creative commons

COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

[UCI]I804:24011-200000234386

2007년 8월

박사학위 논문

비디오 데이터의 객체 움직임 인식을 위한 의미적 이벤트 검출

조선대학교 대학원

전자계산학과

신 주 현

비디오 데이터의 객체 움직임 인식을 위한 의미적 이벤트 검출

Semantic Event Search
for Object Motion Recognition of Video Data

2007년 8월 일

조선대학교 대학원

전자계산학과

신 주 현

비디오 데이터의 객체 움직임 인식을 위한 의미적 이벤트 검출

지도교수 김 판 구

이 논문을 이학박사학위신청 논문으로 제출함.

2007년 4월

조선대학교 대학원

전자계산학과

신 주 현

신주현의 박사학위논문을 인준함

위원장　조선대학교 교수 _____

위　원　조선대학교 교수 _____

위　원　호남대학교 교수 _____

위　원　전남대학교 교수 _____

위　원　조선대학교 교수 _____

2007년　6월

조선대학교 대학원

목 차

ABSTRACT

Ⅰ. 서 론 ····· 1

A. 연구배경 및 필요성 ····· 1

B. 연구내용 및 범위 ····· 4

C. 논문의 구성 ····· 6

Ⅱ. 관련 연구 ····· 7

A. 비디오 데이터의 의미 정보 분석 ····· 7

1. 시각적 특징 정보에 의한 의미정보 분석 ····· 7

2. 시·공간 특징에 의한 의미정보 분석 ····· 10

B. 객체 움직임에 대한 의미적 모델링 ····· 15

1. 시·공간 관계 기반의 저차원 특징과 개념적 통합 ····· 16

2. 온톨로지를 이용한 움직임 표현 ····· 20

Ⅲ. 객체 움직임과 이벤트 표현을 위한 모델링 ····· 28

A. 객체 움직임 의미 인식을 위한 프레임웍 ····· 28

B. 제안된 객체 움직임 모델링 ····· 31

1. 비디오 데이터의 개념적 구조 ····· 31

2. 동사 어휘와 객체 움직임에 대한 정의 ····· 32

3. 객체 움직임과 이벤트 표현을 위한 관계 구조 ····· 35

Ⅳ. 비디오 데이터의 저차원 처리 ····· 37
A. 비디오 데이터의 전처리 ····· 37
1. Shot 검출 ····· 38
2. 움직임 객체에 대한 특징값 추출 ····· 46
3. 메타데이터 생성 ····· 53
B. 제안된 움직임 인식을 위한 전처리 과정 ····· 56
1. Shot 분할과 키프레임 추출 ····· 56
2. 움직임 객체의 특징값 추출 ····· 57

Ⅴ. 제안된 모델을 이용한 움직임 인식과 이벤트 검출 ····· 60
A. 객체 움직임 인식을 위한 규칙 ····· 60
1. 객체 위상관계에 의한 움직임 규칙 정의 ····· 60
2. 이벤트 검출을 위한 규칙 ····· 68
B. 움직임 객체에 대한 규칙 적용 ····· 70
1. 정의된 규칙에 의한 시뮬레이션 ····· 70
2. 움직임 인식에 의한 비디오 데이터 인덱싱 ····· 72
C. Ontology 적용 비디오 검색 기법 ····· 74
1. 이벤트 온톨로지 구축 ····· 74
2. 비디오 의미적 검색을 위한 OWL/RDF(S) 표현 ····· 77

Ⅵ. 실험 및 결과 분석 ····· 81
A. 실험 환경 ····· 81
B. 제안된 모델링을 이용한 이벤트 검출 ····· 83
1. 비디오 프레임별 움직임 인식 ····· 83

2. 이벤트 검출 ····· 92
C. 온톨로지 적용 이벤트 검색 ····· 94
D. 모델링 기법들의 비교 분석 ····· 96

Ⅶ. 결 론 및 향후 연구 ····· 98

[참 고 문 헌] ····· 100

그 림 목 차

(그림 2.1) RGB 컬러 3차원 입방체 ····· 8
(그림 2.2) 객체의 MBR ····· 10
(그림 2.3) Egenhofer의 8가지 공간 관계 ····· 11
(그림 2.4) Allen의 13가지 시간관계 ····· 12
(그림 2.5) 위상관계에 의한 개념적 표현 ····· 13
(그림 2.6) 시각적 특징과 시·공간 관계에 대한 통합 프로세스 ····· 13
(그림 2.7) 시각 객체의 위상관계 표현 ····· 14
(그림 2.8) COBRA 비디오 데이터 모델의 계층구조 ····· 17
(그림 2.9) 추상적 개념의 비디오 데이터 모델 ····· 18
(그림 2.10) 축구 비디오의 “Penalty Goal” 장면 ····· 19
(그림 2.11) 지식기반의 상호 매핑 구조 ····· 21
(그림 2.12) 움직임 동사의 계층 구조 ····· 22
(그림 2.13) 워드넷(WordNet) 모션 동사에 대한 계층 구조 ····· 23
(그림 2.14) 당구 비디오의 온톨로지 구축을 위한 의미적 표현 ····· 25
(그림 2.15) Protégé 3.1에서 당구 비디오 온톨로지 구축 화면 ····· 26
(그림 2.16) 당구 온톨로지 OWL 파일 ····· 27
(그림 3.1) 객체 움직임 의미인식을 위한 프레임웍 ····· 30
(그림 3.2) 비디오의 개념적 계층 구조 ····· 32
(그림 3.3) 정지영상내의 객체관계 ····· 33
(그림 3.4) 연속적인 프레임에 대한 움직임 ····· 33
(그림 3.5) 기본적인 공관 관계 ····· 34
(그림 3.6) 움직임 동사의 계층구조 ····· 35
(그림 3.7) 비디오 Shot의 이벤트 모델링 ····· 36
(그림 4.1) 비디오 데이터의 전처리 과정 ····· 38

(그림 4.2) 프레임 간 히스토그램 차이 ····· 40
(그림 4.3) 움직임 벡터를 이용하여 이전 영상으로부터 DC 값 복원 ····· 42
(그림 4.4) MPEG의 프레임 시퀀스 ····· 43
(그림 4.5) foreman 영상의 움직임 벡터 ····· 47
(그림 4.6) 움직임 벡터의 2차원 히스토그램 ····· 48
(그림 4.7) 매크로 블록 단위 레이블링 ····· 49
(그림 4.8) 입력영상에 대한 레이블링 한 영상 ····· 50
(그림 4.9) 각 영역별로 번호가 지정되어 있는 탁구 영상 ····· 50
(그림 4.10) 객체 영상 중에 선택된 객체만 나타난 영상 ····· 51
(그림 4.11) 움직임 객체 특징값 추출을 위한 전처리 과정 ····· 56
(그림 4.12) Shot 분할과 프레임 추출 ····· 57
(그림 4.13) 움직임 객체의 특징값 추출 과정 ····· 59
(그림 5.1) 규칙정의를 위한 객체의 MBR 표현 ····· 62
(그림 5.2) 단일 프레임에 대한 움직임 인식 ····· 71
(그림 5.3) 연속프레임에 대한 움직임과 이벤트 검출 ····· 72
(그림 5.4) 비디오 데이터 인덱싱 구조 ····· 73
(그림 5.5) 비디오 Shot 데이터의 인덱싱 예 ····· 73
(그림 5.6) 이벤트 온톨로지 구조 ····· 75
(그림 6.1) 실험 인터페이스 ····· 82
(그림 6.2) Enter 이벤트를 포함한 Shot내의 대표 프레임 ····· 84
(그림 6.3) Shot3에 대한 객체 움직임 위치값 추출 과정 ····· 85
(그림 6.4) Shot3에 대한 단일 프레임 움직임 인식 ····· 85
(그림 6.5) Shot3에 대한 연속 프레임 움직임 인식 ····· 86
(그림 6.6) Shot5에 대한 객체 움직임 위치값 추출 과정 ····· 87
(그림 6.7) Shot5에 대한 단일 프레임 움직임 인식 ····· 87
(그림 6.8) Shot5에 대한 연속 프레임 움직임 인식 ····· 88
(그림 6.9) Shot7에 대한 객체 움직임 위치값 추출 과정 ····· 89

(그림 6.10) Shot7에 대한 단일 프레임 움직임 인식 ····· 89
(그림 6.11) Shot7에 대한 연속 프레임 움직임 인식 ····· 90
(그림 6.12) 비디오 Shot의 프레임별 움직임 인식 정확률 그래프 ····· 91
(그림 6.13) 비디오 Shot에 대한 이벤트 검출 정확률 그래프 ····· 93
(그림 6.14) 검색 시스템 구성도 ····· 94
(그림 6.15) 온톨로지 기반의 비디오 Shot 검색 결과 ····· 96

표 목 차

(표 2.1) 객체 사이의 VST 관계 ····· 15
(표 2.2) “패널티 골”에 대한 시·공간 관계 규칙 정의 ····· 19
(표 2.3) Visual information에 대한 의미관계 매핑 ····· 23
(표 3.1) 워드넷의 공간관계 동사 분류 ····· 35
(표 4.1) MPEG 비디오의 메타데이터 형식 ····· 55
(표 4.2) 객체 움직임에 대한 MBR 추출 알고리즘 ····· 58
(표 5.1) 단일프레임에 대한 객체 관계에 의한 움직임 규칙 ····· 62
(표 5.2) 단일프레임 규칙에 대한 알고리즘 ····· 64
(표 5.3) 연속프레임에서 인식 가능한 움직임 규칙 ····· 65
(표 5.4) 연속 프레임 규칙에 대한 알고리즘 ····· 67
(표 5.5) 이벤트 규칙에 대한 알고리즘 ····· 69
(표 5.6) 이벤트에 대한 도메인별 동사 수집 ····· 76
(표 5.7) 이벤트에 대한 OWL/RDF(S) 표현 ····· 78
(표 5.8) 비디오 Shot에 대한 클래스 정보 ····· 79
(표 5.9) 비디오 Shot에 대한 이벤트를 위한 OWL/RDF(S) ····· 80
(표 6.1) 프레임별 움직임 인식 실험 결과 ····· 83
(표 6.2) 비디오 Shot에 대한 프레임 단위의 움직임 인식과 실패 ····· 91
(표 6.3) 이벤트에 대한 연속된 움직임 패턴 ····· 93
(표 6.4) 기존 연구와의 비교 분석 ····· 97

ABSTRACT

Semantic Event Search for Object Motion Recognition of Video Data

Shin, Ju-Hyun
Advisor: Prof. Kim, Pan-Koo, Ph. D.
Department of Computer Engineering
Graduate School of Chosun University

For semantic search of video data, semantic event representations of object motion are being actively studied. However, the existing studies are mostly about content-based video search using low-level features and predefining notes for the video data, and inadequate to interpret and process the conceptual high-level motion of human beings. In this paper, relationship between objects is represented by a hierarchical structure of linguistic concepts to provide semantic representations of video events, and a method that maps high-level motion information, which is for semantic shot detections of video, to low-level features of the video is suggested.

In this paper, first, conceptual relationship between verbs is represented using a hierarchical structure to represent motion event of object in the video shots. Event verbs are located on top of the hierarchical structure, and verbs for each event are arrayed as instances. Second, rules of the intermediate

level that links low-level features to high-level motion information are defined to detect concept-based shots from video images. For that reason, characteristic values of moving objects are extracted throughout the video data processing. Motion recognition rules suggested in this paper define primary rules for spatial relationship using locations of still objects and moving objects, and, for recognition of detailed actions, motion of each consecutive frame is recognized by distances and directions between objects. In this way, events represented by shots are extracted, and, consequently, indexing of the shots is carried out by analyzing consecutive patterns of the recognized motion. This is to allow users to use indexes when they want to search videos related to certain events.

Third, when searching events, ontology is established to semantically search events which are represented differently depending on application. The lexical relationship between video shots and events, indexed for the final ontology search, is described with the standard language of the semantic web OWL/RDF(S). his is to provide semantic search and inference using characteristics of IS-A relationship and synonyms in ontology, and convenient search services for users. For semantic event detection of video data suggested in this paper, motion recognition experiments of object have been carried out in two ways. The average precision rate of motion recognition for each shot was 85% in the test with the modelling suggested and precision rate for event detection was 80%.

Ⅰ. 서 론

A. 연구 배경 및 목적

현재 우리는 언제 어디서나 자유롭게 접근, 처리, 이용할 수 있는 생활 속의 컴퓨팅 시대를 살고 있다. 이는 컴퓨터의 급속한 발전과 초고속통신망의 빠른 보급의 결과이며 이로 인해 미래학자들의 예견보다 더욱 빨리 미래의 컴퓨터가 현실화 되고 있다. 이에 발맞춰 멀티미디어 시대의 도래도 고속화되었으며, 멀티미디어 검색 시스템의 기술들 또한 급속한 발전을 이루고 있다.

그리고 인터넷과 웹의 발전으로 이러한 대량의 멀티미디어 데이터가 온라인상에 존재하고 있으며, 이러한 방대한 멀티미디어 데이터를 쉽고 편리하게 검색하고자 하는 사용자의 요구가 매우 증가하고 있다. 더욱이 유비쿼터스 패러다임(ubiquitous paradigm)의 출현으로 사용자 중심의 컴퓨팅 기술의 발달과 요구가 급증하고 있어 사용자의 요구에 따른 검색을 위해 많은 분야에서 다양한 연구가 진행되어지고 있으며, 연구되는 기술들의 진보로 본 분야는 성장속도의 가속화를 이루고 있다.

멀티미디어의 형태 중 비디오 데이터는 많은 관심과 연구의 대상이 되고 있으며, 다른 멀티미디어 데이터들의 모든 정보를 시간의 흐름을 통해 하나의 정보로 묶고 있다. 그리고 나날이 늘어나는 비디오 데이터에 대한 효과적인 검색을 위한 관리와 처리 방법들의 기술 개발이 시급하다. 특히, 전자 도서관, 디지털 방송, 주문형 비디오 시스템, 의료 정보 시스템, 원격교육, 의미적 비디오 정보 검색 등의 다양한 분야에서 디지털 비디오 처리 시스템의 성능 향상 및 분석을 위한 많은 방법론이 연구되고 있다. 뿐만 아니라 최근 UCC(user created content)의 출현으로 여러 분야에서 비디오 데이터들이 대량으로 생성되고 있으며, 사용자 요구에 따른 비디오 데이터를 저장, 처리, 인덱싱하여 의미적으로 검색하는 기술들의 필요성이 더욱더 크게 대두되고 있다.

기존의 멀티미디어 데이터 검색 방법 중 가장 대표적인 방법은 내용 기반의 검색 방법이다. 비디오 데이터는 구조가 매우 복잡하고, 시간 흐름의 특성을 고려하기

때문에 현재의 기술은 주로 색상, 질감, 모양, 궤적 등의 저차원 특징을 이용한 방법들의 연구가 활발히 진행되고 있다. MPEG-7에서는 비디오 데이터 검색을 위해 색상, 질감, 모양 및 객체들의 공간적 위치 관계(spatial location relationship) 등의 특징을 이용하여 시간적 속성을 고려한 처리 기술을 위해 비디오 분할(video segmentation), 특징 추출(feature extraction), 색인(indexing), 질의(query) 및 검색(retrieval) 등의 기술을 이용한다. Swain은 히스토그램 인터섹션 기법을 이용하였고[1], Huang은 컬러 코렐로그램을 제안하여 효율적인 영상 검색 기법을 고려하였다[2,3]. 그리고 Pass는 영상 내 화소(pixcel)간의 인접성을 특징으로 이용하였다[4]. 이렇듯 많은 내용 기반 검색 연구들은 저차원 특징들을 고려한 검색 방법들이 주를 이룬다.

그러나 이러한 저차원의 특징을 이용한 내용 기반의 검색 기술 방법은 시간의 흐름을 갖는 비디오 데이터의 의미적 내용을 정확하게 표현하기 어렵고, 검색을 위한 의미적 정보를 인식하는데도 한계가 있어 사용자의 요구를 분석하고 사용자가 원하는 의미 기반의 검색 방법으로 미흡한 실정이다. 게다가, 시간적 흐름의 특성이 고려된 시공간적인 특성의 정확한 파싱이 힘들고, 비디오의 의미적 분할 및 검색 등에 대한 접근이 매우 어렵다.

위의 미흡점을 보완하기 위해 비디오 데이터 내에 포함된 의미 정보를 추출(extraction), 표현(representatin), 인식(recognition) 및 처리(processing)할 수 있는 고차원 수준의 의미 분석 방법론이 출현하기 시작하였다. 비디오 데이터에 포함된 의미 정보를 이해하기 위한 특징으로 기존의 색상, 질감, 형태, 궤적 등의 저차원 특징 정보뿐만 아니라 시간 흐름을 가진 비디오 내의 구성 요소인 객체들의 이벤트(event), 움직임(motion) 등의 고차원의 의미 정보를 추출하고 인식할 수 있는 메커니즘이 필요하다.

이러한 메커니즘 중 객체 움직임의 의미적 인식을 위한 연구로 시간 논리(temporal logic)[5], 구간 대수(interval algebra)[6] 등을 이용한 논리 기반(logic-based) 방법과 유한상태머신(FSM : finite state machine), 베이지안 네트워크(bayesian networks), HMM(hidden markov model)[7]을 이용한 상태 머신(status machine) 방법 등을 이용해 이벤트를 표현한다. 또한, 현재 멀티미디어 데이터의 내용기반 검색을 위한

표현 방식에 관한 표준안으로 만들어진 MPEG-7[8]에서는 저차원의 특징뿐만 아니라 시·공간적 관계 표현, 이벤트 검출에 이르기까지 의미적 인식을 위한 시도가 있었으나 이는 메타데이터(metadata) 형태로 멀티미디어 객체에 대해 단순히 키워드를 부여하는 방법으로 그 내용을 표현하는 정도로, 진정한 의미적 내용을 표현하고 이해하는 것은 현재로서는 불가능하다.

따라서 비디오 데이터 내의 개념 정보를 의미적으로 인식하기 위해서는, 비디오 내의 고차원 정보인 객체 이벤트의 의미적 검출과 움직임의 인식을 위한 개념적 표현 방법이 필수적이다. 지금까지 개발된 객체 움직임의 분리·추적 기술 등을 토대로 객체들 간의 시·공간적 특징기반의 이벤트의 표현을 위해 인간의 의사표현 수단인 언어를 이용하여 이벤트를 모델링하고 온톨로지(ontology)를 구축한다. 그리고 비디오 데이터의 저차원 특징 추출을 이용하여 고차원 정보와 저차원 특징의 매핑을 위한 규칙기반(rule based)의 중간계층(intermediate level)을 정의한다. 이와 같이 중간 계층를 정의함으로써 비디오 데이터의 자동 나레이션 생성 및 의미 기반 검색에 응용할 수 있음을 기대한다. 이러한 객체의 움직임 이벤트들을 의미적으로 인식할 수 있도록 제시하고, 다시 말하면, 디지털 비디오 데이터 내용 중 움직임 객체에 대해서 시간 흐름에 따른 언어 형태로 표현하여 비디오 내용의 의미적 인식기술을 제안한다.

B. 연구 내용 및 범위

본 논문의 연구 내용은 비디오 데이터의 객체 움직임 인식과 이벤트 검출을 위한 움직임 모델링에 관한 것으로서 비디오 내의 객체 움직임 인식을 위해 인간의 언어가 가진 어휘의 의미적 관계를 이용하여 움직임 모델을 제시하고, 이를 비디오의 저차원 특징과 매핑함으로써 사용자의 관심도가 높은 이벤트를 검출함으로써 의미적 검색에 활용할 수 있는 방안을 연구하였다.

먼저, 비디오내 객체 움직임에 대한 의미인식을 위해 움직임 모델링 기법을 제안한다. 제안하는 모델링 방법은 비디오내 움직임 객체와 정지 객체 사이의 공간관계를 connect, disconnect, overlap, contain 등 4가지 형태로 분류하고, 인간이 사고 할 수 있는 언어적 지식을 이용하여 객체 움직임에 대한 개념관계를 정의한다. 제안하는 기법은 인간의 인지적이고 개념적 입장을 고려한 의미적인 해석이자 표현 방법으로 비디오내의 객체 움직임을 대표할 수 있는 이벤트(event) 단위로 움직임 동사(motion verb)간의 관계를 정의한다. 비디오 정보의 이벤트는 객체간의 공간 관계(spatial relation), 거리(distance), 객체의 움직임 경로(moving path) 등의 특징적인 변화에 따라 각각 다르게 인지되고 해석되어진다. 따라서 다중 객체(multiple object)간의 시간적(temporal), 공간적(spatial) 움직임에 대한 의미적 표현을 정의한다. 하나의 이벤트는 연속된 시.공간내의 객체간의 움직임으로 구성되고 이러한 의미적 관계를 계층적 모델로 표현하여 의미인식이 가능하게 한다.

다음으로 제안하는 움직임 모델링 기법을 통해 사용자가 의미적 검색을 하기 위해서는 비디오 데이터의 저차원적 특징과 매핑할 수 있는 방안이 필요하다. 이를 위해 비디오 개념 구조상의 프레임을 중심으로 움직임 객체의 위치값(position value)을 기준으로 움직임 규칙을 제안했다. 움직임 객체의 위치값 추출을 위해서는 비디오 데이터의 저차원 처리단계가 필요하다.

먼저 Shot 분할과 프레임 추출을 위해 컬러 히스토그램 변화, DCT 계수 기법을 본 연구를 위한 비디오 영상의 Shot 분할을 위해 활용하였다. 움직임에 대한 규칙정의를 위해 비디오 영상의 키프레임을 추출하여 프레임에 존재하는 다중 객체의 위치값과

시간, 프레임번호에 대해 데이터베이스를 생성한다. 생성된 데이터베이스로부터 각각의 프레임에 위치하는 두 객체가 가지는 좌표값의 차를 구한다음 차에 대한 일정한 거리를 계층모델의 최하위에 위치하는 인스턴스와 의미적으로 매핑할 수 있도록 규칙을 정의한다.

