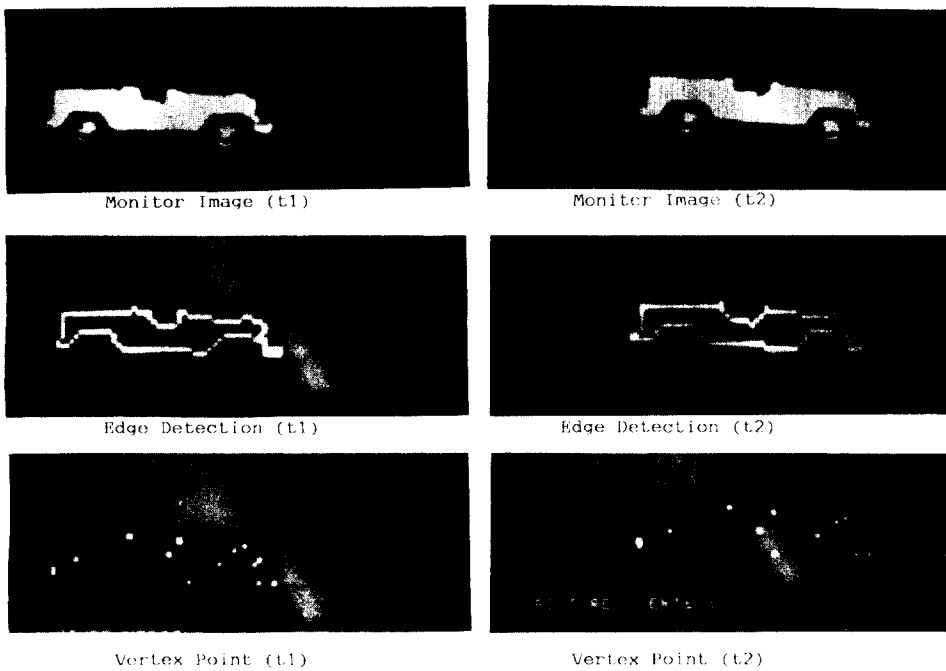


그림17. (b) 지프모델  
Fig. 17. (b) JEEP model.

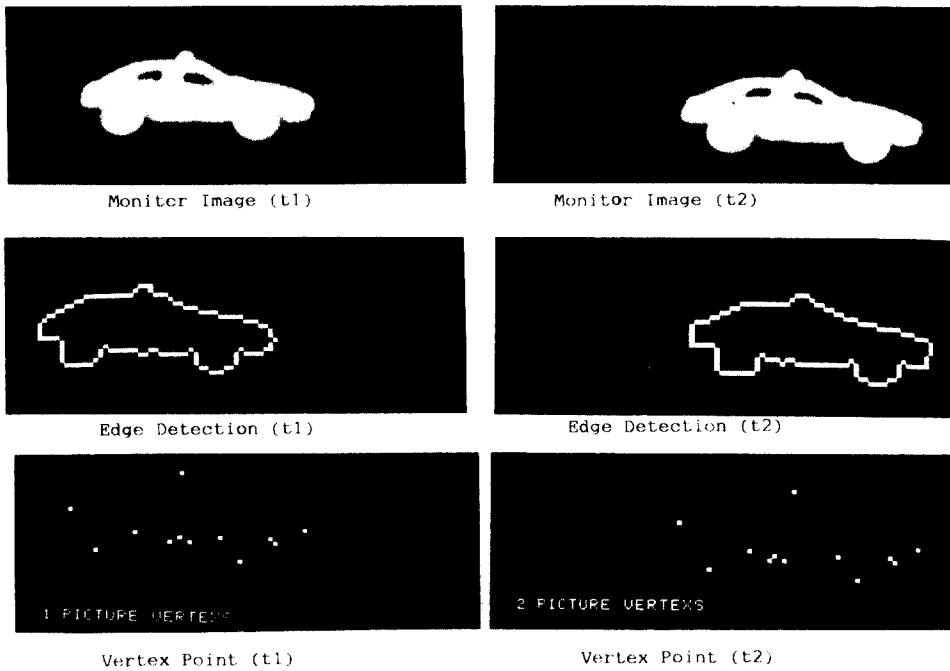


그림17. (c) 자동차모델  
Fig 17. (c) Model CAR.

