

긴급 상황 시나리오 해석을 위한 독립 객체의 규칙 기반 및 확률적 이벤트 인식

이준철[†], 최창규^{**}

요 약

기존의 이벤트 인식은 한정된 규칙 기반으로 이루어졌고, 시나리오 해석은 확률 자료의 크기로 많은 학습 시간이 필요했다. 본 논문에서는 객체로부터 특징 벡터를 추출하고 각 객체의 행동 양식을 분석하여 현재 객체의 이벤트를 인식하는 방법과 확률 모델을 기반으로 본 논문에서 정의한 긴급 상황 시나리오를 해석할 수 있는 방법을 제안한다. 독립 객체의 이벤트 규칙은 주-이벤트, 움직임-이벤트, 상호-이벤트, 그리고 'FALL DOWN' 이벤트로 구성되며, 객체의 특징 벡터와 베이지안 네트워크에 의해 학습된 분할 움직임 방향 벡터(SMOV)를 통해 정의된다. 긴급 상황 시나리오는 현재 이벤트의 상태와 사후 확률에 의해 분석된다. 본 논문에서는 기존 방법에 비해 다양한 이벤트를 정의하였고 이벤트 간의 독립성을 높여 확장성이 용이하도록 하였다. 그리고 객체 추적만을 통해 얻을 수 없는 의미론적 정보를 규칙과 확률을 기반으로 획득할 수 있었다.

Rule-based and Probabilistic Event Recognition of Independent Objects for Interpretation of Emergency Scenarios

Jun-Cheol Lee[†], Chang-Gyu Choi^{**}

ABSTRACT

The existing event recognition is accomplished with the limited systematic foundation, and thus much longer learning time is needed for emergency scenario interpretation due to large scale of probability data. In this paper, we propose a method for rule-based event recognition of an independent object(human) which extract a feature vectors from the object and analyze the behavior pattern of each object and interpretation of emergency scenarios using a probability and object's events. The event rule of an independent object is composed of the Primary-event, Move-event, Interaction-event, and 'FALL DOWN' event and is defined through feature vectors of the object and the segmented motion orientated vector (SMOV) in which the dynamic Bayesian network is applied. The emergency scenario is analyzed using current state of an event and its post probability. In this paper, we define diversified events compared to that of pre-existing method and thus make it easy to expand by increasing independence of each events. Accordingly, semantics information, which is impossible to be gained through an

Key words: event recognition(이벤트 인식), emergency scenario(긴급 상황 시나리오), visual surveillance system(비전 감시 시스템)

※ 교신저자(Corresponding Author): 이준철, 주소: 경기도 부천시 원미구 심곡동 424(420-735), 전화: 032)610-3483, FAX: 032)610-3218, E-mail: jclee@bc.ac.kr
접수일: 2007년 9월 19일, 완료일: 2008년 1월 2일
[†] 정회원, 부천대학 인터넷과 부교수

^{**} (주)테크바일 개발 팀장
(E-mail: cgchoi@techbile.com)

※ 본 연구는 2007년도 부천대학 교내 학술연구비 지원으로 수행되었음

1. 서 론

일상생활에서 사용하는 많은 종류의 정보들 중에서 영상과 같은 시각 정보가 차지하고 있는 비율은 매우 크며, 가장 효율적인 정보 전달력을 가지고 있다. 처음에는 정지 영상을 대상으로 다양한 형태의 영상 정보로 처리하였으나 H.263, MPEG 등의 표준안이 제시되면서 동영상에 관련된 연구들이 증가하였다[1-3]. 동영상은 정지 영상에 비해 처리하는 데이터의 양이 많을 뿐만 아니라 영상의 내용이 시간적으로 변하는 특성을 가지고 있다. 그러므로 원하는 영상 정보를 쉽게 검색하거나 저장할 수 있는 방법, 고용량의 동영상 압축 기술, 다시점 기술(multi-view point) 등에 대한 다양한 연구들이 활발하게 이루어지고 있다[4-5].

최근에는 동영상에서 상대적으로 움직임이 많은 객체를 추출하거나 객체의 특징 분석을 통해 컴퓨터 비전을 자동화 시스템으로 확장하는 동영상 처리 기술이 포함되는 비전 감시 시스템(visual surveillance system)이 연구되고 있다[6-11]. 기존의 비전 감시 시스템은 CCTV, 적외선 카메라 등과 같은 영상 획득 장치를 이용하여 주차장 관리에서부터 건물의 방범 시스템, 도심 도로와 고속도로 등의 교통 감시 시스템, 산업 현장에서 영상 분석을 통한 공정 자동화 시스템 등의 다양한 분야에 적용되고 있다. 그러나 CCTV를 이용한 일반적인 감시 시스템은 항상 사람의 개입에 의한 수동적인 조작이 필요하며 순수한 동영상 정보로만 관리되므로 동영상의 다양한 특징 정보를 활용한 검색이나 비교 등의 고급화된 영상 정보 분석이 어렵다. 또한, 특별한 예외 상황이 발생할 경우 저장되어 있는 영상 정보들을 인위적으로 순차 검색을 하므로 상황 해결에 많은 시간이 소모된다. 이와 같은 감시 시스템의 문제점은 향후 시스템 자동화를 통해 특정 환경에서 필요한 정보를 추출하고, 객체의 이벤트를 인식하고, 객체의 행동을 분석하는 시나리오 해석 분야가 관심을 가지게 되었다[9,12-13]. 시나리오 해석은 객체의 행동 이벤트를 기반으로, 인식된 객체의 행동 이벤트를 확률 모델 또는 직관적인 접근 등을 이용하여 미리 설정된 시나리오와 얼마나 유사한 것인가를 자동으로 해석하는 분야이다. 객체의 시나리오 해석은 시간의 흐름에 따른 객체의 상태가 미리 정의된 상태 전이 과정과 얼마나 유사한가를 판단하는 것이다.

본 논문에서는 실내 또는 실외 환경에서 추출된

객체의 특징 벡터를 분석하여 일반적으로 발생할 수 있는 객체의 이벤트를 정의 및 인식하고 긴급 상황에서 발생하는 “사람이 쓰러지다” 또는 “강도를 당하다”의 시나리오를 해석하는 방법을 제안한다. 제안한 방법은 입력 영상에서 인식된 객체로부터 특징 벡터를 구하고 시간에 따른 특징 벡터값의 변화를 분석하여, 정의된 이벤트의 규칙에 따라 해당 객체의 이벤트를 인식한다. 특징 벡터는 매 프레임 또는 주기적 프레임 단위로 추출되며, 객체의 이벤트는 시간에 따라 현재 상태를 유지하거나 다른 이벤트로 전이가 된다. 그리고 긴급 상황에 대한 시나리오를 정의하고, 인식된 객체의 이벤트를 바탕으로 미리 정의된 시나리오에 가장 근접하는 상황이 발생하는 것을 자동으로 해석할 수 있도록 한다. 정의된 시나리오는 해당 객체의 시간에 따른 이벤트 전이와 객체의 특징 벡터를 확률적 모델을 통해 해석한다.

본 논문의 구성은 다음과 같다. 2장에서는 이벤트 인식과 시나리오 해석에 대한 기본 개념과 관련 연구들을 설명한다. 3장에서는 본 논문에서 제안하는 객체의 이벤트 정의 및 인식, 시나리오 해석 방법을 제안하고 이벤트와 확률 모델을 이용한 긴급 상황의 시나리오 해석 방법을 제시한다. 4장에서는 본 논문에서 제안한 이벤트 인식과 긴급 상황의 시나리오 해석에 대한 실험 결과를 제시하고, 마지막으로 5장에서는 결론을 맺고 향후 연구에 대해 기술한다.

2. 이벤트 인식과 시나리오 해석 방법의 기본 개념 및 관련 연구

이벤트 인식이란 특정 객체가 시간 흐름에 따라 발생하는 움직임을 사전에 정의된 규칙과 특징 벡터를 분석하여 객체가 어떤 움직임을 나타내고 있는가를 인식하는 것이다[13]. 이벤트 인식은 카메라의 위치, 실내 또는 실외 환경, 독립 객체, 그룹 객체 등의 여러 요소에 따라 다양한 형태로 연구되고 있다[8,12,14]. 이벤트 인식 분야는 이러한 요소와 사람과 같은 객체가 행동할 수 있는 행동 패턴 양식이 다양하기 때문에 제한된 환경을 가진 비전 감시 시스템에만 적용되고 있는 실정이다.

2.1 규칙 기반의 독립 객체 이벤트 인식 및 관련 연구

규칙 기반의 이벤트 인식은 독립적으로 나타난 객

체를 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 인식한다. 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 적용한다.

A. Amer[8]는 실내 또는 실외에서 발생할 수 있는 독립 객체의 행동 분류를 4개의 도메인으로 나누고, 각각의 도메인을 다시 세부적인 이벤트로 정의하였다. 모든 이벤트는 특징 벡터값을 기준으로 규칙이 정해지고 해당 객체가 시간에 따른 특징 벡터값이 어느 기준에 맞느냐에 따라 인식된다. W^4 는 객체의 수직/수평 히스토그램을 이용하여 객체의 이벤트를 인식한다[15]. 미리 정의된 템플릿의 수직/수평 히스토그램과 현재 인식된 객체의 수직/수평 히스토그램 간의 유사도를 구하여 가장 높은 값을 현재 객체의 이벤트로 인식한다.

독립 객체에 대한 이벤트 인식은 해당 객체의 특징 벡터를 매 프레임 또는 주기적으로 추출하여 미리 정의된 규칙을 적용하여 가장 적합한 이벤트를 현재 객체의 이벤트로 정의하므로 실시간의 비전 감시 시스템의 응용에 적합하다. 이 방법은 확률 기반의 이벤트 인식과 다르게 특별한 학습 과정이 필요 없고, 특징 벡터를 바로 추출하기 때문에 연산 시간이 적은 장점을 가진다. 그러나 객체가 행할 수 있는 행위는 다양하고 이를 모든 규칙에 적용하기는 불가능하다.

2.2 확률을 통한 이벤트 인식 및 관련 연구

규칙 기반 이벤트 인식은 처리 속도가 빠른 장점은 있으나 적용할 수 있는 환경이 한정되는 문제점을 가진다. 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 베イズ 이론을 바탕으로 노드와 방향성을 가지는 아크로 구성된 방향성 그래프이다[16]. 각 노드는 발생할 수 있는 행위에 대한 조건부 확률 테이블(CPT, conditional probability table)을 가지고, 아크는 노드와 노드간의 확률 의존성을 나타낸다. 베이지안 추정은 자식 노드에 대한 현재 증거를 바탕으로 부모 노드의 사후 확률이 결정되는데, 일반적으로 우도(likelihood) 함수를 통해 수행된다[17].