다음으로 사용자가 보고자 하는 비디오는 최소한 Shot 단위의 동영상이기 때문에 각각의 프레임에 대한 움직임 패턴을 통해 이벤트로 인식될 수 있는 규칙이 필요하다. 따라서 인스턴스에 정의한 각각의 규칙에 대해 연속된 패턴을 분석하여 이벤트 규칙을 재정의함으로써 비디오 Shot에 대한 최종 이벤트를 추출한다. 이와 같이 추출된 이벤트는 온톨로지[21]를 이용한 비디오 데이터의 의미적 검색을 위해 Shot 단위로 인덱싱을 수행한다.

온톨로지를 이용한 비디오 데이터의 의미적 검색을 위해 meet, leave, enter, exit 등과 같은 이벤트 온톨로지와 스포츠, 자연, 교통 등과 같은 도메인 온톨로지를 구축하고 각 비디오 Shot에 대한 온톨로지를 구축한다. 구축된 온톨로지는 시맨틱 웹의 표준인 OWL/RDF(S) 형태의 메타데이터로 표현하여 온톨로지를 이용한 의미적 검색이 가능하게 한다.

마지막으로 제안된 움직임 모델링 기법에 대한 성능평가를 위해 실제 영상의 움직임 객체의 좌표값 추출을 통해 제시된 규칙을 적용하여 움직임 인식과 이벤트 검출에 대해 실험한다. 또한 제안된 모델링 방법에 의해 실험된 움직임 인식을 통한 이벤트에 대해 비디오 Shot 단위로 인덱싱을 수행하여 온톨로지에 의한 의미적 검색이 가능함을 보인다. 사용자의 관심대상인 meet, leave, enter, exit 등과 같은 이벤트 검출을 위해서는 각각의 비디오 프레임에 존재하는 객체의 의미인식이 선행되어야 한다. 따라서 비디오 프레임에 대한 의미인식과 주요 의미 정보를 이용한 검색은 사용자가 원하는 비디오 Shot을 검출하는데 있어 검색의 질을 보다 향상시킨다. 아울러, 비디오 객체 움직임에 대한 인식 기술은 움직임 정보에 대한 자동 색인이나 자동 나레이션 생성에도 매우 유용하게 적용될 수 있다.

C. 논문의 구성

본 논문의 구성은 다음과 같다. 제 1장 서론에 이어 제 2장에서는 비디오 데이터의 객체 움직임 모델링에 대한 기존 연구에 대해 기술한다. 특히, 비디오 데이터의 객체간 위상관계에 의한 움직임 표현을 위해 사용된 모델링 기법을 소개한다.

제 3장에서는 좀더 고차원적이고 사용자 인지적인 움직임을 통해 이벤트 표현이 가능한 모델링 기법을 제안한다. 제한된 모델링 방법은 비디오내 객체 움직임에 대한 의미 인식을 위해 객체들간의 시·공간적 특징기반의 이벤트에 대해 인간의 의사표현 수단인 언어를 이용하여 움직임을 표현하고 이벤트 설명을 위한 계층적 형태로 나타낸다. 움직임 표현을 위해 비디오의 프레임을 중심으로 단일프레임에서 인식 가능한 움직임과 연속된 프레임의 패턴을 통해 분석이 가능한 움직임 동사를 선별하였으며 사용자의 주요 관심 대상인 이벤트를 표현하기 위해 움직임의 시작-과정-결과 형태로 나타내어질 수 있는 대표 동사를 사용하였다. 제안된 모델링 기법은 이벤트와 관련된 비디오 Shot을 검출하는데 사용된다.

제 4장에서는 제시된 사용자 중심의 고차원적 모델과 저차원 특징을 매핑하기 위해서는 비디오의 전처리 과정이 필요하다. 따라서 Shot 검출 과정과 비디오 영상에서의 움직임 객체 추출기법을 통해 실험에 사용될 메타데이터 생성 과정을 기술한다.

제 5장에서는 비디오 데이터내의 객체 움직임을 인식하기 위한 방법으로 기준 객체에 대한 움직임 객체의 위치값을 통해 객체사이의 공간관계와 거리, 방향성을 고려하여 움직임 인식을 위한 규칙을 제안하고 알고리즘을 구현한다. 최종적으로 비디오 Shot이 의미하는 이벤트에 대한 규칙을 재 정의한다.

추가적으로 비디오 검색에 대한 사용자 만족도를 높일 수 있는 의미적 검색 방법으로 이벤트에 대해 도메인 온톨로지를 구축하고 OWL/RDF(S)로 표현한다.

제 6장에서는 제안된 모델링을 이용한 이벤트 검출과 온톨로지를 이용한 이벤트 검출에 대해 실험한다. 실험을 위해 실제 비디오 영상에 대해 프레임단위의 움직임 인식율과 이벤트 검출에 대해 실험하고 평가한다.

마지막으로, 7장에서는 본 논문의 결론 및 향후 연구 방향에 대해서 기술한다.

Ⅱ. 관련 연구

비디오 데이터에 대한 사용자의 관심이 높아짐에 따라 비디오에서 객체 움직임의 의미적 인식을 위한 다양한 연구가 이루어지고 있다. 본 장에서는 비디오 데이터의 의미 정보 분석을 위해 고려되는 시각적 특징정보와 시·공간적 특징 정보에 의한 연구를 중심으로 사용자 입장을 고려한 고차원적 의미 인식에 대한 기존 모델링 기법에 대한 연구들을 살펴본다.

A. 비디오 데이터의 의미 정보 분석

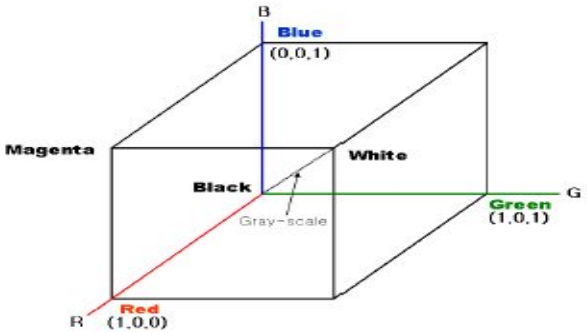
비디오 데이터의 인덱싱과 검색을 위해 비디오 데이터의 내용을 의미적으로 분석하고 인식하기 위한 기술이 대두되었다. 특히, TV 방송국, 비디오 처리 하드웨어 제공업체 등 비디오 데이터 처리를 요구하는 다양한 분야의 출현으로 다양하고 복합적인 구조를 가지는 비디오 데이터를 효과적으로 모델링하고, 대량의 비디오 데이터를 효율적으로 저장 및 검색할 수 있는 기능이 요구되고 있다. 비디오 데이터의 내용 인식을 위한 방법으로 색상, 움직임, 절대위치(absolute position) 등과 같은 시각적 특징 정보를 이용한 방법과 비디오내에 존재하는 객체들 사이의 공간적, 시간적 특징 정보를 이용한 방법 등이 있다.

1. 시각적 특징정보에 의한 의미정보 분석

비디오 데이터는 색상, 움직임, 절대 위치 등과 같은 시각적 특징정보를 가지고 있다. 이러한 특징정보는 비디오의 내용을 좀더 의미적으로 표현하는데 사용된다.

a. 색상 정보

색상이란 영상을 볼 때 즉각적으로 인지되는 시각 특징이며 밝기, 색도, 채도의 세 가지 속성을 가지는 객체를 식별하게 하는 하나의 특징이다. 색상 특징을 나타내는 대표적인 방법으로 컬러 히스토그램에 의한 방법이 있다[9][10]. 컬러 히스토그램은 영상의 하위 수준 색상 속성을 기술하는 가장 전통적인 방법이다. 컬러 히스토그램의 계산 방법은 각 색상 값에 얼마나 많은 화소가 속하는지 계수하는 통계적 방법을 사용한다. 영상간의 유사성은 각 화소가 갖는 색상 값(color value)의 차이에 의해 결정된다. 컬러 히스토그램으로써 표현되는 색상 분포는 영상을 나타내는 객체의 각 구성 요소에 대한 내용보다는 전체적인 영상의 내용을 나타내는데 적당하다. 색상 정보를 이용한 대표적인 모델로는 RGB 색상 모델이 있다[11]. 이는 (그림 2.1)과 같이 빛의 삼원색인 빨강, 초록, 파랑을 기본색으로 하여 3차원 좌표공간으로 되어 있으며 빛의 가산 혼합 원리에 의해 색을 만든다.



(그림 2.1) RGB 컬러 3차원 입방체

(그림 2.1)에서 검정(B)색은 원점(0,0,0)이며, 흰색(W)은 입방체의 반대쪽이다. 명암도는 검정색에서 흰색으로 이어지는 선을 따라 표현된다. 컬러 채널 당 8비트를 가지는 24비트의 그래픽스 컬러 시스템에서 빨간색은 (255,0,0)이며, RGB 컬러 3차원 입방체 상에서는 (1,0,0)이다. 컬러 입방체 사이에서의 빨강, 초록, 파랑 값의 범위는

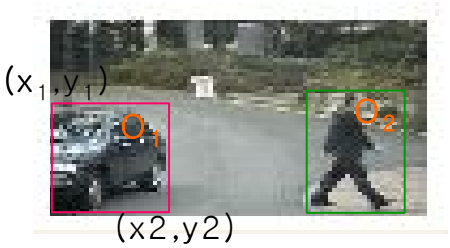
$0 \leq R, G, B \leq 1$ 이다. RGB 색상 모델은 간단하지만, 빨강, 초록, 파랑의 컬러 요소들의 상호관계가 너무 크다는 단점을 가지고 있다. 그러나 주어진 컬러 영상은 초기에 특별한 변환을 가하지 않으며 R, G, B값을 보유하고 있기 때문에, 영상을 구성하는 각 픽셀의 RGB 색상 공간에서의 분포 특성은 검색과정에 유용하게 사용될 수 있으며, RGB의 세 가지 색상채널별 히스토그램을 일렬로 정렬한 컬러 히스토그램을 가장 널리 사용하고 있다.

b. 움직임 정보

움직임 정보는 비디오 데이터의 인덱싱과 검색을 위한 핵심 정보라 할 수 있다. 움직임 정보를 이용한 기존 연구에서는 카메라의 움직임만을 고려하거나[12][13] 객체의 움직임만을 고려한 연구[14][15][16]가 대부분이며 아주 일부의 연구에서 카메라, 객체 두 가지 움직임을 모두 고려하였다[17].

c. 절대 위치

비디오 데이터에는 하나이상의 객체가 존재한다. 이러한 객체들 사이의 공간적, 시간적 관계 정의를 위해 비디오 구조의 프레임내에 위치하는 객체의 2차원적 위치를 나타내기 위해서 절대위치가 사용된다. MBR(Minimum Bounding Rectangle)은 비디오 구조에 존재하는 객체의 공간적, 시간적 관계를 이용하기 위한 가장 적절한 값이 된다. (그림 2.2)의 2차원 프레임 Kj의 MBR에 의한 객체 O_1의 위치를 나타내기 위해서는 (x1,y1), (x2,y2)로 표현되어 진다. 사용자 관심대상의 비디오 Shot은 연속적인 프레임으로 구성되어 지며 각 프레임내의 객체의 절대 위치값을 통해 객체의 시・공간적 관계를 이용하여 움직임에 대한 인식이 가능해 진다.



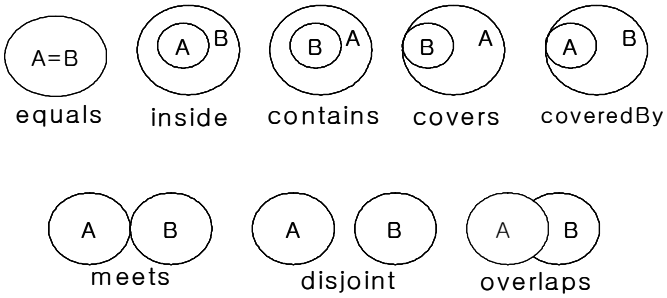
(그림 2.2) 객체의 MBR

2. 시・공간적 특징에 의한 의미정보 분석

시・공간관계는 시간의 흐름에 따라 객체의 위상관계와 방향관계로 객체를 표현하는 것을 말한다. Egenhofer[6]는 객체사이의 관계를 8가지 형태의 공간적 관계로 정의하였으며 Allen은 13가지 시간적 관계를 정의함으로써 객체의 움직임을 표현했다[5].

a. 공간적 특징

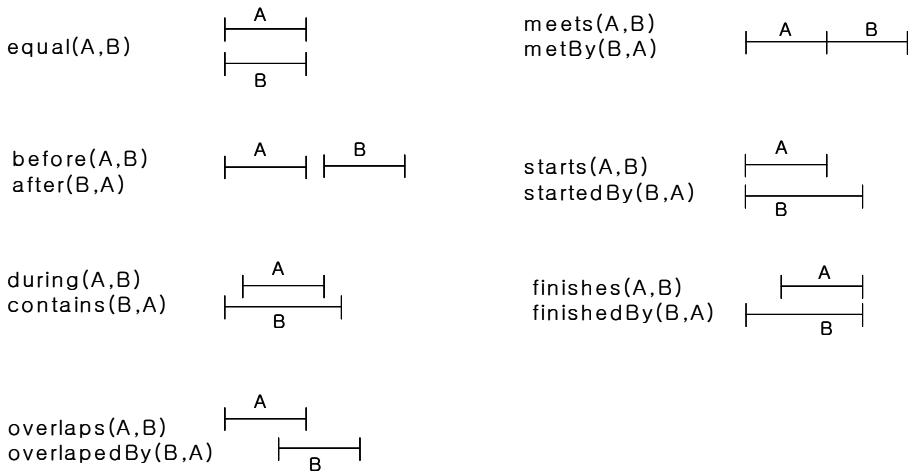
Egenhofer는 2차원 프레임 내의 비디오 객체 A와 B의 절대 위치값인 MBR에 의해 (그림 2.3)과 같은 8가지의 공간적 관계(spatial relation)를 제안했다. 또한 두 객체사이의 공간적 관계 표현을 위해 Description Logic을 사용하였다. 예를 들어 객체 A와 B사이의 공간적 관계 정의를 위해서는 A의 위치값 (A_{x1},A_{y1}), (A_{x2},A_{y2})와 B의 위치값 (B_{x1},B_{y1}), (B_{x2},B_{y2})가 얻어진다. 따라서 두 객체의 공간적 관계가 contains의 관계인 경우 정의된 DS에 의해 $(A_{x1},<B_{x1} \wedge A_{x2}>B_{x2}) \wedge (A_{y1},<B_{y1} \wedge A_{y2}>B_{y2})$와 같이 표현했다.



(그림 2.3) Egenhofer의 8가지 공간 관계

b. 시간적 특징

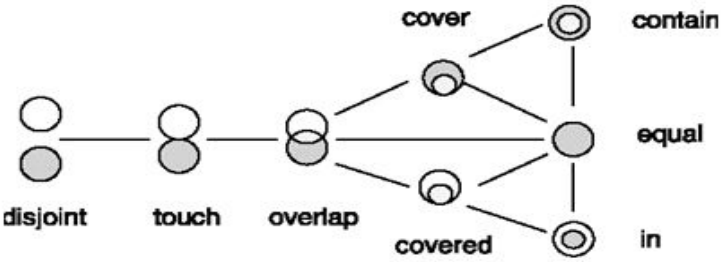
일반적으로 객체를 비디오 데이터베이스에 저장하기 위해서는 시간적 간격(temporal interval)이 존재한다. 즉 객체가 데이터베이스 저장되는 순간의 시간(valid time)과 처리되는 시간(transaction time) 등 두 가지 형태의 시간이 존재한다. 따라서 Allen은 객체의 시간적 간격을 표현하기 위해 13가지의 관계와 그에 따른 구간 기반 대수를 제안했다[47]. 이러한 관계는 객체사이의 이벤트에 대한 상대적 시간을 테스트 하는데 필요하게 된다. 예를 들어 "비디오 씬(scene)내에 객체 O_1, O_2, O_3의 순으로 나타나다 어느 순간 O_2후에 O_3가 사라진 장면이 들어있는 비디오를 찾아라"와 같은 질의를 위해서는 before, after 등과 같은 시간적 관계가 요구된다. (그림 2.4)는 Allen의 시간적 관계를 나타내고 있다. 객체 A와 B의 I간격사이의 시간관계 after(A, B)를 구간 대수로 정의하면 $(A_{i1}>B_{i2})$로 정의되며 overlaps(A, B)는 $(A_{i2}>B_{i1} \wedge A_{i2}<B_{i2}) \wedge A_{i1}<B_{i1})$으로 정의된다.



(그림 2.4) Allen의 13가지 시간관계

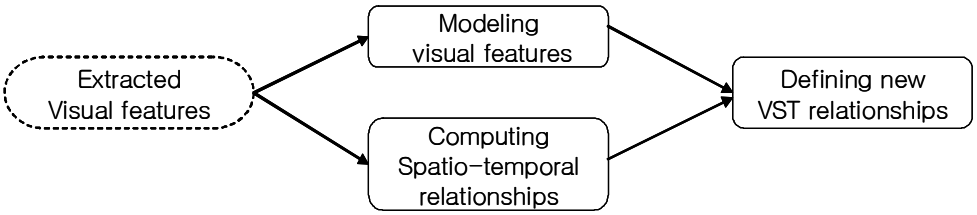
c. 위상관계

위상관계(topological relationships)는 Allen의 시간관계와 Egenhofer의 공간관계를 통합한 개념으로 시간의 흐름에 따른 움직이는 객체의 공간적 관계를 나타내는 것으로 기본이 되는 위상관계는 움직이는 두 객체 사이의 관계로 나타낼 수 있다. 이를 시간에 따라 나누면 한 객체가 움직이고 있을 때 다른 객체는 멈추고 있다고 말할 수 있으며, 이러한 과정들의 반복을 통해 움직이는 객체 사이의 관계를 표현할 수 있다. 시・공간 관계를 통한 객체의 위상관계에 관한 많은 연구가 진행되었다. Egenhofer 는 (그림 2.5)와 같이 시간의 흐름에 따른 공간적 관계를 제안하여 객체의 위상관계를 표현했다[18]. 예를 들어 특정 시간동안 두 객체가 touch 상태이고 이시간의 인터벌이 끝나는 시점에서 계속적으로 존재한다면 이웃하는 두 객체 사이의 위상관계는 touch, disjoint 또는 overlap 상태가 된다. 이에 따라 touch 상태의 연속되는 시간 관계에 의해 meet(m)과 met(mi), during(d)와 contain(di), start(s)와 started(si), finish(f)와 finished(fi)와 같이 연속적으로 수행될 수 있다.



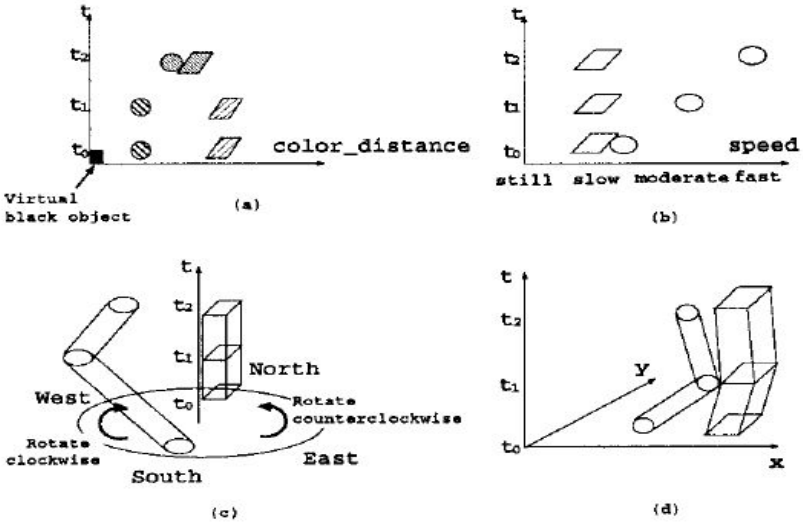
(그림 2.5) 위상관계에 의한 개념적 표현

Zaher Aghbar는 비디오 Shot내의 시각적 특징(color, motion, texture, shape, etc)과 시・공간적 관계를 통합한 Visual-Spatio-Temporal(VST) 모델을 제안했다[39].



(그림 2.6) 시각적 특징과 시・공간 관계에 대한 통합 프로세스

VST 모델은 먼저, 시각적 특징정보인 컬러, 속도, 방향, 그리고 위치에 대해 비디오 Shot내의 객체에 대한 위상 관계를 정의하였다. 컬러 위상 공간의 경우 프레임 단위의 두 객체 관계를 컬러값의 거리를 통해 표현하였으며, 속도 위상 공간은 프레임 단위의 두 객체에 대한 속도를 slow, moderate, fast로 구분하여 속도를 표현하였다. 방향 위상 공간은 하나의 Shot에 존재하는 객체에 대해 9차원적 공간 즉, none, north, northeast, east, southeast, south, southwest, west, and northwest로 표현하였다. 위치 위상 공간(position topological space)에 대한 정의는 비디오 Shot에 존재하는 객체의 위치 표현을 2차원 공간으로 나타냈다. (그림 2.7)은 시각적 특징들에 대한 위상 관계 표현을 나타내고 있다.



(그림 2.7) 시각 객체의 위상관계 표현

시・공간 관계 정의는 Egenhofer의 공간관계와 Allen의 시간관계를 이용하였으며 새로운 객체 사이의 VST 관계는 (표 2.1)에서와 같이 정의하였다.

(표 2.1) 객체 사이의 VST 관계

Relationship set	Member Relationships
relColor	sameColor, lighter, darker
relSize	sameSize, bigger, smaller
relPosition	samePosition, above, below, toTheRight, toTheLeft
relSpeed	sameSpeed, slower, faster
relDirection	sameDirection, opposite, perpendicular, sameWithAngle, oppositeWithAngle, towards, awayFrom
relAppearanceTime	inBefor, inWith, inAfter, outBefore, outWith, outAfter

이와 같은 VST관계는 비디오 Shot의 유사검색을 위해 Egenhofer[18]의 위상관계에 의한 Neighborhood 그래프로 표현하여 유사검색에 사용된다.

