

# 독립 객체의 이벤트 인식을 위한 규칙 기반 및 확률적 접근 방법

고재형<sup>0\*</sup> 최창규\* 조정현\*\* 김승호\*

\*경북대학교 컴퓨터공학과, \*\*영남이공대학 컴퓨터정보기술계열

jhko<sup>0</sup>@mmlab.knu.ac.kr, cgchoi@mmlab.knu.ac.kr, petercho@ync.ac.kr, shkim@knu.ac.kr

## Rule-based and Probabilistic Access Method for Event Recognition of Independent Objects

Jaehyung Ko<sup>0\*</sup> Changgyu Choi\* Jeonghyun Cho\*\* Sungho Kim\*

\*Dept. of Computer Engineering, Kyungpook National University

\*\*Division of Computer Information Technology, Yeungnam College of Science & Technology

### 요 약

본 논문에서는 실내 또는 실외 환경에서의 비전 감시 시스템에 응용할 객체의 이벤트 인식 방법을 정의한다. 객체의 이벤트를 실시간으로 적용하기 위해서는 연산 시간이 적어야하며 이벤트 인식 정확도를 또한 높여야 한다. 기존의 규칙 또는 확률 기반 이벤트 인식 방법은 객체의 움직임 이벤트에 대해 한정적이거나 객체의 특징 벡터 추출이 규칙에 어긋나는 경우 오류가 발생하여 이벤트 인식의 정확도가 떨어지는 문제점이 있다. 본 논문에서는 시간의 흐름에 따른 독립 객체의 특징 벡터와 확률 정보를 통해 정해진 규칙에 가장 가까운 상황에 대해 계속적으로 이루어지는 이벤트 인식 방법을 제안한다.

### 1. 서 론

일상생활에서 사용하는 많은 종류의 정보들 중에서 동영상과 같은 시각 정보가 차지하고 있는 비율은 매우 크며, 가장 효율적인 정보 전달력을 가지고 있다. 최근에는 동영상에서 상대적으로 많은 움직임을 보이는 객체를 추출하거나 객체의 특징 분석을 통해 컴퓨터 비전을 자동화 시스템으로 확장하는 동영상 처리 기술이 연구되고 있다[1,2]. 객체 인식은 감시 시스템에서 객체의 특징을 추출하여 행동을 인식 및 분석하기 위해 이용된다. 그리고 더 나아가 객체의 의미 해석 또는 영상 내에서의 이벤트 탐지 등에 활용할 수 있다.

본 논문에서는 독립 객체의 단위를 사람으로 설정하고 독립 객체의 이벤트 인식을 위한 방법을 제안한다. 이벤트 인식을 위한 전처리 과정인 배경 모델링, 객체 인식, 그리고 객체 추적에 대한 연구는 오래전부터 진행이 되었고 현재 많은 부분에서 상당한 결과를 보이고 있다[3,4]. 독립 객체에 대한 이벤트 인식에 대해 A. Amer와 R. Cucchiara가 제안한 방법은 객체의 움직임에 대해 한정적으로 이벤트를 적용하거나 바운딩 박스의 크기 등 특징 벡터 추출이 규칙에 어긋나는 경우에는 이벤트 인식 과정의 오류로 작용되어 이벤트 인식의 정확도가 떨어지는 문제점이 있다[1,5].

본 논문에서는 실내 또는 실외 환경에서 발생하는 객체의 일반적인 이벤트를 정의하고 특징 벡터와 정의된 이벤트 규칙을 통해 이벤트를 인식할 수 있는 방법을 제안한다. 입력 영상에서 인식된 객체로부터 속도, 방향,

색상, 바운딩 박스 크기, 중심점 등의 특징 벡터를 구하고 시간에 따른 특징 벡터값의 변화를 분석하여, 정의된 이벤트 규칙에 따라 프레임별 단위로 해당 객체의 이벤트를 인식한다.

본 논문의 구성은 2장에서는 이벤트를 인식하는 방법에 대한 이해와 관련 연구들을 설명하고, 3장에서는 본 논문에서 제안하는 객체의 이벤트 정의 및 인식 방법을 제안한다. 4장에서는 본 논문에서 제안한 이벤트 인식에 대해 실험한 결과를 기술하고, 마지막으로 5장에서는 결론을 맺고 향후 연구에 대해 기술한다.

### 2. 이벤트 인식 방법의 이해 및 관련 연구

#### 2.1 규칙 기반의 독립 객체 이벤트 인식 및 관련 연구

규칙 기반의 이벤트 인식은 독립적으로 나타난 객체를 추적하고 동일 객체로부터 특징 벡터를 추출하여 매 프레임 또는 주기적으로 시간의 흐름에 따라 객체의 행동을 규칙에 따라 인식하는 것이다. 이 방법은 객체의 특징 벡터를 기반으로 미리 이벤트에 대한 규칙을 정의하고 시간에 따른 객체의 특징 벡터가 어느 규칙에 만족하는가를 판단하여 해당 객체의 이벤트를 인식하는 것이다. 즉 규칙 기반의 이벤트 인식 방법은 현재 환경에서 객체가 수행할 수 있는 행동에 대해 여러 가지의 규칙을 정의하고 매 프레임 객체의 특징 벡터를 추출하여 각 규칙에 적용하게 된다. 기존에 연구된 규칙 기반의 독립 객체 이벤트 인식 방법은 A. Amer에 의해 제안된 방법과 I. Haritaoglu에 의해 제안된 W<sup>4</sup>의 통계적 방법이 있

다.