R. V. Babu[18]는 입력 영상에서 연속적으로 들어오는 프레임 내의 객체로부터 움직임 기록 영상(움직임 기록 영상, motion history image), 움직임 흐름 영상(MFH, motion flow history)을 정의한 후 특징 벡터를 추출한다. 이 특징 벡터들을 베이지안 네트워크로 학습한 후 WALK, RUN, JUMP 등의 이벤트를 해석하였다. G. Guler[11]는 MEET, Approach, MOVE TOGETHER, Move away 등의 이벤트를 정의하고 객체의 크기 변화, 객체 간의 거리 변화, 객체의 이동거리를 관측치로 하여 이벤트를 인식하였다. 제안된 방법은 각 이벤트 간의 상태 정리가 발생하고 각 관측치로부터 전이 확률을 정의하였다.

확률을 통한 이벤트 인식은 독립된 객체의 특징 벡터를 이용하여 인식하는 방법보다 정확성이 뛰어나다. 그 이유는 확률적인 접근을 하기 때문에 예상할 수 없는 상황에 대한 대처가 용이하기 때문이다. 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 기반의 이벤트 인식 방법에 비해 실시간에 적용하기에는 문제점이 있다.

2.3 시나리오 해석과 시나리오 해석 이벤트의 기본 개념 및 관련 연구

시나리오 해석은 이벤트 인식이 된다는 가정 하에서, 특정 환경에서 특정 행위가 발생한다는 것을 전제로 객체의 각 행위를 상태로 정의하고, 해당 객체가 상태의 흐름에 따라 특정 행위를 수행했는가를 인식한다. 객체의 행위는 다양하게 나타나기 때문에 모든 경우에 대한 시나리오를 설정하는 것은 불가능

하다. 따라서 특정 상황에 맞는 시나리오를 작성하고 해당 시나리오가 해석되면 객체는 특정 행동을 했다는 것으로 판단한다.

초기 이벤트에서 확률적인 접근 또는 객체의 특징 벡터의 변화에 따라 다음 이벤트로 전이되고 이 상태에서 다시 다음 이벤트로 전이되면 해당 시나리오가 발생했다고 할 수 있다. 시나리오 해석 이벤트는 정의된 시나리오에서 발생할 수 있는 독립 객체의 움직임 이벤트를 나타낸다. 즉, 본 논문에서 정의한 “사람이 쓰러지다”의 긴급 상황 시나리오 내에서 발생할 수 있는 객체의 특별한 행동 이벤트를 나타낸다. 각 이벤트의 인식은 시간의 흐름에 따라 독립 객체의 특징 벡터와 확률 정보를 통해 정해진 규칙에 가장 가까운 상황에 대해 계속적으로 이루어진다.

S. Hongeng[12]은 “사람에게 접근하여 그 사람이 가지고 있는 물건을 훔치다(Stealing)” 시나리오를 Converse, Approach1, Approach2, Blocking, Tacking_Object의 5개 이벤트로 구분하였다. “Stealing” 시나리오는 은닉 마코프 모델(HMMs, hidden markov models)을 사용하여 현재 관측치로부터 가장 높게 발생하는 이벤트가 시간의 흐름에 따라 어떤 순서로 발생했는지를 확인하여 해석된다. I. Ersoy[9]는 사람과 차에 대해 방향, 속도, 거리 특징 벡터를 사용하여 T시간 후의 객체 시나리오를 해석하는 방법을 제안하였다. 제안된 방식은 확률 모델을 통해 접근하지 않고 시간에 따른 특징 벡터를 분석하여 발생된 이벤트의 AND 또는 OR 연산에 따라 시나리오가 정의된다.

시나리오 해석에 대한 연구는 현재 다른 비전 감시 시스템 분야에 비해 많은 연구가 이루어지고 있지 않다. 그 이유는 객체 추적, 이벤트 인식 등의 전처리가 잘 되어야만 가능하고, 실제 발생할 수 있는 시나리오가 다양하고 정의된 시나리오 내에서 객체가 반드시 해당 이벤트를 일으키지 않기 때문이다. 따라서 시나리오 해석을 위해서는 확률 모델의 접근을 통한 시나리오의 정의와 객체의 시간에 따른 상태 전이를 정확하게 인식할 수 있는 이벤트 처리가 필요하다.

3. 독립 객체의 이벤트 인식 및 긴급 상황 시나리오 해석

실내 또는 실외 비전 감시 시스템에서 인식된 독립

객체가 행동할 수 있는 기본 움직임 이벤트를 정의하고 인식하는 방법과 기본 움직임 이벤트를 기반으로 긴급 상황에서 발생하는 시나리오를 정의하고 해석하는 방법을 제시한다. 제안하는 이벤트는 일반적으로 실내 또는 실외에서 발생할 수 있는 독립 객체의 행위이고 각 객체의 행위를 규칙으로 정의한다. 긴급 상황에서의 시나리오는 “사람이 갑자기 쓰러지다”, “강도를 당하다” 등 긴급하게 발생하는 상황이고 이를 일반적인 상황과 구분할 수 있도록 해석하게 된다.

독립 객체의 이벤트 인식 및 시나리오 해석을 위해서는 객체 인식 및 추적을 위한 전처리 과정, 특징 벡터 추출, 이벤트 규칙을 통한 이벤트 인식, 그리고 베이지안 네트워크를 적용한 이벤트 인식과 시나리오 해석으로 나누어진다. 그림 1은 독립 객체의 이벤트 인식 및 시나리오 해석 방법의 개요를 보여준다.

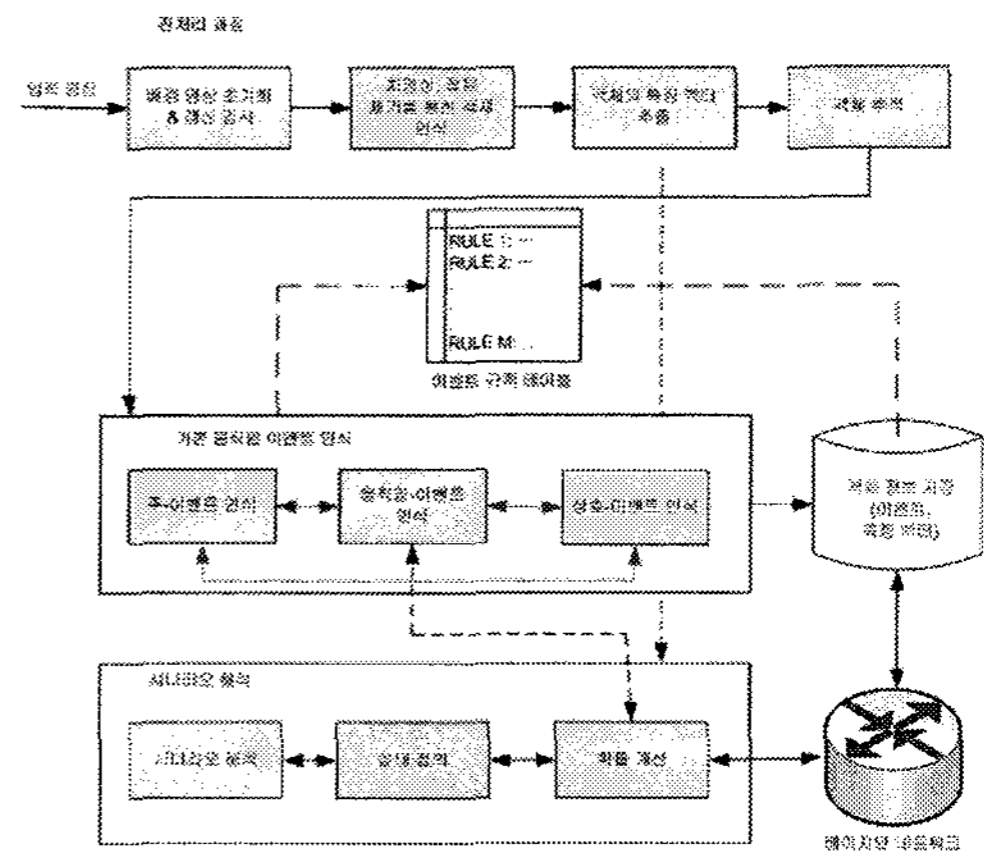


그림 1. 방법의 개요

3.1 전처리 과정과 특징 벡터 추출

전처리 과정은 입력 영상으로부터 객체를 분리하고 동일 객체를 추적하기 위한 과정으로, 배경 모델링, 객체 인식, 객체로부터의 특징 벡터 추출, 동일 객체 추적으로 구성된다. 독립객체의 이벤트를 정의하기 위해 기본이 되는 것은 객체의 특징 벡터이다. 동일한 객체로부터 특징 벡터를 추출해야 해당 객체의 이벤트를 정확하게 인식할 수 있다. 입력 영상에 대한 배경 모델링과 객체 인식 방법은 [10]에서 제안된 방법을 이용한다. 그림 2는 이와 같은 전처리 과정이다.

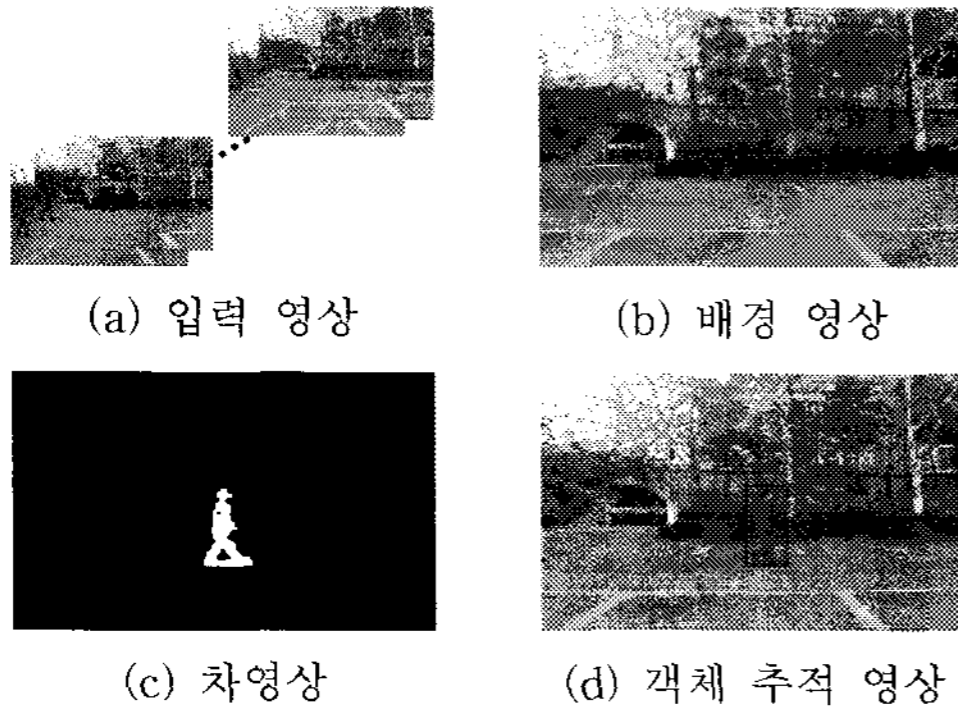


그림 2. 이벤트 인식을 위한 전처리 과정

객체의 특징 벡터는 독립 객체의 이벤트를 정의하는데 가장 중요한 요소이다. 표 1에서 독립 객체의 이벤트를 정의하기 위한 특징 벡터들을 나타내고, 각 특징 벡터의 특성에 따라 이벤트의 규칙이 정의되고 이를 통해 시간의 흐름에 따른 객체의 이벤트가 인식된다.