VST모델은 비디오의 움직임 객체에 대한 시간의 변화에 따른 시각적 특징 정보의 궤적과 객체들간의 관계 궤적을 조합하여 나타내는 내용 궤적 모델링 기법으로 객체관계를 표현하기 위해 모든 VST관계 속성별로 그래프를 그려 유사도를 계산해야 하며 또한 정확한 궤적 표현과 검색이 어렵다는 문제점을 가진다.

B. 객체 움직임에 대한 의미적 모델링

비디오 데이터에 대한 의미적 검색을 위해 객체 움직임의 의미적 이벤트 표현에 대한 연구가 활발하게 진행되고 있다. 기존의 연구들은 저차원 특징을 이용한 내용기반 영상검색이나 비디오 데이터에 미리 주석을 정의하는 방법들이 대부분이며, 영상을 바라보는 인간의 인지적이고 개념적인 움직임 표현을 해석하고 처리할 수 있는 방법들이 미흡하다. 비디오 데이터의 객체 움직임 표현에 대한 저차원적 특징을 사용자의 관심대상인 비디오구조의 scene이 가진 이벤트를 표현하기 위해 제1절에서 제시한 Allen의 시간관계와 Egenhofer의 공간관계를 통합한 위상관계 정의를 통해 객체의 움직임을 고차원적으로 표현하기 위한 연구와 객체의 움직임을 시·공간적 관계 기반의 의미표현을 위해 영어의 동사가 가진

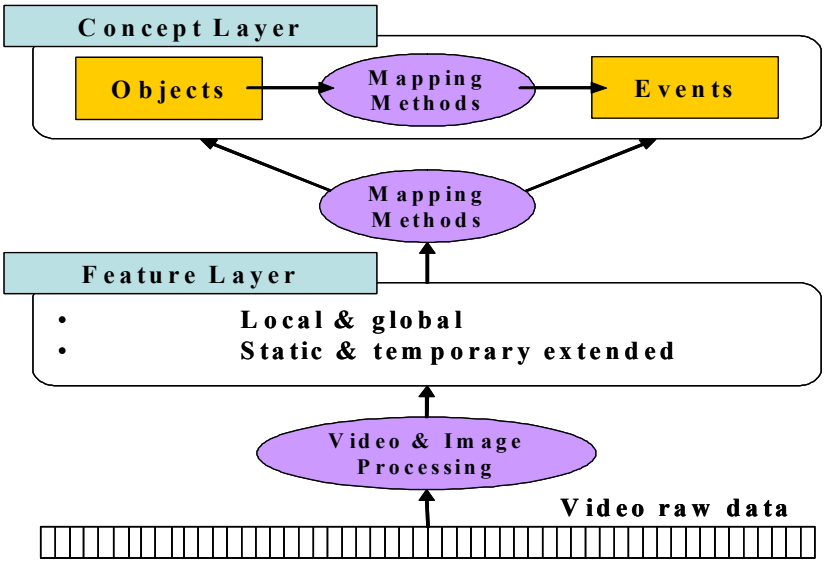
어휘관계를 온톨로지로 표현함으로써 비디오 내의 객체 움직임을 인식하고자 하는 연구가 시도되고 있다.

1. 시・공간 관계 기반의 저차원 특징과 개념적 통합

비디오내 움직임 객체의 의미표현을 위한 방법으로 객체의 시・공간 관계를 이용한 다양한 연구가 진행되어 왔다. 복잡한 비디오 구조에서 사용자 인지적인 의미표현을 위해서는 고차원적 개념과 이를 저차원 특징과 매핑할 수 있는 방안이 필요하다. 이를 위해 움직임 객체간의 시・공간적 규칙을 정의하여 의미적 표현을 하기 위한 연구가 시도되고 있다.

a. COBRA 비디오 모델에 의한 시・공간적 이벤트 인식

M.Andrea Rodriguez[19]는 low-level features와 high-level concept을 매핑하기 위한 방안으로 내용기반 검색(COBRA) 모델을 제안했다. 제안된 모델은 (그림 2.8)과 같이 Video raw data, Feature layer, concept layers로 구성된다. 이 연구의 핵심은 Feature layer에서 비디오 영상으로부터 저차원 특징값인 colors, textures, shapes, , motion 등을 자동 추출한다. concept layer에서는 비디오내의 이벤트 검출을 위한 규칙으로 객체에 대한 특징 값과 시・공간적 관계에 대한 규칙을 정의하였다. 사용된 규칙은 Allen과 Egenhofer가 제안한 시・공간 관계 정의에 의해 위상관계(equal, inside, cover, overlap, touch, disjoint와 두개의 inverse covered_by와 contains)를 사용하였으며 방향관계를 위해 유클리디언 거리 변환 방법뿐만 아니라 north, south, west, ease, north_ease, north-west, south-east, south-west 등을 사용하였다. 예를 들어 A와 B 시간동안 의 인터벌을 나타내기 위해 before(A, B, n), overlaps(A, B, n)과 during(A, B, n) 등 3개의 파라메터를 통해 정의했다. 여기에서 n은 A와B에 대한 시작시점에서의 시간적 차이를 나타낸다.

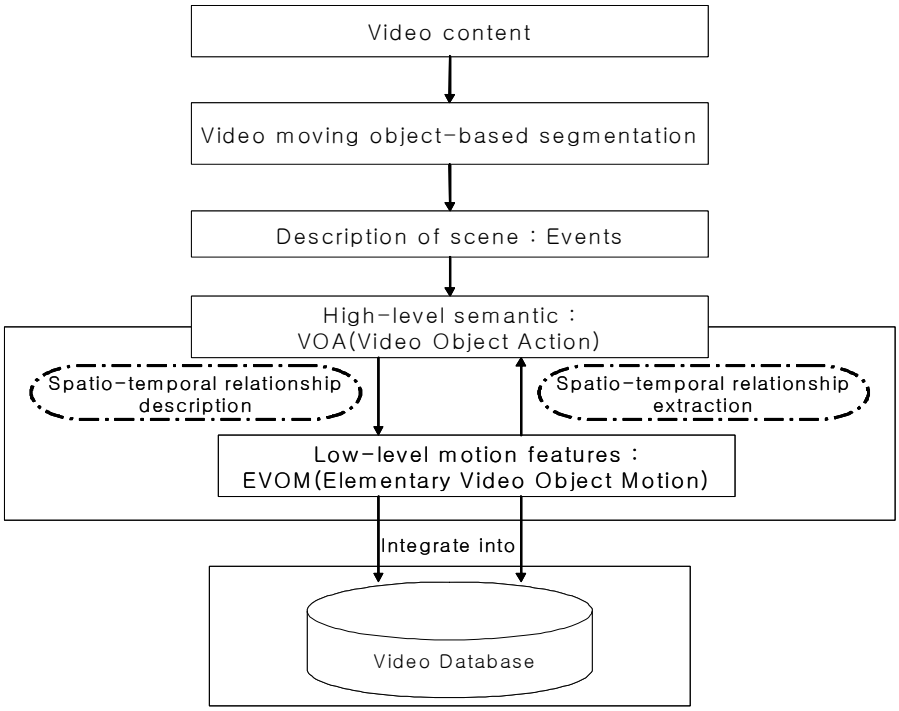


(그림 2.8) COBRA 비디오 데이터 모델의 계층구조

이 연구의 미진한 부분은 시·공간적 규칙을 수작업으로 정의했을 뿐만 아니라 내용 기반의 의미표현을 중점적으로 했기 때문에 사용자가 의미적 검색을 하기 위해서는 다양한 속성정보를 제시해야 한다. 따라서 영상내의 특징에 대한 의미적 표현은 시도되었으나 사용자 입장이 고려된 고차원적인 접근방법은 미흡한 상태라 할 수 있다.

b. 시·공간 규칙에 의한 객체 움직임 의미 표현

(그림 2.9)는 시·공간 관계에 의해 움직임 객체를 설명하기 위해 계층구조의 추상적 비디오 데이터 모델을 제안했다[44]. 비디오 시작 프레임과 마지막 프레임 사이에 나타나는 이벤트 설명을 위해 객체 움직임에 대한 저차원적 특징을 행동 등과 같이 고차원적으로 표현하기 위한 방법으로, 비디오 데이터베이스로부터 객체 움직임에 대한 특징(EVOM)값을 자동으로 추출하여 VOA로 매핑하기 위한 시·공간 관계 정의를 통해 사용자 관심도가 높은 이벤트를 추출할 수 있는 모델이다.

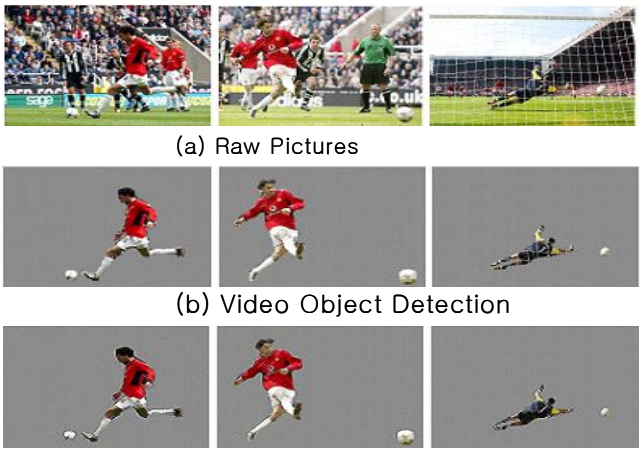


(그림 2.9) 추상적 개념의 비디오 데이터 모델

제안한 모델을 통해 축구비디오의 두 객체사이의 상호관계를 정의하기 위해 VOA (kicking, pushing, pulling...)과 EVOM(approach, touch...) 등과 같은 몇 가지 액션을 정의했다. 따라서 "Penalty Goal" 이벤트 설명을 위해서는 O_1(player), O_2(ball), O_3(net) 등 3가지의 객체의 시・공간 관계를 정의하고 객체와 이벤트에 대한 의미적 표현을 다음과 같이 정의했다.

- e=(event10, the penalty goal, frame0-frame42, penalty area, first half : 40:56)
- O1=(object1;[name:VanNistelooy];[role:Forward]:
  [event: Penalty Goal]]; [locater : #frame0];[location: Penalty Area])
- O2=(object2;[name:ball]:[event:PenaltyGoal];
  [locater:#frame(0,14,28,42)];[location:Penalty Area])
- O3 = (object3;:#frame(42));[location:PenaltyArea])

(그림 2.10)은 “Penalty Goal"에 대한 키프레임으로 (a)는 비디오 원본데이터, (b)는 움직임 객체만 추출한 상태, (c)는 두객체의 시・공간 관계에 의해 Tracking한 상태를 나타내고 있으며 위에서 정의한 의미적 표현을 저차원 특징과 매핑하기 위한 방안으로 (표 2.2)에서와 같이 두 액션에 대해 6가지의 시・공간적 표현을 통해 규칙을 제안했다.



(a) Raw Pictures

(b) Video Object Detection

(c) Video Object Tracking

(그림 2.10) 축구 비디오의 “Penalty Goal” 장면

(표 2.2) “패널티 골”에 대한 시・공간 관계 규칙 정의

이벤트 : 패널티 골		
Action	Motions	시・공간관계 규칙
Kick	Approach	((O1x.{d,di}O2x∧O1y{d,di,s,si,e}O2y)∨ (O1x{d, di, s, si, f, fi e} O2x∧O1y{d, di} O2y)
	Touch	((O1x.{m,mi} O2x∧O1y{d, di, s, si, m, mi e}O2y)∨ (O1x{d, di, s, si, f, fi e} O2x∧O1y{m, mi} O2y)
	Disjoint	O1x{b, bi} O2x∨O1y{b ,bi} O2y
Getting Goal	Meet with	((O2x.{d, di} O3x∧O1y{d, di, s, si, e}O3y)∨ (O2x{d, di, s, si, f, fi e} O3x∧O2y{d, di} O3y)
	Overlap	O2x{d, di, s, si, f, fi, o. oi. e} O3x∧O2y{d, di, s, si, f, fi,o,oi, e} O3y
	Go into	O2x{d} O3x∧O2y{d} O3y

이와 같은 시·공간에 의한 이벤트 규칙은 몇 가지 한계점을 가진다. 객체 사이의 위상 관계를 위한 매개변수 처리가 정규화 되지 않았을 뿐만 아니라 각각의 객체가 위치하는 프레임에 대한 정보를 사전에 명시해야 된다. 수 많은 비디오처리를 위해 프레임 정보를 일일이 수작업으로 분석해야 하는 어려움이 따른다. 또한 저차원 특징값과 매핑을 위한 명확한 방법론에 대한 제시가 되지 않음으로서 검증에 대한 문제점을 가진다.

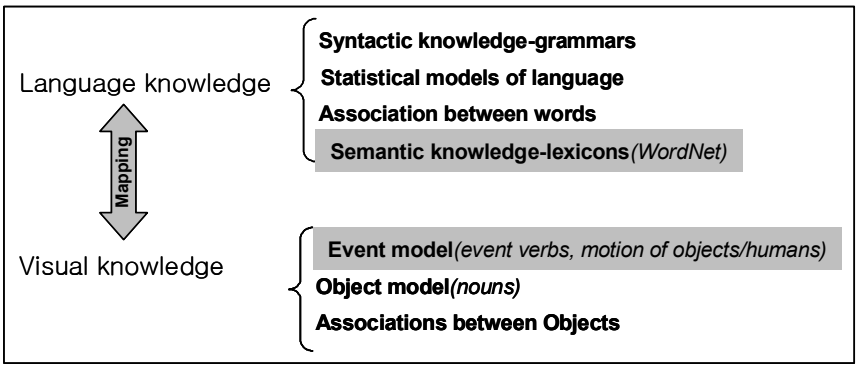
2. 온톨로지를 이용한 움직임 표현

최근 비디오내 객체 움직임 인식을 위해 온톨로지를 이용한 방법이 부각되고 있다. 온톨로지의 기본 개념은 철학의 존재론으로부터 시작되었다. 실제에 대한 구성 개체간의 관계에 대한 표현 및 의미를 부여하기 위하여 온톨로지는 발전하였으며, 이러한 기본 개념으로부터 사람과 컴퓨터 사이에서의 공통된 이해를 위한 필요성 때문에 온톨로지의 개념이 더욱 대중화 되고 있다. 움직임을 표현하기 위한 많은 연구에서 영어의 동사를 의미단위로 분류하여 시각적 특징에 대한 의미 표현을 하기 위한 연구가 시도되면서 대표적인 어휘 온톨로지인 워드넷을 기반으로 특정 도메인 온톨로지를 이용한 연구가 진행되고 있다[21-25][35-38].

a. 지식기반 매핑관계를 이용한 객체 움직임 표현

지능형 멀티미디어 처리를 위한 시스템인 CONFUCIUS 에서는 지식기반의 언어와 시각적 모델을 통합하여 3D 에니메이션 동작을 표현하기위한 구조를 제시했다. (그림 2.11)은 언어적 지식과 시각적 지식을 통합하기 위한 모델을 나타내고 있다. Language knowledge에서는 워드넷[22]과 LCS 데이터베이스[26]를 사용하며 Visual knowledge는 3D 에니메이션에서 요구되는 정보로 구성된다. Object model은 사물, 장소, 사람 등과 같은 명사로 구성되었으며 Event model의 경우 사물 또는 사람 객체의 움직임을 표현 지식과 walk, jump, give, push 등과 같은 행동을 고차원적으로

표현하기 위한 이벤트 동사로 구성되었다.

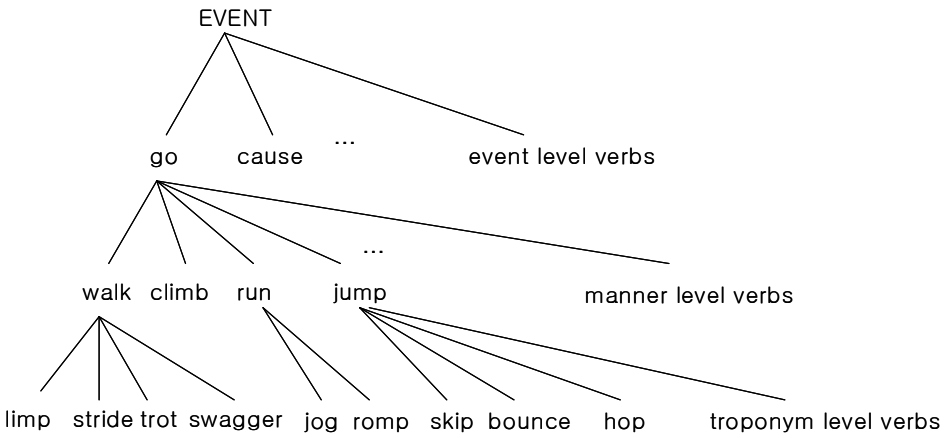


(그림 2.11) 지식기반의 상호 매핑 구조

(1) 영어 동사의 의미관계에 기반 한 온톨로지

의미적 분류를 통해 구성된 많은 영어 동사에 대한 온톨로지 종류가 있다. 그중 객체의 동작 표현을 위한 대표적인 동사 클래스는 워드넷[31] 동사 클래스와 Levin[38]의 동사 클래스이다. Minhua ma[27]에서는 지식기반의 이벤트 모델에서 분류하고 있는 Action verb의 4가지 형식을 5가지 범주로 분류하였으며 그중 사람과 다른 객체의 동작을 표현하는 클래스에 중점을 두었다. 이러한 motion verb 클래스의 특징 분석을 통해 (그림 2.12)와 같이 3단계의 움직임 동사에 대한 계층구조를 제시하였다.

최상위 계층인 event level은 기본객체의 움직임을 나타내는 go, move와 동사로 구성되며, 중간 계층인 manner level에서는 사람 또는 객체의 움직임 방법을 나타내는 동사로 구성된다. 최하위 계층인 troponym level은 manner level 동사의 속도, 상태, 동작 기간, 반복 등을 설명하는 동사로 구성된다. 이러한 범주의 온톨로지는 사람, 사물 어느 것에도 한정되지 않고 다양한 객체의 움직임 표현에 적용될 수 있다.



Levels	Verbs	Basic-level verb for ...
event level	go, move	Atomic object
manner level	walk, jump	Human/non-atomic object
troponym level	limp, stride	Human

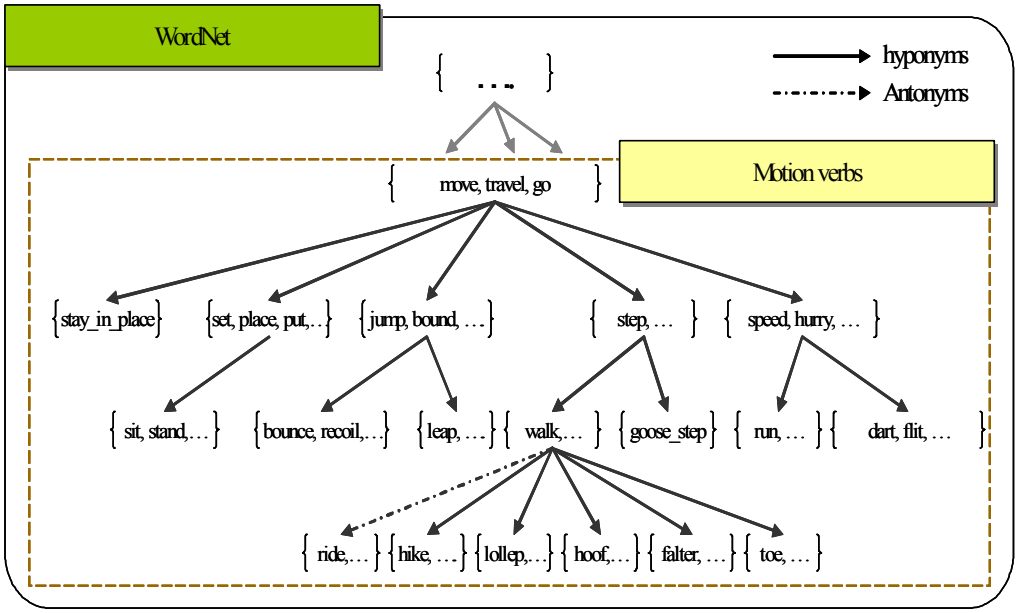
(그림 2.12) 움직임 동사의 계층 구조

(2) 워드넷의 의미관계 표현

워드넷(WordNet)이란 단어간의 관계를 의미중심으로 표현하는 어휘 데이터베이스이다[22]. 워드넷은 영어에 대한 명사, 동사, 형용사, 부사를 포함하여 120,000개의 Synset(유의어 집합)으로 구성되어 있다. 동사의 경우 의미 영역을 기준으로 15개의 범주로 나뉘었으며 14개의 범주는 행동이나 사건을 기술하는 동사로 이루어져 있다. 워드넷의 모션 동사 구조에서 “walk" 동사의 위치는 L+1계층에 위치하며 walk의 상위계층으로 L+2 계층의 {move, travel}은 walk를 하위어로 갖는다. L0계층으로 march, strut, traipse, amble, mosey, slouch 등이 있다. (그림 2.13)은 워드넷 동사의 움직임을 5개의 클래스 {stay_in_place, set, bound, step, speed}로 표현했다. 이러한 클래스는 사람, 사물, 방향 또는 속도와 같은 객체 속성을 구분하는데 사용된다.

Miyoung Cho[23]에서는 객체, 움직임, 방향, 속도 등의 시각정보와 워드넷 동사와

의 의미관계를 매핑할 수 있는 방안에 대해 (표 2.3)와 같이 움직임 중 sit과 stand에 대한 어휘로써 인간의 앉은 동작과 서는 동작을 표현하고자 했다. 이는 단순히 고차원적인 접근 방법이며 움직임 인식을 위한 저차원의 특징값의 표현이 모호할 뿐만 아니라 사람의 움직임 표현을 위한 요소가 단순함으로써 제시된 sit, down을 위한 인식을 위해서는 더욱 세부적인 요소가 필요하다.



