

Difference Picture를 이용한 이동벡터의 추정과 이동물체의 추출

(A Displacement Vector Estimation and Moving Object Extraction Using Difference Picture)

張 淳 華*, 金 鍾 大*, 金 聖 大*, 金 在 均*

(Soon Hwa Jang, Jong Dae Kim, Seong Dae Kim and Jae Kyoon Kim)

要 約

본 논문은 DP(difference picture)를 이용하여 이동벡터를 추정하고 이동물체를 추출하는 새로운 방법을 제안한다. 먼저 두 연속 영상에서 물체의 경계와 DP영역의 경계와의 관계를 분석하고 이로부터 이동벡터를 추정하는 알고리즘이 제안된다. 그리고 이 알고리즘으로 추정된 이동벡터를 이용하여 이동물체를 DP로부터 직접 추출한다.

제안된 알고리즘은 원래의 밝기 영상을 처리하지 않으므로 수행시간이 짧고 따라서 실시간 구현에 적합하다. 실험 결과로부터 DP만 잘 추출된다면 제안된 알고리즘은 복잡한 배경, 빠르거나 느린 운동, 회전운동 그리고 움직임 영역이 아닌 곳에서의 occlusion등의 상황에서도 잘 적용됨을 알 수 있다.

Abstract

This paper proposes new algorithms for the estimation of displacement vector and moving object extraction using difference picture. First, the relations between the boundary of moving objects in two consecutive images and the boundary of difference picture regions are analyzed, then displacement vector estimation algorithm is proposed. Using the estimated displacement vector, moving objects are directly extracted from difference picture.

Since the proposed algorithms do not process gray-valued image, they have a short processing time and are suitable to real time processing. From the experimental results, we observed that, if difference picture is well extracted, the proposed algorithms work well even in the circumstances of complex background, fast or slow motion, rotation etc., including occlusion where is not moving area.

*正會員, 韓國科學技術院 電氣 및 電子工學科
(Dept. of Electrical and Elec. Eng., KAIST)
接受日字: 1987年 6月 10日

I. 서 론

연속 영상에서 물체의 이동벡터를 추정하고 이동물체를 추출하는 것은 운동 영상해석(dynamic scene

analysis)의 기본 단계인데 방법에 따라 크게 화소를 이용한 방법(pixel based method)과 특징을 이용한 방법(feature based method)으로 나눌 수 있다.^{11,12} 화소를 이용한 방법에는 다시 시공간 경사방법(spatio-temporal gradient method)과 정합법(matching)으로 나눌 수 있는데 전자는 속도가 작다는 가정을 해야 하고 후자는 계산시간이 길다는 단점이 있다.^{13,14} 화소를 이용한 방법으로부터 이동벡터를 추정한 후 이동물체를 추출하려면 화소 단위의 grouping이 필요하다.¹⁵ 특징을 이용한 방법은 이동벡터와 이동물체를 특징 영역에서 추출하므로 비교적 빠른 방법이 되나 각 영상에서 영역분할(segmentation)과 특징추출(feature extraction)을 해야하는 문제가 발생한다.^{17,18}

위의 방법과는 다르게 두 연속 영상으로부터 DP(difference picture)를 구하고 이를 이용하여 운동 영상을 해석하는 방법도 많이 연구되어 왔다.^{19,20} DP란 두 연속영상의 차의 절대치를 임계치화(thresholding)하여 얻은 이진영상(binary image)을 말한다. 따라서 두 영상사이의 변화가 있는, 즉 움직임이 있는 부분만 DP의 영역으로서 나타나게 되므로 영상에서 처리해야 할 영역이 그만큼 줄어들게 되어 위의 두 방법보다는 비교적 빠른 방법이 된다. DP를 이용한 대표적인 방법은 Aggarwal등이 제안한 region growing을 이용한 방법^{21,22}과 Jain이 제안한 DP 영역의 경계를 해석하여 물체의 이동정보를 추출하는 방법²³이 있다.

Region growing을 이용한 방법은 DP로부터 움직이는 물체가 존재하는 영역중 일부를 찾아내고, 여기에서 region growing을 할 출발점과 물체의 밝기의 통계 특성을 구하여 원래의 밝기 영상(gray-value image)에서 region growing을 행하는 방법이다. 이 방법은 출발점이 정확하게 추출될 경우는 안정하게 동작하나, 한 물체에 의한 DP가 여러개로 나뉘어져 있을 경우 출발점을 찾기가 어렵고 DP와 원래 영상 사이를 빈번하게 access하게 된다. Jain이 제안한 decision tree를 이용하는 방법은 DP 영역의 경계를 과거 영상의 경계와 현재 영상의 경계로 분류한 후 두 경계 화소의 비를 비교하여 물체의 움직임 정보 즉 병진운동(translation), 다가섬(append), 멀어짐(receding), 겹침(occlusion) 등을 알아낸다.

본 논문에서는 먼저 이동 물체에 의해서 생긴 각 DP 영역들의 이동벡터를 추정한 후 이로부터 이동물체를 추출하는 알고리즘을 제안한다. 이 방법은 DP로부터 직접 이동벡터와 이동물체를 추출하므로 원래의 밝기 영상을 처리하지 않아 빠른 처리 속도를 얻을 수 있다는 장점이 있다.

II. 본 론

1. DP와 DP 경계의 추출

두 연속 영상을 각각 $F_{i-1}(x, y)$, $F_i(x, y)$ —이하 (x, y) 는 생략함—라고 할때 DP_i 는 다음과 같이 얻어진다.

$$DP_i = 0 \text{ if } |F_i - F_{i-1}| < T_1$$

$$1 \text{ if } |F_i - F_{i-1}| \geq T_1$$

임계치 T_1 은 F_{i-1} 과 F_i 사이의 밝기 변화를 고려하여 하나 두 연속 영상 사이의 시간 간격이 짧으면 밝기 변화도 이에 비례해 작아진다. 본 논문에서는 실험을 통하여 10에서 30사이의 값을 사용하였다.

만일 영상에 잡음이 많거나 처리시간의 단축을 요할 경우에는 영상을 축소하여 DP를 구할 수도 있는데, 즉 영상의 $N \times N$ block이 DP의 1 화소에 대응되고 DP의 화소는 F_{i-1} , F_i 의 대응하는 두 block의 평균과 분산으로부터 다음과 같이 구하는 방법도 있다.²⁴

$$DP_i = 0 \text{ if } \frac{[(S_p + S_c)/2 + ((m_p - m_c)/2)^2]}{(S_p \times S_c)} < T_2$$

$$1 \text{ else}$$

여기서

m_p, m_c : 과거 영상과 현재 영상의 block의 평균(mean)

S_p, S_c : 과거 영상과 현재 영상의 block의 분산(variance)

DP 영역의 경계는 항상 움직인 물체의 경계와 일치하는데, 현재 영상의 물체의 경계와 일치할 수도 있고 과거 영상의 물체의 경계와 일치할 수도 있다. 이를 알아내는 방법은 F_i 와 F_{i-1} 의 edge 영상을 구하고 DP_i 의 경계와 같은 위치에서 F_i 의 edge 값의 크기가 클 경우는 현재 영상의 경계가 되고 F_{i-1} 의 edge 값의 크기가 클 경우는 과거 영상의 경계가 된다.