A. Amer가 제안한 방법은 실내 또는 실외에서 발생할 수 있는 독립 객체의 행동 분류를 4개의 도메인으로 나누고, 각각의 도메인을 다시 세부적인 이벤트로 정의하였다[1].  $W^4$ 는 미리 정의된 템플릿의 수직/수평 히스토그램과 현재 인식된 객체의 수직/수평 히스토그램간의 유사도를 구하여 가장 높은 값을 현재 객체의 이벤트로 인식하게 된다[6]. 두 방법 모두 확률을 통한 이벤트 인식과 다르게 특별한 학습 과정이 필요 없고, 특징 벡터를 바로 추출하기 때문에 연산 시간이 적은 장점을 가진다. 그러나 한정된 규칙을 기반으로 하는 특징 벡터이기 때문에 일반적인 비전 감시 시스템에 완전히 적용할 수 없고 제한된 환경에서만 가능하다는 문제점이 있다.

### 2.2 확률을 통한 이벤트 인식 및 관련 연구

확률을 통한 접근 방식은 이전의 다양한 영상으로부터 특징 벡터를 학습하고 이를 통해 획득한 사전 확률 값을 적용하여, 새롭게 입력되는 영상에서 발생하는 이벤트를 저장된 데이터베이스와 비교하여 가장 큰 사후 확률 값을 가지는 이벤트를 해당 객체의 이벤트로 인식하는 것이다. 따라서 확률을 통한 이벤트 인식은 어떤 특징 벡터를 어떻게 학습시키는가가 중요한 요소가 된다. 본 논문에서는 확률을 통한 이벤트 인식을 위하여 베이지안 네트워크를 사용한다.

베이지안 네트워크는 베이즈 이론을 바탕으로 노드와 방향성을 가지는 아크(arc)로 구성된 방향성 그래프이다 [7]. 각 노드는 발생할 수 있는 행위에 대한 조건부 확률 테이블(CPT, Conditional Probability Table)을 가지고, 아크는 노드와 노드간의 확률 의존성을 나타낸다.

베이지안 네트워크를 구성하는 방법은 현실 문제에서 발생할 수 있는 이벤트를 노드로 정의하고 해당 이벤트가 발생하기 위한 조건들 또한 노드와 노드간의 상관관계에 따라 아크를 설정한다. 이 모델을 바탕으로 학습을 통해 각 노드의 조건부 확률 값을 구하고 사후에 발생하는 증거를 이용하여 사후 확률 값을 추정하게 된다. 각 조건부 확률 테이블은 학습을 통해 얻어진 확률 값으로 구성된다.

각 노드의 조건부 확률 테이블 확률 계산은 결합 확률 분포(joint probability distribution)로 결정되는데, 부모 노드와 자식 노드간의 체인 규칙이 존재한다. 확률을 표현하는 방법은 식 (1)과 같이 나타내고, 각  $x$ 는 노드  $X$ 가 발생할 수 있는 확률 값이다( $X_1 = x_1$ ).

$$\begin{aligned}
 P(x_1, x_2, \dots, x_n) \\
 &= P(x_1) \times P(x_2 | x_1) \times \dots \times P(x_n | x_1, \dots, x_{n-1}) \\
 &= \prod_i P(x_i | x_1, \dots, x_{i-1})
 \end{aligned}
 \tag{1}$$

베이지안 추정은 자식 노드에 대한 현재 증거를 바탕으로 부모 노드의 사후 확률이 결정되는데, 일반적으로 우도 함수(likelihood function)를 통해 수행된다[2].

확률을 통한 이벤트 인식은 독립된 객체의 특징 벡터를 이용하여 인식하는 방법보다 정확성이 뛰어나다. 그 이유는 확률적인 접근을 하기 때문에 예상할 수 없는 상

황에 대한 대처가 용이하기 때문이다. 또한 확률적으로 여러 가지 상황을 정의했기에 현재 발생한 이벤트가 이미 정의된 이벤트 중에 특정한 이벤트에 가깝다는 것으로 결론을 내므로 신빙성이 높아진다.

R. V. Babu는 입력 영상에서 연속적으로 들어오는 프레임 내의 객체로부터 MHI(motion history image), MFH(motion flow history)를 정의한 후 특징 벡터를 추출한다. 이 특징 벡터들을 베이지안 네트워크로 학습한 후 WALK, RUN, JUMP 등의 이벤트를 해석하였다[8]. G. Guler는 MEET, Approach, MOVE TOGETHER, Move away 등의 이벤트를 정의하고 객체의 크기 변화, 객체간의 거리 변화, 객체의 이동 거리를 관측치로 하여 해당 이벤트를 인식하였다. 제안된 방법은 각 이벤트간의 상태 정리가 발생하고 각 관측치로부터 전이 확률을 정의하였다[9].