(그림 2.13) 워드넷 모션 동사에 대한 계층 구조

(표 2.3) Visual information에 대한 의미관계 매핑

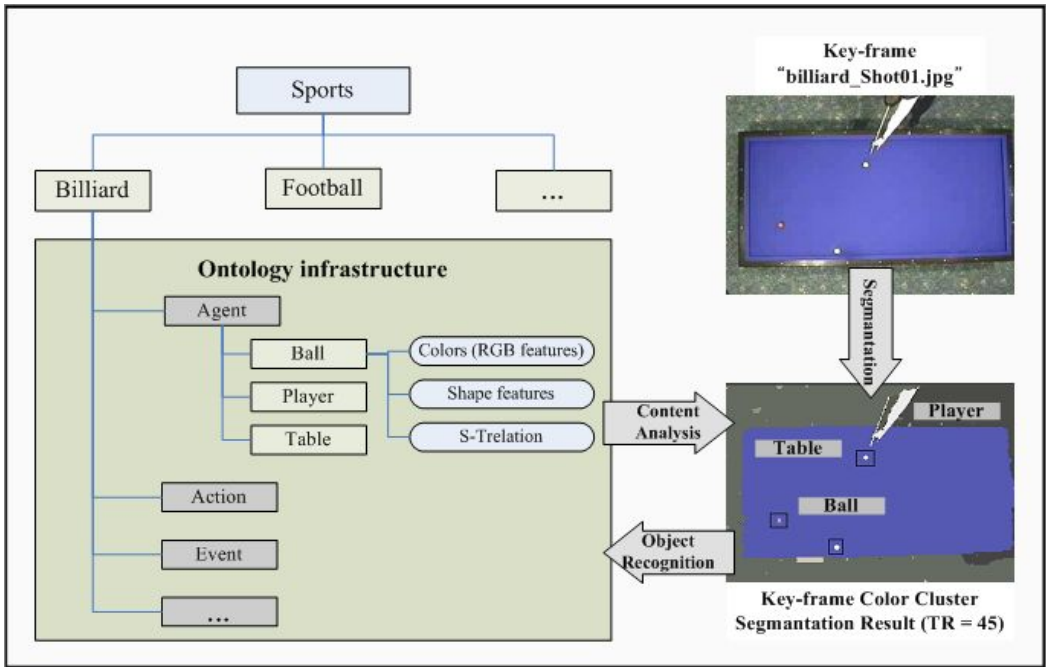
Visual information	Element	Attribute
object	person(noun)	-
surrounding	-	none, indoor, outdoor
motion	move place sit stand	-
motion speed	-	none, slow, fast
motion direction	-	north, south, west, east

b. 도메인 온톨로지에 의한 객체 움직임 표현

지식을 표현하기 위한 방법으로 온톨로지를 적용하는 연구가 증가하면서 특정 영역(domain)에 존재하는 개념들과 그들의 속성, 상호 의미적 관계에 대한 정보를 가지는 도메인 온톨로지 활용이 높아지고 있다. 개념과 개념들 사이의 관계를 정의하고 이를 언어적으로 표현하기 위해서는 시맨틱 웹 언어인 OWL/RDF(S)를 사용한다. 시맨틱 웹(Semantic Web)은 문서의 의미를 명백하게 기계가 이해할 수 있는 형태로 표현하여 컴퓨터가 웹 자원들을 효율적으로 관리할 수 있게 하는데 목적이 있다. 시맨틱 웹을 담당하는 W3C에서는 온톨로지 구축을 위한 언어로 OWL/RDF(S)를 표준으로 제정하였으며 이 언어가 가진 어휘를 사용하여 구축된 온톨로지를 표현함으로써 의미적 검색은 물론 추론이 가능해진다.

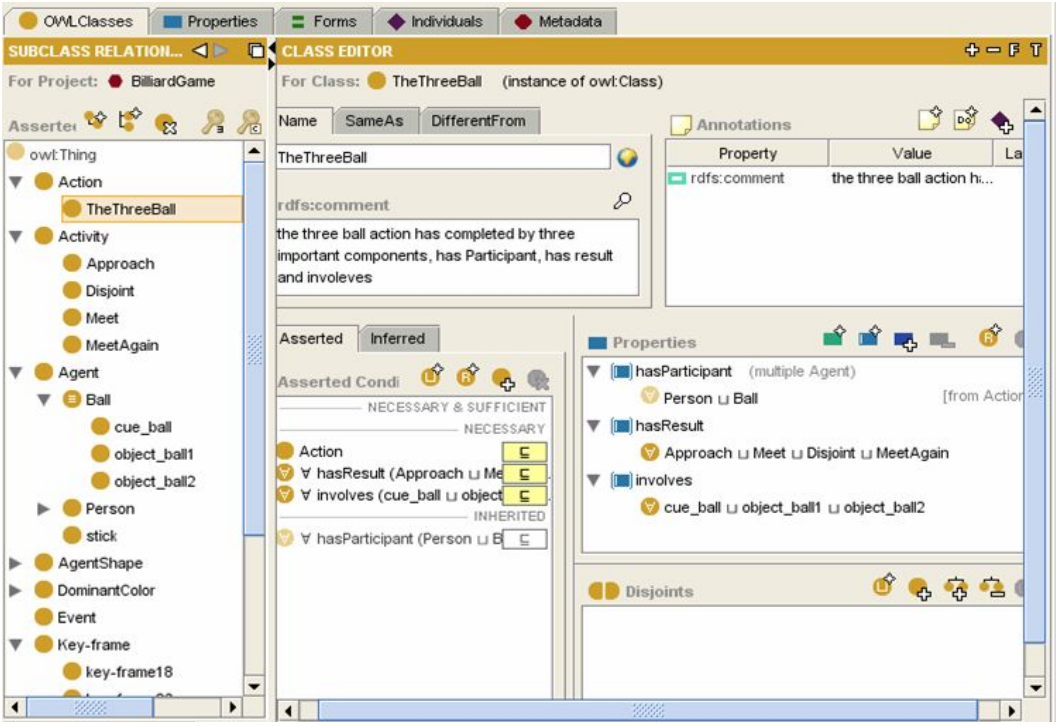
Dan Song[25]에서는 당구 비디오내의 “Three Ball" 이벤트 표현을 위해 볼, 선수, 테이블 등의 객체 관계를 표현하기 위한 도메인 온톨로지를 구축하고 OWL/RDF(S)로 표현함으로써 의미적 검색이 가능함을 보였다. (그림 2.14)는 비디오 내용 표현을 위한 온톨로지를 나타낸다. Billiard 도메인 내에 클래스 계층으로 Agent, Action, Event 등을 정의하였으며, 비디오 프레임내의 DS(Description Schema)를 참조하여 에이전트 내에 Ball, Player, Table 등과 같은 인스턴스를 정의하였다.

인스턴스의 특징 정보로 컬러, 윤곽, 시공간 관계와 같은 저차원 정보는 인스턴스의 프로퍼티로 정의앴다. (그림 2.14)의 온톨로지 계층구조는 오른쪽 비디오 키프레임내의 객체정보를 중심으로 표현되었으며, "billiard_Shot01.jpg" 에 대한 키프레임에는 3개의 Ball, Table, Player 등의 객체가 있으며 각 객체의 특징정보를 도메인 온톨로지 내에 프로퍼티로 생성하고 이는 비디오 데이터의 의미적 표현을 위해 사용된다.



(그림 2.14) 당구 비디오의 온톨로지 구축을 위한 의미적 표현

(그림 2.15)는 온톨로지 구축을 위한 에디터 Protégé 3.1을 통해 객체사이의 관계를 정의하였다. "Three-cushion"에 대한 규칙 정의를 위해 Person과 Ball은 hasPaticipant 이며, cue_ball과 object_ball1, object_ball2를 포함해야 하며, 이들은 서로 approach∪Meet∪Disjoint∪MeetAgain 관계를 갖어야 함을 정의하였다.



(그림 2.15) Protégé 3.1에서 당구 비디오 온톨로지 구축 화면

이와 같은 온톨로지 저작도구를 사용하여 작성한 Class는 시맨틱 웹 언어인 OWL/RDF(S)로 표현되고, 작성된 OWL은 온톨로지 파서에 의해 파싱이 이루어지며 의미적 검색 시 추론 기능을 제공하기도 한다. (그림 2.16)은 당구 비디오 온톨로지에 대한 OWL 내용의 일부이다.

```
<?xml version="1.0"?>
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdf="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:owl="http://www.w3.org/2002/07/owl#"
        xmlns="http://220.67.222.71/BilliardGame.owl">
    xmlns="http://220.67.222.71/BilliardGame.owl">
    <owl:Ontology rdf:about=""/>
    <owl:Class rdf:ID="player">
        <rdfs:subClassof>
          <owl:Class rdf:ID="Person"/>
        </rdfs:subClassof>
    </owl:Class>
    <owl:Class rdf:ID="object_ball2">
        <rdfs:subClassof>
          <owl:Restriction>
            <owl:onProperty>
                <owl:ObjectProperty rdf:ID="color"/>
            </owl:onProperty>
            <owl:allValuesFrom>
                <owl:Class rdf:ID="Red"/>
            </owl:allValuesFrom>
          </owl:Restriction>
        </rdfs:subClassof>
        <rdfs:subClassof>
            <owl:Restriction>
                <owl:onProperty>
                <owl:ObjectProperty rdf:ID="AgentShape_"/>
            </owl:onProperty>
            <owl:allValuesFrom>
                <owl:Class rdf:ID="Round"/>
            </owl:allValuesFrom>
          </owl:Restriction>
        </rdfs:subClassof>
        <rdfs:subClassof>
            <owl:Class rdf:ID="Ball"/>
        </rdfs:subClassof>
    </owl:Class>
```

(그림 2.16) 당구 온톨로지 OWL 파일

Ⅲ. 객체 움직임과 이벤트 표현을 위한 모델링

본 장에서는 비디오내의 객체 움직임 인식을 통해 의미적 이벤트를 표현하기 위한 모델링 기법에 대해 설명한다. 먼저, 객체 움직임 인식을 위한 프레임웍을 제시하고 모델링을 수행한다. 제시하는 이벤트 표현을 위한 객체 움직임 인식 방법은 앞 장에서 언급하였던 객체의 시·공간 관계에 의한 의미 표현 방법으로 인간의 인지적이고 개념적 입장을 고려한 의미적인 해석이자 표현 방법으로 고차원적 접근방법이다. 좀더 사용자 인지적인 움직임 인식을 위해 영어 동사의 어휘를 통해 객체관계를 표현하고 이벤트 검출을 위한 관계구조를 제시한다.

A. 객체 움직임 의미 인식을 위한 프레임웍

본 논문에서 제시하는 프레임웍은 비디오내 움직임 객체에 대한 의미를 사용자가 이해할 수 있는 고차원적인 표현을 위해 인간이 사고할 수 있는 언어적 지식을 이용하여 의미적 개념관계를 계층적으로 표현한다. 또한 고차원의 개념을 컴퓨터가 이해하기 위해서는 저차원 특징과 매핑할 수 있는 방안이 필요하다. (그림 3.1)은 본 논문에서 제안하는 비디오 데이터의 객체 움직임 인식을 위한 프레임웍으로 움직임에 대한 저차원적 특징값에 의한 규칙을 통해 영어의 동사와 매핑함으로써 비디오 영상으로부터 의미적 이벤트를 검출하기 위한 구조를 나타내고 있다. 프레임웍에서 제시하는 모델은 상위레벨의 움직임 위상 관계 정의를 위한 움직임 개념관계 모델링 부분과 규칙기반 매핑을 위한 intermediate level, 비디오 영상에 대한 전처리를 위한 Video & Image Process 부분과 최종 사용자 인터페이스 부분으로 구성되어 있다.

a. Video & Image Process(비디오 전처리 단계)

비디오 영상에 대한 전처리 과정을 통해 Shot 분할과 키 프레임을 추출하고 객체 움직임 추출을 수행한다. 이 단계의 핵심은 본 논문에서 제안하는 객체 움직임에 대한 의미인식을 위해 객체 움직임 추출 알고리즘을 사용하여 자동으로 MBR을 추출하고 그에 따른 로그를 생성하는 것이다. 이 단계에 대한 내용은 본 논문의 4장에서 기술한다.

b. High Level(움직임 개념관계 모델링)

사용자가 비디오 데이터의 의미적 검색을 위해서는 사용자 인지적인 모델링 방법이 필요하다. 본 단계에서는 객체 사이의 관계를 인간이 인지할 수 있는 공간적 관계로 표현하여 이벤트 중심의 모델링을 위해 영어 동사의 어휘관계를 움직임 인식을 위한 개념관계로 표현하여 고차원적 모델링을 수행한다. 이 단계에 대한 내용은 본 논문의 3장에서 기술한다.

c. Intermediate Level(규칙기반 매핑)

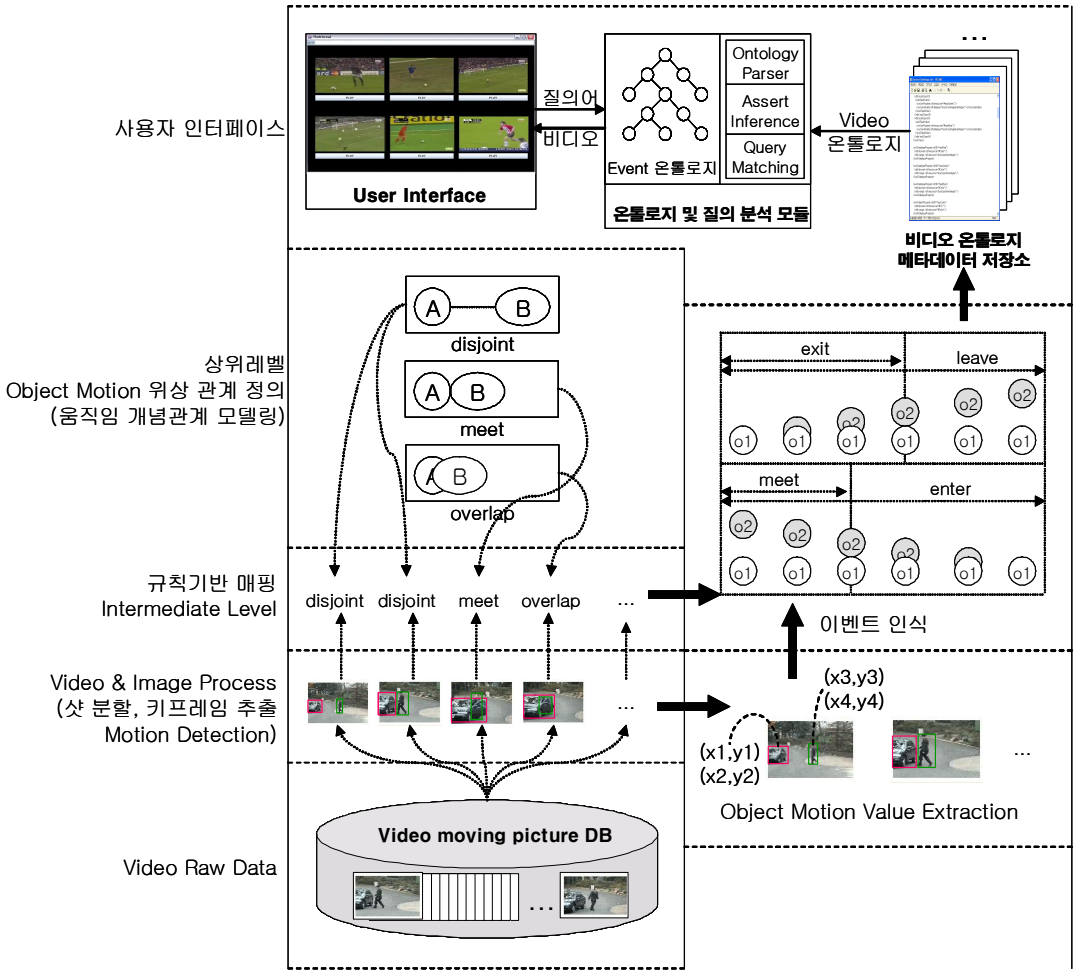
움직임 개념관계 모델링을 통해 제시된 고차원적 이벤트 모델은 최하위의 저차원 특징과 매핑되어야 실제 비디오 영상에 대한 객체 움직임 표현이 가능 하다. 본 단계에서는 video&image process 단계에서 추출된 객체 움직임 값과 상위레벨에서 제시된 움직임 동사와 매핑할 수 있는 움직임 규칙을 정의한다.

또한 최종적으로 인식된 이벤트에 대해 비디오 Shot단위로 인덱싱을 수행하여 온톨로지를 위한 의미적 검색에 활용할 수 있도록 한다. 본 내용은 5장에서 기술한다.

d. User interface(사용자 인터페이스)

비디오 데이터의 의미적 검색을 위해 이벤트 온톨로지를 구축하고 움직임 분석을 통해 인덱싱된 비디오 Shot에 대해 OWL/RDF(S)로 작성된 온톨로지 메타데이터 저장소를 통해 최종 이벤트에 대한 비디오 영상 검색을 수행한다.

본 내용은 5장과 6장에서 기술된다.



(그림 3.1) 객체 움직임에 대한 의미인식을 위한 프레임웍

B. 제안된 객체 움직임 모델링

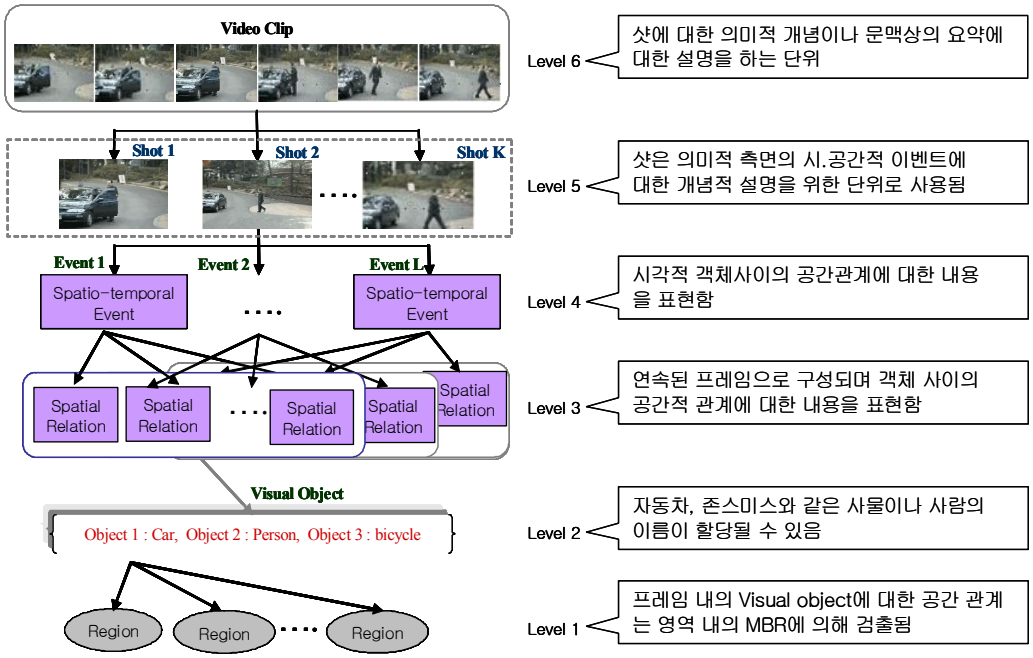
본 절은 (그림3.1)의 최상위 레벨에서 표현하고자 하는 움직임 개념관계 모델링 방법에 대해 설명한다. 비디오 데이터는 개념적으로 복잡한 구조를 가지고 있기 때문에 먼저, 개념적 구조를 통해 객체와 움직임에 대한 특징을 분석해 본다. 또한 사용자의 관심대상인 두 객체가 만나고 있는 장면, 사람이 건물안으로 들어가는 장면 등을 검출하기 위한 방법으로 움직임 특징에 의한 meet, enter와 같은 이벤트를 정의하기 위한 방법에 대해 설명한다.

1. 비디오 데이터의 개념적 구조

일반적으로 비디오의 개념적 구조 정보는 하나의 비디오 클립에서부터 최하위에 위치하는 프레임내의 여러 개의 영역으로 정의되며 각각은 비디오의 의미적 표현을 위한 기본 단위로 사용되고 있다. (그림 3.2)는 비디오 데이터의 개념적 계층구조를 나타내고 있다.

프레임내의 시각적 객체에 대한 공간관계는 영역내의 MBR에 의해 검출된다. 시·공간적 관계에 의한 의미적 이벤트는 시각적 객체 사이의 공간관계에 대한 내용이며, Shot은 의미적 측면의 시·공간적 이벤트에 대한 개념적 설명을 위한 단위로 사용된다. 최상위 계층은 비디오 클립으로 Shot에 대한 의미적 개념이나 문맥상의 요약에 대한 설명이다. 오브젝트 레벨에서는 car, john smith와 같은 사물이나 사람의 이름이 할당될 수 있다. 이 레벨에서 개념관계 처리가 요구되며 이러한 통합적 기술은 아직 충분한 연구가 되지 않았으며 주석을 사용하여 처리되고 있다. 객체간의 공간적 관계에 대한 개념적 표현을 위해서는 특정 프레임에서 일어나는 행위에 대한 의미적 분석이 필요하다. 시·공간적 이벤트의 개념적 설명은 “누가 일정기간 동안 무엇을 하고 있는지”와 같은 정보를 포함한다. 이와 같이 하나의 주제에 대한 개념적 설명은 두 번째 Shot 레벨에서 설명할 수 있다. 하나의 Shot은 시간과 공간 안에서 연속적인 동작을 표현하는 연속된 프레임을 가지게

되므로 Shot을 구분함으로써 프레임에 포함된 정보의 내용 변화를 알아낼 수 있다. 비디오 파서 알고리즘은 이러한 정보의 내용 변화를 발견하여 Shot 경계를 알아 낼 수 있어야 한다. 두개의 연속적인 비디오 프레임들의 각 기준 사이의 차이가 임계값을 초과했는지에 따라서 Shot 경계가 식별된다.