각 특징 벡터의 추출 방법은 [10]에서 제안한 방법을 이용한다. 질감 템플릿 영상은 t_1 시간 동안 객체의 질감 부분을 겹쳐서 나타낸 것으로 W^d 에서 사용된 이차원 동적 템플릿(2D dynamic template)을 사용한다[15]. 질감 템플릿 영상은 t_1 주기 동안 가장 최근의 질감 템플릿 영상으로 갱신되고, 객체의 이벤트가 OCCLUDE로 발생되면 동일 객체를 추적하기 위해 질감 템플릿 매칭을 수행한다. 분할 움직임 방향 벡터는 움직임 기록 영상을 통해 독립 객체의 움직임 기울기(motion gradient)를 구하여 특정 부분의 방향 성분을 추출하는 특징 벡터이다[19]. 분할 움직임 방향 벡터는 픽셀 단위의 연산에 비해 특징 벡터의 차원이 줄고 연산 시간이 빠른 장점을 가진다.

표 1. 객체로부터 추출되는 특징 벡터

특징 벡터	설 명
테두리 상자	인식된 객체의 둘레 정보
중심점	객체의 중심점 좌표값
속도	객체의 평균 이동 속도
방향	객체의 이동 방향
색상	객체의 RGB 대표 색상 정보
질감 템플릿 영상	겹침을 해결하기 위한 질감 영상
분할 움직임 방향 벡터	동일한 방향의 움직임을 가지는 벡터 성분. 0~360°로 표현됨.

3.2 독립 객체의 이벤트 정의

독립 객체의 이벤트는 특징 벡터 정보들을 기반으로 규칙을 정한다. 정의된 이벤트는 실내 또는 실외 환경에서 독립 객체가 일반적으로 발생할 수 있는 이벤트로 ‘나타나다’, ‘사라지다’, ‘걷다’, ‘뛰다’ 등의 행위를 표현하는 기본 움직임 이벤트와 “넘어지다”의 시나리오 내에서 객체의 특별한 행동이 발생하는 이벤트인 시나리오 해석 이벤트이다. 각 이벤트의 인식은 시간의 흐름에 따라 독립 객체의 특징 벡터와 확률 정보를 통해 정의한 특징 벡터를 객체로부터 추출하고 추출된 특징 벡터에 확률 정보를 더하여 그림 3에서 정의한 이벤트 규칙에 확률적으로 가장 가까운 이벤트를 현재의 이벤트로 인식한다.

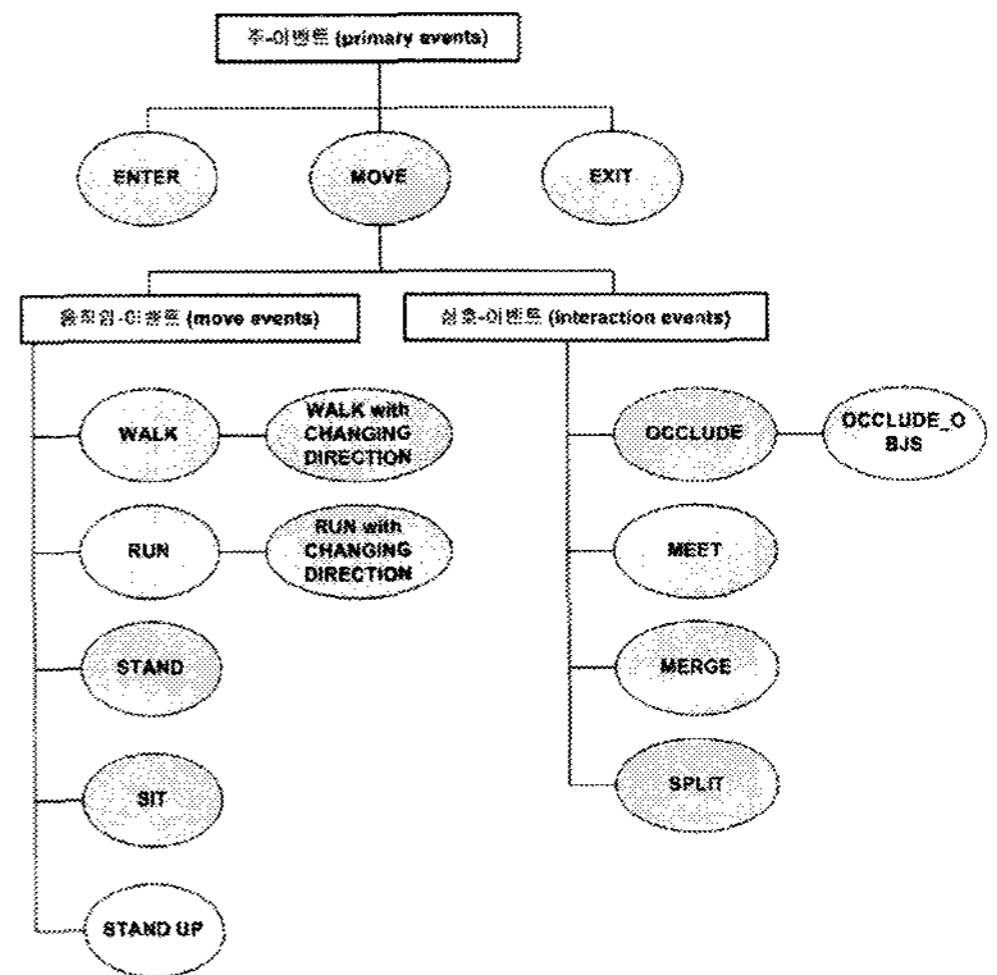


그림 3. 규칙 기반에 의해 정의된 이벤트

주-이벤트(primary events)는 독립 객체의 기본이 되는 움직임으로 객체가 인식되는 과정에서부터 객체가 영상 내에서 사라지는 이벤트들로 구성된다. 움직임-이벤트(move events)는 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. 상호-이벤트(interaction event)는 두 개 이상의 독립 객체에서 발생하는 이벤트로, 객체간의 겹침 현상, ‘만나다(MERGE)’, ‘같이 걷다(MEET)’ 등의 객체 간에 상호 영향을 주는 이벤트와 ‘분리되다(SPLIT)’ 등과 같이 다시 이전의 독립 객체로 돌아가

는 이벤트로 구성된다. 상호-이벤트가 발생하는 시점에서는 동일 객체의 추적을 위해 질감 템플릿 영상을 이용한다.

3.2.1 ENTER

ENTER 이벤트는 현재 인식된 객체 O_i 이 이전 프레임에 존재하지 않고(식 (1)), $t_{ENTER}(t_1)$ 시간 동안 현재 프레임 내에 존재했을 경우이거나 객체의 중심점 CP_i 가 프레임의 테두리가 되는 I_{Border} 내에 존재하지 않는 시간이 t_{ENTER} 이상일 때 발생한다(식 (2)). 해당 객체는 이벤트 발생과 함께 새로운 객체로 데이터베이스에 등록되고 특징 벡터와 이벤트 정보가 저장된다.

● 규칙 1. ENTER 이벤트

$O_i^c \notin I_p$	(1)
$TIME (O_i^c \in I_c) \geq t_{ENTER}$ or	(2)
$TIME (CP_i \notin I_{Border}) \geq t_{ENTER}$	

3.2.2 MOVE

객체 O_i 의 현재 이벤트가 ENTER 이벤트이고(식 (3)), 해당 객체가 $t_{MOVE}(t_1)$ 시간 이상 영상 내에 존재할 경우 MOVE 이벤트가 발생한다(식 (4)). 해당 이벤트는 MOVE 이벤트가 발생하는 동안 객체의 속도 특징 벡터 정보를 추출하고, 속도의 특징에 따라 움직임-이벤트로 전이하게 된다.

● 규칙 2. MOVE 이벤트

$EVENT (O_i^c)_p = ENTER$	(3)
$TIME(O_i^c \in I_p \text{ and } O_i^c \in I_c) \geq t_{MOVE}$	(4)

3.2.3 EXIT

객체가 이전 프레임에서는 존재했으나 현재 프레임에 존재하지 않으면 해당 객체는 NOT FOUND가 되고(식 (5)), 이후 입력 프레임에 대해 동일 객체를 찾지 못해 식 (6)과 같이 객체가 현재 영상에 존재하지 않는 시간이 $t_{EXIT}(t_1)$ 이상이 되면 해당 객체는 사라진 것으로 판단하고 EXIT 이벤트를 발생한다. EXIT 이벤트가 발생한 객체는 데이터베이스에 저장된 모든 정보가 삭제되는데, 이것은 독립 객체의 이

벤트 인식 과정의 연산 시간을 줄이기 위한 것이다.

● 규칙 3. EXIT 이벤트

$O_i^c \in I_p \text{ and } O_i^c \notin I_c$	(5)
식 (5)를 만족했을 시, $TIME (O_i^c \notin I_p \text{ and } O_i^c \notin I_c) \geq t_{EXIT}$	(6)

3.2.4 WALK/RUN

WALK와 RUN 이벤트는 MOVE 이벤트 이후에 객체가 현재 영상 내에 계속 존재하고(식 (7), 식 (10)), 평균 속도 $f(V_i^{avg})$ 가 μ_{WALK}^{min} 이상이고 μ_{WALK}^{max} 이하인 경우는 WALK가(식 (8)), μ_{RUN}^{min} 이상이고 μ_{RUN}^{max} 이하인 경우에는 RUN이 된다(식 (11)). 두 이벤트는 특성이 거의 비슷하지만 평균 속도는 RUN 이벤트가 WALK 이벤트 보다 빠른 특성이 있다. 그리고 분할 움직임 방향 벡터는 머리의 경우 거의 같지만 상체와 하체가 RUN 이벤트가 WALK 이벤트 보다 급격한 경사를 가진다. 만약 객체의 분할 움직임 방향 벡터의 사후 확률이 WALK가 될 경우에는 WALK 이벤트(식 (9))가 되고, RUN 이벤트의 사후 확률이 가장 클 경우에는 RUN 이벤트가 된다(식 (12)). 즉 식 (9)와 식 (12)의 사후 확률값과 평균 속도에 의해 결정된 이벤트의 결과를 가중치를 두어 결정한다.