R. V. Babu와 G. Guler의 확률을 통한 이벤트 인식 방법은 독립된 객체의 특징 벡터를 이용하여 인식하는 방법보다 정확성이 뛰어나다는 장점이 있다. 그러나 확률 모델들은 어떤 데이터로 어떻게 학습하는가에 따라 정확도의 차이가 보이고, 학습에 대한 시간이 많이 소요되는 문제점이 있다. 확률 모델의 노드 또는 상태의 개수에 따라 현재 인식된 이벤트를 해석하는 과정의 연산 시간이 많기 때문에 규칙 기반 이벤트 인식 방법에 비해 실시간에 적용하기에는 문제점이 있다.

### 3. 독립 객체의 이벤트 인식

객체의 특징 벡터는 독립 객체의 이벤트를 정의하는데 가장 중요한 요소이다. 각 특징 벡터의 특성에 따라 이벤트의 규칙이 정의되고, 이를 통해 시간의 흐름에 따른 객체의 이벤트가 인식된다. 본 논문에서는 표 1에서 정의한 특징 벡터를 객체로부터 추출하고 추출된 특징 벡터에 확률 정보를 더하여 그림 1에서 정의한 이벤트 규칙에 확률적으로 가장 가까운 이벤트를 현재의 이벤트로 인식하는 방법을 제안한다.

표 1. 객체로부터 추출되는 특징 벡터

특징 벡터	설명
바운딩 박스 (bounding box)	인식된 객체의 둘레 정보
중심점(center point)	객체의 중심점 좌표값
속도(velocity)	객체의 평균 이동 속도
방향(direction)	객체의 이동 방향
색상(color)	객체의 RGB 대표 색상 정보
질감 템플릿 영상 (texture template)	겹침 현상을 해결하기 위한 질감 영상
분할 움직임 방향 벡터 (segmented motion orientation vector)	동일한 방향의 움직임을 가지는 벡터 성분

독립 객체의 이벤트 인식은 정의된 규칙과 시간의 흐름에 따른 객체의 특징 벡터에 따라 진행된다. 주-이벤트는 독립 객체의 가장 기본이 되는 움직임으로 객체가 인식되는 과정에서부터 객체가 영상 내에서 사라지는 이벤트들로 구성된다. 영상 내에 객체가 존재하기 시작하면서 *들어오다(ENTER)* 이벤트가 인식되고, 객체가 존재하는 동안 *움직이다(MOVE)* 이벤트가 발생한다. 객체가 영상 내에서 사라지게 되면 *나가다(EXIT)* 이벤트가 발생한다. 주-이벤트 사이에서 발생하는 움직임-이벤트, 상호-이벤트는 해당 객체의 특징 벡터를  $t$  주기 프레임 또는 매 프레임마다 추출하고 객체의 현재 이벤트와 정의된 규칙을 통해 상태 전이가 된다. 인식된 해당 객체의 이벤트는 데이터베이스에 계속 저장되고 *EXIT* 이벤트가 발생하는 시점에서 더 이상의 객체 추적을 수행하지 않는다.

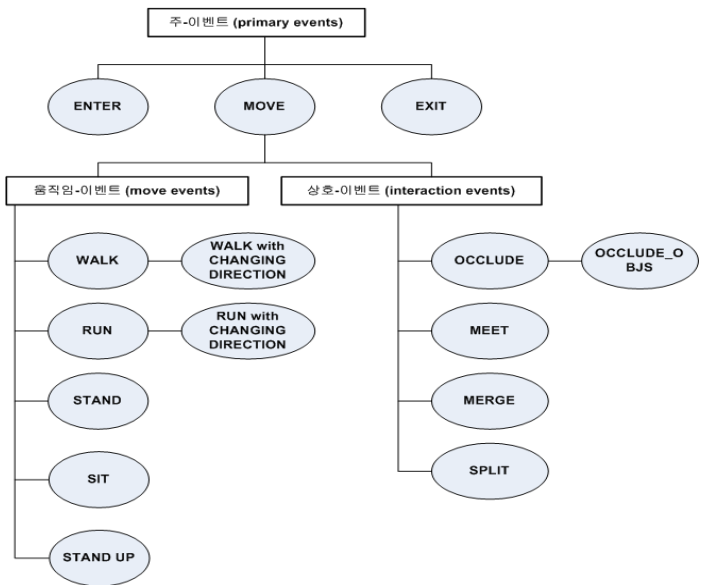


그림 1. 규칙 기반에 의해 정의된 이벤트

움직임-이벤트는 객체의 속도 특징 벡터와 함께 분할 움직임 방향 벡터를 함께 사용하여 *걷다(WALK)* 또는 *뛰다(RUN)*, *서있다(STAND)*, *앉다(SIT)*, *서다(STAND UP)* 이벤트로 구분한다. 각 이벤트는 객체의 속도가 차이나지만 *WALK*와 *RUN* 이벤트를 구분하기는 모호한 경우가 있다. 따라서 본 논문에서는 움직임-이벤트의 보다 정확한 구분을 위하여 분할 움직임 방향 벡터를 베이지안 네트워크를 통해 학습하여 현재 분할 움직임 방향 벡터가 어떤 움직임이 있는지를 사후 확률로 추정하는 방법을 사용한다.