위와 같이 하여 얻어진 DP 영역과 DP 영역의 경계를 적절하게 표현(representation)하기 위하여 본 논문에서는 DP 영역의 경계를 시계 반대 방향으로 추적하면서 경계 길이 부호(chain run-length code)를 추출하였다. 단 영역을 이루는 화소의 갯수가 적을 때에는 처리하지 않는다. 본 논문에서는 10개 이하의 화소들로 이루어진 영역은 처리하지 않았다. 하나의 run은 8-connected 된 과거 혹은 현재 영상의 경계 화소들의 집합이며 run의 길이는 해당하는 경계화소의 갯수를 나타낸다. 과거 영상의 경계화소들로 이루어진 run을 과거경계, 현재 영상의 경계화소

들로 이루어진 run을 현재경계라 명명한다.

4-connected 화소들로 이루어진 DP 영역은 현재 영상에서 이동물체가 될 수도 있고 배경이 될 수도 있다. 전자를 O-type 영역, 후자를 B-type 영역이라 한다. 볼록한(convex) 물체가 움직일 경우 O-type 영역은 현재경계가 과거경계보다 길며 B-type 영역인 경우는 이와 반대가 된다. 그러나 이동물체의 오목한(concave) 부분에 의해 생기는 DP는 이 사실이 성립하지 않고 영상에 잡음이 생긴 경우등을 고려하여 DP 영역의 과거경계와 현재경계의 길이의 비가 적당한 임계치 사이에 있을 때는 X-type 이라 정의 한다. 따라서 DP 영역의 type은 다음과 같이 정해진다.^(10,11)

- O-type if 현재경계의 길이/과거경계의 길이 > 1 + T3
 - B-type if 현재경계의 길이/과거경계의 길이 < 1 - T3
 - X-type else
- 여기서 $0 < T3 < 1$

임계치 T3의 값은 물체의 움직임이 2 화소 이상일 경우 이론적으로 0이 될 수 있으나 영상의 잡음등을 고려하여 0.2~0.3 정도의 값을 사용하였다. 그림 1에 DP의 영역과 경계를 물체의 운동에 따라 분류하여 나타내었고 DP 및 DP 경계를 추출하기까지의 알고리즘은 다음과 같다.

[알고리즘 1] DP 및 DP 경계의 추출

1. 연속 영상 F_{i-1} , F_i 를 얻는다.
2. 차영상 $|F_{i-1} - F_i|$ 를 구한 후 이를 임계치 T1을 기준으로 이진 영상화하여 DP를 구한다.
3. 구한 DP에서 4- 혹은 8-connected 화소들로 이루어진 영역들을 구한다.
4. 화소의 갯수가 10 이하인 영역들을 제거한다.
5. 각 영역의 경계화소의 좌표에서 F_{i-1} 과 F_i 에 sobel edge operation을 행한다. 만일 F_i 에서의 edge 값이 F_{i-1} 에서의 edge 값보다 크면 현재경계의 화소, 그렇지 않으면 과거경계의 화소가 된다.
6. 각 영역의 경계를 추적(boundary following) 하면서 연속된 현재경계 화소들의 집합을 현재경계, 과거경계의 화소들의 집합을 과거경계로서 기록한다.
7. 현재경계 혹은 과거경계의 화소수가 0이면 해당 영역을 제거한다.
8. 만일 (현재경계의 화소수)/(과거경계의 화소수)가 $1 + T3$ 보다 크면 해당 영역을 O-type 영역, $1 - T3$ 보다 작으면 B-type 영역, 그렇지 않으면 X-type 영역으로 기록한다.

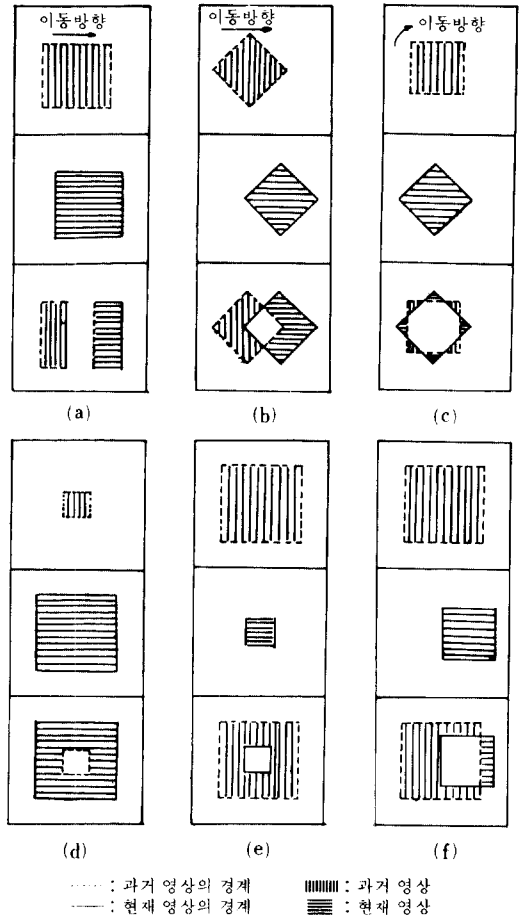


그림 1. 물체의 여러가지 운동에 의한 DP
 (a) 병진운동 1 (b) 병진운동 2
 (c) 회전운동 (d) 다가섬
 (e) 멀어짐
 (f) 병진운동과 멀어짐 운동이 동시에 일어남

Fig. 1. DP due to various motions of object.
 (a) Translation 1. (b) Translation 2.
 (c) Rotation. (d) Approaching.
 (e) Receding.
 (f) Translation and receding at the same time.

2. DP 경계의 해석
 DP 영역의 경계는 항상 이동물체의 경계의 일부 분이다. 그러나 이동물체의 경계가 항상 DP 영역의 경계에 나타나는 것은 아닌데, 그 이유는 현재 영상의 이동물체의 경계와 과거 영상의 이동물체의 경계가 겹치기 때문이다.
 이것은 다음 두가지 경우에 나타난다.
 (1) 물체가 자신의 경계 방향으로 움직인 경우(그림 2의 (a))

(2) (1)의 경우가 아니면서 과거 영상의 이동물체의 경계와 그곳으로 이동한 현재 영상의 이동물체의 경계가 접치는 경우(그림 2의 (b))

대개의 경우 (2)와 같이 물체의 경계가 정확히 접치는 경우는 거의 없으나 (1)과 같은 경우는 물체가 자신의 경계 방향으로 직선 운동을 할 경우 자주 나타나게 된다. 따라서 위와 같은 경우를 제외하면 이동물체의 경계는 항상 DP 영역의 경계에 나타나고, DP는 2 장의 영상으로부터 얻어졌으므로 DP영역의 현재경계와 과거경계 사이에는 서로 부합하는 쌍이 존재한다. 부합하는 경계가 같은 DP 영역내에 존재할 때 *matched boundary*, 다른 DP 영역에 존재할 때 *totally displaced boundary*, 그리고 (1)과 같이 물체가 경계 방향으로 움직여 부합하는 경계가 사라진 경계를 *motion directed boundary*라 명명한다 (이하 각각 MB, TDB, MDB로 약함). 또 MDB와 TDB가 있을 경우에는 항상 MB의 양쪽 끝에만 나타난다.