1. 휘도치 변화에 기인하는 차화상 오차를 줄이기 위한 방법으로서는 화소의 비트데이터를 재배열한 결과 양질의 차화상을 추출할 수 있었다.
2. 동일물체 판별은 첫번째 추출된 차화상과 두번째 추출된 차화상의 휘도치 오차를 20% 이내로, 버텍스 수의 오차를  $\pm 2$  개 이내로, 버텍스와 버텍스 길이의 오차는 10% 이하로 결정하였을 때 복잡한 물체도 동일물체 판별이 가능하다.
3. 컴퓨터내에 좌표계를 설정하여 실제거리와 대응시키므로써 물체의 이동거리를 측정할 수 있다.
4. 이동거리와 샘플링과 샘플링 시간간격을 이용하여 속도를 환산할 수 있다.
5. 이동속도가 빠른 물체를 측정하기는 현재 곤란하지만 이정도 프레임 주파수가 높은 센서를 사용하면 충분히 가능하다.
6. 본 연구에서는 2 차원내에서 직선 운동을 하고 있는 물체를 대상으로 측정이었다. 따라서 물체가 회전하며, 변위되어가는 것도 측정하고자 계속연구 중이다.

參 考 文 獻

- [1] R. Jain. "Extraction of motion information from peripheral-processes." *IEEE Trans. on PAMI* vol. PAMI-3, no. 5, pp. 489-503, Sept., 1981.
- [2] WB, Thompson. "Combining motion and contrast for segmentation." *IEEE Trans. PAMI.*, vol. Pami-2, no. 6, pp. 543-549. Nov., 1980.
- [3] GR. Legters, JR. and T.Y. Young. "A mathematical model for computer image tracking." *IEEE Trans. on PAMI.*, vol. PAMI-4, no. 6, pp. 583-594. Nov., 1982.
- [4] M. Yachida, M. Asada and S. Tsuji, "automatic analysis of moving image." *IEEE Trans. PAMI.*, vol. PAMI-3, no. 1, pp. 12-20, Jan. 1981.
- [5] Byong won HWANG and Sadao TAKABA. "Real-time measuring of pedestrian flow using processing of ITV image," *The Trans. IECE, JAPAN.* vol. J66-D, no. 8, pp. 917-924. Agust.
- [6] C. Cafforio and F. Rocca. "Methods for measuring small displacements of television

- image.” *IEEE Trans.*, vol. IT-22, no. 5, pp. 573-579, Sept., 1976.
- [7] Y. Ninomiya and Y. Ohtsuka. “A motion compensated interframe coding scheme for television pictures.” *IEEE Trans. on Communication.*, vol. COM-30, no. 1, pp. 201-211, Jan., 1982.
- [8] I.L. Ayalai Da. Ortoni J.B. Larson. and D.F. “Elliott moving Target Tracking using Symbolic Registration”, *IEEE Trans. on PAMI.* vol. PAMI-4, no. 5 pp. 515-520, Sept., 1982.
- [9] C.L. Fennema and W.B. Thompson’ “Velocity determination in scenes containing several moving objects,” *Comput. Graphics Image Processing*, Pox vol. 9, pp. 301-315, Apr. 1979.
- [10] R.A. Samy. and C.A. Bojjo. *Moving Object Recognition Using Motion Enhanced HOUGH Transform.* Digital Signal Processing-84, V. Capplini and A.G. Constantinides Editors, North-Holland, pp. 770-775, 1984.
- [11] P. Speor, *Moving Object Detection by Temporal Frame Defference Accumulation* Digital signal Processing-84, V. Cappellini and A.G. Constantinides editors, North-Holland, pp. 900-907, 1984.
-