분할 움직임 방향 벡터를 구하기 위해서는 먼저 영상에 대해서 움직임 분할 과정을 수행하여야 한다. 움직임 기울기의 정보를 모든 픽셀에 적용 하였을 때 연산에 대한 시간이 많이 소요되는 문제점을 유사한 방향으로의 움직임을 가지는 신체 부위를 하나의 움직임으로 표현하게 함으로써 움직임 기울기의 특징 벡터 정보를 줄여 보다 빠른 연산을 가능하게 하는 과정을 움직임 분할이라고 한다[10]. 움직임 분할은 가장 최근 실루엣의 경계

부분을 따라가면서 임계치 시간 내에 움직임이 있었던 부분을 하나의 값으로 레이블을 하여 동일한 레이블 값을 가진 영역을 하나의 분할 영역으로 마스크가 되는 것이다.

분할 움직임 방향 벡터는 움직임 분할 영역이 만들어지면 각 영역의 움직임 방향 특징 벡터를 식 (2)를 이용하여 구하게 되고 각 영역에 대해 하나의 값으로 표현된다. 독립 객체에는 한 개 이상의 분할 움직임 방향 벡터가 존재하며 저장되는 정보는 기준선을 기준으로 시계 방향으로의 각도( $\theta$ )로 표현되고 식 (3)으로 나타낸다.

$A_i$ 는  $i$ 객체의 분할 움직임 방향 벡터를 표현하고,  $A_i^j$ 는  $i$ 번째 객체의  $j$ 번째 분할 움직임 방향값이다.

$$A_i(x,y) = \arctan\left(\frac{S_y(x,y)}{S_x(x,y)}\right) \quad (2)$$

$$SMOV_i = smov_1, smov_2, \dots, smov_j, \dots, smov_m \quad (3)$$

$$smov_j = 360 - A_i^j$$

그림 2에서 설계된 베이지안 네트워크의 구성은 움직임이 있는 *WALK/RUN* 이벤트와 움직임이 거의 없는 *STAND* 이벤트로 나눈다.

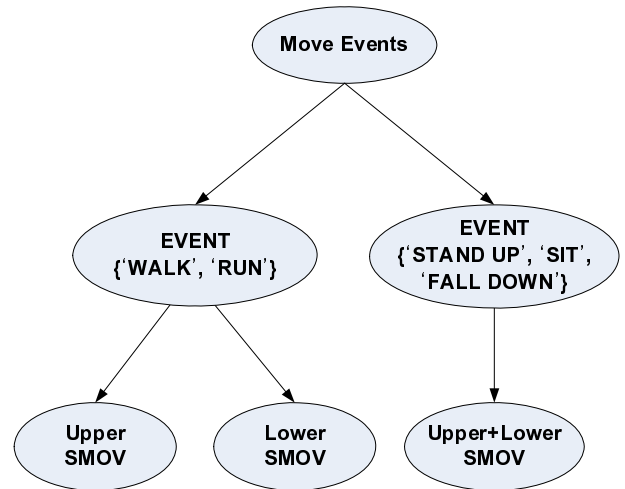


그림 2. 움직임-이벤트를 구분하기 위한 베이지안 모델

*WALK/RUN* 노드는 객체의 상위와 하위 부분의 분할 움직임 방향 벡터를 따로 학습하고, *STAND* 노드는 객체의 전체적인 움직임에 영향을 받기 때문에 신체의 구분 없이 분할 움직임 방향 벡터가 학습된다. 베이지안 모델에서 자식 노드는 움직임이 존재하는 경우 분할 움직임 방향 벡터의 학습 결과를 통해 *WALK*와 *RUN* 이벤트의 사전 확률 값이 만들어지고, 움직임이 없는 *STAND*는 *SIT*, *STAND UP*, *넘어지다(FALL DOWN)*로 만들어진다.

분할 움직임 방향 벡터는 현재 분할된 움직임 벡터의 방향 성분(각도)을 가지고 있고, 이 정보를 통해 각 이벤트에 대한 사전 확률을 구한다. 사후 확률의 결정은 객

체의 현재 분할 움직임 방향 벡터를 증거로 하여 해당 객체의 이벤트를 추정하게 된다. 하나의 객체에는 한 개 이상의 분할 움직임 방향 벡터가 추출된다. 만약 독립 객체에서  $m$ 개의 분할 움직임 방향 벡터를 추출하였다면 각 이벤트의 사후 확률은  $m$ 개의 평균으로 정의하고, 사후 확률 값이 가장 최고가 되는 이벤트가 현재 객체의 이벤트가 된다. 사후 확률을  $Bel(E)$ 라고 가정하고 현재의 증거를 분할 움직임 방향 벡터라 하고 찾고자 하는 이벤트의 가설을  $E$ 라고 한다면 현재 독립 객체의  $i$ 번째 분할 움직임 방향 벡터에 대한  $j$ 이벤트의 사후 확률을 구하는 식은 식 (4)와 같이 정의한다.