(그림 3.2) 비디오의 개념적 계층 구조

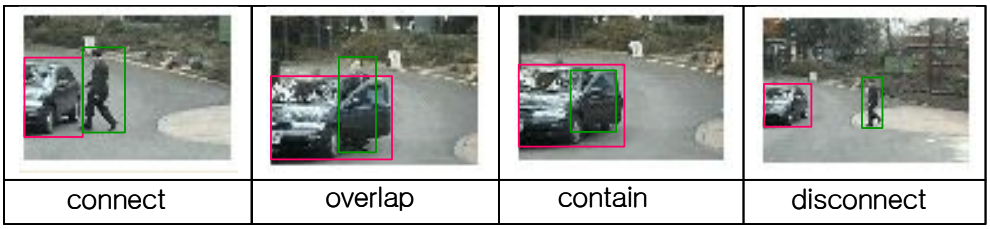
2. 동사 어휘와 객체 움직임에 대한 정의

본 논문에서 제시하는 모델링 방법은 비디오 데이터의 움직임 인식을 위해 객체사이의 관계를 설명하기 위한 동사를 수집하고 객체사이의 관계를 의미적으로 표현하기 위해서 객체관계와 가장 유사한 어휘를 가진 동사를 적용하는 방법이다. 제안하는 움직임 모델링을 위해 먼저, 정지영상 내에서 표현이 가능한 객체사이의 공간관계를 정의한다. 다음으로 본 논문의 최종 목표는 이벤트를 추출하는데 있기 때문에 연속적인 프레임에 대한 패턴을 통해 거리와 방향성 등을 고려하여 최종적인

움직임을 인식함으로써 비디오 Shot이 가진 이벤트를 추출할 수 있다.

a. 정지영상에서 객체관계 분석

객체의 움직임 의미 인식을 위해 정지객체와 움직임 객체로 구성된 비디오를 대상으로 각각의 프레임내에 존재하는 객체사이의 관계를 분석한 결과 (그림 3.3)과 같이 connect, overlap, contain, disconnect 상태로 파악되었다.



(그림 3.3) 정지영상내의 객체관계

b. 연속적인 프레임에 대한 움직임 패턴 분석

비디오 Shot이 의미하는 이벤트를 추출하기 위해 연속적인 프레임에 대한 패턴을 통해 거리와 방향성을 고려한 결과 (그림 3.4)와 같이 정지영상에서 disconnect 상태인 객체관계가 come toward, go away, approach, separate로 분석되었다.

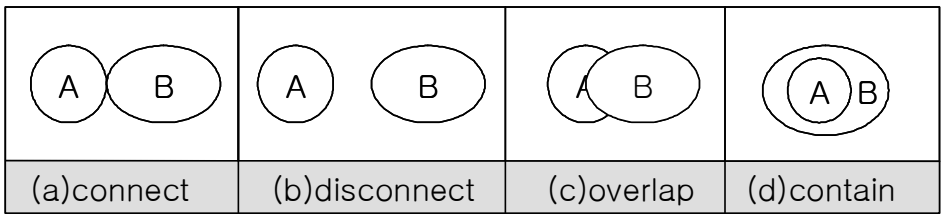
(그림 3.4) 연속적인 프레임에 대한 움직임

c. 연속된 움직임에 대한 이벤트 표현

정지영상과 연속된 프레임의 패턴에서 분석된 객체 움직임을 통해 이벤트를 표현하기 위한 규칙으로 초기단계(Initial State : IS), 중간 단계(Mid-State : MS), 마지막 단계(Final State : FS)와 같이 정의하였다. 초기 단계(IS)는 전체 움직임에 대한 최소 상태이고, 중간 단계(MS)는 특정 움직임에 대한 의미를 인식하는데 있어서 핵심적인 단계이고, 마지막 단계(FS)는 움직임에 대한 의미적인 해석이 종료된 상태이다. 따라서 움직임의 의미적인 기술은 SMD(Semantic Motion Description)={IS, MS, FS} 과 같이 표현할 수 있다. (그림 3.5)와 같이 meet 이벤트에 대한 연속된 움직임 상태를 meet={come toward, approach, touch}로 나타낼 수 있다.

d. 워드넷 내의 움직임 동사 수집

정지영상과 연속된 프레임에 대해 객체 관계를 분석한 결과 사람이 인지할 수 있는 공간관계는 기본적으로 connect, disconnect 와 overlap, contain 네 가지 형태임을 알 수 있었다. 이에 본 논문에서는 객체 움직임에 대한 정확한 표현을 위해 워드넷을 기반으로 정지영상에서의 공간관계 표현이 가능한 동사와 장소의 이동, 방향, 거리의 의미를 가진 동사를 중심으로 수집하였다. (표3.1)은 네 가지 형태의 기본적 공간관계를 세부적으로 설명하기 위하여 의미상 유사한 단어들로 분류했으며 각각의 단어에 대한 동의어들을 파악하여 의미적 검색에 적용한다.



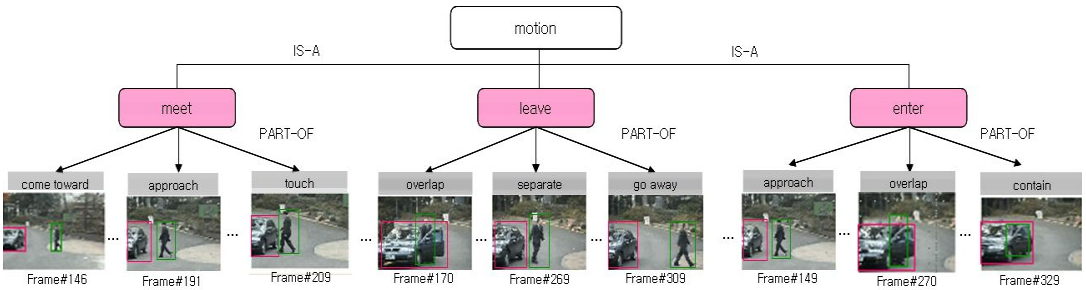
(그림 3.5) 기본적인 공관 관계

(표 3.1) 워드넷의 공간관계 동사 분류

공간관계 상태	워드넷 동사
connect	connect, touch, meet, contact, link, tie, link up, fasten, touch, adjoin, meet, contact, buss, osculate
disconnect	disconnect, come_toward, go_away, separate, approach, disjoint, enter, leave, go_across
overlap	overlap
contain	contain, include

3. 객체 움직임과 이벤트 표현을 위한 관계 구조

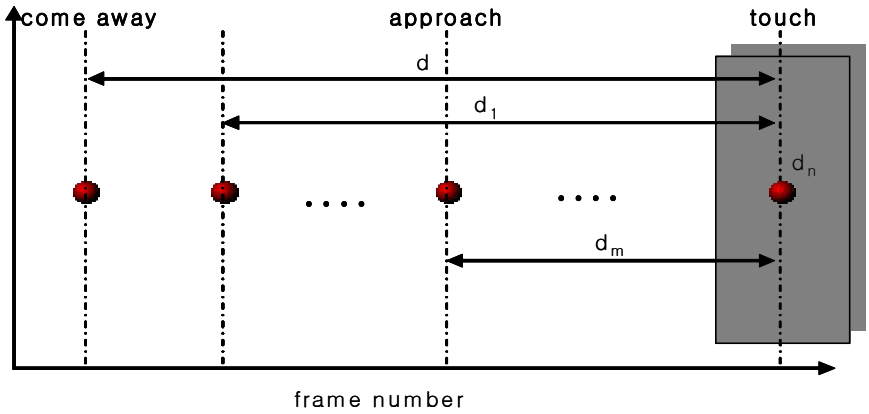
이벤트는 연속적인 프레임에 대해 움직임을 분석함으로써 인식할 수 있다. 본 논문은 객체의 움직임을 인식하여 이벤트를 검출하는데 목적이 있기 때문에 (그림 3.5)의 공간관계를 이용하여 SMD={IS, MS, FS} 형태를 통해 이벤트를 표현하기 위한 구조로 동사의 개념관계를 IS-A와 PART-OF 관계를 가진 계층구조로 나타낸다. (그림 3.6)은 meet, leave 와 enter 이벤트에 대한 계층구조를 나타내고 있다. enter의 경우 '누군가 차안으로 들어가는 장면'을 검색하기 위해서는 {come toward, approach, touch}와 같은 연속적인 패턴을 가진 움직임에 가진 Shot에 대해 검출이 가능하다.



(그림 3.6) 움직임 동사의 계층구조

enter의 경우 come toward 로부터 시작하여 approach→touch→overlap으로 진행될 수 있으나 overlap을 대표 어휘로 나타냈으며 leave는 overlap→touch→separate→go away로 진행될 수 있다. 이와 같은 이벤트에 대한 계층 구조는 사용자 중심의 고차원적 의미표현 방법이며 이러한 인간의 사고를 컴퓨터가 이해하기 위해서는 비디오 데이터의 저차원 특징과의 매핑 방안이 요구된다. 즉, 비디오 데이터의 저차원 특징 중 프레임내에 존재하는 특징 객체의 위치값과 거리, 방향 관계를 고려하여 (그림 3.6)에서 제시한 계층구조의 최하위 인스턴스와 매핑할 수 있는 규칙 정의를 통해 의미적 이벤트 추출이 가능해진다.

(그림 3.7)은 meet 이벤트에 대해 Shot내의 연속적인 프레임을 통해 모델링한 예를 보이고 있다. 움직임 객체와 정지객체가 있다고 가정할 때 첫 번째, d > d1 조건이 만족되면 come toward로 정의하고 dn =< 0 조건이면 touch로 dm은 approach로 정의한다. 여기에서 m은 중간거리를, n은 최종거리를 나타낸다.



(그림 3.7) 비디오 Shot의 이벤트 모델링

이와 같은 모델링을 통해 축구, 야구, CCTV 등의 비디오로부터 특정 장면 검색 및 비디오 제작기술 등 다양한 응용분야에서 사용될 수 있을 것으로 기대된다.

Ⅳ. 비디오 데이터의 저차원 처리

3장에서 제시한 고차원적 의미표현 모델을 통해 비디오 데이터를 추출하기· 위해서는 비디오 데이터의 저차원 특징과 매핑할 수 있어야 한다. 본 장에서는 (그림 3.1)의 프레임웍에서 제시하는 최하위의 “Low level Feature Video Raw Layer” 에 해당하는 부분으로 비디오 데이터의 저차원적 의미정보를 추출하기 위한 방법에 대해 설명한다. 먼저, 비디오 영상에 대한 전처리 과정을 소개하고 움직임 객체 추출을 위한 기존 연구에 대해 살펴본다. 마지막으로 본 논문에서 제안하는 움직임 인식에 대한 실험을 위해 특징값 추출을 위한 과정을 설명한다.

A. 비디오 데이터의 전처리

비디오 데이터의 특정 객체에 대한 컬러, 방향, 속도, 위치값 등 시각적 특징값을 추출하기 위해서는 전처리(preprocessing)단계를 거쳐야 한다. 먼저 파싱 단계를 통해 비디오 영상으로부터 Shot을 분리한다. 하나의 Shot은 시간과 공간 안에서 연속적인 동작을 표현하는 프레임을 가지게 되므로 Shot을 구분함으로써 프레임에 포함된 정보의 내용 변화를 알아낼 수 있다. 특징추출 단계는 분할된 Shot에서 이미지 분석 기법들을 사용하여 카메라 정보와 색상 히스토그램 등 주로 저차원의 정보들을 자동으로 추출하는 단계이다. 색인단계에서는 추출된 특징들을 비디오의 구조에 대한 표시자로 삽입하여 검색 시에 이용한다. 비디오 색인 기법으로는 고차원 기법과 저차원 기법이 있다. 고차원 기법은 미리 정의된 색인항목들의 집합들에 시간적인 흐름에 따라 비디오를 첨부하거나 주석을 붙이는 기법이고, 저차원 기법은 비디오로부터 움직임이나 색상과 같은 시청각적 특성을 추출하여 비디오 검색을 위한 키로 사용하는 기법이다. 메타 데이터 구축 방법에는 XML과 시맨틱 웹의 표준인 OWL/RDF(S)로 표현해주는 방법 등이 있다. (그림 4.1)은 이러한 비디오 데이터의 전처리 단계를 보여주고 있다.

(그림 4.1) 비디오 데이터의 전처리 과정

1. Shot 검출

Shot은 시간과 공간 안에서 연속적인 행동을 표현하는 연속된 프레임들 의미한다. Shot을 구분함으로써 프레임에 포함된 정보의 내용 변화를 알아낼 수 있다. 비디오 파서 알고리즘은 이러한 정보의 내용 변화를 발견하여 Shot의 경계선을 알아 낼 수 있어야 한다[42]. 비디오 데이터의 Shot 검출 방법으로는 압축되지 않은 비디오데이터에 대한 방법과 압축된 비디오 데이터를 위한 방법으로 구분할 수 있다. 압축되지 않은 비디오에 대한 방법으로는 픽셀 차에 의한 검출, 히스토그램을 이용한 검출, 에지를 이용한 검출, 비디오 형성 모델에 근거한 검출, 움직임 벡터를 이용한 검출 등이 있으며 압축된 비디오에 대한 방법에는 DCT의 DC를 이용한 검출, I, P, B를 이용한 검출, 비트열 간의 정보를 이용한 검출 등이 있다. 다음은 가장 대표적인 Shot 검출 방법인 히스토그램을 이용한 방법과 DCT의 DC를 이용한 방법, I, P, B를 이용한 검출 방법에 대해 설명한다.

a. 히스토그램에 의한 비교

히스토그램 비교법은 Shot 검출의 대표적인 방법으로 영상의 밝기나 컬러의 확률 분포도인 히스토그램을 구성한 후 히스토그램 간의 차이를 구하는 방법이다. 히스토그램 비교법에 사용되는 차이 측정 측도는 (식 4.1)과 같다.

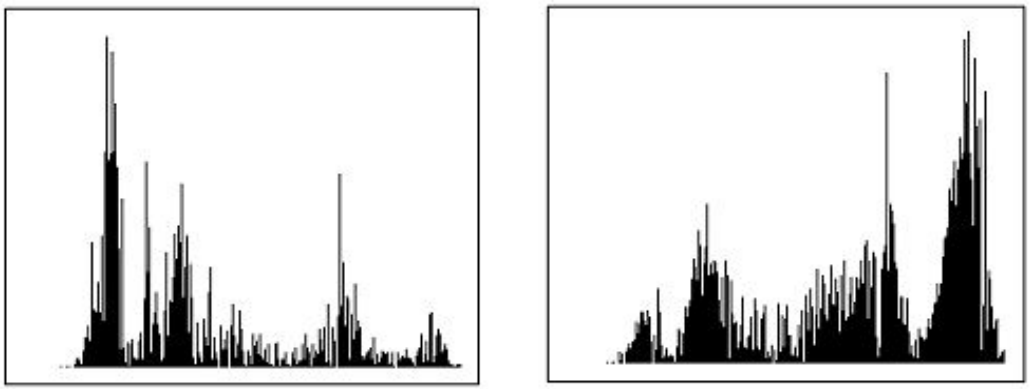
$$Di = \sum_{j=0}^{G} |Hi(j) - Hi-1(j)| > T \qquad \text{식 (4.1)}$$

D_i는 Shot 검출을 위한 차이 측정 측도이고 j는 픽셀의 밝기 구간 (0-255)을 나타내고 G는 최대값의 밝기값을 나타낸다. H_{i-1}는 i-1번째 프레임의 밝기 히스토그램이고 H_i는 i번째 프레임의 히스토그램에 해당된다. 따라서 두 히스토그램 간의 차이의 합이 임계치 T 이상인 경우가 Shot 경계지점이다. 그리고 칼라 히스토그램을 이용할 경우에는 (식 4.2)와 같이 각 R R R G B별로 히스토그램을 구성하여 차이를 계산한다.

$$Di(c) = \sum_{j=0}^{G} |Hi(c)(j) - Hi-1(c)(j)| > T \qquad \text{식 (4.2)}$$

$$Di = \sum_{c \in r, g, b} Di(c) \qquad \text{식 (4.3)}$$

히스토그램 비교법은 픽셀 밝기 차이에 의한 검출법에서 문제가 되었던 움직임에 민감하다는 단점을 어느 정도 극복하고 있다. 즉 히스토그램은 영상의 전체적인 분위기를 파악하는 것으로 영상을 이해하는데 있어서 영상내의 물체나 카메라의 움직임이 존재하는 경우라도 밝기 분포나 색상 분포는 거의 일정하기 때문이다.



(그림 4.2) 프레임 간 히스토그램 차이

히스토그램 비교법은 픽셀 밝기 차이에 의한 방법과 마찬가지로 점진적 컷을 검출하는 방법으로는 적합하지 않다. Zhang[28][29]은 점진적 컷을 검출하기 위한 방법으로 이중 비교법을 제안하였다. 이 방법에서는 두개의 임계치를 설정하고 두 프레임 간의 히스토그램 차이가 큰 임계치 T_1보다 크면 단순 컷이라 판단하고 프레임 간의 차이가 작은 임계치 T_2보다 크면 점진적 컷 후보로 설정한 후 일정 기간동안 T_2보다 큰 히스토그램 차이를 축적시킨 값이 T_1보다 크면 점진적 컷이 발생한 것으로 판단한다.

히스토그램을 이용한 차이 측정 측도는 위에서 기술한 것 이외에도 Shot 검출 과정을 수월하기 위하여 차이 값을 강조한 카이 제곱 비교법(chisquared comparison), 히스토그램 교차법(histogram intersection) 등이 있고 픽셀 간 밝기 차의 히스토그램을 구성하여 검출하는 방법도 있다.

b. DCT의 DC를 이용한 방법

한 장의 영상은 일정한 크기의 정방영역으로 나뉘어져 각 영역에 대해 변환처리를 행함으로써 영역내의 평균값(영역 전체가 같은 값을 가지는 화면성분, DC)으로부터 매우 정교한 최고주파수의 영상성분에 이르기까지 여러 주파수의 영상성분으로 분해 된다. 이 분해 과정을 “직교변환”이라고 하는데, 보다 정교한

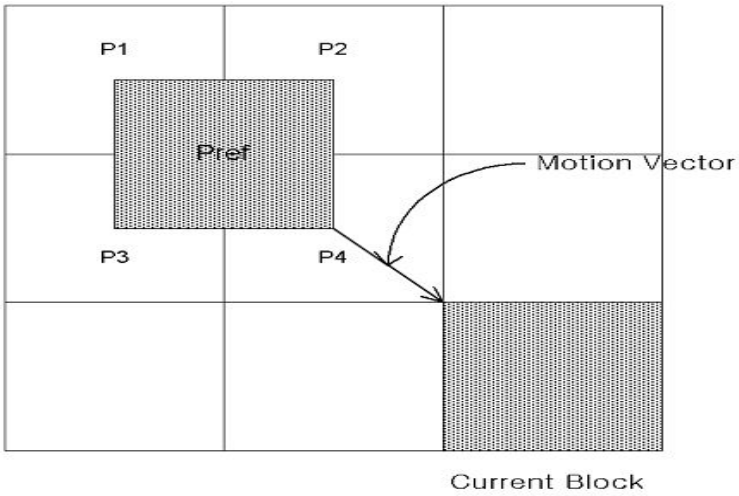
성분은 고주파인 성분이라고 이해하면 된다. 직교변환은 1960년대이래 여러 종류가 제안되었는데, 그 중 DCT(Discrete Cosine Transform)가 영상압축에 가장 효과적인 방식으로 꼽히고 있다. 직교변환(특히 DCT)의 장점은 변환 전에는 화면에 불규칙하게 퍼져 있던 화소 값(예를 들어 휘도)이 변환 후에는 저주파항 쪽으로 집중되는 경향이 있다는 것이다[30]. 따라서 고주파항들을 버리는 조작을 통해 정보손실이 거의 없이도 정보압축을 할 수 있게 되었다. MPEG이나 움직임 JPEG의 경우 8x8블록의 크기에 대해서 2차원 DCT 변환을 사용한다. 이 블록을 *B(i, j)*라 표시할 때, DC 값은 *B(0,0)*으로서 영상 *X(i,j)*의 평균적인 밝기의 8배에 해당하는 값을 가지고 있다.

$$B(0,0) = \frac{1}{8} \sum_{i=1}^{8} \sum_{j=1}^{8} X(i,j) \qquad \text{식 (4.4)}$$

DC 값은 화면 전체에 대한 정보를 담고 있으므로 이러한 DC 값을 이용하여 복원한 DC 영상 역시 원래 화면과 거의 비슷하게 나타난다. DC 값만을 이용하여 원래의 영상을 나타내면 8x8 블록을 DC 값 하나로만 나타내므로 원래 영상의 *M/8 X N/8* 크기를 갖는다. 움직임 JPEG의 경우나 MPEG의 I 프레임의 경우에 있어서 DC 영상을 복원하는 것은 각 블록의 *B(0,0)*값을 취해 8로 나누어주기만 하면 되므로 매우 간단하다. 그러나 P 프레임과 B 프레임의 경우 B(0,0)값은 이전 프레임과의 화소값의 차를 DCT한 값이므로 원래 영상의 DC 값을 나타내지 않는다. 따라서 P 프레임과 B 프레임의 경우에는 움직임 벡터에 의해 보상된 DC 값을 찾아줄 필요가 있다.

움직임 벡터에 의해 이전 영상을 복원하지 않고 현재 블록의 DC 계수를 복원하기 위해 이전 영상과 겹쳐진 블록들에 대해서 다음과 같이 구한다. 움직임 벡터가 8의 배수의 형태를 갖는다면 현재 블록은 이전 영상의 임의의 블록과 정확히 일치하여 이전 영상에서 일치하는 블록의 DC 값을 그대로 사용하면 되지만 움직임 벡터는 임의의 값을 가지므로 (그림 4.3)과 같이 4개의 블록이 겹치게 된다. 따라서 현재 영상의 블록은 이전 영상의 겹치는 네 개의 블록을 이용하여 근사화 시킬 수

있다[31].