그림 3에 이들의 구성을 나타내었다.

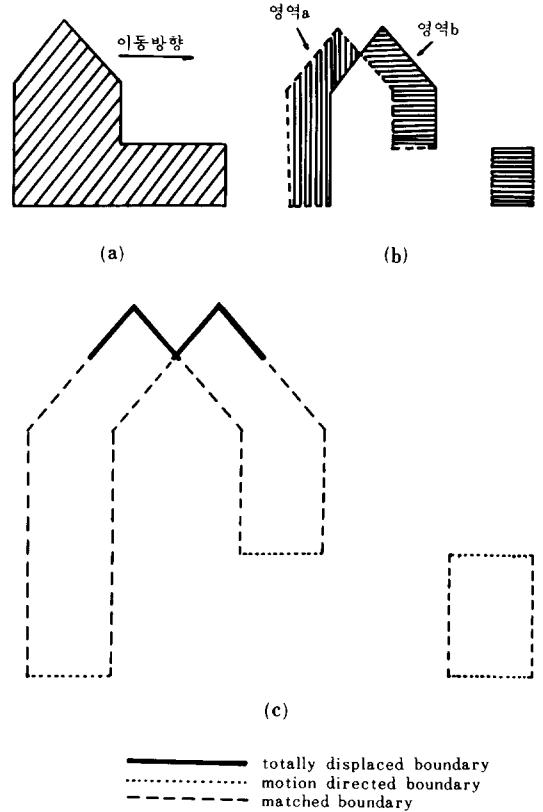


그림 3. DP 경계의 구성

- (a) 이동물체
 - (b) DP 영역
 - (c) 물체의 형태에 따른 DP 경계의 구성
- Fig. 3. Structure of DP boundary.
- (a) Moving object.
 - (b) DP region.
 - (c) Structure of DP boundary due to object shape.

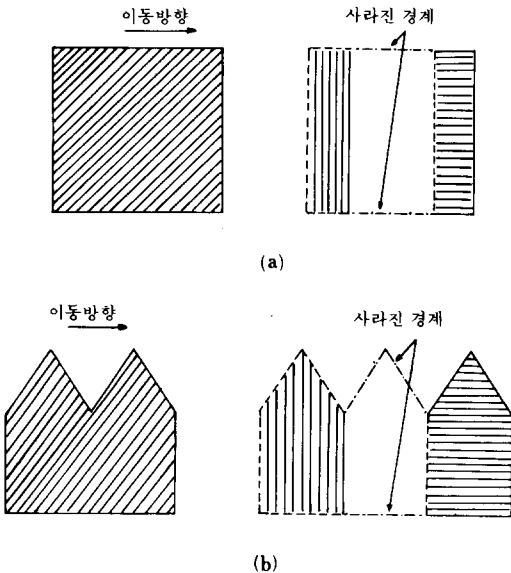


그림 2. 물체의 경계가 DP에서 사라지는 경우
(a) 물체가 자신의 경계 방향으로 움직임
(b) 물체의 경계끼리 겹침

Fig. 2. Case which object boundary is disappeared in DP.

- (a) Object moves in the direction of its boundary.
- (b) Object boundary is overlapped.

위에서 분류한 DP영역의 경계는 물체의 볼록한 (convex) 부분과 오목한 (concave) 부분에 따라 특성이 다르게 나타난다.

볼록한 부분에 의해 생기는 DP 영역은 그림 3(b)의 영역a와 같으며, 이 경우 현재경계와 과거경계 사이에는 상호간에 포함관계를 이루는데 B-type 영역인 경우에는 현재경계가 과거경계에 포함되며 O-type 영역인 경우에는 이와 반대가 된다. 포함하는 경계는 그림 3(c)에서와 같이 TDB, MDB, MB로 구성되고 포함되는 경계는 MB로만 이루어진다. 볼록한 부분의 특징은 이와같이 TDB, MDB가 현재경계와 과거경계중 한 쪽에만 존재한다는 것이다.

오목한 부분에 의해 생기는 DP 영역은 그림3(b)의 영역b와 같다. 볼록한 부분의 경계와 달리 현재경계, 과거경계간에 포함관계가 성립하지 않는다. 양쪽 모두 TDB, MDB, MB를 가질 수 있다.

TDB는 다른 영역으로 옮겨간 경계이므로 DP 영역들 중에는 공통의 TDB를 가지고 있는 영역의 쌍이 존재한다. 이때 두 영역은 서로 연결되었다고 한다. 그림 4의 (a)는 두 영역이 연결된 경우이고(b)는 여러개의 영역이 서로 연결된 경우를 나타낸다. 그림에서처럼 연결된 DP 영역은 8-connected 되어 있는데 이때 두 영역이 만나는 점에서는 반드시 각 영역의 현재경계와 과거경계가 만나는 점이 된다. 현재경계와 과거경계가 만나는 점을 교차점(crossing point)이라 하고 교차점들 중에서 연결된 DP 영역이 만나는 점을 연결점(connected point)이라 명명한다.

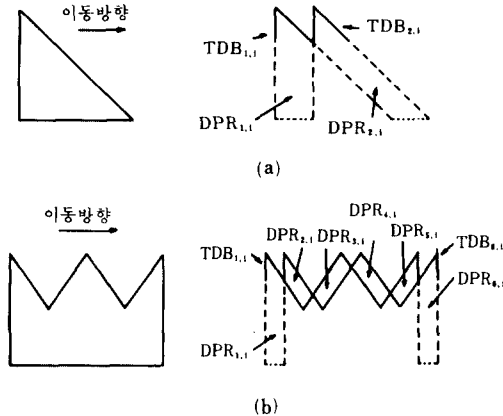


그림 4. 연결된 DP 영역

- (a) 2 개의 DP 영역이 연결된 예
- (b) 6 개의 DP 영역이 연결된 예

Fig. 4. Connected DP regions.

- (a) Example of two connected regions.
- (b) Example of six connected regions.