● 규칙 4. WALK 이벤트

$O_i^c \in I_p \text{ and } O_i^c \in I_c$	(7)
$\mu_{WALK}^{min} \leq f(V_i^{avg}) \leq \mu_{WALK}^{max}$	(8)
$EVENT \left(\max_{\forall (E_1, E_2, \dots, E_j, \dots, E_n)} Bel(E_j) \right)_c = WALK$	(9)

● 규칙 5. RUN 이벤트

$O_i^c \in I_p \text{ and } O_i^c \in I_c$	(10)
$\mu_{RUN}^{min} \leq f(V_i^{avg}) \leq \mu_{RUN}^{max}$	(11)
$EVENT \left(\max_{\forall (E_1, E_2, \dots, E_j, \dots, E_n)} Bel(E_j) \right)_c = RUN$	(12)

여기에서 객체의 평균 속도의 최소, 최대값을 μ_{WALK}^{min} , μ_{WALK}^{max} , μ_{RUN}^{min} , μ_{RUN}^{max} 로 정의하고, 정의된 평균 속도의 최소, 최대값은 10개 영상의 WALK와 RUN 이벤트로 판단되는 객체들의 평균 속도를 통해 구한다.

3.2.5 STAND

STAND 이벤트는 객체가 영상 내에 계속 존재하

고(식 (13)), 평균 속도 $f(V_i^{avg})$ 가 WALK 이벤트의 최소 평균 속도인 μ_{WALK}^{min} 보다 낮을 경우에 발생한다(식 (14)). 객체의 분할 움직임 방향 벡터는 거의 기울기가 존재하지 않기 때문에 식 (14)와 식 (15)를 통해 WALK 이벤트와 RUN 이벤트를 구분한다.

● 규칙 6. STAND 이벤트

$$O_i^c \in I_p \text{ and } O_i^c \in I_c \quad (13)$$

$$f(V_i^{avg}) < \mu_{WALK}^{min} \quad (14)$$

$$EVENT \left(\max_{\forall (E_1, E_2, \dots, E_j, \dots, E_n)} Bel(E_j) \right)_C = STAND \quad (15)$$

3.2.6 SIT

SIT 이벤트는 WALK나 RUN 중에 발생하는 것은 불가능하기 때문에 반드시 이전 이벤트는 STAND가 되어야 한다(식 (16)). 속도가 μ_{WALK}^{min} 보다 작고 분할 움직임 방향 벡터는 전체적으로 아래쪽으로 방향을 가지게 되므로 사후 확률값이 WALK, RUN, STAND UP 이벤트와 구분된다(식 (17)).

● 규칙 7. SIT 이벤트

$$EVENT (O_i^c)_p = STAND \quad (16)$$

$$EVENT \left(\max_{\forall (E_1, E_2, \dots, E_j, \dots, E_n)} Bel(E_j) \right)_C = SIT \quad (17)$$

3.2.7 STAND UP

해당 이벤트는 객체가 벤치나 의자 등에 ‘앉다’ 이벤트인 SIT인 상태(식 (16))에서 일정 시간 후에 객체가 다시 ‘일어서다’라는 이벤트로 전이할 때 발생한다. STAND UP 이벤트에서 가장 큰 특징은 분할 움직임 방향 벡터가 위쪽으로 향하는 것이다.

● 규칙 8. STAND UP 이벤트 규칙

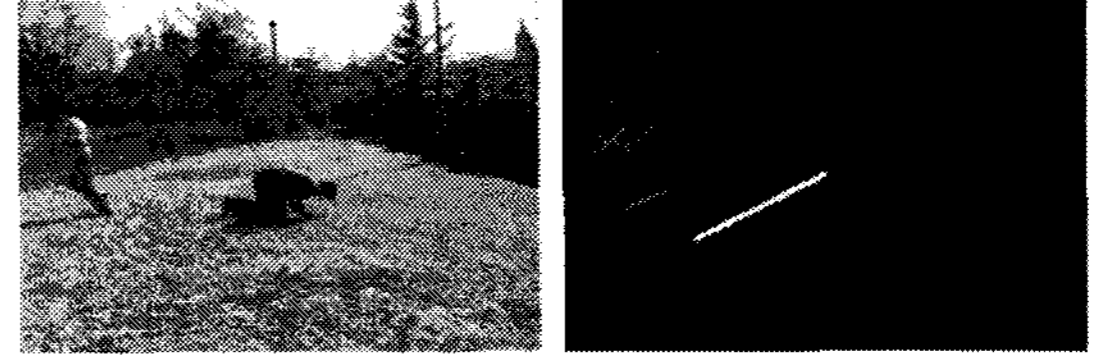
$$EVENT (O_i^c)_p = SIT \quad (18)$$

$$EVENT \left(\max_{\forall (E_1, E_2, \dots, E_j, \dots, E_n)} Bel(E_j) \right)_C = STAND UP \quad (19)$$

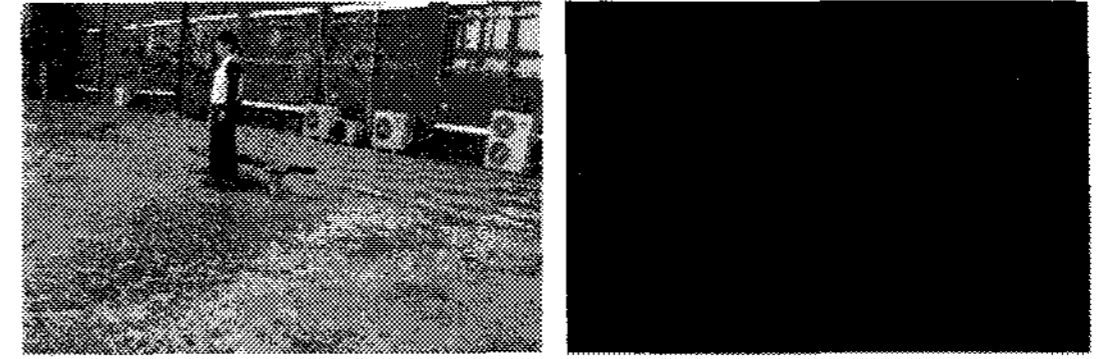
그림 4는 움직임-이벤트인 WALK, RUN, STAND, SIT, STAND UP에 대한 입력 영상과 각 영상의 분할 움직임 방향 벡터의 예다. 그림 4의 왼쪽은 입력 영상이고, 오른쪽은 해당 입력 영상에 대한 움직임 기록 영상과 분할 움직임 방향 벡터를 표현한 것이다.



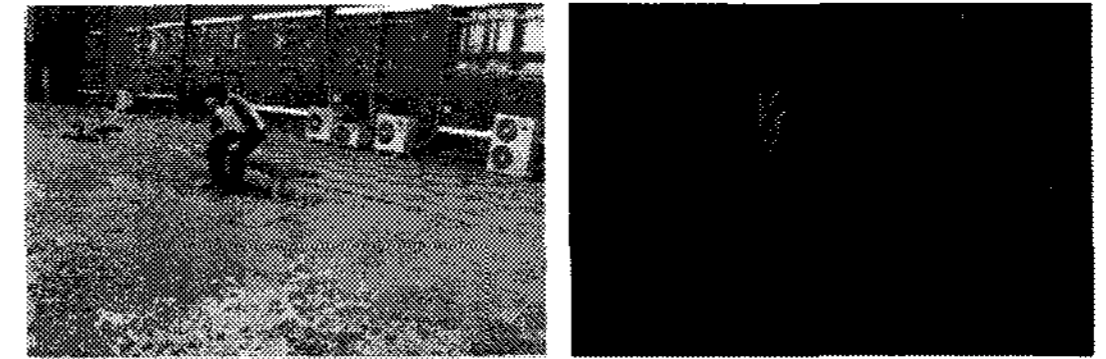
(a) WALK 이벤트



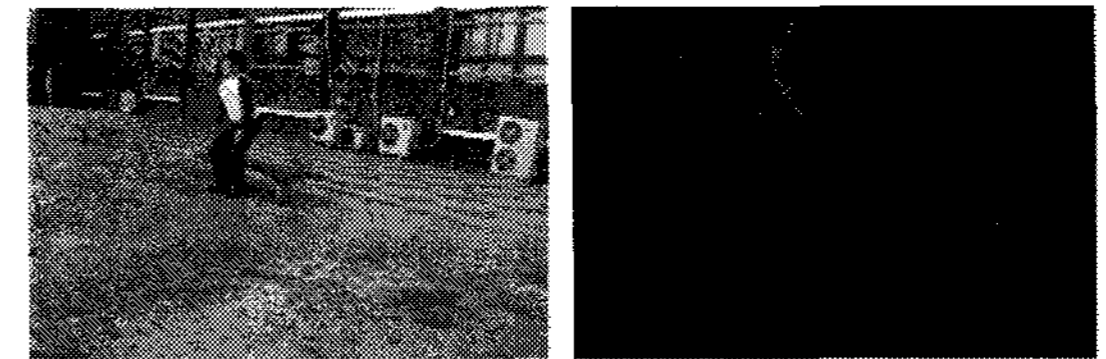
(b) RUN 이벤트



(c) STAND 이벤트



(d) SIT 이벤트



(e) STAND UP 이벤트

그림 4. 움직임-이벤트 입력 영상과 분할 움직임 방향 벡터

3.2.8 OCCLUDE/OCCLUDE_OBJJS

객체 O_i 와 O_j 는 영상 내에 계속 존재하고 새로운 객체 O_k 가 ENTER 이벤트를 발생하고(식 (20)), 객체 O_i 와 O_j 의 테두리 상자가 객체 O_k 의 테두리 상자 내에 포함되고(식 (21)), 객체의 이동 방향이 서로 다른 경우(식 (22))에 객체 O_i 와 O_j 는 OCCLUDE 이벤트가 발생하고 객체 O_k 는 OCCLUDE_OBJJS 이벤트가 발생한다.

● 규칙 9. OCCLUDE/OCCLUDE_OBJS 이벤트

$$O_i^c \in I_p \text{ and } O_i^c \in I_c \text{ and } O_j^c \in I_p$$

$$\text{and } O_j^c \in I_c \text{ and } \text{EVENT } (O_k^c)_c$$

$$= \text{ENTER} \tag{20}$$

$$\text{FOR } l = i \text{ to } j$$

$$f(B_k^{up}) < f(B_l^{up}) \text{ and}$$

$$f(B_k^{down}) > f(B_l^{down}) \text{ and}$$

$$f(B_k^{left}) < f(B_l^{left}) \text{ and}$$

$$f(B_k^{right}) > f(B_l^{right}) \tag{21}$$

$$f(D_i) \neq f(D_j) \tag{22}$$

i 와 j 는 겹침이 발생한 객체 인덱스이고, k 는 겹쳐서 새로 생성된 객체의 인덱스.

3.2.9 MERGE

객체 O_i 와 O_j 의 이전 이벤트가 OCCLUDE이고 (식 (23)), 각 해당 객체의 방향 특징 벡터가 동일하고 속도 특징 벡터가 WALK의 최소 평균 속도인 μ_{WALK}^{min} 이상일 경우(식 (24)) MERGE로 판단하며, MERGE의 의미는 ‘합쳐지다’ 보다는 ‘같이 걸어가다’로 해석할 수 있다. 이때에도 동일 객체를 추적하기 위하여 질감 템플릿 매칭을 수행한다.