(그림 4.3) 움직임 벡터를 이용하여 이전 영상으로부터 DC 값 복원

DC 영상은 DCT 계수중 DC 값만을 사용하므로 겹치는 네 개의 블록으로부터 DC 값만을 근사화 시키면 된다. DC 값은 움직임 벡터에 의한 이전 블록들의 DC 값과 거의 비슷한 값을 나타내므로 다음과 같이 근사화 시킬 수 있다.

$$(DCT(\mathrm{Pr}\,ef))00 = \frac{1}{64}\sum_{i=1}^{4} hi \cdot wi \cdot (DCT(Pi)00 \qquad \text{식 (4.5)}$$

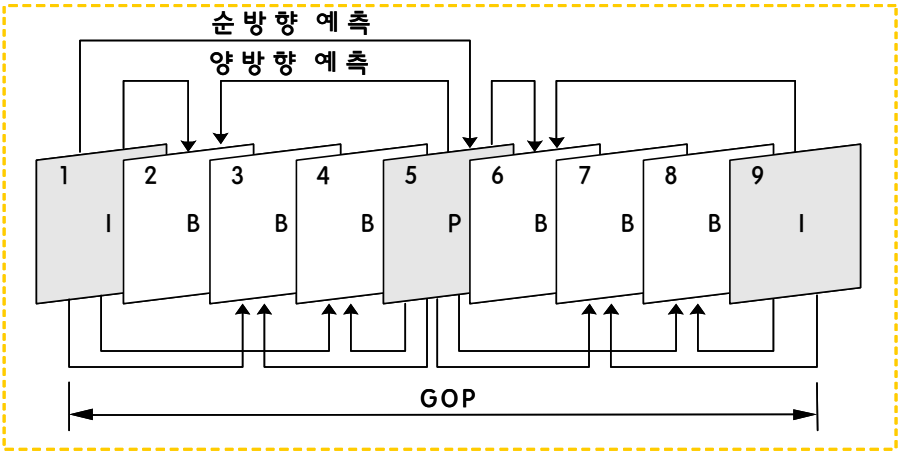
(식 4.5)에서 h_i와 w_i 는 이전 블록과 겹치는 부분의 세로와 가로의 크기를 화소단위로 나타낸 값과, $(DCT(P_{ref}))_{00}$는 근사화 하려는 블록의 DC 값, $(DCT(P_i))_{00}$ 는 이전영상에서 겹치는 블록의 DC 값을 나타낸다. 이렇게 얻은 연속적인 DC 값을 가지고 Shot을 검출한다.

c. I, P, B를 이용한 방법

움직임을 추정하는데 사용되는 참조 픽처에 따라 앞의 프레임을 참조하는 순방향

예측(forward prediction)과 뒤의 프레임을 참조하는 역방향 예측(backward prediction)이 있다.

MPEG에서는 세 가지 picture 형식을 지원하는데, 움직임 추정방법을 사용하지 않는 I picture(intra-coded picture)와 순방향 예측만을 사용하는 P picture(predictive-coded picture), 양방향 예측을 이용하는 B picture(bidirectinally predictive-coded picture)가 있다.



(그림 4.4) MPEG의 프레임 시퀀스

■ **I-Picture**

I picture는 인트라 코딩을 하고 다른 프레임을 참조하지 않는다. 장면 전환이 두 개의 연속적인 I picture에서 발생할 때 두 I picture의 전체적인 특성이 비슷한 경우라도 부분적인 특성은 다를 것이다. 그러므로 매크로 블록의 비트 율(bit rate)를 조사하면 두 개의 연속적인 프레임 사이의 차이를 검출 할 수 있다.

$$d(R,S) = \sum_{i,j} |r(i,j) - s(i,j)| \qquad \text{식 (4.6)}$$

$(R = r(i,j),\ S = s(i,j)$: 레이트메트릭스)

$$RIn = \frac{d(R_n, R_{n+1})}{T_n}$$

R_n (n=1,2, ...N+1) 는 레이트 메트릭스의 시퀀스이다. 레이트 차이의 시퀀스를 RI_n(n=1,2...N)이라 한다. $r(i, j)$는 이미지의 $(i,j)_{th}$ 매크로 블록의 비트들의 수이다.

■ **P-Picture**

P picture는 이전 P picture 또는 I picture를 참조하는 순방향 예측을 한다. 두 연속적인 P picture에서 장면 전환이 발생했으면 현재 프레임의 성분은 이전 프레임과 완전히 다르기 때문에 순방향 예측을 하기 힘들다. 그러므로 대부분의 매크로 블록들은 커다란 예측 에러를 가지기 때문에 인트라 코딩을 하게 될 것이다. 물론 현재 프레임의 비트 레이트는 상당히 증가할 것이다. 그러므로 두 연속적인 P picture에서 비트 레이트의 증가 여부를 조사함으로써 장면 전환을 검색할 수 있다. 레이트 차이의 메트릭스를 (식 4.7)과 같이 정의한다.

(RP_n, n=1,2...N)

$$RPn = \frac{d(R_n, R_{n-1})}{T_n} \qquad \text{식 (4.7)}$$

(T_n는 시퀀스의 i^{th} picture에 의해서 요구되는 비트의 총 수이다)

■ **B-Picture**

B picture는 이전과 미래의 P picture 또는 I picture를 참조하여 예측을 하게 된다. 장면 전환이 발생했을 때, 현재 프레임의 성분은 이전 프레임과 크게 다를 것이고 미래의 프레임과는 비슷한 성분을 가질 것이다. 그러므로 현재 대부분의 매크로 블록들은 미래의 P picture 또는 I picture에 의존할 것이고 역방향 예측을 할 것이다.

예측은 한 방향으로만 하기 때문에 움직임 보상의 코딩 효율은 매우 낮을 것이고

이전 B picture와 비교했을 때 비트 레이트는 증가할 것이다.

$$BP_i = \frac{N_b(i)}{N_m(i)} \qquad \text{식 (4.8)}$$

N_b와 N_m는 각각 역방향 움직임 예측 블록의 수와 i번째 프레임에서 움직임 예측 블록의 총 수를 나타낸다. 위의 I, P, B picture의 특성을 이용해서 다음과 같은 방법으로 장면 전환을 검출할 수 있다.

(1) 현재 GOP(Group of Pictures)에 있는 I picture와 다음 GOP에 있는 I picture를 비교함으로써 장면 전환을 검색할 수 있다. 만약 의심되는 picture가 발견되지 않는다면 GOP에 있는 P, B picture를 검색하지 않고 다음 GOP를 검색한다.

(2) 만약 I picture에서 비트 레이트가 증가하면 현재 GOP안에 있는 P picture를 검색하게 된다. 장면 전환으로 의심되는 picture가 없다면 현재 GOP와 다음 GOP I picture에서 비트 레이트가 증가하면 B picture를 검색하게 된다. 이때 여러 가지 상황이 발생하게 된다.

① peak가 하나 발생하면 장면 전환으로 선언한다.
② peak가 여러 개의 B picture에서 발생하면 장면 전환으로 맨 처음의 peak를 가지는 B picture를 선언한다.
③ B picture에서 장면 전환으로 의심되는 picture가 발견되지 않는다면 ②에서 발견된 P picture를 장면 전환 picture로 선언한다. 만약 P picture에서도 비트 레이트가 큰 picture가 발견되지 않는다면 다음 GOP에 있는 I picture를 장면 전환 picture로 선언한다.

GOP는 트릭모드를 실현 가능케 하기 위해 MPEG-1에 새로 부가된 구조로서 몇 장의 picture를 한 묶음으로 한 GOP를 단위로 랜덤 액세스가 가능하게 하고 있다.

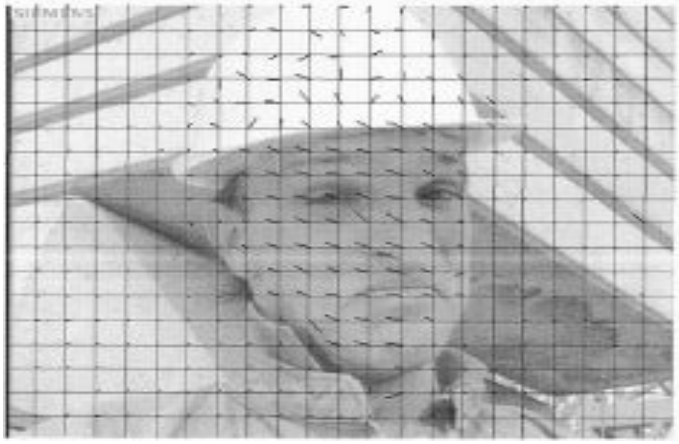
2. 움직임 객체에 대한 특징값 추출

본 논문에서는 비디오내 객체의 움직임을 기반으로 연구를 수행하기 때문에 움직임 객체에 대한 특징값 추출을 위한 기존 방법에 대해 설명한다. 비디오 데이터를 분석하여 움직이는 객체를 추출하는 데는 많은 시간이 걸린다. 압축된 비디오 데이터는 압축되지 않은 비디오 보다 계산적으로 간단하고 처리시간이 빠르다는 장점을 가지고 있다. 압축된 비디오 비트 스트림에서 제공하는 정보는 DCT 계수와 움직임 벡터, 움직임 예측 오차와 같은 이미지의 형태에 관한 많은 정보를 제공한다. 이러한 정보는 압축된 비디오 데이터의 움직임 객체를 검출하는데 이용할 수 있다[32][33][34]. 다음은 압축 비디오 신호가 가진 정보 중 매크로블록의 움직임 벡터로 2차원 히스토그램을 생성 및 분석하여 간단한 연산으로 효과적인 움직임 객체를 추출 할 수 있는 방법을 설명한다. 먼저, 압축된 영상 신호에서 움직임 벡터를 추출하여 2차원 히스토그램을 생성 및 분석해 배경영역과 객체영역을 분리한다. 분리된 객체영역에 복잡도를 검사하고 객체추출 성공으로 판단된 객체영역을 1차적인 모델로 지정하여 객체영역에 해당되는 이미지를 객체모델로 지정하고 객체영역의 16x16 매크로 블록의 개수를 저장한다. 다음으로 프레임에서 복잡도 검사 및 객체 영역 평균 블록 개수를 검사하여 하나라도 만족하지 못했을 경우에 이전 프레임에서 지정된 객체 모델과 현재 프레임간의 정합연산을 통해 객체를 추출한다.

a. 배경영역과 움직임 객체 영역의 분리

압축 영상에서 배경영역과 움직이는 객체의 영역을 분리하는 방법은 움직임 벡터를 해석함으로써 배경 움직임 벡터와 객체의 움직임 벡터를 구분할 수 있다. (그림 4.5)는 foreman 영상에 대하여 이전 프레임과 현재 프레임간의 움직임 예측과정의 전탐색 블록매칭으로 생성된 움직임 벡터를 보여주고 있다. 16x16 매크로 블록당 하나의 움직임 벡터가 생성된 것을 알 수 있는데 여기서 각각의 움직임 벡터는 움직이는 객체의 방향과 크기를 알 수 있다. 이러한 움직임 벡터의 누적치를

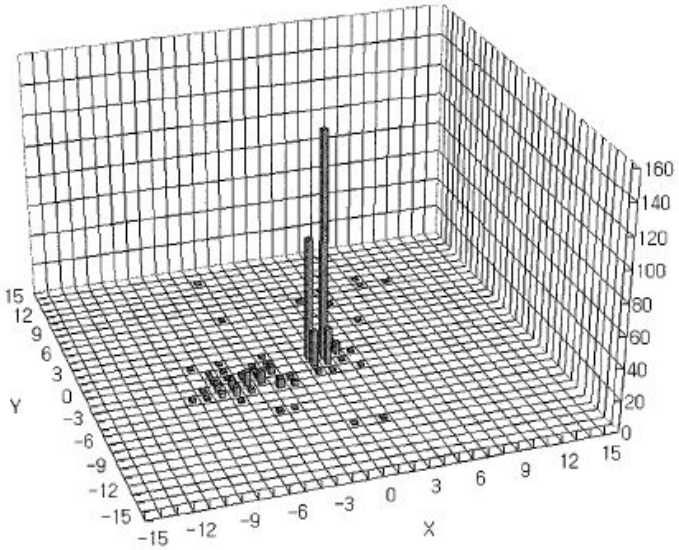
분석하여 움직이는 객체 영역과 배경 영역을 분리할 수 있다.



(그림 4.5) foreman 영상의 움직임 벡터

다음단계로 배경과 객체 영역을 분리하기 위해 움직임 벡터의 2차원 히스토그램을 구하여 가장 높은 빈도수를 갖는 움직임벡터를 배경영역의 움직임벡터로 결정한다. 이 경우 배경영역을 정하는 것은 2차원 히스토그램에서 가장 빈도수가 높은 움직임 벡터를 구하고, 이 움직임 벡터를 갖는 블록을 배경에 해당하는 블록으로 정한다. 먼저 각 블록의 움직임 벡터에 대해 2차원 히스토그램을 구한다. 구해진 2차원 히스토그램을 바탕으로 가장 빈도수가 높은 벡터를 얻을 수 있는데 이 벡터를 배경 벡터 $B(x,y)$라 하고 이 $B(x,y)$를 갖는 블록을 배경블록으로 정의하며, 배경블록들로 구성된 영역을 배경영역으로 정의한다.

배경블록을 제외한 나머지 벡터들 중 빈도수가 높은 것을 얻을 수 있는데, 이 경우 배경벡터가 아닌 나머지 벡터들을 움직이는 객체의 벡터라고 정의한다. 압축된 비디오 신호에서 매크로 블록의 움직임 벡터를 취득하여 2차원 히스토그램을 생성하고 2차원 히스토그램에서 누적치가 가장 많은 곳을 기준으로 3x3을 배경벡터라 판단하고 이러한 움직임 벡터를 포함하고 있는 영역을 배경영역으로 지정하여 배경영역을 제외한 나머지 영역에서 움직임 객체영역을 추출한다. (그림 4.6)은 실험 영상에 대해 움직임 벡트의 2차원 히스토그램을 나타낸 것이다.



(그림 4.6) 움직임 벡터의 2차원 히스토그램

(그림 4.6)에서 알 수 있듯이 0(zero) 근처에서 가장 누적치가 높게 나오며, 배경의 움직임 벡터는 (1, 0) 근처임을 알 수 있고, 배경을 제외한 나머지 벡터중에서 (-6, -2) 근처에 상당량의 누적치가 있음을 볼 수 있다. 반면 다른 곳에서도 소수의 누적치가 있으나 큰 누적치가 아니므로 배경으로 처리한다. 따라서 foreman 영상의 경우 (-6, -2)의 움직임 벡터 값을 갖는 부분이 움직임 객체일 것이라는 추측이 가능하며, 움직임 벡터의 방향과 크기로 객체 분할이 가능한 것으로 유추할 수 있다.

b. 움직임 객체의 선택

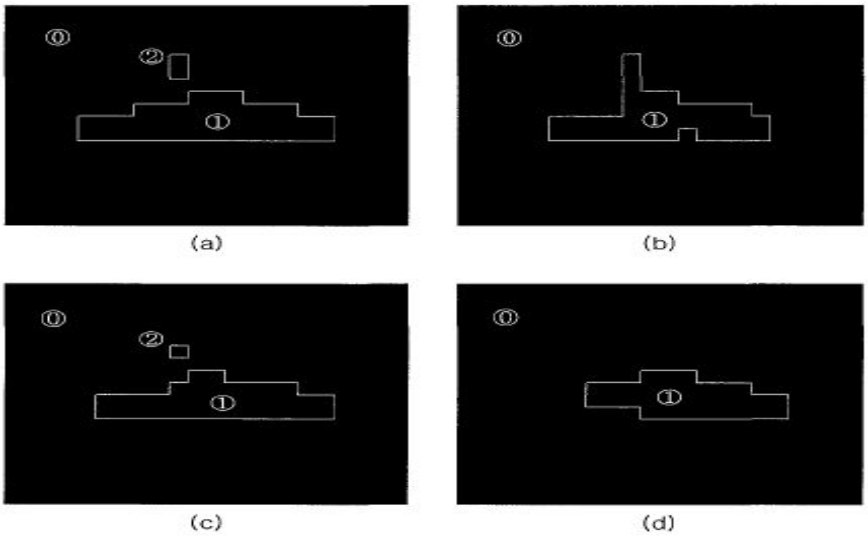
비디오 영상으로부터 배경과 움직이는 객체가 분리 되면 추출하고 싶은 움직임 객체를 선택하여 추출할 수 있어야 한다. 다음은 추출된 움직임 객체의 움직임 벡터를 이용하여 객체를 선택하여 추출하는 방법에 대해 설명한다. 움직임 벡터를 이용하여 16x16 매크로 블록 당 레이블링을 하면 각 배경블록과 움직임 객체블록에 번호가 할당된다. 매크로 블록단위에서 레이블링을 적용하면 가장 넓은 영역을 가지고 있는 배경 부분이 0번으로 할당되며 그다음으로 큰 영역이 1번, 그다음 큰

영역이 2번으로 차례대로 각 영역별 번호가 부여된다. 번호가 지정된 각각의 움직임 객체 중에서 추출을 원하는 움직임 객체를 선택하면 움직임 객체의 움직임 벡터와 밝기 정보를 이용하여 선택한 움직임 객체를 추출하게 된다. (그림 4.7)은 매크로 블록단위 레이블링을 보여주고 있다. 여기서 배경 영역은 0, 객체는 1과 2의 번호로 각각 할당되어 영역을 구분 지을 수 있기 때문에 각각의 객체 영역을 선택할 수 있다.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	2	2	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	2	2	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	2	2	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

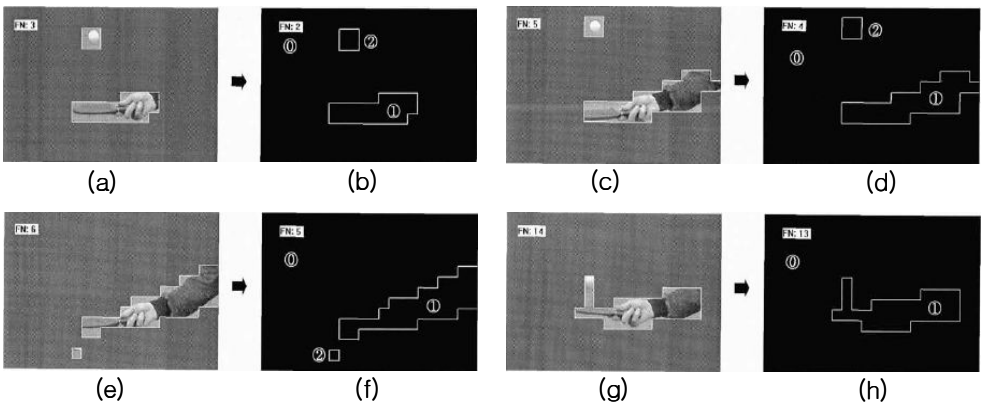
(그림 4.7) 매크로 블록 단위 레이블링

(그림 4.8)은 입력영상(coastgurd)에서 배경과 분리된 객체 영역의 움직임 벡터를 이용해서 분할된 움직임 객체에 대한 레이블링 결과 영상을 나타내고 있다. (a), (c)는 두 개의 움직임 객체가 존재하는 것을 나타내는데 가장 큰 영역이 배경이므로 0번으로 할당되고 그 다음 가운데에 보이는 두 번째 큰 영역이 1번, 그 다음 세 번째 영역이 2번으로 각각 할당된다. (b), (d)는 영역 중 가장 큰 곳을 차지하고 있는 배경역역이 0번으로 할당되고 두 번째 큰 영역이 1번으로 각각 할당되어 선택이 가능하게 된다.



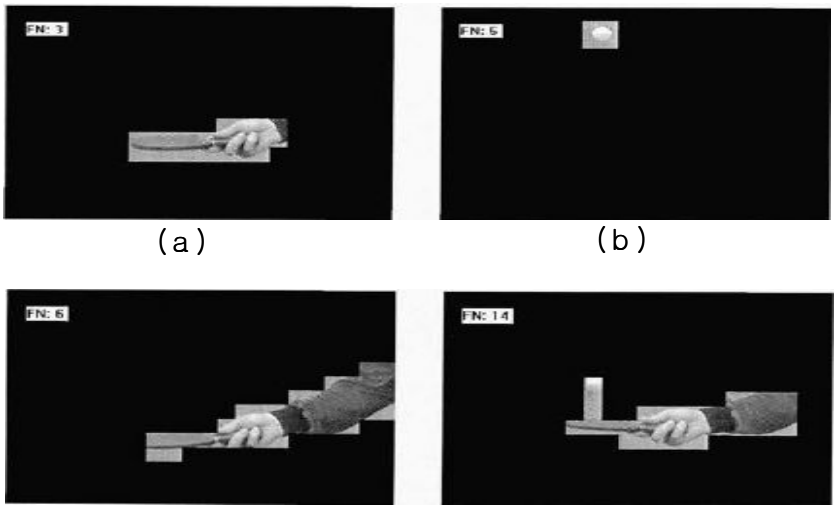
(그림 4.8) 입력영상에 대한 레이블링 한 영상

(그림 4.9)는 탁구 실험 영상을 입력하여 배경 영역과 객체 영역을 분리한 다음 레이블링을 하는 과정을 보여주고 있다. (a)는 움직이는 객체가 두 개임을 알 수 있는데 이 영상을 레이블링을 했을 때 역시 배경을 제외한 두 개의 영역이 (b)와 같이 각 영역에 대하여 번호로 정해진다. 따라서 번호로 지정된 영역을 선택하면 번호에 따른 객체 영역이 선택되어 선택된 객체 영역만 추출이 가능해진다. (b)는 (a), (d)는 (c), (f)는 (e), (h)는 (g)를 각각 레이블링하여 나타난 결과 영상이다.