$$Bel(E_j) = \frac{P(E_j | SMOV_i)}{P(SMOV_i)} = \frac{P(SMOV_i | E_j)P(E_j)}{P(SMOV_i)} \quad (4)$$

식 (5)는  $m$ 개의 분할 움직임 벡터에 대한  $j$ 이벤트의 사후 확률 값을 구하는 식을 정의한 것이다.

$$Bel(E_j) = \frac{\sum_{k=1}^m P(E_j | SMOV_k)}{m} = \frac{\sum_{k=1}^m \frac{P(SMOV_k | E_j)P(E_j)}{P(SMOV_k)}}{m} \quad (5)$$

따라서  $n$ 개에 대한 모든 이벤트  $E_1, E_2, \dots, E_n$ 은 식 (6)을 통해 사후 확률 값이 가장 높은 이벤트를 현재의 이벤트로 정의한다.

$$Bel(EVENT) = \max_{\forall (E_1, E_2, \dots, E_j, \dots, E_n)} Bel(E_j) \quad (1 \leq j \leq n) \quad (6)$$

현재 객체의 움직임-이벤트는 객체의 움직임이 존재하는 이벤트인 WALK/RUN의 경우와 움직임이 거의 없는 SIT/STAND UP/FALL DOWN으로 구분되어 결정된다. 즉, WALK/RUN의 경우는 객체의 속도 특징 벡터에 의해 결정된 이벤트 결과와 분할 움직임 방향 특징 벡터의 사후 확률 값의 조합으로 결정되기 때문에 두 특징 벡터에 대한 가중치가 적용된다. 반면, SIT/STAND UP/FALL DOWN 이벤트는 속도 특징 벡터가 존재하지 않으므로 사후 확률 값에 의해 결정된다.

식 (7)은 WALK/RUN 이벤트를 결정하는 식이다. 여기서  $\alpha$ 는 가중치이고,  $EVENT\_VEL()$ 는 현재 객체  $O_i$ 의 속도 특징 벡터에 의해 결정된 이벤트 결과와 베이저안 네트워크를 통해 결정된 이벤트 결과가 동일한 경우 1이 되고, 서로 다른 이벤트 결과가 나타나면 0이 된다. 식 (7)은 현재 이벤트가 WALK/RUN인 경우 속도 특징 벡터와 사후 확률에 의해 현재 객체의 이벤트를 결정하고, 움직임이 없는 이벤트인 STAND UP/SIT/FALL DOWN 이벤트는 사후 확률에 의해 객체의 이벤트가 결정된다.

최종적으로 현재 객체의 움직임-이벤트는 식 (7)의  $RES(O^i)$  결과 값이 움직임-이벤트 결정 임계치  $\beta$  이상일 경우에 식 (7)에 의해 결정된 이벤트 결과가 현재 객

체의 이벤트가 된다. 식 (8)에서  $EVENT(RES(O_i))$ 는 식 (7)에서  $RES(O^i)$ 의 결과값이 가장 높은 때의 이벤트 결과를 가진다.

$$RES(O_i) = \begin{cases} \alpha EVENT\_VEL(O_i^c) + (1-\alpha)Bel(EVENT) & \text{현재 이벤트가 'WALK' 또는 'RUN'인 경우} \\ Bel(EVENT) & \text{그 외의 움직임-이벤트인 경우} \end{cases} \quad (7)$$

$$0 \leq \alpha \leq 1, \quad 0 \leq RES(O_i) \leq 1$$

$$EVENT(O_i^c)_c = \begin{cases} EVENT(O_i^c)_p & RES(O_i) < \beta \\ EVENT(RES(O_i)) & RES(O_i) < \beta \end{cases} \quad (8)$$

$$0 \leq \beta \leq 1$$

영상 내에 존재하는 모든 객체는 독립적으로 각각의 이벤트 전이를 수행하고 있다. 상호-이벤트는 두 개 이상의 독립 객체에서 발생하는 이벤트로, 객체간의 겹침 현상(OCCLUDE), 만나다(MEET), 같이 걷다(MERGE) 등의 객체 간에 상호 영향을 주는 이벤트와 분리되다(SPLIT) 등과 같이 다시 이전의 독립 객체로 돌아가는 이벤트로 구성된다.

상호-이벤트는 각각의 객체가 겹침 현상 이후에 발생된 것이기 때문에 객체의 바운딩 박스의 특징 벡터와 속도, 방향 특징 벡터를 통해 상호-이벤트 또는 움직임-이벤트로 전이가 가능해진다.

OCCLUDE 이벤트가 발생하는 시점부터 객체 추적을 위하여 템플릿 매칭을 수행한다. 템플릿 매칭은 OCCLUDE\_OBJS 이벤트를 가진 겹침 현상으로 인해 새롭게 생성된 객체 내에서 OCCLUDE, MEET, MERGE, SPLIT 이벤트에 해당 되는 객체의 질감 템플릿 영상과의 차이 값을 구하는 것이다.