● 규칙 10. MERGE 이벤트

$$\text{EVENT } (O_i^c)_p = \text{OCCLUDE} \text{ and}$$

$$\text{EVENT } (O_j^c)_p = \text{OCCLUDE} \tag{23}$$

$$f(V_i^{avg}) \geq \mu_{WALK}^{min} \text{ and } f(V_j^{avg}) \geq \mu_{WALK}^{min}$$

$$\text{and } f(D_i) = f(D_j) \tag{24}$$

3.2.10 MEET

MEET 이벤트는 MERGE 이벤트와 마찬가지로 해당 객체의 이전 이벤트가 OCCLUDE 상태이지만 MERGE 이벤트와 다른 점은 속도 특징 벡터값이 μ_{WALK}^{min} 보다 작다는 것이다. 즉, 두 개 이상의 객체가 겹침 현상이 발생한 후 움직이지 않고 있는 상태이며, 이 때 ‘서로 이야기하다’의 의미로 해석이 된다. 질감 템플릿 매칭을 통해 동일 객체의 추적은 계속 이루어진다.

● 규칙 11. MEET 이벤트

$$\text{EVENT } (O_i^c)_p = \text{OCCLUDE} \text{ and}$$

$$\text{EVENT } (O_j^c)_p = \text{OCCLUDE} \tag{25}$$

$$f(V_i^{avg}) < \mu_{WALK}^{min} \text{ and } f(V_j^{avg}) < \mu_{WALK}^{min} \tag{26}$$

3.2.11 SPLIT

객체의 이벤트가 MERGE 또는 MEET 상태에서 하나 이상의 객체로 분리될 때 SPLIT 이벤트가 발생한다. 현재 객체의 이벤트가 SPLIT이 되기 위해서는 먼저, OCCLUDE_OBJS 이벤트인 객체(O_k)의 바로 가까운 곳에(th_{dist}) 새로운 객체(O_l)가 나타나고, 이 시점에서 객체 O_k 의 이전 시점에서의 테두리 상자 크기와 현재 시점에서의 테두리 상자 크기를 비교한다. 그리고 독립 객체 O_i , O_j 와 O_k 간의 템플릿 매칭 (식 (30))을 통해 동일 객체의 존재 여부를 확인한다. 만약 동일 객체가 존재하지 않으면 템플릿 매칭 함수는 FALSE를 결과로 리턴한다.

3.3 이벤트 인식

독립 객체의 이벤트 인식은 정의된 규칙과 시간의 흐름에 따른 객체의 특징 벡터에 따라 진행된다. 영상 내에서 객체의 이벤트는 객체가 존재하기 시작하면서 ENTER 이벤트가 인식되고 객체가 사라지면서 EXIT 이벤트가 발생한다. 주-이벤트 사이에서 발생하는 움직임-이벤트, 상호-이벤트는 해당 객체의 특징 벡터를 t_1 주기 프레임 또는 매 프레임마다 추출하고 객체의 현재 이벤트와 정의된 규칙을 통해 상태 전이된다. 인식된 해당 객체의 이벤트는 데이터베이스에 계속 저장되고 EXIT 이벤트가 발생하는 시점에서 더 이상의 객체 추적을 수행하지 않는다.

움직임-이벤트는 객체의 속도 특징 벡터와 함께 분할 움직임 방향 벡터를 함께 사용한다. WALK 또는 RUN, STAND, SIT, STAND UP 이벤트는 객체의 속도가 차이나지만 WALK와 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. 그림 5는 움직임-이벤트를 구분하기 위한 베이지안 모델이다.

설계된 베이지안 네트워크의 구성은 움직임이 있는 WALK/RUN 이벤트와 움직임이 거의 없는 STAND 이벤트로 나눈다. 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. 그림 5에서 자식 노드는 움직임이 존재하는 경우 분할 움직임 방향 벡터의 학습 결과를 통해 WALK와 RUN 이벤트의 사전 확률 값이 만들어지고, 움직임이 없는 STAND는 SIT, STAND UP, 그리고 시나리오 해석에 필요한 이벤트인 FALL DOWN으로 만들어진다.

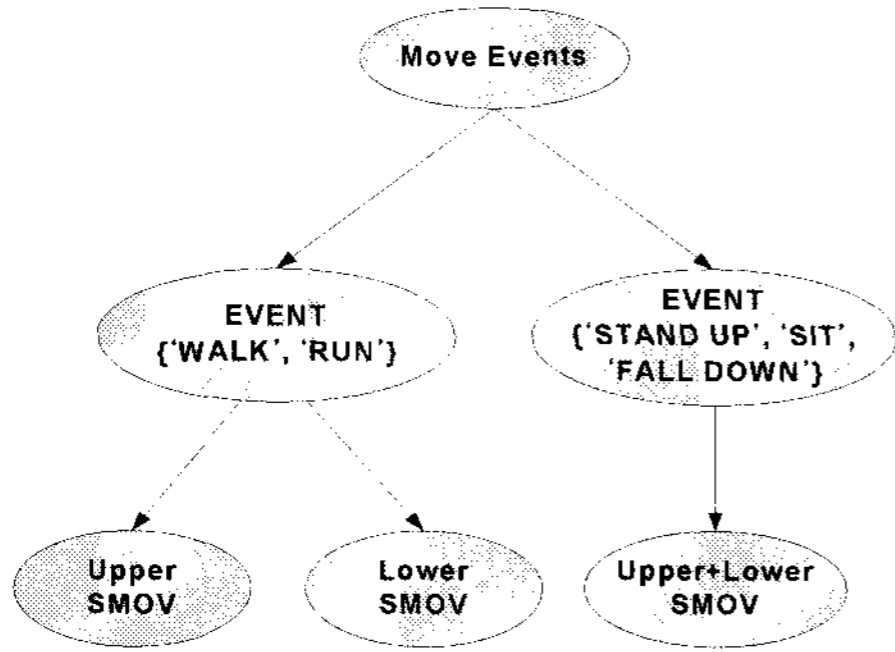


그림 5. 움직임-이벤트를 구분하기 위한 베이저안 모델

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 객체의 현재 분할 움직임 방향 벡터를 증거로 하여 해당 객체의 이벤트를 추정한다. 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 (27)과 같이 정의한다.

$$\begin{aligned}
 Bel(E_j) &= P(E_j | SMOV_i) \\
 &= \frac{P(SMOV_i | E_j)P(E_j)}{P(SMOV_i)} \quad (27)
 \end{aligned}$$

식 (28)은 m 개의 분할 움직임 벡터에 대한 j 이벤트의 사후 확률 값을 구하는 식을 정의한 것이다.

$$\begin{aligned}
 Bel(E_j) &= \frac{\sum_{k=1}^m P(E_j | SMOV_k)}{m} \\
 &= \frac{\sum_{k=1}^m \frac{P(SMOV_k | E_j)P(E_j)}{P(SMOV_k)}}{m} \quad (28)
 \end{aligned}$$

따라서 n 개에 대한 모든 이벤트 E_1, E_2, \dots, E_n 은 식 (29)를 통해 사후 확률 값이 가장 높은 이벤트를 현재의 이벤트로 정의한다.

$$Bel(EVENT) = \max_{\forall (E_1, E_2, \dots, E_j, \dots, E_n)} Bel(E_j) (1 \leq j \leq n) \quad (29)$$

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 테두리 상자의 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 차이값을 구한다.

OCCLUDE_OBJS 이벤트를 가진 객체의 테두리 상자의 시작점을 $f(x,y)$ 라 하고, 해당 객체의 질감 템플릿 영상의 시작점을 $TTI(x,y)$ 라고 할 때 템플릿 매칭은 식 (30)과 같이 정의한다. 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_OBJS의 테두리 상자 밖에 존재하면 비교를 하지 않는다. 그리고 배경에 해당되는 부분은 0으로 계산한다.

$$\begin{aligned}
 & \text{Template}(f, TTI) \\
 & = \min_{\forall(m,n)} (f(x+m, y+m) \cdot TTI(i, j)) \\
 & = \min_{\forall(m,n)} \left(\frac{\sum_{i=0}^{M-1-N} \sum_{j=0}^{N-1} f(x+m+i, y+n+j) - TTI(i, j)}{255 \times M \times N} \right) \quad (30) \\
 & \quad -5 \leq m, n \leq 5
 \end{aligned}$$

3.4 긴급 상황 시나리오 정의 및 해석

본 논문에서 정의한 긴급 상황 시나리오는 실내 또는 실외 환경에서 발생하는 상황을 기본 움직임 이벤트를 바탕으로 독립 객체들의 이벤트 전이 상태를 체크하여 해당 객체의 긴급한 상황을 해석하고 이를 알려준다. 긴급 상황은 다음과 같이 독립 객체가 갑자기 쓰러지는 경우로 정의한다. 즉, 첫 번째는 객체가 길을 걷다가 갑자기 넘어지는 경우 바로 일어나지 않고 쓰러져 있다면 큰 부상이거나 지병으로 인한 긴급 상황이다. 두 번째는 객체가 걸침 현상 발생 후 한 객체는 갑자기 쓰러지고 다른 한 객체는 걸거나 뛰어가며 쓰러진 객체는 일정 시간이 지나도 일어나지 못하는 경우이다.

본 논문에서는 위와 같은 상황에 대한 긴급 상황 시나리오를 정의하고 현재 입력되는 영상에 대해 어떤 시나리오가 발생했는가를 추정하게 된다. 그리고 “넘어지다”라는 시나리오를 해석하기 위해 FALL DOWN 이벤트를 정의한다.

긴급 상황 시나리오는 독립 객체가 길을 걸거나 뛰어 갈 때 장애물 또는 사람과 부딪쳐서 넘어지는 경우, 또는 갑작스럽게 쓰러지는 경우 등 여러 가지 상황이 실내 또는 실외 환경에서 발생할 수 있다. “객체가 쓰러지다(FALL DOWN)”라는 시나리오는 독립 객체가 스스로 쓰러지는 경우와 다른 객체에 의해 쓰러지는 경우로 나눈다. 독립 객체가 쓰러지는 경우는 장애물에 의한 경우와 갑자기 지병 등에 의해 쓰러지는 경우로 나눌 수 있고, 다른 객체에 의해 쓰러지는 경우는 살인, 폭력 등에 의한 것으로 해석할 수 있다.

그림 6에서 타원은 기본 움직임 이벤트에 해당되고 녹색으로 표시된 것은 다른 객체에서 발생하는 이벤트가 된다. 그리고 타원이 점선인 경우에는 해당 상태인지를 확인하기 위한 특징 벡터로, 분할 움직임 방향 벡터, 테두리 상자, 그리고 쓰러진 후 객체의 움직임이 발생하기 전까지의 시간으로 나타낸다.