(그림 4.9) 각 영역별로 번호가 지정되어 있는 탁구 영상

움직임 객체를 선택하게 되면 레이블링이 되어있는 각 객체 영역의 번호에 따라서 선택되어진 부분만 객체 검출이 가능하게 된다. (그림 4.10)은 이와 같은 연산 과정에 의해서 선택된 객체 영역만 검출된 영상을 부여주고 있다. (그림 4.9)와 (그림 4.10)를 연관해서 보면, (그림 4.9)(b)에서 객체 영역 1번을 선택한 부분이 (그림 4.10)(a)의 영상이다. (그림 4.10)(b)는 (그림 4.9)(b)에서 탁구공에 해당되는 2번 객체 영역을 선택하여 나타난 결과이고 (그림 4.10)(c), (그림 4.10)(d)도 각각 (그림 4.9)(f), (그림 4.9)(h)의 1번 객체 영역을 선택하여 선택한 객체 영역만 나타난 영상임을 알 수 있다.



(a) (b)

(c) (d)

(그림 4.10) 객체 영상 중에 선택된 객체만 나타난 영상

c. 움직임 객체의 유효성 검사와 모델 생성

선택된 움직임 객체에 대해 배경과 분리된 움직임 객체영역이 움직임 객체인지에 대한 유효성 검사 단계를 거친다. 유효성 검사 단계는 첫째 움직임 객체의 경계부의 공간적 복잡도를 조사하여 복잡도가 임계치보다 클 경우에 객체에서 제외시킨다.

둘째 장면전환이 있기 전까지 프레임에서 추출되어진 움직임 객체가 계속 존속한다고 가정하고 이전 프레임까지 추출된 움직임 객체 블록의 평균 개수를 구하여 검사하는 프레임에서 객체가 없다고 판별 될 때 이전 프레임까지 추출되어진 객체의 밝기정보를 이용하여 검사하는 프레임과 일치하는지 MAD(Mean Absolute Difference) 정합연산을 하여 판별한다. 이러한 과정을 통해 유효성이 결정되었을 때 객체 추출 판단 기준과 일치하거나 판단 기준 안에 포함된다면 객체 추출 성공, 아니면 객체 추출 실패로 간주하고 다음 프레임에 대한 움직임 객체 추출을 계속한다.

이러한 배경 비율 검사와 복잡도 검사 그리고 객체 블록 평균 검사를 거쳐 움직임 객체 추출로 성공한 프레임에 대해서 객체 영역을 추출하여 객체 모델로 지정하고 다음 프레임에서 움직임 객체 추출을 계속 시도하게 된다. 만약 현재 프레임에서 움직임 객체 추출이 성공한다면 이전 프레임에서 성공한 움직임 객체의 블록 개수와 현재 프레임 객체 블록의 개수를 비교하여 더 많은 블록의 개수를 가지고 있는 프레임에서 추출된 움직임 객체를 객체 모델로 지정하게 된다. 즉, 전체적인 장면 변화가 있지 않는 이상 객체 모델은 좀 더 완벽한 객체의 형태로 바뀌게 된다. 지정된 객체 모델은 움직임 객체 추출 검사에 만족하지 못한 경우에 정합 연산을 수행하기 위한 모델 영상이다. 움직임 정합 연산으로 템플레이트 정합(template matching)을 사용하여 움직임 객체를 추출할 수 있다. 템플레이트 정합에는 보통 MAD(Mean Absolute Difference)와 MSD(Mean Square Difference)가 있다. MAD는 차이의 절대 값을 더하고 MSD는 차이의 제곱 값을 더하는 방법이다. 템플레이트 영상과 겹쳐진 부분의 영상 밝기 특징이 서로 비슷하다면 MSD나 MAD 값이 0에 가까운 값이 나오고 그렇지 않을 경우 값이 커지기 때문에 0에 가까운 값이 나올 때 영상의 위치를 알 수 있다. 그러나 MSD보다 MAD의 연산 처리량이 적기 때문에 보통 MAD를 많이 사용한다. 다음은 움직임 객체 부분의 영상과 움직임 객체 추출이 실패했을 때의 영상을 서로 겹쳐놓고 대응되는 픽셀의 밝기정보를 서로 빼서 차이값을 더하는 방법으로 검사하고자 하는 영상과 겹쳐진 부분의 영상 밝기값이 서로 비슷하다면 MAD값은 0에 가까운 값이 나올 것이고 서로 다르면 두 값은 커질

것이다. 이에 대한 MAD는 (식 4.9)와 같다.

$$MAD(dx, dy) = \frac{1}{MN} \sum_{i=0}^{M} \sum_{j=0}^{N} |P(x_i, y_i) - G(x_j + dx, y_j + dy)| \qquad \text{식 (4.9)}$$

dx, dy = search position of motion vector

$P(x_i, y_i)$ = MxN macroblock position

$G(x_i, y_i)$ = MxN reference macroblock position

위의 M과 N은 영상의 가로 세로 크기이고 P와 G는 각각 템플레이트 영상과 템플레이트 영상에 겹쳐질 영상이다. 다시 말하면 P는 이전 프레임에서 움직임 객체 추출이 성공한 움직임 객체 영역부분만 추출된 움직임 객체 모델이고 G는 움직임 객체 추출이 실패했을 때의 현재 프레임이다.

3. 메타데이터 생성

비디오는 저장 공간을 줄이기 위해 압축되어, 프레임(frame)의 순서대로 저장된다. 저장된 데이터는 가공되지 않은 또는 해석되지 않은 원시 데이터이다. 모든 비디오 데이터는 자신에 대한 정보를 메타데이터를 통하여 표현하고, 사용자는 어떠한 질의를 수행하더라고 메타데이터를 통하여 실제 데이터에 접근하게 되므로 메타데이터는 사용자와 비디오 데이터를 연결하는 매개자 역할을 수행한다. 메타데이터는 미디어 객체들에 대해 특징 추출 기능들을 적용함으로써 생성할 수 있다. 이 특징 추출 기능들은 내용 의존적인(content dependent), 내용 묘사적(content descriptive), 내용 독립적인 기법들을 사용한다.

a. 내용 의존적 메타데이터

이 형식의 메타데이터는 비디오 정보의 가공되지 않은 특징을 표현하다. 비디오

프레임들의 순서에 대한 내용 의존적인 메타데이터로는 카메라의 동작, 카메라 높이, 밝기 정도, 비디오 순서 안에서의 객체들의 트랙이 있다. 개별적인 프레임에 대한 메타데이터는 색상 히스토그램과 같은 프레임 특징들로 구성된다. 비슷한 방법으로 회색조 이미지를 표현하기 위하여 회색 단계 히스토그램이 사용된다.

b. 내용 묘사적 메타데이터

비디오 프레임들의 순서에 대한 내용 묘사적인 메타데이터로는 카메라 Shot의 거리, Shot 각도, Shot의 움직임, 행위 설명, Shot내의 객체들의 형식 등이 있다. 개별적인 프레임에 대한 메타데이터는 프레임의 밝기, 색상, 질감, 프레임 안의 객체들의 형식, 객체에 대한 서술 등과 같은 특징들로 구성된다.

c. 내용 독립적 메타데이터

이 메타데이터는 비디오 전체에 적용 가능한 특징들을 기술한다. 이 메타데이터에는 비디오의 제작 날짜, 제작자 이름, 감독의 이름, 비디오에 들어간 경비 등이 포함된다.

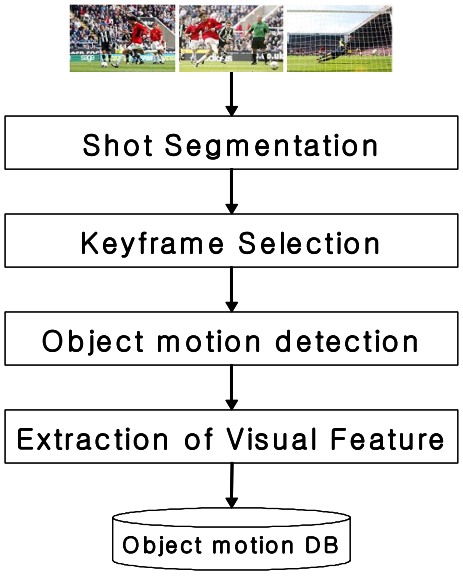
(표 4.1)은 MPEG 비디오 영상에 대한 메타데이터로 XML를 이용하여 표현되었다.

(표 4.1) MPEG 비디오의 메타데이터 형식


```xml
<?xml version="1.0" encoding="iso-8859-1" ?>
- <Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
    xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
  - <Description xsi:type="ContentEntityType">
    - <MultimediaContent xsi:type="VideoType">
      - <Video>
        - <TemporalDecomposition>
          - <VideoSegment>
            - <MediaTime>
                <MediaTimePoint>T00:00:00:0F30000</MediaTimePoint>
                <MediaIncrDuration
                  mediaTimeUnit="PT1001N30000F">150</MediaIncrDuration>
              </MediaTime>
            - <TemporalDecomposition>
              - <VideoSegment>
                - <MediaTime>

                      <MediaTimePoint>T00:00:02:14074F30000</MediaTimePoint>
                  </MediaTime>
                </VideoSegment>
              </TemporalDecomposition>
            </VideoSegment>
          </TemporalDecomposition>
        </Video>
      </MultimediaContent>
    </Description>
  </Mpeg7>
```

B. 제안된 움직임 인식을 위한 전처리 과정

본 절에서는 비디오 데이터의 객체 움직임 인식에 관한 실험을 위해 수행된 비디오 움직임 인식 전처리 과정에 대해 기술한다. 실험에 사용될 비디오 영상은 정지객체와 움직임 객체를 포함하고 있으며 다음과 같은 단계를 거쳐 특징값을 추출한다.



(그림 4.11) 움직임 객체 특징값 추출을 위한 전처리 과정

1. Shot 분할과 키 프레임 추출

본 논문의 핵심은 사용자의 관심이 높은 이벤트를 포함한 Shot 단위의 영상을 검색하는데 있다. 따라서 비디오 영상을 Shot 단위로 분할하여 객체 움직임 인식을 위한 특징값을 추출한다. Shot 분할과 프레임 추출을 위해 IBM에서 제공하는 “VideoAnnEx Annotation Tool”을 사용하였다. 사용된 Shot 분할 기법은 MPEG video에 사용되는 DCT 블록기반 알고리즘이 적용되었다. 사용된 Tool이 제공하는

기능은 Shot 분할과 프레임 추출 외에도 주석(annotation)을 위한 각종 속성정보를 제공한다. (그림 4.12)는 정적 객체(car)와 움직임 객체(person)를 포함하고 있는 비디오에 대해 분할된 Shot과 프레임을 나타낸다.

(그림 4.12) Shot 분할과 프레임 추출

2. 움직임 객체의 특징값 추출

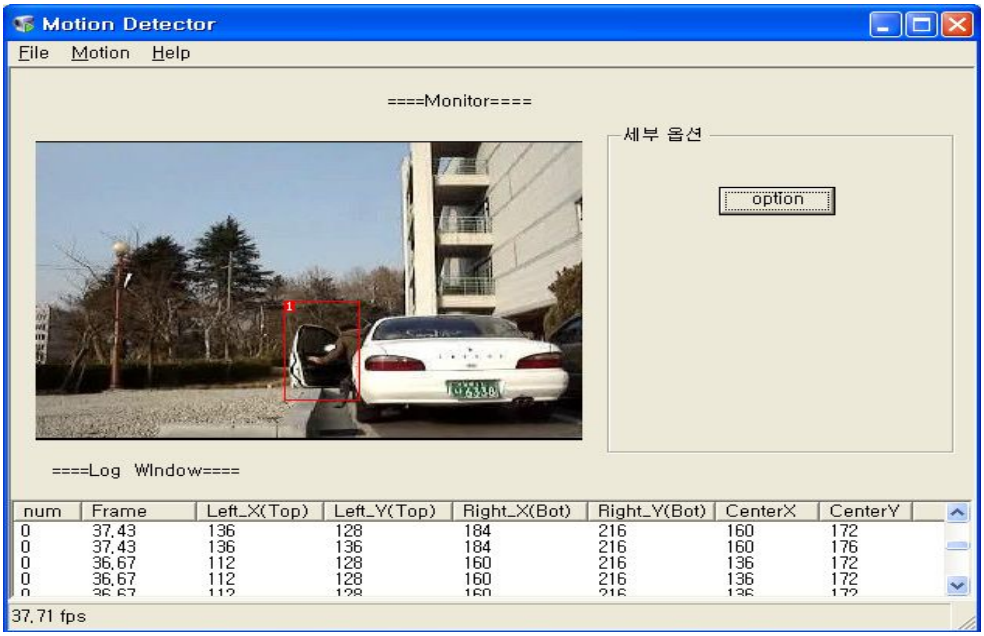
움직임 객체의 특징값 추출을 위해서는 비디오에서 특정 객체의 분할과 움직임 예측에 대한 작업이 이루어져야 한다. 본 논문에서는 배경과 움직이는 객체를 분리하여 이들의 컬러 값의 차이를 이용하는 Andrew Kirillov[20]의 객체 추출 알고리즘을 사용하였다. 하지만 알고리즘의 성격상 트래킹을 중심으로 설계되었으며 본 논문에서는 객체 움직임에 따른 위치값을 추출하는데 주안점이 있기 때문에 다음과 같은 몇 가지 조건에 대한 사항을 고려하여 알고리즘을 수정함으로써 객체 움직임 추출이 가능했다.

(표 4.2)에서 먼저, 한 프레임에서 인식할 수 있는 객체수를 지정했으며(5~8행), 움직임에 대한 최소, 최대영역(12~19행)을 주었다. 주어진 영역을 최초 인식된 객체로 인정하고 다음 인식된 객체와 비교한다(34~41행). 두 영역의 관계가 overlap 또는 touch 상태가(38~41행) 되면 초기 인식된 객체에 대한 움직임으로 인정하고 두 번째 인식된 영역을 다시 기준영역으로 변경한 후(42~48행) 위치에 대한 로그를 저장한다. 이와 같은 과정을 반복하면서 마지막 움직임까지 계속 인식된 상태에 대한 위치값을 로그로 저장하여 제안된 규칙을 적용한 움직임 인식에 사용된다.

(표 4.2) 객체 움직임에 대한 MBR 추출 알고리즘

```
1   // draw each rectangle
2
3   foreach ( Rectangle rc in rects )
4    {
5     // 객체 인식 개수 지정
6       if ((n < ack_row - 2) && (rc.Width > 50) && (rc.Height > 50) && (rc.Width < 300) &&
7                                                                          (rc.Height < 250))
8          {
9           //motion.MainForm.ret_Mhandle().define_num
10          int pt = rc.Y - rc.Height
11
12         // 초기 객체 영역 설정 부분
13           if (init == 0)
14             {
15                memRc[var_num].X = 0;  // 최소 기준 점 X좌표
16                memRc[var_num].Y = 0;  // 최소 기준점 Y좌표
17                memRc[var_num].Width = 300;  // X축 최대 크기
18                memRc[var_num].Height = 200; // Y축 최대 크기
19              }
20
21              int ptm = memRc[var_num].Y - memRc[var_num].Height
22
23        // 움직임 객체 위치 영역 인식 부분
24       Recognize_Rec.RelationInstance RI =new Recognize_Rec.RelationInstance(
25             (uint)memRc[var_num].Left ,
26             (uint)memRc[var_num].Y,
27             (uint)memRc[var_num].Right,
28              (uint)ptm,
29              (uint)rc.Left,
30              (uint)rc.Y,
31              (uint)rc.Right,
32              (uint)pt
33                );
34  // 처음 인식된 객체의 MBR값과 초기 객체 영역의 MBR값 비교
35
36       Recognize_Rec.STATE RSST = RI.result_pattern();
37
38          if (
39             (RSST == Recognize_Rec.STATE.OVERLAP) ||  // 두 영역이 overlap 상태인지
40             (RSST == Recognize_Rec.STATE.TOUCH))       // touch 상태인지 체크
41             {
42   // 초기 위치값과 다음 위치값의 관계가 overlap, touch 상태이면 객체의 움직임으로 인정
43       하고 이전 좌표값 로그로 저장
44
45                memRc[var_num].X = rc.X
46                memRc[var_num].Y = rc.Y
47                memRc[var_num].Width = rc.Width
48                memRc[var_num].Height = rc.Height

                                  ……
```

(그림 4.13)은 이와 같은 알고리즘을 적용해 실제 비디오 영상에 대한 객체 움직임을 탐지한 좌표값에 대해 로그를 만드는 과정을 보이고 있다. 로그 형태는 우측 영상에 대한 프레임번호, MBR, 프레임 시간 등의 정보를 추출한다. 이와 같이 추출된 값은 대표프레임에 대한 값만을 취하여 프레임 단위의 특징값 데이터베이스를 생성한다. 이는 본 논문에서 제안하는 객체 관계에 대한 규칙을 적용하는데 사용된다.



(그림 4.13) 움직임 객체의 특징값 추출 과정

Ⅴ. 제안된 모델을 이용한 움직임 인식과 이벤트 검출

본 논문에서 제시하는 움직임 인식과 이벤트 검출 방법은 제 3장에서 제시하였던 고차원적 움직임 모델을 저차원적 특징과 매핑하기 위해 정지객체와 움직임 객체 사이의 관계에 대한 규칙을 통해 움직임 패턴을 인식하는 방식이다. 본 논문에서는 움직임 인식 규칙 정의를 위해 두개의 객체, 움직임 객체와 정지객체에 대해 정의한다. 움직임 패턴에 대한 분석을 위해 1차적으로 각각의 프레임에 대해 정지상태의 객체관계를 인식한다. 2차적으로 각 프레임에 대한 세부적인 움직임 인식을 위해 이전 프레임과 이후 프레임의 객체 관계를 통해 최종적인 움직임을 인식한다. 이와 같이 각 프레임에 대한 움직임에 대해 영어의 동사와 매핑하고 Shot이 가진 이벤트 검출을 위해 연속적인 프레임의 움직임 패턴을 분석한다.

A. 객체 움직임 인식을 위한 규칙

비디오 데이터의 객체 움직임 인식을 위한 규칙 정의를 위해 제 4장에서 추출한 객체의 좌표값을 이용해 두 객체가 가지는 좌표값의 차를 구한다음 차에 대한 일정한 거리를 통해 각각의 프레임에 대해 (그림 3.6)의 최하위 인스턴스와 매핑한다. 이와 같이 인스턴스와 매핑할 수 있는 규칙으로는 먼저 단일 프레임에서 인식 가능한 규칙을 구하고 연속 프레임에 대한 전·후 패턴을 적용한 규칙을 정의한다. 이와 같은 두 단계를 거쳐 인식된 움직임에 대해 각 프레임별로 움직임 인덱싱을 수행한다.

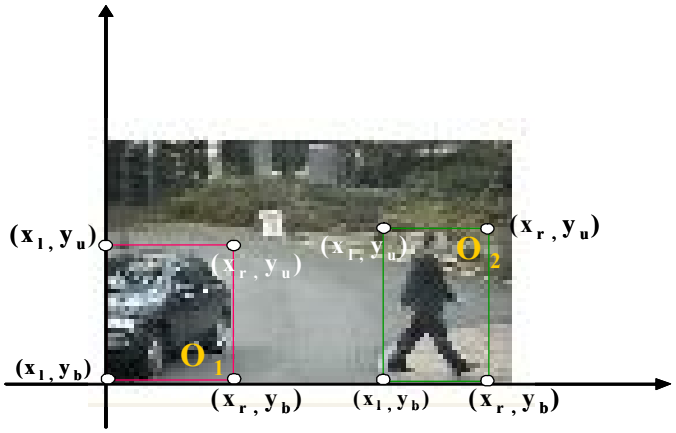
1. 객체 위상관계에 의한 움직임 규칙 정의

본 논문에서 정의하고자 하는 객체 움직임 인식을 위한 규칙을 위해 다음과 같은

몇 가지 조건을 적용한다.

- [out 조건]
  - 객체관계가 서로 떨어져 있는 상태
  - Separate, go away, approach, stop 어휘를 포함함
- [toucht 조건]
  - 각 객체의 접촉면이 맞닿아 있는 상태
- [In 조건]
  - 두 객체가 서로 포함관계에 있는 상태
- [overlap 조건]
  - 두 객체관계가 교집합이 존재하는 상태
- [separate 조건]
  - 이전 프레임 상태가 touch상태로 인식되며 현 상태는 바로 후 상태인 조건
  - 첫프레임 에서는 절대로 나타날 수 없는 상태
- [approach 조건]
  - 움직임 객체가 고정객체 쪽으로 다가가고 있는 상태
- [go away 조건]
  - 움직임 객체가 고정객체로부터 멀어지고 있는 상태
- [come toward 조건]
  - 첫프레임에서 나타날 수 있으며 다음 프레임을 고려했을 때 고정객체 쪽으로 다가가려 하는 순간의 상태

객체 움직임 규칙을 위한 요소로서 정지객체 O_1과 움직임 객체 O_2에 대한 공간적 거리를 이용하고자 하기 때문에 두 객체의 거리 산출을 위해 MBR에 대한 4개의 좌표점을 (그림 5.1)과 같이 표현한다. 즉, 좌측 상단 좌표를 (x_1, y_u), 좌측 하단 좌표를 (x_1, y_b)라고 표현하고, 우측 상단 좌표를 (x_r, y_u), 우측 하단 좌표를 (x_r, y_b)라고 표현한다. 두객체의 거리 산출을 위한 기준점 좌표값은 0으로 한다.



(그림 5.1) 규칙정의를 위한 객체의 MBR 표현

a. 단일 프레임에서 인식 가능한 규칙

단일 프레임에서 인식 가능한 움직임은 out, in 상태와 touch, overlap 상태이며 이를 위한 규칙은 다음과 같다.