3. 이동벡터의 추정

연결된 두 DP 영역은 항상 TDB가 존재하며 그것이 한쪽 영역에서 현재경계에 속해 있으면 다른 한쪽 영역에서는 반드시 과거경계에 속해 있음을 알 수 있다. 또한 MB도 자신의 DP 영역내에 부합하는 경계가 존재하며 마찬가지로 한 쪽이 현재경계에 속해 있으면 다른 한 쪽은 과거경계에 속해 있다. 즉 TDB와 MB는 물체의 경계가 이동하여 생긴 경계의 쌍이다. 그러나 MDB는 현재경계와 과거경계간에부

합하는 쌍이 존재하지 않는다. 따라서 연결된 DP 영역들에서 MDB만 제거한 후 과거경계와 현재경계 사이의 평균 좌표로부터 물체의 이동벡터를 추출할 수 있다.

실제의 알고리즘에서는 연결된 DP 영역과 MDB를 찾는 것이 중요한 문제가 된다. 연결된 DP 영역이 8-connected 되어있다 하더라도 digital 영상에서는 4-connected된 경우, 8-connected된 경우 그리고 이웃하지 않는 경우도 생기게 된다. 따라서 알고리즘에서는 각 영역의 교차점간의 거리가 적당한 임계치 T4보다 작을 때 두 영역을 연결된 DP 영역으로 한다.

MDB는 볼록한 부분에서는 항상 MB의 양쪽 끝에만 존재하고, 또 TDB와 함께 현재경계와 과거경계중 한쪽에만 몰려있으므로 쉽게 찾아낼 수 있으나, 오목한 부분이 있을 때에는 DP 영역의 국부적인 성질만으로는 유일하지 않을 수도 있게된다. 그러나 물체의 이동벡터가 물체의 형태에 비해 크지 않다면 전체 경계 화소중에서 MDB가 차지하는 비율이 충분히 작을 것이다. 이 가정하에서 먼저 연결된 모든 DP 영역들의 과거경계와 현재경계의 평균 좌표로부터 근사적인 이동벡터 V'를 구한 후 이를 이용하여 MDB를 추정하여, 추정한 MDB를 제외한 과거경계와 현재경계의 평균 좌표로부터 DP 영역의 이동벡터 V를 구한다. 알고리즘은 다음과 같다.

[알고리즘 2] 이동벡터의 추정

1. DP 영역들의 경계를 추적하면서 현재경계에서 과거경계로 혹은 과거경계에서 현재경계로 바뀌는 부분의 좌표를 교차점으로 기록한다.
2. 모든 교차점의 쌍으로부터 이들의 공간적 좌표간의 거리가 임계치 T4 보다 작은 쌍을 연결된 DP 영역으로서 기록한다.
3. k번째 연결된 DP 영역들의 집합 DPCR_k를 찾는다.
4. DPCR_k에 속한 모든 영역들의 근사적인 이동벡터 V'를 다음과 같이 구한다.

$$V'_k = \text{현재경계의 평균좌표} - \text{과거 경계의 평균좌표}$$

for all DP regions in DPCR_k

5. 연결점이 아닌 교차점으로부터 V'의 크기만큼 현재경계와 과거경계를 따라 추적하면서 MDB를 추정하여 이를 제외한 경계를 '현재경계', '과거경계'라 한다. MDB는 V'와 방향이 같거나 반대이면서 현재경계와 과거경계의 양 끝 부분에 위치한 경계이다.
6. 이동벡터를 다음과 같이 추정한다.

$$V_k = \text{현재경계'의 평균좌표} - \text{과거경계'의 평균 좌표}$$

for all DP regions in DPCR_k

그림 5는 이동벡터가 추출되는 과정을 그림으로 나타내었다.

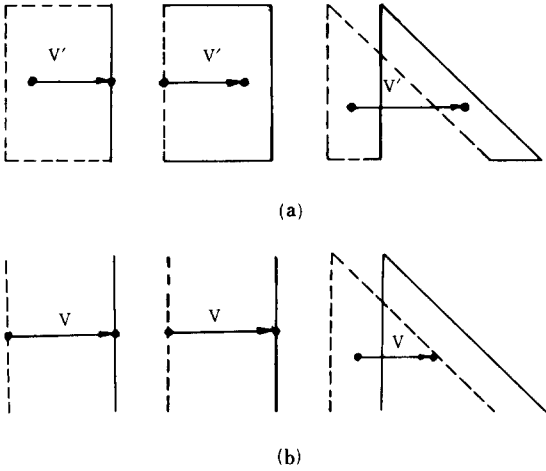


그림 5. 이동벡터의 추정 예
 (a) 근사적인 이동벡터
 (b) MDB를 제거한 후의 이동벡터
Fig. 5. Example of displacement vector estimation.
 (a) Approximate displacement vector.
 (b) Displacement vector after MDB is removed.

4. 이동물체의 추출

본 절에서는 앞에서 구한 이동벡터를 이용하여 DP 에 나타나지 않은 물체의 경계를 찾아냄으로써 이동물체를 추출하는 방법을 알아본다. 이동물체 추출방법은 물체의 운동 형태에 따라 다음과 같이 구분할 수 있다.

(1) 물체의 병진 운동에서 DP 영역내에 TDB가 존재하지 않을 경우는 물체가 자신의 경계방향으로 움직인 경우인데 이때 경계중의 일부가 사라지면서 DP 영역은 2개 이상으로 나뉘어진다. 사라진 경계는 운동 방향의 경계이고 DP 영역의 교차점과 교차점 사이에 존재한다. 따라서 물체의 이동벡터만 알면 교차점을 이동벡터 방향으로 연장하여 사라진 경계를 찾아낼 수 있다. 이때 교차점에서 연장한 직선은 대응하는 영역의 교차점과 만난다. 특히 대응하는 두 영역이 각각 B-type과 O-type이면 현재경계의 시점은 반드시 시점과, 종점은 반드시 종점과 만난다.

(2) 물체가 병진운동을 하면서 TDB가 존재할 경우는 반드시 대응하는 연결된 DP 영역이 존재하고, 따라서 대응하는 두 연결점을 연결시키면 된다.

(3) 회전 운동이 일어날 경우에는 경계끼리 겹치지 않는다면 모든 DP 영역은 연결되어 있으므로 대

응하는 연결점끼리만 연결시키면 이동물체를 추출할 수 있다.

(4) 물체가 멀어지거나 혹은 다가서는 운동을 할 경우에는 앞에서의 경계해석이 적용되지 않는다. 그러나 경계끼리 겹치지만 않는다면 그림 1의 (f)와 같이 모든 영역이 연결되어 있거나, (d) (e) 와 같이 현재 경계와 과거경계중 하나가 다른 하나를 둘러싸고 있다. 따라서 전자의 경우 대응하는 교차점 끼리만 연결시키고 후자인 경우에는 현재경계 혹은 과거경계로부터 이동물체를 추출할 수 있다.

본 알고리즘에서는 DP 내의 모든 교차점 쌍에 대해 위의 사실에 의하여 연결된 cost 를 구하여 그 값이 최소인 교차점 쌍부터 차례로 연결하였다. (2) (3),(4)번의 경우처럼 두 교차점이 연결점인 경우는 반드시 연결되어야 하므로 cost 는 0 이다. 그러나 (1) 과 같이 물체가 자신의 경계방향으로 움직여 경계가 사라졌을때 생긴 두 DP 영역은 이동벡터의 방향, 크기 그리고 영역의 type 으로부터 cost 를 계산한다.