질감 템플릿 영상은  $t$ 시간 동안 객체의 질감 부분을 겹쳐서 나타낸 것으로  $W^4$ 에서 사용된 이차원 동적 템플릿을 사용한다[6]. 식 (9)는 인식된 객체의 회색조 질감 정보를 이용하여 생성한 질감 템플릿 영상을 의미하고, 모양 정보는 바운딩 박스와 배경 영상간의 차를 통해 배경 부분만을 제외한 영상이 된다.

$$TTI_t^f(x,y) = \frac{I_t(x,y) + TTI_t^{f-1}(x,y)}{t} \quad (1 < t \leq t+1) \quad (9)$$

$I(x,y)$ 는 현재 입력 프레임에서 객체로 분류된 픽셀의 값으로 질감 정보를 나타내고,  $TTI_t^f(x,y)$ 는  $t$ 번째 질감 템플릿 영상을 나타낸다.

OCCLUDE\_OBJS 이벤트를 가진 객체의 바운딩 박스의 시작점을  $f(x,y)$ 라 하고, 해당 객체의 질감 템플릿 영상의 시작점을  $TTI(x,y)$ 라고 할 때 템플릿 매칭은 식 (10)와 같이 정의하였다.

동영상에서 한 프레임 간의 시간 간격이 짧기 때문에 객체의 움직인 거리 또한 크게 증가하지 않는다는 특성과 한 프레임 이후에 해당 객체가 어느 지점으로 이동하는지 알 수 없기 때문에 템플릿 매칭의 시작점인  $(m,n)$

좌표는 하나로 결정할 수 없다. 따라서 템플릿 매칭을 수행할 때 시작 좌표는  $(m \pm 5, n \pm 5)$ 로 하여 차이가 가장 최소가 되는 지점이 동일 객체가 존재하는 것으로 판단한다. 단 템플릿 매칭의 속도를 개선하기 위하여  $(m, n)$  좌표가 *OCCLUDE\_OBJS*의 바운딩 박스 밖에 존재하면 비교를 하지 않는다. 그리고 배경에 해당되는 부분은 0으로 계산한다.

$Template(f, TTI)$

$$= \min_{\forall(m,n)} (f(x+m, y+n) \bullet TTI(i, j))$$

$$= \min_{\forall(m,n)} \left( \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(x+m+i, y+n+j) - TTI(i, j)}{255 \times M \times N} \right) \quad (10)$$

$$-5 \leq m, n \leq 5$$

4. 실험 결과

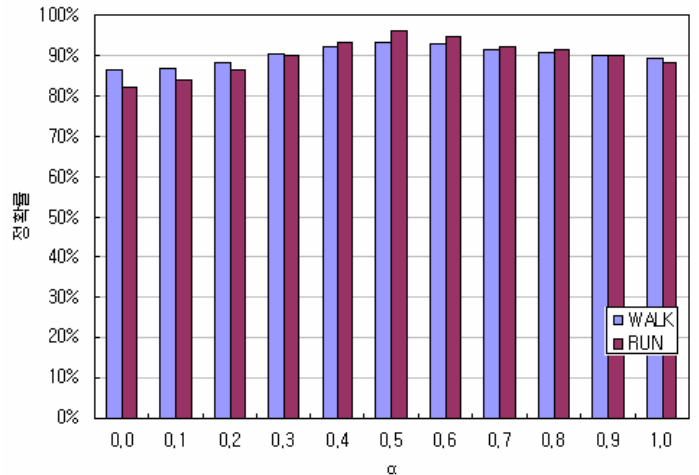
제안한 실내 또는 실외 환경에서 발생하는 기본 움직임 이벤트에 대한 실험을 한다. 이벤트를 발생시키는 독립 객체의 단위는 사람이고 정의된 이벤트는 독립 객체를 바탕으로 두고 있기 때문에 여러 명이 동시에 나타나는 그룹 객체는 존재하지 않는다. 실내 또는 실외의 고정된 카메라를 통해 영상을 입력 받아 실험을 진행한다. 실험에 사용되는 영상은 15개를 사용하고, 각 영상은 최소 600프레임에서 1000프레임으로 구성된다. 영상의 크기는  $360 \times 240$ (pixel)이고, 영상의 속도는  $30 \text{ frame/sec}$ 이다.

실험 영상에 대해 독립 객체의 기본 움직임 이벤트 인식의 정확률을 보여주고, 움직임-이벤트인 *WALK*, *RUN*, *SIT*, *STAND UP* 이 베이지안 네트워크를 통해 분할 움직임 특징 벡터가 어떤 확률 분포를 가지는가를 제시한다. 움직임-이벤트를 인식하기 위한 변수인  $\alpha$ 와  $\beta$ 값을 정의하기 위해, 각각 0.1씩 증가하면서 이벤트 인식의 정확률이 얼마인가를 확인하였다

식 (7)에서 사용된  $\alpha$ 값은 *WALK*와 *RUN* 이벤트에만 적용되는 값이고, 그림 3의 정확률은  $\alpha$ 값을 0.1씩 증가하면서 시스템이 인식한 결과와 사람이 인식한 결과의 비율을 나타낸다.