“객체 A”가 영상에 나타난 후 WALK/RUN 또는 STAND 상태로 전이한 상황에서 분할 움직임 방향 벡터가 아래쪽 방향 성분으로 나오고 테두리 상자의 수평 비율이 수직 비율보다 큰 경우 해당 객체는 FALL DOWN 이벤트로 전이한다. 동일한 경우로 다른 객체 B와 OCCLUDE 이벤트가 발생한 후 MEET 상태에서 분할 움직임 특징 벡터와 테두리 상자의 비율이 FALL DOWN의 조건에 맞으면 “객체 A”는 “쓰러지다”로 판단된다. FALL DOWN 이벤트가 발생한 시점부터 시간을 측정하여 임계치 내에 일어나는 경우에는 정상적인 것으로 판단하고, 그렇지 않을 때는 쓰러진 객체에 문제점이 있다고 보고 “위급 사항(emergency)”으로 판단하여 알람 등의 이벤트를 알려준다.

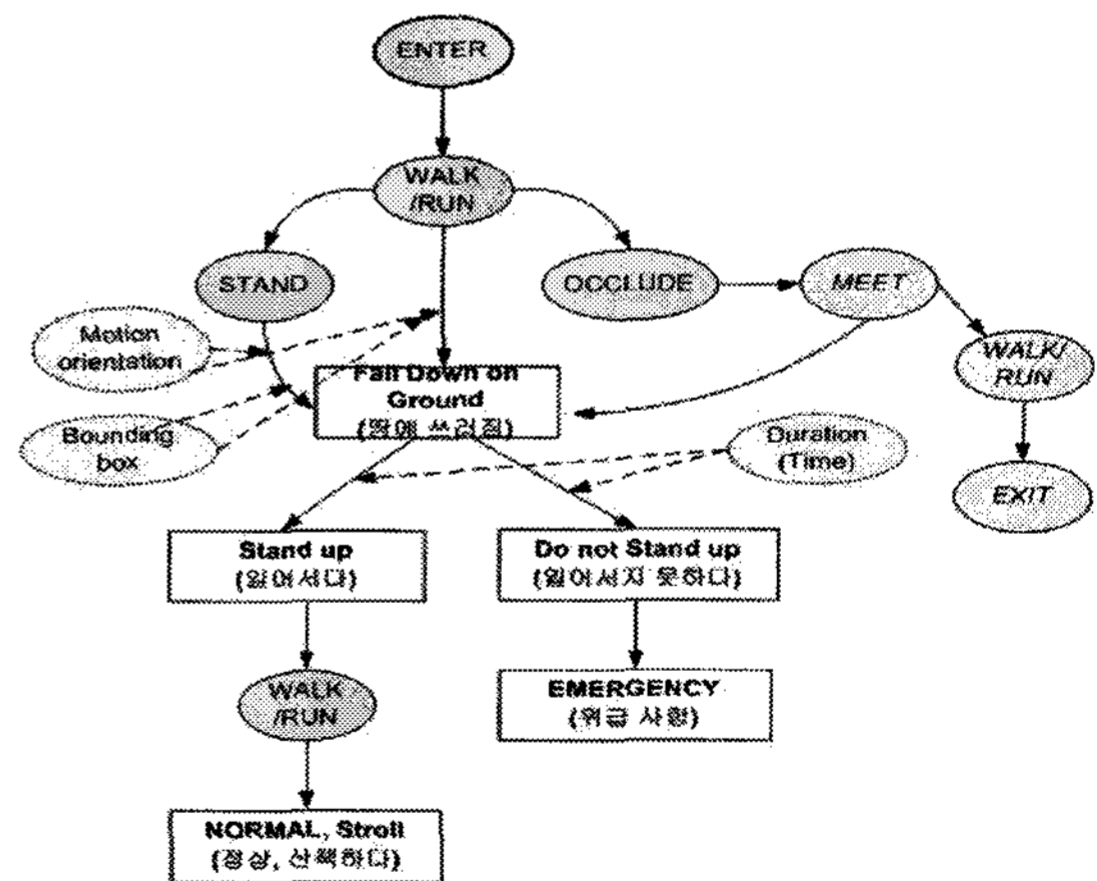


그림 6. “객체가 쓰러지다” 시나리오의 상태도

독립 객체의 이전 이벤트가 STAND 이고, 분할 움직임 방향 벡터의 사후 확률이 FALL DOWN 일 때, “해당 객체는 쓰러지다”의 이벤트가 발생한다. 이 확률을 구하기 위하여 앞서 설명한 베이지안 모델에서 자식 노드에 FALL DOWN에 대한 이벤트를 하나 추가하게 된다. 그리고 테두리 상자의 비율을 나타내는 $BoundRatio(O_i)$ 는 객체 O_i 의 테두리 상자 크기가

$M \times N$ 일 때 M/N 의 값을 반환한다. 따라서 객체가 STAND 또는 WALK/RUN 이벤트 일 때는 테두리 상자의 수직 길이가 수평 길이보다 크기 때문에 1보다 작은 값이 되지만, FALL DOWN에서는 수평 길이가 수직 길이보다 더 크게 나타나 1보다 큰 값이 나오게 된다. FALL DOWN 이벤트는 식 (31)과 같이 정의한다.

● 규칙 12. FALL DOWN 이벤트

$$EVENT(O_i^c)_p = STAND$$

$$EVENT\left(\max_{v(E_1, E_2, \dots, E_n)} Bel(E_j)\right)_C = FALL\ DOWN \quad (31)$$

$$BoundRatio(O_i^p) < BoundRatio(O_i^c)$$

4. 실험 결과

이벤트를 발생시키는 독립 객체의 단위는 사람이 고 정의된 이벤트는 독립 객체를 바탕으로 하기 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 600 프레임에서 1000 프레임으로 구성된다. 영상의 크기는 360×240(pixel)이고, 영상의 속도는 30 frame/sec이다. 실험 영상에 대해 독립 객체의 기본 움직임 이벤트 인식의 정확률을 보여주고, 움직임-이벤트인 WALK, RUN, SIT, STAND UP이 베이지안 네트워크를 통해 분할 움직임 특징 벡터가 어떤 확률 분포를 가지는가를 제시한다. 그리고 해당 객체의 긴급 상황 시나리오가 시간에 따라 어떤 이벤트를 발생하고 확률적으로 어떻게 분포되는지를 실험 결과를 통해 보여준다.

4.1 기본 움직임 이벤트 실험 결과

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 7은 기본 움직임 이벤트에 대한 정확률의 결과이다.

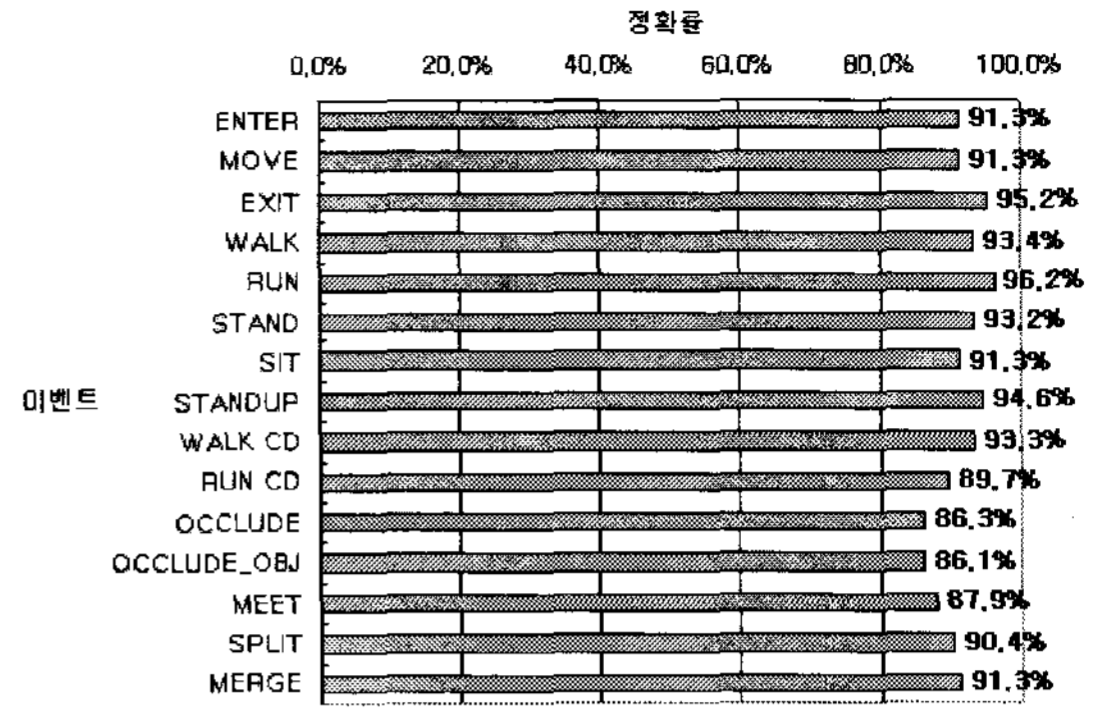


그림 7. 15개 실험 영상에 대한 이벤트 결과

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.

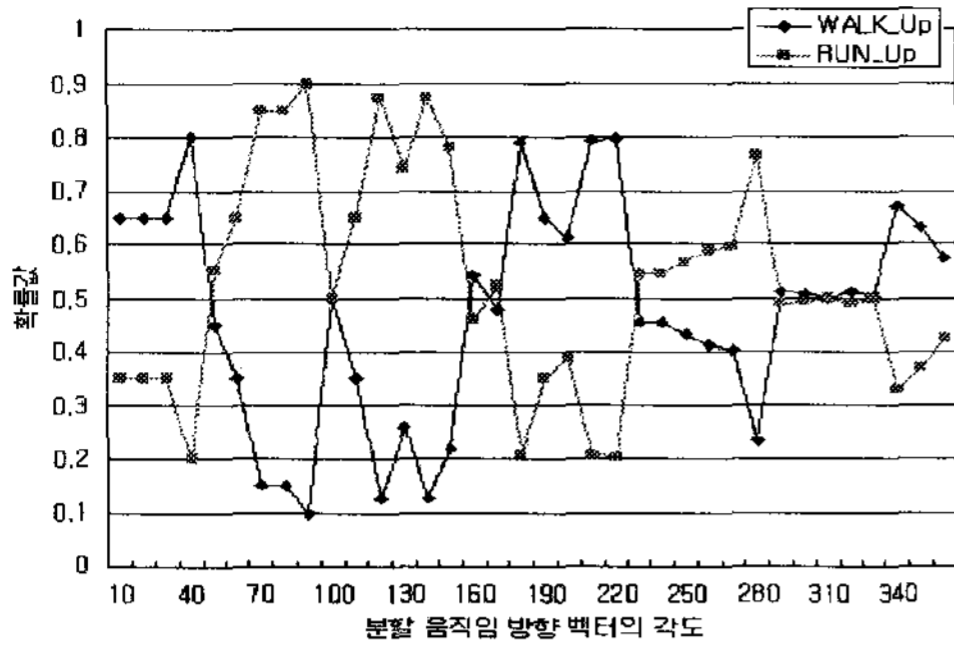
상호-이벤트인 SPLIT, MERGE, MEET가 주-이벤트와 움직임-이벤트보다 정확도가 떨어지는 원인은 실험 영상에서 실제 발생된 해당 이벤트의 빈도수가 전체의 20% 정도로 객체 인식 오류, 겹침 현상에 의한 객체 추적의 오류 등이 발생할 경우 움직임-이벤트에 비해 상대적으로 정확도가 떨어지기 때문이다.