(1)과 같은 경우 다음 사실이 성립한다.

1. 현재경계와 과거경계의 교차점에서 이동벡터 방향으로 연장했을 때 대응하는 DP 영역의 교차점과 만난다.
2. 대응하는 두 DP 영역의 이동벡터는 같다.
3. DP 영역이 한 쪽이 B-type(O-type)일 때 다른 하나는 O-type(B-type) 혹은 X-type 이 된다. X-type 이 생기는 경우는 오목한 물체일 때 생기게 된다.

따라서 두 교차점간의 연결 cost 는 다음 세 변수에 의해 결정된다.

$X_{k,1}$: k번째 교차점에서 1번째 교차점으로 연결한 직선이 k번째 교차점이 속하는 영역의 이동벡터와 이루는 각도.

$Y_{k,1}$: 대응하는 두 영역의 이동벡터의 차이.

$Z_{k,1}$: 두 영역의 type 사이의 부합 정도.

B-O 혹은 O-B : 연결 확률이 높음.

B(O)-X 혹은 X-X : 연결될 수 있음.

B-B 혹은 O-O : 연결될 수 없음.

변수 X와 Y는 연속인 값을 가지나 Z는 불연속적인 값을 갖는다. 본 논문에서는 위의 세 변수와 연결 cost $c_{k,1}$ 과의 관계를 선형적으로 근사화하였다. 따라서 앞에서의 사실과 함께, DP 내의 k 번째 교차점으로부터 1번째 교차점으로 연결될 cost 는 다음과 같다.

$$c_{k,1} = m_1 * X_{k,1} + m_2 * Y_{k,1} + m_3 * Z_{k,1}$$

단

(1) 두 교차점이 연결점이면 $c_{k,1} = c_{1,k} = 0$ 으로 한

다.

(2) $k=1$ 이거나 두 교차점이 같은 영역에 속해 있으면 $c_{k,1} = c_{1,k} = \infty$ 로 한다.

(3) 두 교차점이 속한 영역의 type 이 각각 O-type, B-type 혹은 B-type, O-type 이고, 두 교차점이 각각 자신이 속한 경계부호의 시점과 종점 혹은 종점과 시점이면 $c_{k,1} = c_{1,k} = \infty$ 으로 한다.

k 번째 교차점과 1 번째 교차점이 서로 연결될 cost $C_{k,1}$ 은 k 번째 교차점에서 1 번째 교차점으로 연결될 cost $c_{k,1}$ 과 1 번째 교차점에서 k 번째 교차점으로 연결될 cost $c_{1,k}$ 에 의존한다. 본 논문에서는 이들의 합으로서 cost $C_{k,1}$ 을 정하였다.

$$C_{k,1} = C_{1,k} = c_{k,1} + c_{1,k}$$

알고리즘상에서는 임계치 T5를 정하여 그 이상의 cost에 대해서는 처리를 하지 않는다. 임계치 T5는 처리한 교차점 쌍의 연결 결과의 신뢰도에 대한 기준이 된다. 본 논문에서는 $C_{k,1}$ 의 평균값으로 T5를 정하였다. 알고리즘은 다음과 같다.

[알고리즘 3] 이동물체의 추출

(1) 모든 교차점들을 모은다.

(2) k 번째 교차점에서 보았을 때 1 번째 교차점과 연결된 cost $c_{k,1}$, $k=0,1,\dots,N, 1=0,1,\dots,N$ 을 계산한다. 여기서 N은 DP 내의 교차점의 총 수이다.

(3) k 번째 교차점과 1 번째 교차점이 연결될 cost

$$C_{k,1} = C_{1,k} = c_{k,1} + c_{1,k}$$

를 계산한다.

(4) cost가 가장 최소인 하나의 교차점 쌍이 임계치 T5보다 작을 때 이들을 기록하고 cost table에서 제외시킨다.

(5) 최소인 cost가 T5를 넘으면 (6)번으로, 그렇지 않으면 (4)번으로 간다.

(6) (4)번에서 구한 모든 교차점 쌍끼리 연결한 직선과 현재경계에 의하여 만들어진 곡선이 폐곡선일 경우 물체의 경계로서 기록하고 그렇지 않을 경우는 이동 물체 추출이 실패한 경우이다.

그림 6에서 예를 든 영상의 현재경계와 과거 경계를, 표 1에서 각 변수값과 cost $c_{k,1}$, $c_{1,k}$ 를 나타내었고, 표 2에서 cost $C_{k,1}$ 과 처리과정을 보였다. 이 영상에 대한 실험 결과는 그림 8에 나타나 있다.

III. 실험 결과 및 고찰

실험에서 사용한 각 cost와 cost의 계수 그리고 임계치 T1, T3, T4, T5는 다음과 같다.

$X_{k,1}$: 이동벡터와 두 교차점을 연결한 직선이 이루

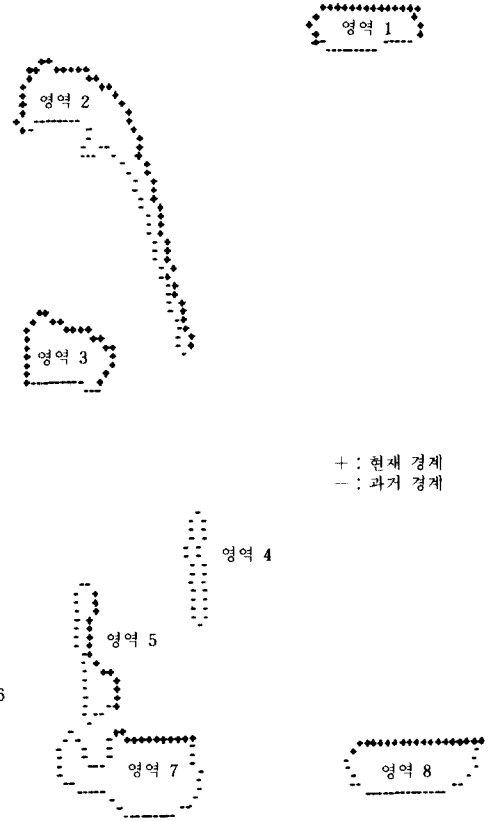


그림 6. 이동물체 추출 알고리즘을 적용한 영상의 DP 영역의 경계

Fig. 6. Boundary of DP regions of image tested for moving object extraction algorithm.

는 각 (degree)

$Y_{k,1}$: 두 교차점이 속한 영역의 이동벡터의 차이 (angle difference + 10* magnitude difference)

$Z_{k,1}$: 두 교차점이 속한 영역의 type 사이의 차이

O-B, B-O: 0

B-X, O-X, X-X: 30

O-O, B-B: 90

T1: 10~30

T3: 0.2~0.3

T4: 1~4

T5: cost $C_{k,1}$ 의 평균

위의 기준은 몇번의 실험과 수정을 반복할 결과이다. 이 기준을 적절하게 정하는 것은 더 연구되어야 할 문제이다.