$\alpha$ 값이 0.4까지는 이벤트 인식 정확률이 서서히 증가하다가 0.5에서 가장 높은 정확률을 보이고 서서히 감소하는 것을 알 수 있다. 이것은 속도 특징 벡터 또는 분할 움직임 방향 벡터만을 사용할 경우 *WALK* 또는 *RUN* 이벤트를 잘못 인식할 수 있다는 것을 의미한다. 따라서, 가장 높은 정확률을 보이고 있는 0.5를  $\alpha$ 값으로 정한다.

그림 4에서 *WALK*와 *RUN* 이벤트는  $\alpha$ 값에 의해 이미 가장 좋은 인식 결과를 가지고 있기 때문에  $\beta$ 값에 영향을 받지 않는다. 그러나 *STAND UP*, *SIT*, *FALL DOWN* 이벤트는 현재 계산된 사후 확률에 따라 이벤트가 인식되기 때문에  $\beta$ 값의 변화에 영향을 받는 것을 알 수 있다. 실험 결과, *STAND UP*, *SIT*, *FALL DOWN* 이벤트의



정확률이  $\beta$ 값이 0.7을 기준으로 서서히 증가하다가 감소  
그림 3.  $\alpha$ 값의 변화에 따른 *WALK*와 *RUN* 이벤트 인식 정확률

하는 것을 볼 수 있다. 그 이유는 객체가 *STAND UP*, *SIT*, *FALL DOWN* 이벤트를 발생하는 초기 시점과 끝나는 시점에서 각 이벤트들이 사후 확률이 비슷하게 나타나기 때문에  $\beta$ 값이 작을 경우 잘못된 이벤트로 인식하기 때문이다. 또한  $\beta$ 값이 0.7보다 큰 경우에는 이벤트 사후 확률이  $\beta$ 보다 작아서 이전 이벤트를 유지하기 때문에 정확률이 낮은 문제점이 있다. 따라서  $\beta$ 값이 0.7 이상일 때 가장 좋은 이벤트 결과를 얻을 수 있다는 것을 실험을 통해 확인할 수 있다.

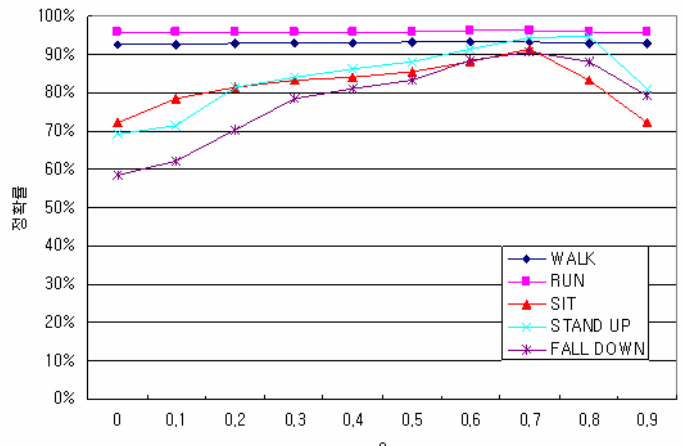


그림 4.  $\beta$ 값의 변화에 따른 움직임-이벤트 인식 정확률 ( $\alpha=0.5$ )

기본 움직임 이벤트가 발생하게 되면, 각각의 이벤트가 전이 가능한 이벤트로 전이 과정을 수행하였고, 정확도는 실제 사람이 인식한 결과와 비교하여 100%를 기본으로 나타내었다. 15개의 실험 영상에 나타난 객체 수는 모두 74개이며 정확도는 각각의 객체에 대해 매 프레임의 이벤트 결과를 바탕으로 실제 사람이 인식한 결과와의 비율을 측정하였다. 그림 5는 기본 움직임 이벤트에 대한 정확률의 결과를 보여주고 있다.

겹침 현상이 발생했을 때 하나의 객체가 다른 객체를

전체적으로 겹쳐져 템플릿 매칭을 수행하여도 동일 객체를 찾지 못해서 해당 객체가 EXIT 이벤트로 전이되고, 겹침 현상이 끝난 후에 새로운 객체로 인식되어 ENTER 이벤트가 발생하기 때문에 주-이벤트와 움직임-이벤트는 90% 이상의 정확률을 나타내지만, 상호-이벤트는 90% 미만의 결과가 나타난다. 따라서 겹침 현상에서 동일 객체를 추적하지 못할 경우 전체적으로 이벤트의 정확도를 떨어뜨리는 원인이 된다.

속도 정보는 움직임-이벤트를 구분하는데 필요한 특징 벡터였고, 베이지안 네트워크의 사전 학습 결과와 사후 결과를 통해 이를 확인하였다.

제안한 독립 객체의 이벤트 인식 방법은 실내 또는 실외 환경에서 발생하는 일반적인 객체의 행동을 의미론적으로 해석이 가능하다. 또한 본 연구 결과는 보다 다양한 이벤트를 정의하는데 확장이 가능하고 그룹 객체의 처리 방법에 대한 추가적인 연구, 그리고 여러 가지 시나리오를 정의하여 비전 감시 시스템 내에서 발생하는 다양한 환경에서 객체의 의미론적 해석이 될 수 있을 것으로 기대한다.