4.2 긴급 상황 시나리오 실험 결과

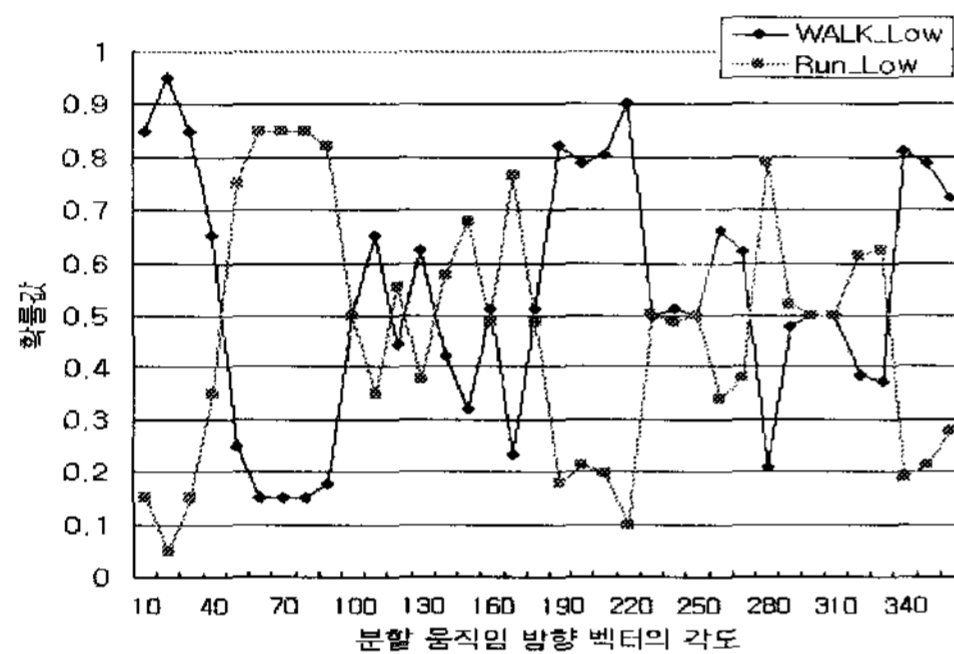
분할 움직임 방향 벡터의 확률은 10개 실험 영상을 베이지안 네트워크를 통해 학습하였다.

그림 8은 WALK와 RUN 이벤트에 대해 객체의 상체 부분과 하체 부분의 이벤트 사전 확률을 나타내고, 그림 9는 객체의 상체 부분과 하체 부분을 구분하지 않고 SIT/STAND UP/FALL DOWN 이벤트의 방향에 대한 확률을 보여준다.

그림 8에서 WALK 이벤트는 10~40°, 160~220°, 330~350°에서 RUN 이벤트보다 높은 확률이 나왔다. 그 이유는 WALK 이벤트의 경우 상체 또는 하체 부분의 분할 움직임 방향 벡터가 RUN 이벤트일 때 보다 움직이는 각도의 크기가 작기 때문이다. 즉



(a) 분할 움직임 방향 벡터의 각도에 따른 확률(하체 부분)



(b) 분할 움직임 방향 벡터의 각도에 따른 확률(상체 부분)

그림 8. 분할 움직임 방향 벡터의 각도에 따른 WALK/RUN의 확률

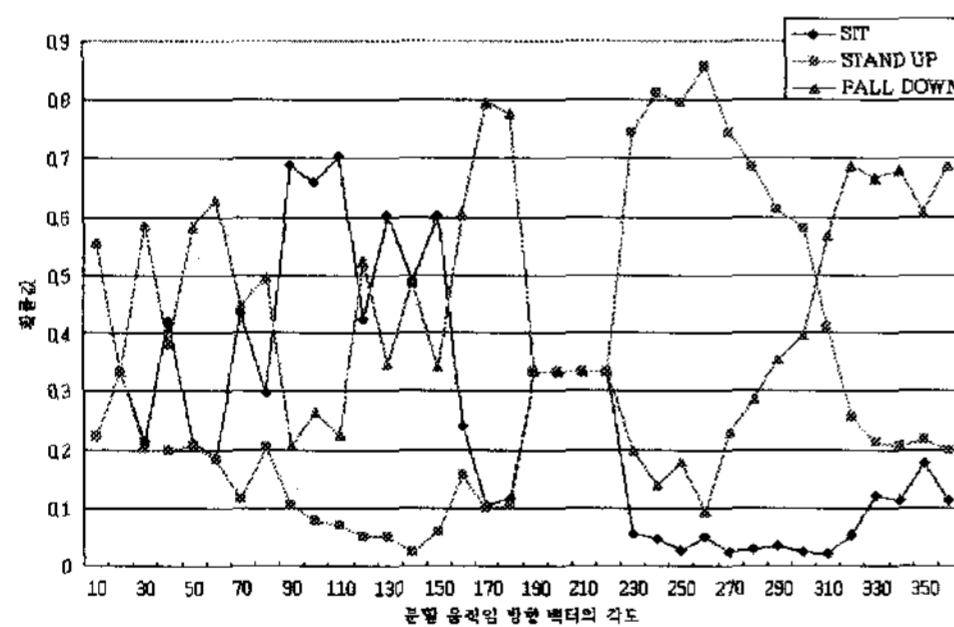


그림 9. 분할 움직임 방향 벡터의 각도에 따른 SIT/STAND UP/FALL DOWN의 확률

WALK에서 팔 또는 다리의 움직임 폭이 RUN 이벤트인 경우에 비해서 작다는 것이다. 또한 확률 값이 주기를 가지는 것은 객체가 좌측에서 우측으로 이동할 때의 분할 움직임 방향 벡터와 우측에서 좌측으로 이동할 때의 분할 움직임 방향 벡터가 180°정도의 주기를 가지기 때문이다.

“사람이 쓰러지다”의 시나리오가 정상적으로 해석되는가를 확인하기 위하여 실험하였다.