그림 7은 카메라에서 얻은 다각형 영상에 대한 실

丑 1. cost C_{k1}
Table 1. cost C_{k1} .

1	1		2		3		5		7		8			
	S	E	S	E	S	E	S	E	S	E	S	E		
1	S	X		14	14	63	69	41	47	34		8		
		Y	∞	∞	57	87	23	23	42	42	42	∞	3	∞
		Z			30	30	30	30	30	30	30		0	
		T			101	131	116	92	113	119	76		11	
1	E	X		14	14	52	61	33	38		32		9	
		Y	∞	∞	42	89	23	23	42	42	∞	42	∞	3
		Z			30	30	30	30	30	30		0		0
		T			86	123	105	114	105	190		74		12
2	S	X	44	29			72	80	13	22	11	12	38	24
		Y	14	14	∞	∞	10	10	29	29	29	29	12	12
		Z	30	30			30	30	30	30	30	30	30	30
		T	88	73			112	120	72	81	70	71	80	66
2	E	X	79	77			22	0	10	9	17	10	40	32
		Y	14	14	∞	∞	10	14	29	29	29	29	12	12
		Z	30	30			30	30	30	30	30	30	30	30
		T	123	121			62	44	69	68	76	69	82	74
3	S	X	41	31	64	31			10	5	22	10	56	45
		Y	23	23	10	10	∞	∞	19	19	19	19	21	21
		Z	30	30	30	30			30	30	30	30	30	30
		T	94	84	104	71			59	54	71	59	107	96
3	E	X	47	40	72	8			24	26	34	23	62	53
		Y	23	23	10	10	∞	∞	19	19	19	19	21	21
		Z	30	30	30	30			30	30	30	30	30	30
		T	100	93	112	48			73	75	83	72	113	104
5	S	X	4	5	11	35	26	41			84	0	69	73
		Y	42	42	29	29	19	19	∞	∞	0	0	40	40
		Z	30	30	30	30	30	30			30	0	30	30
		T	76	77	70	84	75	90			114	0	119	143
5	E	X	9	1	12	35	22	43			65	36	81	39
		Y	42	42	29	29	19	19	∞	∞	0	0	40	40
		Z	30	30	30	30	30	30			30	30	30	30
		T	81	73	71	94	71	92			95	66	151	159
7	S	X	3		16	42	39	51	85	65			65	68
		Y	42	∞	29	29	19	19	0	0	∞	∞	40	40
		Z	0		30	30	30	30	30	30			90	90
		T	45		75	101	88	100	115	95			195	198
7	E	X		5	12	35	26	40	0	36			66	68
		Y	∞	42	29	29	19	19	0	0	∞	∞	40	40
		Z		0	30	30	30	30	0	30			90	90
		T		47	71	94	75	89	0	66			196	198
8	S	X	6		27	30	35	43	74	62	78	77		
		Y	3	∞	12	1	21	21	40	40	40	40	∞	∞
		Z	0		30	30	30	30	30	30	90	90		
		T	9		69	72	88	94	144	132	238	237		
8	E	X		7	14	21	26	34	70	54	75	75		
		Y	∞	3	12	12	21	21	40	40	40	40	∞	∞
		Z		0	30	30	30	30	30	30	90	90		
		T		10	56	63	77	85	140	124	205	205		

X: 이동벡터와 두 교차점을 연결한 직선이 이루는 각(degree)
 Y: 두 교차점이 속한 영역의 이동벡터의 차이 (angle difference + 10* magnitude difference)
 Z: 두 교차점이 속한 영역의 type 사이의 차이

T: cost C_{k1}
 S: 현재경계의 경계 부호의 출발점인 교차점
 E: 과거경계의 경계 부호의 끝나는 점인 교차점

표 2. cost C_{ki}
Table 2. cost C_{ki} .

k \ 1	1		2		3		5		7		8		
	S	E	S	E	S	E	S	E	S	E	S	E	
1	S	∞	∞	186	154	210	192	189	200	121	∞	20	∞
	E	∞	∞	159	244	189	198	182	263	∞	121	∞	22
2	S	186	159	∞	∞	216	332	142	152	145	142	149	122
	E	154	244	∞	∞	133	88	153	172	177	163	154	137
3	S	210	189	216	133	∞	∞	134	125	159	134	195	173
	E	192	198	332	88	∞	∞	163	167	183	161	207	189
5	S	189	182	142	153	134	163	∞	∞	229	0	283	283
	E	200	263	152	172	125	167	∞	∞	190	132	282	283
7	S	121	∞	145	177	159	183	229	190	∞	∞	433	403
	E	∞	121	142	163	134	161	0	132	∞	∞	433	403
8	S	20	∞	149	154	195	207	283	283	433	433	∞	∞
	E	∞	22	122	137	173	189	283	283	403	403	∞	∞

$$C_{k,i} \quad C_{k,i} = C_{i,k} = C_{k,i} + C_{i,k}$$

험 결과이다. 삼각형과 오각형 영상은 두 DP 가 서로 연결되어 있고 따라서 두 영역의 이동벡터는 같다. 그림 7 (g) 는 물체의 이동으로 인하여 사라진 경계를 각 영역의 교차점들 사이의 cost 를 구하여 찾아낸 영상이다. 그림 7 (h) 는 추출된 이동물체이다.

그림 8 은 배경과 occlusion 이 일어난 영상의 실험 결과이다. 이 경우 기존의 region growing 방법으로 이동물체를 정확히 추출할 수 없으나, 본 논문의 알고리즘은 운동 영역에서의 occlusion 만 없으면 이동물체가 추출된다. 그림 8 (f) 의 추출한 이동벡터와 실제의 이동벡터와의 차이는 차영상을 임계치화 할 때 임계치에 의한 오차라 생각된다. 이는 임계치를 적절히 조절함으로써 줄어든 것이다. 그림 8 (h) 의 움직인 물체를 추출한 영상과 실제 영상의 경계와의 차이는 교차점과 교차점 사이를 직선으로 근사화한 사실과 DP 에서 작은 영역을 제거할 때 생긴 결과이다.

그림 9 는 균일한 밝기를 갖지 않은 영상에 대한 처리 결과이다. 위에서 두번째 DP 영역이 자동차 지붕 부분의 반사로 인한 밝기 값의 변화로 인하여 생긴 영역이다. 이 영역은 현재경계로만 이루어져 있어서 처리하지 않았다.

실험은 Data General MV-10000 computer 에서 행해졌고 사용한 언어는 C 이다. DP 를 구하기까지는 실시간으로 구현할 수 있으므로 그 이후의 과정에 대해서 프로그램을 최적화하지 않은 상태에서 주요

알고리즘의 수행시간은 다음과 같다.