참 고 문 헌

[1] A. Amer, E. Dubois, and A. Mitiche, "Rule-Based Real-time Detection of Context-independent Events in Video Shots," Real-Time Imaging, Vol. 11 No. 2, pp. 244-256, 2005.  
 [2] L. Wang, W. Hu, and T. Tau, "Recent Developments in Human Motion Analysis," Pattern Recognition, Vol. 36, No. 3, pp. 585-601, 2003.  
 [3] T. Yang, S. Z. Li, Q. Pan, and J. Li, "Real-time Multiple Objects Tracking with Occlusion Handling in Dynamic Scenes," Proceeding of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 970-225, 2005.  
 [4] L. Q. Xu, J. L. Landabaso, and B. Lei, "Segmentation and Tracking of Multiple Moving Objects for Intelligent Video Analysis," BT Technology Journal, Vol. 22, No. 3, pp. 140-150, 2004.  
 [5] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani, "Probabilistic Posture Classification for Human-Behavior Analysis," IEEE Transaction on System, Man, and Cybernetics - Part A: System and Humans, Vol. 35, No. 1, pp. 42-54, 2005.  
 [6] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-Time Surveillance of People and Their Activities," IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, pp. 809-830, 2000.  
 [7] K. B. Korb and A. E. Nicholson, Bayesian Artificial Intelligence, A CRC Press Company, New York, 2004.  
 [8] R. V. Babu and K. R. Ramakrishnan, "Recognition of Human Actions Using Motion History Information Extracted from the Compressed Video," Image and Vision Computing, Vol. 22, pp. 597-607, 2004.  
 [9] S. Guler, W. H. Liang, I. A. Pushee, "A Video Event Detection and Mining Framework," Second IEEE Workshop on Event Mining, Vol. 4, No. 4, pp. 42-483  
 [10] G. R. Bradski and J. W. Davis, "Motion Segmentation and Pose Recognition with Motion History Gradients," Machine Vision and Application, Vol. 13, No. 3, pp. 174-184, 2002.

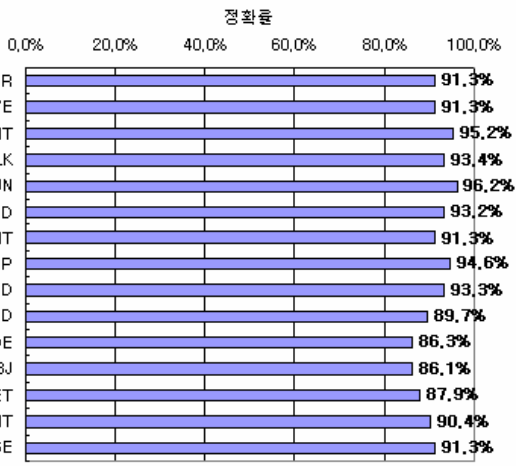


그림 5. 15개 실험 영상에 대한 이벤트 결과 ( $\alpha=0.5, \beta=0.7$ )

상호-이벤트인 SPLIT, MERGE, MEET가 주-이벤트와 움직임-이벤트보다 정확도가 떨어지는 원인은 실험 영상에서 실제 발생된 해당 이벤트의 빈도수가 전체의 20% 정도로 객체 인식 오류, 겹침 현상에 의한 객체 추적의 오류 등이 발생할 경우 움직임-이벤트에 비해 상대적으로 정확도가 떨어지기 때문이다.

5. 결 론

기존에 연구된 독립 객체의 규칙 기반 및 확률적 이벤트 인식 방법은 미리 정의된 이벤트 규칙에 적용하거나 확률 모델을 이용하는 방법이기 때문에 한정된 환경에서만 적용되거나 실시간으로 적용하기 문제점이 있다.

본 논문에서는 객체로부터 특징 벡터를 추출하고 각 객체의 행동 양식을 분석함으로써 현재 객체에서 어떤 이벤트가 발생했는가를 인식하는 방법을 제안하였다. 제안된 방법은 먼저 객체로부터 행동 양식을 분석하기 위한 특징 벡터를 추출하고 이를 바탕으로 독립 객체의 이벤트를 정의하였다. 이벤트 규칙은 추출된 특징 벡터를 이용하여 주-이벤트, 움직임-이벤트, 상호-이벤트를 정의하고 현재 객체의 특징 벡터와 확률을 이용하여 객체의 이벤트를 인식하게 된다. 정의된 이벤트를 기반으로 현재 객체의 이벤트를 인식하고 객체의 움직임을 확률적으로 해석함으로써 보다 정확한 이벤트 인식이 되도록 제안하였다.

제안한 기본 움직임 이벤트가 실험 영상 내에서 적절하게 인식되는지를 확인한 결과, 주-이벤트와 움직임-이벤트의 이벤트 인식 정확도는 90% 이상이었고, 상호-이벤트는 85% 이상의 정확도를 보였다. 그리고 SMOV와