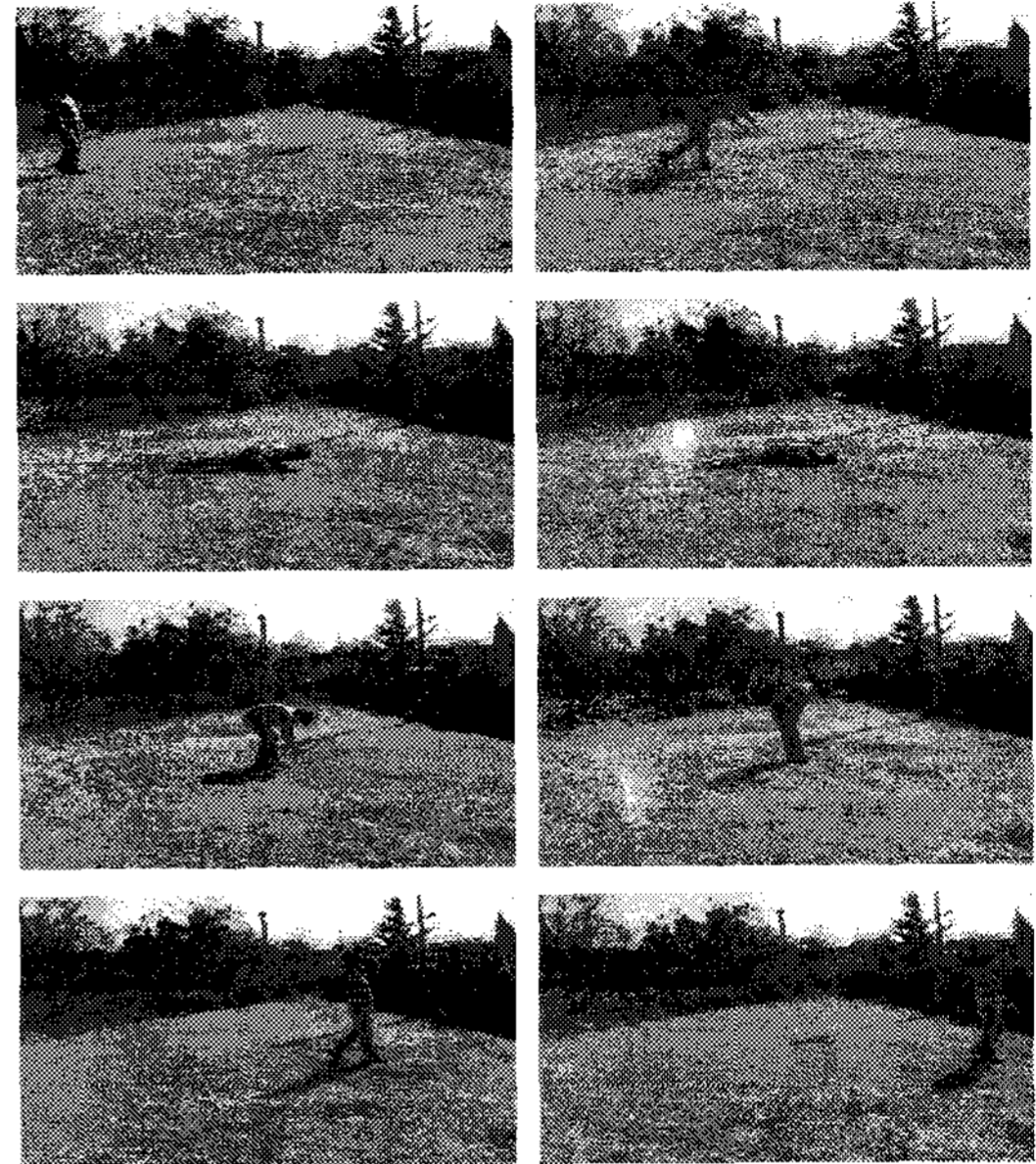


그림 10. 시나리오 해석을 위한 실험 영상의 키 프레임

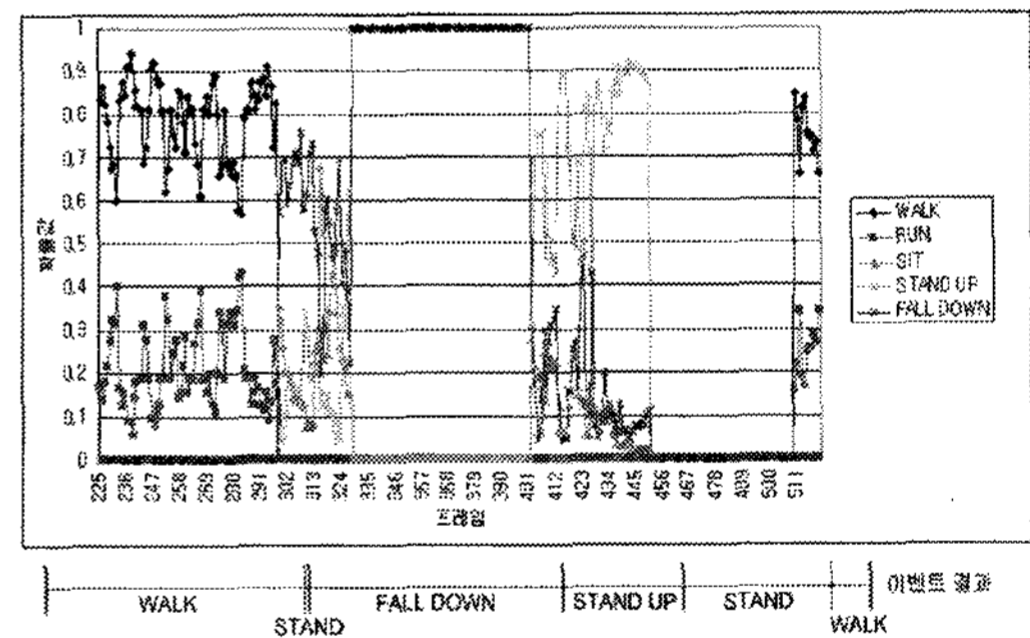


그림 11. 실험 영상에 대한 “객체 1”의 이벤트 사후 확률 결과

그림 10의 실험 영상은 객체가 걸어가다 넘어지는 것이다. 그림 11에서 “객체 1”은 290 프레임까지는 WALK 이벤트 상태로 진행되다가 STAND로 이벤트 전이가 발생한 후 325 프레임 전에서 넘어진다. 이 때 SIT, STAND UP, FALL DOWN 이벤트의 분할 움직임 방향 벡터의 사후 확률은 SIT와 FALL DOWN 이 높게 나타났지만 테두리 상자의 변화가 FALL DOWN 이벤트 규칙에 적합하기 때문에 FALL DOWN 이벤트로 전이가 되었다. 그리고 400 프레임 이후에 “객체 1”은 STAND UP 이벤트가 되고 510 프레임 이후에는 다시 WALK 이벤트로 전이가 되었다. 따라서 실험 영상은 “정상적으로 산책하다”라는 시나리오로 해석이 된다.

4.3 기존 방법과의 비교 분석

본 논문에서 제안한 독립 객체의 이벤트 인식과 긴급 상황 시나리오 해석 방법이 기존 제시된 방법들과 비교 분석하였다.

표 2에서 기호 은 해당 방법이 이벤트의 상태 전이 또는 시나리오 해석을 지원한다는 의미를 가진다. 표 2에서 보듯이 본 논문에서 제안한 이벤트 인식은 실내 또는 실외에서 가능하고 규칙 기반과 확률 기반을 동시에 사용함으로써 연산 시간의 감소와 이벤트 인식의 정확도를 높였다. 또한 이벤트 인식 과정에서 상태 전이를 정의함으로써 해당 객체의 다음 이벤트를 인식하는데 빠르고 정확하게 수행할 수 있다.

표 2에서 시나리오 해석은 [9]와 [12]만이 가능함으로 시나리오 해석에 대한 분야의 연구가 이벤트 인식에 비해 상대적으로 적음을 알 수 있다. 그러나 두 방법은 시나리오 해석을 위해 이벤트를 정의하였고, 시나리오의 상태 전이가 미리 정의된 순서와 일치하지 않으면 정확한 해석이 불가능한 문제점이 있었다. 이것은 정의된 시나리오와 이벤트가 상당한 의존 관계를 가지고 있기 때문이다. 제안한 시나리오 해석 방법은 FALL DOWN 이벤트를 제외한 기본 움직임 이벤트가 시나리오와 독립적으로 인식되기 때문에 다양한 시나리오 해석으로의 확장이 용이한 장점을 가진다.

표 2. 방법들의 환경 및 이벤트 인식과 시나리오 해석 여부

제안 방법	지역	이벤트			시나리오	
		개수	접근 방법	상태 전이	해석 여부	이벤트 관계
A. Amer et al.[8]	실내 실외	20	규칙 기반	<input type="checkbox"/>	<input type="checkbox"/>	해당 사항 없음
I. Ersoy et al.[9]	실외	5	규칙 기반	<input type="checkbox"/>	<input checked="" type="checkbox"/>	의존적
S. Guler et al.[11]	실외	7	확률 기반	<input checked="" type="checkbox"/>	<input type="checkbox"/>	해당 사항 없음
S. Hongeng et al.[12]	실외	11	확률 기반	<input type="checkbox"/>	<input checked="" type="checkbox"/>	의존적
R. Cucchiara et al.[14]	실내	4	확률 기반	<input checked="" type="checkbox"/>	<input type="checkbox"/>	해당 사항 없음
W ⁴ [15]	실외	4	규칙 기반	<input type="checkbox"/>	<input type="checkbox"/>	해당 사항 없음
R. V. Babu et al.[18]	실내	5	확률 기반	<input type="checkbox"/>	<input type="checkbox"/>	해당 사항 없음
제안한 방법	실내 실외	16	규칙 기반 확률 기반	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	독립적

5. 결 론

본 논문에서는 객체로부터 특징 벡터를 추출하고 각 객체의 행동 양식을 분석함으로써 현재 객체가 어떤 이벤트를 발생했는가를 인식하고, 확률 모델을 통해 긴급한 상황에서의 시나리오를 해석할 수 있는 방법을 제안하였다. 실험 결과 이벤트 인식의 정확도는 주-이벤트와 움직임-이벤트는 90% 이상이었고, 상호-이벤트는 85% 이상의 정확도를 보였다. 그리고 분할 움직임 방향 벡터와 속도 정보는 움직임-이벤트를 구분하는데 필요한 특징 벡터였고 베이지안 네트워크의 사전 학습 결과와 사후 결과를 통해 이를 확인하였다. “사람이 쓰러지다”, “강도를 당하다”의 긴급 상황 시나리오의 실험 결과는 독립 객체의 이벤트와 사후 확률을 이용하여 현재의 객체 상태를 해석할 수 있었다. 본 논문에서 제안한 방법은 객체의 의미론적 해석이 가능하고 다양한 비전 감시 시스템의 의미론적 해석 분야에 적용이 가능할 것으로 기대한다.

참 고 문 헌

[1] ITU-T Recommendation H.263, *Video Coding for Low Bit Rate Communication*, Ver. 2, 1998.

[2] ISO/IEC 13818, *Information Technology: Generic Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s (MPEG-1)*, 1993.

[3] ISO/IEC 14496-2, Amendment 1, *Information Technology - Coding of Audio-Visual Objects - Part 2: Visual (MPEG-4 Visual)*, 2001.

[4] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC Video Coding Standard,” *IEEE Transaction on Circuits and System for Video Technology*, Vol. 13, No.7, pp. 560-576, 2003.

[5] ISO/IEC JTC1/SC29/WG11, “Survey of Algorithms Used for Multi-view Video Coding (MVC),” *WG11 document N6909*, Hong Kong, China, Jan. 2005.

[6] W. Hu, T. Tan, L. Wang, and S. Maybank, “A Survey on Visual Surveillance of Object Motion and Behaviors,” *IEEE Transaction on Systems,*

Man and Cybernetics Part C, Vol.34, Issue.3, pp. 334-352, 2004.

[7] T. B. Moeslund and E. Granum, "A Survey of Computer Vision-based Human Motion Capture," *Computer Vision and Image Understanding*, Vol.81, No.3, pp. 231-268, 2001.

[8] 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.

[9] I. Ersoy, F. Bunyak, and S. R. Subramanya, "A Framework for Trajectory Based Visual Event Retrieval," *Proceeding of the International Conference on Information Technology: Coding and Computing*, Vol.4, No.4, pp. 42-48, 2003.

[10] S. H. Kang, J. W. Lee, J. H. Cho, and S. H. Kim, "A Determining Method of Factors for Effective Real Time Background Modeling," *The 6th Asia Pacific International Symposium on Information Technology (APIS)*, Jan. 2007.

[11] S. Guler, W. H. Liang, and I. A. Pushee, "A Video Event Detection and Mining Framework," *Second IEEE Workshop on Event Mining*, Vol. 4, No.4, pp. 42-48, 2003.

[12] S. Hongeng, R. Nevatia, and F. Bremond, "Video-Based Event Recognition: Activity Representation and Probabilistic Recognition Method," *Computer Vision and Image Understanding*, Vol.96, pp. 129-162, 2004.

[13] G. Medioni, I. Cohen, F. Bremond, and R. Nevatia, "Event Detection and Analysis from Video Streams," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol.23, No. 8, pp. 873-889, 2001.

[14] 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.

[15] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Real-Time Surveillance of People and Their

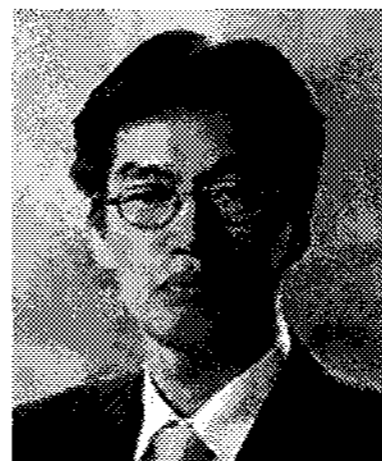
Activities," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol.22, No. 8, pp. 809-830, 2000.

[16] V. Tzouvaras, G. Tsechpenakism G. Stamou, and S. Kollias, "Adaptive Rule-Based Recognition of Events in Video Sequences," *Proceeding of the International Conference on Image Processing*, Vol.2, pp. 607-610, 2003.

[17] K. B. Korb and A. E. Nicholson, *Bayesian Artificial Intelligence*, A CRC Press Company, New York, 2004.

[18] 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.

[19] 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.



이 준 철

1988년 경북대학교 통계학과 졸업(이학사)
 1991년 경북대학교 컴퓨터공학과 졸업(공학석사)
 2007년 경북대학교 컴퓨터공학과 졸업(공학박사)
 1995년~현재 부천대학 인터넷과 부교수

관심분야 : 감시시스템, 멀티미디어, 다시점 동영상



최 창 규

1998년 대구대학교 전자계산학과 졸업(공학학사)
 2000년 경북대학교 컴퓨터공학과 졸업(공학석사)
 2006년 경북대학교 컴퓨터공학과 졸업(공학박사)
 2006년~현재 (주)테크바일 개발팀장

관심분야 : 멀티미디어, 무선 프로토콜, 모바일