사용 영상

Image size : 128 * 128

Gray level : 256

DP 영역의 수 : 8

처리된 DP 영역의 수 : 6

DP 영역의 총 경계 화소의 수 : 252

처리시간

이동벡터 벡터 추정 : 0.04 second

Cost 계산 : 0.01 second

이동물체 추출 : 0.02 second

IV. 결 론

DP 를 이용하여 움직이는 물체의 이동벡터를 추정하고 이동물체를 추출하는 새로운 방법을 제안하였다. 실험 결과에서 알 수 있듯이 DP 가 정확히 추출되기만 하면 거의 정확하게 이동벡터를 추정하고 이동물체를 추출할 수 있었다. 또 배경이 복잡하거나 물체의 움직임이 크거나 작을 때 혹은 회전운동일 때 도 잘 동작하며 부분적인 occlusion 문제를 해결할 수 있었다.

기존의 방법은 DP 를 해석한 후 다시 밝기영상을 처리하는데 반하여 제안한 방법은 DP 로부터 직접 이동벡터와 이동물체를 추출함으로써 실시간에 가까운 처리시간을 얻었다. 제안한 방법은 자동 추적장

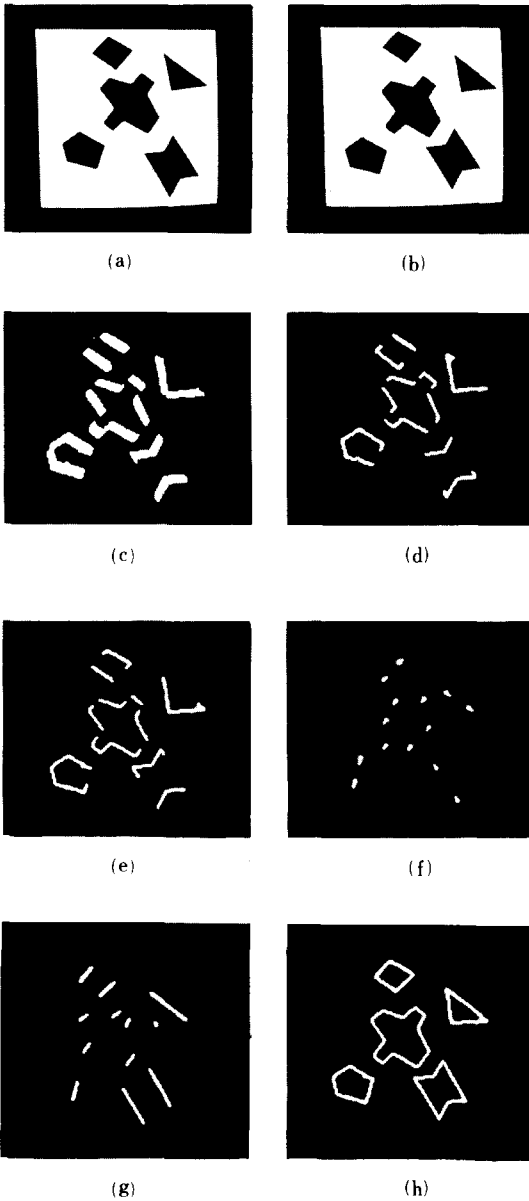


그림 7. 다각형 영상에 대한 실험결과

- (a) 과거영상 (b) 현재 영상
 (c) DP (d) 과거 경계
 (e) 현재 경계 (f) 이동벡터
 (g) 교차점을 연결한 영상
 (h) 이동물체를 추출한 영상
- Fig. 7. Simulation results of polygonal image.
 (a) Previous frame. (b) Current frame.
 (c) DP. (d) Previous boundary.
 (e) Current boundary.
 (f) Displacement vector.
 (g) Image connecting crossing point.
 (h) Image extracting moving object.

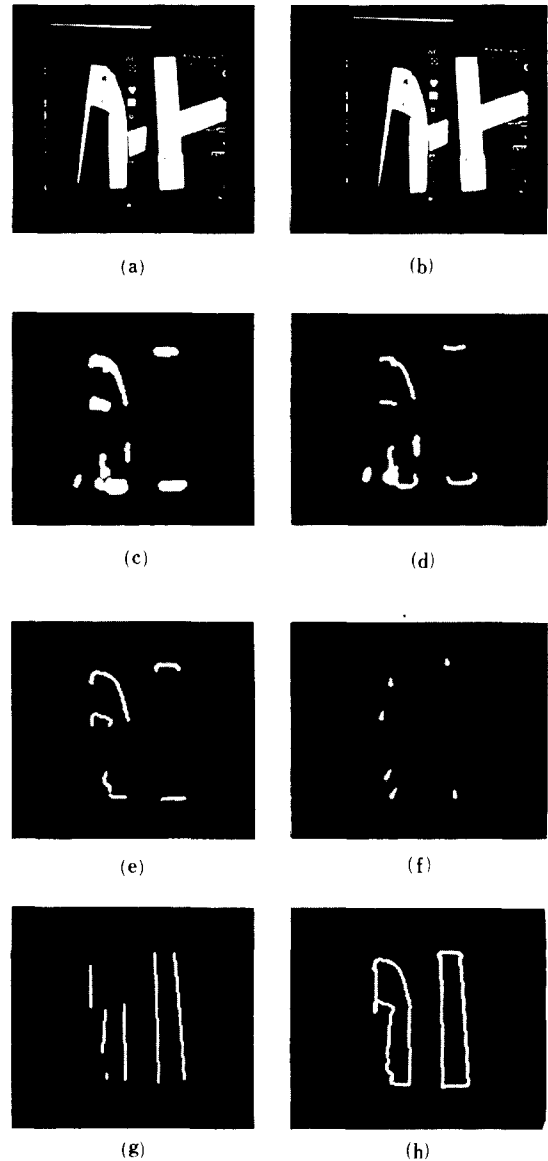


그림 8. 이동물체가 배경과 occlusion된 영상에 대한 실험결과

- (a) 과거 영상 (b) 현재 영상
 (c) DP (d) 과거 경계
 (e) 현재 경계 (f) 이동벡터
 (g) 교차점을 연결한 영상
 (h) 이동물체를 추출한 영상

- Fig. 8. Simulation results of occlusion image between moving objects and background.
 (a) Previous frame. (b) Current frame.
 (c) DP. (d) Previous boundary.
 (e) Current boundary.
 (f) Displacement vector.
 (g) Image connecting crossing point.
 (h) Image extracting moving object.

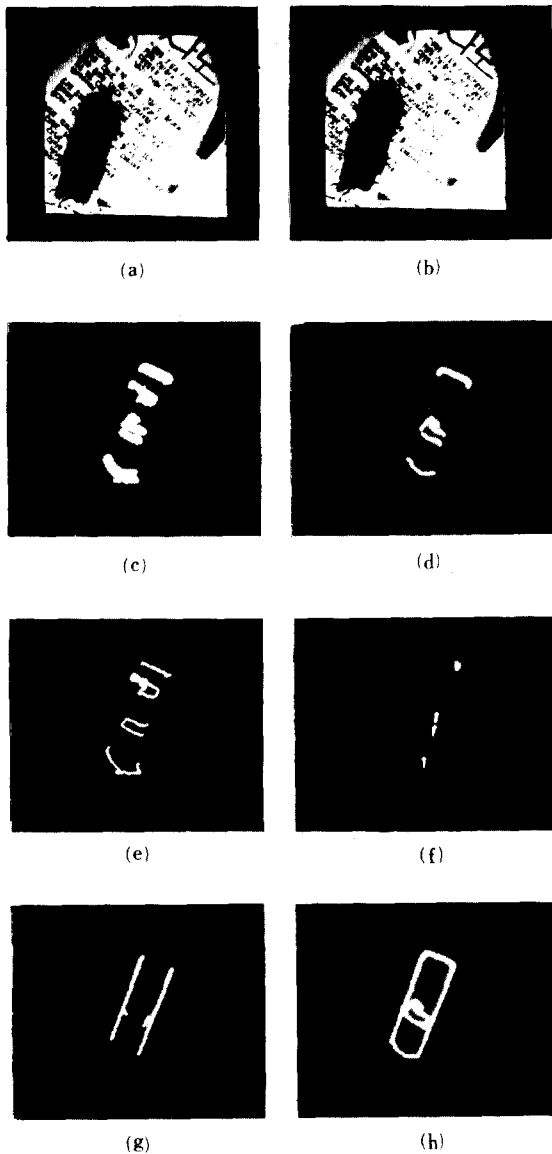


그림 9. 밝기가 균일하지 않은 이동물체에 대한 실험결과

- | | |
|------------------|-----------|
| (a) 과거 영상 | (b) 현재 영상 |
| (c) DP | (d) 과거 경계 |
| (e) 현재 경계 | (f) 이동벡터 |
| (g) 교차점을 연결한 영상 | |
| (h) 이동물체를 추출한 영상 | |

Fig. 9. Simulation results of nonhomogeneous moving object.

- | | |
|--------------------------------------|------------------------|
| (a) Previous frame. | (b) Current frame. |
| (c) DP. | (d) Previous boundary. |
| (e) Current boundary. | |
| (f) Displacement vector. | |
| (g) Image connecting crossing point. | |
| (h) Image extracting moving object. | |

치(automatic target tracking), 원격 감시 장치(remote sensing), 공장 자동화(industrial automation), 교통 감시 장치(traffic monitoring) 등에 응용할 수 있다.

앞으로 DP를 정확히 추출하기 위한 방법과 사용된 임계치들을 적절하게 정하는 방법 그리고 이동물체 추출 알고리즘에 사용된 cost를 최적화하는 것이 필요하다.

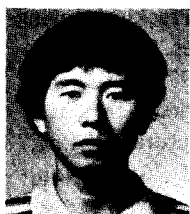
參 考 文 獻

- [1] W.N. Martin and J.K. Aggarwal, "Dynamic Scene Analysis: A Survey," *Computer Graphics and Image Processing*, vol. 7, no. 3, pp. 356-374, June 1978.
- [2] J.K. Aggarwal and W.N. Martin, "Dynamic Scene Analysis," NATO ASI Series, vol. F₂, *Image Sequence Processing and Dynamic Scene Analysis*, Springer-Verlag, 1983.
- [3] J.O. Lim and J.A. Murphy, "Measuring the Speed of Moving Object from television Signals," *IEEE Trans. on Comm.*, vol. 24, no. 4, April 1975.
- [4] C. Cafforio and F. Rocca, "Tracking Moving Objects in Television Images," *Signal Processing*, vol. 1, no. 2, April 1979.
- [5] H.G. Mussman and P. Pirsh and H.G. Gallert, "Advances in Picture Coding," *Proceedings of the IEEE*, vol. 73, no. 4, pp. 523-548, April 1985.
- [6] C.L. Fennema and W.B. Thompson, "Velocity Determination in Scenes Containing Several Moving Objects," *Computer Graphics and Image Processing*, 9, pp. 301-315, 1979.
- [7] M.F. Costabile and C. Gurerra and G. G. Pieroni, "Matching Shapes: A Case Study in Time-Varying Images," *Computer Graphics and Image Processing* 29, pp. 296-310, 1985.
- [8] I.L. Ayala et al., "Moving Target Tracking Using Symbolic Registration," *IEEE transactions on Pat. Anal. Machine Intel.*, PAMI-4, pp. 515-520, 1982.
- [9] R. Jain and W. N. Martine and J.K. Aggarwal, "Segmentation through the Detection of Changes due to Motion," *Computer Graphics and Image Processing* 11, pp. 13-14, 1979.

[10] S. Yalamanchili and W.N. Martin and J.K. Aggarwal, "Extraction of Moving Object Descriptions Via Differencing," *Computer Graphics and Image Processing*, vol. 18, pp. 188-201, 1982.

[11] R. Jain, "Extraction of Motion Information from Peripheral Processes," *IEEE Trans. Pat. Anal. Machine Intel.*, PAMI-3, pp. 489-503, 1981. *

— 著 者 紹 介 —



張 淳 華 (正會員)

1963年 12月 8日生. 1985年 2月 연세대학교 전자공학과 졸업(공학사). 1987年 2月 한국과학기술원 전기 및 전자공학과 석사학위 취득. 현재 한국과학기술원 전기 및 전자공학과 박사과정 재학중. 주

관심분야는 영상처리 및 Computer Vision 등임.



金 聖 大 (正會員)

1953年 12月 26日生. 1977年 2月 서울대학교 전자공학과 공학사. 1979年 2月 한국과학기술원 전기 및 전자공학과 석사학위 취득. 1983年 12月 프랑스 E.N.S.E.E.I.H.

T. 졸업 박사학위 취득. 1984年 1月~6月 프랑스 C.R.E.A.T.I. 연구원. 1984年 9月~현재 한국과학기술원 전기 및 전자공학과 조교수. 주관심분야는 영상처리 및 Computer Vision 등임.



金 鍾 大 (正會員)

1959年 12月 17日生. 1982年 2月 서울대학교 전자공학과 공학사. 1984年 2月 한국과학기술원 전기 및 전자공학과 석사학위 취득. 현재 한국과학기술원 전기 및 전자공학과 박사과정 재학중. 주관심

분야는 영상처리 및 Computer Vision 등임.



金 在 均 (正會員)

1938年 9月 17日生. 1962年 2月 한국 항공대학 응용전자과 공학사. 1967年 2月 서울대학교 대학원 전자공학과 석사학위 취득. 1971年 8月 미국 남가주 대학교 대학원 공학박사학위 취득. 1972年 4月

~1973年 3月 미국 NASA, GSFC 연구원. 현재 한국과학기술원 전기 및 전자공학과 교수. 주관심분야는 영상처리, 정보이론 및 Channel Coding 등임.