(표 5.1) 단일프레임에 대한 객체 관계에 의한 움직임 규칙

```
IF(두 객체의 가로의합 == 두 객체중 가장 큰 X좌표와 가장 작은 X좌표의 차)
{
        IF(두 객체의 세로의 합) < 두 객체중 가장 큰 Y좌표와 가장작은 Y좌표의 차)
                STAT : OUT
        ELSE
                STAT : TOUCH
} //IF
ELSE IF(두 객체의 가로의 합 < 두 객체중 가장 큰 X 좌표와 가장 작은 X 좌표의 차)
        STAT : OUT
ELSE // 두 객체의 가로의 합 > 두 객체 중 가장 큰 X 좌표와 가장 작은 X 좌표의 차
{
        IF(왼쪽 상단 좌표와 오른쪽 하단의 2좌표를 비교하여 어느 한쪽이 다른 한쪽을 포함)
                STATE : IN
        ELSE
        {
                IF(두 객체의 가로의 합) == 두 객체중 가장 큰 X좌표와 가장 작은 X 좌표의 차)
                    STATE : TOUCH
                ELSE IF(두 객체의 가로의 합 < 두 객체의 가장 큰 X좌표와 가장 작은 X 좌표의 차)
                    STATE : OUT
                ELSE
                    STAT : OVERLAP
                    }
}//ELSE
```

[규칙 1] (표 5.1)에서 만약 두 객체의 MBR에 대한 가로의 합과 두 객체 중 가장 큰 X좌표와 가장 작은 X 좌표의 차가 같다면 out 상태와 touch 상태를 정의하기 위해 다음을 비교한다. 두 객체의 세로의 합이 두 객체 중 가장 큰 Y 좌표와 가장 작은 Y 좌표의 차와 비교하여 작으면 out 상태, 크면 touch 상태로 정의한다.

[규칙 2] 만약 두 객체의 가로의 합이 두 객체 중 가장 큰 X 좌표와 가장 작은 X 좌표의 차보다 작다면 out 상태로 정의하고, 두 객체의 가로의 합이 두 객체 중 가장 큰 X 좌표와 가장 작은 X 좌표의 차보다 크면 [규칙 3]을 정의한다.

[규칙 3] 왼쪽 상단 좌표와 오른쪽 하단의 2좌표를 비교하여 어느 한쪽이 다른 한쪽을 포함한다면 in 상태로 정의하고 그렇지 않으면 다음 조건을 비교한다.

만약 두 객체의 가로의 합과 두 객체 중 가장 큰 X 좌표와 가장 작은 X 좌표의 차가 같다면 touch 상태로 정의하고, 두 객체의 가로의 합이 두 객체의 가장 큰 X좌표와 가장 작은 X 좌표의 차보다 작다면 out 상태로 정의한다. 최종적으로 두 객체의 가로의 합이 두 객체의 가장 큰 X 좌표와 가장 작은 X 좌표의 차보다 크면 overlap 상태로 정의한다. (표 5.2)는 이와 같은 단일 프레임 상에서의 객체 관계에 대한 규칙을 알고리즘으로 나타낸 것이다.

(표 5.2) 단일프레임 규칙에 대한 알고리즘

```
ENUM STATE { OUT, TOUCH, OVERLAP, GOAWAY, SEPARATE, IN,APPROACH, STOP}
 STATE STATIC_ACK()
{
    IF{O1(region_w)+O2(region_w)==Max[O1(xr),O2(xr)]-Min[O1(xl),O2(xl)]}
    {
        IF(O1(region_h)+O2(region_h)< Max[O1(yu),O2(yu)]-Min[O1(yb),O2(yb))
            STATE : OUT
        ELSE
            STATE : TOUCH
    }//IF
    ELSE IF{O1(region_w)+O2(region_w)<Max[O1(xr),O2(xr)]-Min[O1(xl),O2(xl)]}
        STATE : OUT

    ELSE // {O1(region_w)+O2(region_w)>Max[O1(xr),O2(xr)]-Min[O1(xl),O2(xl)]}
    {
        IF[O1(region)⊆O2(region))or O1(region)⊇O2(region)]
        STATE : IN
        ELSE
            {
            IF{O1(region_w)+O2(region_w)==Max[O1(xr),O2(xr)]-Min[O1(xl),O2(xl)]}
                STATE : TOUCH
            ELSE IF{O1(region_w)+O2(region_w)<Max[O1(xr),O2(xr)]-Min[O1(xl),O2(xl)]}
                STATE : OUT
            ELSE
                STATE : OVERLAP
            }
    }//ELSE
}//FUNC
```

b. 연속 프레임에서 인식 가능한 규칙

연속 프레임에서 인식 가능한 움직임은 단일 프레임 상에서 out으로 인식된 상태에 대해서 하나의 Shot에 대한 첫 번째 프레임에서 마지막 프레임까지 두 객체의 공간적 거리를 비교하면서 다음과 같은 규칙을 정의한다.

(표 5.3) 연속프레임에서 인식 가능한 움직임 규칙

```
    IF(현재 상태 == OUT)
    {
        IF(프레임  == 첫번째 프레임)
        {
        IF {(현상태의 객체간 중심점 Distance) > (다음 프레임의 객체간 중심점 Distance) }
                STATE : COME AWAY
        IF {(현상태의 객체간 중심점 Distance) < (다음 프레임의 객체간 중심점 Distance) }
                STATE : GO AWAY
        ELSE
                STATE : STOP
    } //IF(첫번째 프레임일 경우)

     IF(프레임 == 마지막 프레임)
      {
        IF(이전 상태 == touch)
                STATE : SEPARATE
        ELSE
                STATE : OUT
      }//IF(마지막 프레임일 경우)
     ELSE // 중간 프레임일 경우
      {
              IF (전상태가 touch 였다면)
                STATE : SEPARATE
                IF {(현상태의 객체간 중심점 Distance) < (다음 프레임의 객체간 중심점 Distance)}
                STATE : GO AWAY
        IF {(현상태의 객체간 중심점 Distance) > (다음프레임 객체간 중심점 Distance) }
                STATE : APPROACH
                ELSE
        {
                IF(현상태의 두객체의 거리 > 다음프레임 상태의 두객체의 거리)
                        STATE : APPROACH
                        ELSE IF(현상태의 두객체의 거리 < 다음프레임 상태의 두객체의 거리)
                        STATE : GO AWAY
                ELSE STOP
    } } }IF(두 객체의 가로의합 == 두 객체중 가장 큰 X좌표와 가장 작은 X좌표의 차)
{
        IF(두 객체의 세로의 합) < 두 객체중 가장 큰 Y좌표와 가장작은 Y좌표의 차)
              STAT : OUT
        ELSE
              STAT : TOUCH
} //IF
ELSE IF(두 객체의 가로의 합 < 두 객체중 가장 큰 X 좌표와 가장 작은 X 좌표의 차)
        STAT : OUT
ELSE // 두 객체의 가로의 합 > 두 객체 중 가장 큰 X 좌표와 가장 작은 X 좌표의 차
{
        IF(왼쪽 상단 좌표와 오른쪽 하단의 2좌표를 비교하여 어느 한쪽이 다른 한쪽을 포함)
              STATE : IN
        ELSE
        {
              IF(두 객체의 가로의 합) == 두 객체중 가장 큰 X좌표와 가장 작은 X 좌표의 차)
                STATE : TOUCH
              ELSE IF(두 객체의 가로의 합 < 두 객체의 가장 큰 X좌표와 가장 작은 X 좌표의 차)
                STATE : OUT
              ELSE
                STAT : OVERLAP
                }
}//ELSE
```

(표 5.3)의 규칙은 크게 프레임이 첫 번째 프레임인 경우와 마지막 프레임인 경우에 대한 세부적인 규칙을 정의한다.

[규칙 1] 첫 번째 프레임인 경우 현 상태의 가장 큰 X 좌표와 가장 작은 X 좌표의 차가 마지막 상태 객체의 가장 큰 X 좌표와 가장 작은 X 좌표의 차보다 큰 경우 come toward 상태로 정의한다.

만약, 현 상태의 가장 큰 X 좌표와 가장 작은 X 좌표의 차와 마지막 상태 객체의 가장 큰 X 좌표와 가장 작은 X 좌표의 차가 같은 경우 go away 상태로 정의하며 그렇지 않은 경우는 stop 상태로 본다.

[규칙 2] 프레임이 마지막 프레임이면서 이전 상태가 touch 라면 separate 상태로 정의하고 아니면 out 상태로 판단하고 다음 조건을 비교한다.

현 상태의 가장 큰 X 좌표와 가장 작은 X 좌표의 차가 마지막 상태 객체의 가장 큰 X 좌표와 가장 작은 X 좌표의 차보다 작은 경우 go away 상태로 정의하며, 현 상태의 가장 큰 X 좌표와 가장 작은 X 좌표의 차가 마지막 상태 객체의 가장 큰 X 좌표와 가장 작은 X 좌표의 차보다 큰 경우 approach 상태로 정의한다.

마지막으로 위 두 가지 규칙 go away와 approach 조건이 아니면 현 상태의 두 객체의 거리가 마지막 상태의 두 객체의 거리보다 크면 approach, 현 상태의 두 객체의 거리가 마지막 상태의 두 객체의 거리보다 작으면 go away로 정의하고 종료한다.

(표 5.4)는 이와 같은 연속 프레임 상에서의 객체 관계에 대한 규칙을 알고리즘으로 나타낸 것이다.

(표 5.4) 연속 프레임 규칙에 대한 알고리즘

```
STATE JUDGESTATE(arg Frame_num)
{
 IF(PST== OUT)
 {   IF(Frame_num==1)
   {
    IF{(PST.Max[O1(xr),O2(xr)]-PST.Min[O1(xl),O2(xl)])>(FST.Max[O1(xr),O2(xr)]-FST.Min[O1(xl),O2(xl)])}
      STATE : COMEAWAY
    IF{(PST.Max[O1(xr),O2(xr)]-PST.Min[O1(xl),O2(xl)])==(FST.Max[O1(xr),O2(xr)]-FST.Min[O1(xl),O2(xl)])}
      STATE : GOAWAY
    ELSE
      STATE : STOP //(NO changed Coordinate)
   }//IF(Frame_num==1)

   IF(Frame_num==Frame_end)
   {      IF(Previos_state==touch)
          STATE : SEPARATE
    ELSE
          STATE : OUT
   }//IF(Frame_num==Frame_end)
   ELSE
   {
     IF(Previos_state==touch)
       STATE : SEPARATE

    IF{(PST.Max[O1(xr),O2(xr)]-PST.Min[O1(xl),O2(xl)])<(FST.Max[O1(xr),O2(xr)]-FST.Min[O1(xl),O2(xl)])}
      STATE : GOAWAY

    IF{(PST.Max[O1(xr),O2(xr)]-PST.Min[O1(xl),O2(xl)])>(FST.Max[O1(xr),O2(xr)]-FST.Min[O1(xl),O2(xl)])}
      STATE : APPROACH
     ELSE
      {
      IF[PST.Distence(O1,O2)>FST.Distence(O1,O2)]
            STATE :  APPROACH
      ELSE IF[PST.Distence(O1,O2)< FST..Distence(O1,O2)]
            STATE : GOAWAY
      ELSE STOP //(NO changed Coordinate)
      }

   }//IF(ELSE)
 }//IF(PST == OUT)
}//JUDGESTATE
```

2. 이벤트 검출을 위한 규칙

본 논문은 각각의 프레임에 대한 객체 움직임 인식을 통해 사용자에게 이벤트와 관련된 Shot을 검색할 수 있도록 하는 것이 최종 목표다. 본 절에서는 연속된 움직임 패턴에 따른 이벤트 규칙을 정의하고 정의된 규칙에 대한 유효성을 실험하기 위해 좌표값에 대한 시뮬레이션을 수행한다.

1. 이벤트에 대한 움직임 패턴

다음은 하나의 Shot에 대한 이벤트를 인식하기 위해 첫 프레임에서 마지막 프레임까지의 움직임 패턴을 분석하여 meet, enter, leave 이벤트에 대한 규칙을 정의한다.

MEET 이벤트 : 먼저 come toward 상태를 반복적으로 체크하여 동일한 움직임에 대해선 하나의 움직임으로 인식한다. 다음으로 approach상태를 반복체크한 후 다음 움직임이 touch로 인식되면 meet 이벤트로 정의한다.

ENTER 이벤트 : 첫 번째 움직임이 approach 상태인가를 체크하고 다음으로 touch||overlap 상태를 반복체크한 후 다음 움직임이 contain 으로 인식되면 enter 이벤트로 정의한다.

LEAVE 이벤트 : 첫 번째 움직임이 overlap 상태인가를 체크하고 다음으로 separate 상태를 반복체크한 후 다음 움직임이 go away 로 인식되면 leave 이벤트로 정의한다.

(표 5.5)는 이와 같은 이벤트 규칙을 알고리즘으로 나타낸 것이다.

(표 5.5) 이벤트 규칙에 대한 알고리즘

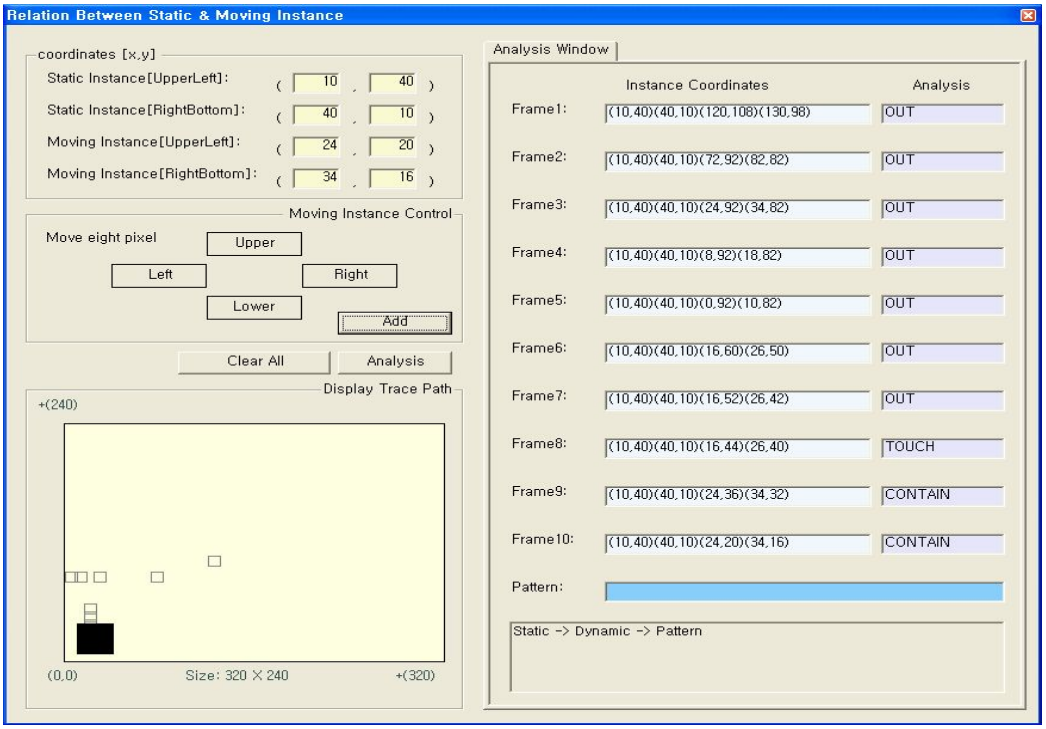
```
//i: 상태표기(샷에대한 시작프레임을 0을 기준으로 한다.)

FOR(i = 처음 프레임 ; i < 전체 프레임-1 ; i++)
{
   switch( TransState[i] )
   {
            case COMEAWAY:
            while(TransState[i] == COMEAWAY)i++;//COMEAWAY의 반복상태를 한상태로 인식한다.
            if(TransState[i] == APPROACH) //만약 다음상태가 APPROACH와 같다면
            {
                        while(TransState[i] == APPROACH)i++; //APPROACH의 반복상태를 한상태로 인식한다.
                        if(TransState[i] == TOUCH) //만약 다음상태가 TOUCH상태와 같다면
                        STATE : MEET //touch의 반복상태는 파악할 필요없이 바로 MEET로 파악하며 패턴파악을
                                    종료
                        else
                        STATE : STATIC_LAST_STATE //만약 COMEAWAY->APPROACH까지 파악되었으나
                        다음정적파악이 패턴과 다르다면 마지막 정적 프레임 인식상태를 찍는다.
            }
                        else
                        STATE : STATIC_LAST_STATE //만약 come toward가 파악되었으나 다음 정적파악상태에
                        실패한다면 마지막 정적 프레임 인식상태를 찍는다.
                        break
                        case 1:
                        while(TransState[i] == APPROACH)i++;
                        if(TransState[i] == OVERLAP)
                        {
                                    while(TransState[i] == OVERLAP)i++;
                                                if(TransState[i] == COMTAIN)
                                                STATE : ENTER
                                                else
                                                STATE : STATIC_LAST_STATE
                        }
                        else
                                    STATE : STATIC_LAST_STATE
                                    break
                        case 6: //overlap
                        while(TransState[i] == OVERLAP)i++;
                        if(TransState[i] == SEPARATE)
                        {
                                    while(TransState[i] == SEPARATE)i++;
                                    if(TransState[i] == GOAWAY)
                                                STATE : LEAVE
                                    else
                                                STATE : STATIC_LAST_STATE
                        }
                        else
                                    STATE : STATIC_LAST_STATE
                        break
                        default:
                        STATE : STATIC_LAST_STATE // 처음정적상태가 패턴익식의 첫 정적프레임상태와 다르다면
                                                / 마지막 상태를 찍는다.
                        break

                                                }
                                    }
```

B. 움직임 객체에 대한 규칙 적용

본 절에서는 제안된 객체 움직임 인식에 대한 규칙에 대한 유효성 테스트를 한다. 이에 따른 방법으로 10개의 프레임으로 구성된 움직임 좌표값을 통해 좌, 우, 위, 아래 방향으로 움직임을 발생시킨 후 제안된 규칙을 적용하여 움직임 인식에 대한 시뮬레이션을 수행한다. 또한 인식된 움직임에 대해 비디오 Shot 데이터에 인덱싱하기 위한 구조를 설명한다.

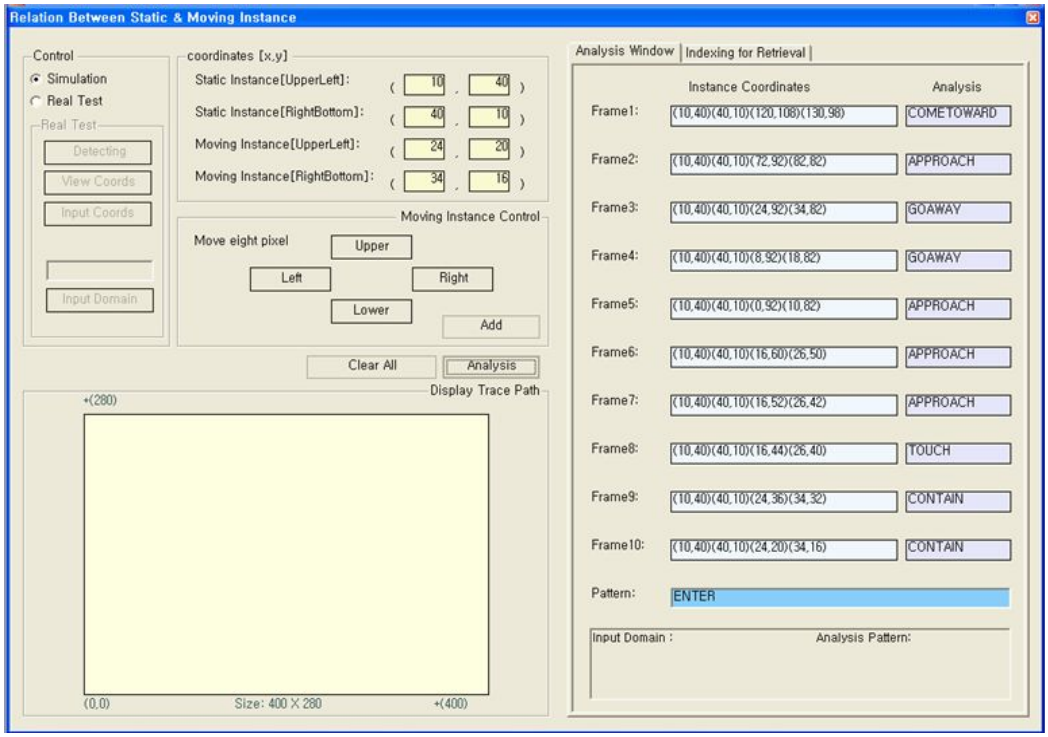
1. 정의된 규칙에 의한 시뮬레이션

앞에서 정의한 움직임 규칙과 이벤트 규칙에 대한 유효성 실험을 위해 10개의 프레임에 대해 객체 위치의 좌표값을 입력하여 시뮬레이션을 수행했다. (그림 5.2)는 각각의 단일 프레임에 대한 움직임 상태를 보여주고 있다. 고정 객체의 초기 MBR값과 이동 객체의 초기 MBR값을 입력한 후 left, right, lower, upper 방향으로 움직임을 발생시킨다. 취득된 좌표값에 대한 움직임은 움직임 객체가 고정 객체 방향으로 다가가고 있는 것을 알 수 있다. 좌표값에 의해 인식된 움직임 중 Frame1~Frame7 까지는 out 상태이므로 연속된 프레임에 대한 규칙을 적용함으로써 최종 움직임 인식이 가능하다.



(그림 5.2) 단일 프레임에 대한 움직임 인식

(그림 5.3)은 위 그림의 out 상태인 프레임에 대해 연속된 전·후 프레임에 대한 패턴을 분석한 후 최종적인 움직임과 이벤트를 나타내고 있다. 각각의 프레임에 대해 come toward, approach, go away, approach, touch, contain 상태로 인식되었으며 최종 Shot에 대한 이벤트 패턴은 Frame5~Frame10까지의 움직임 패턴이 approach, touch, contain 상태로 이벤트는 Enter로 인식되었다.

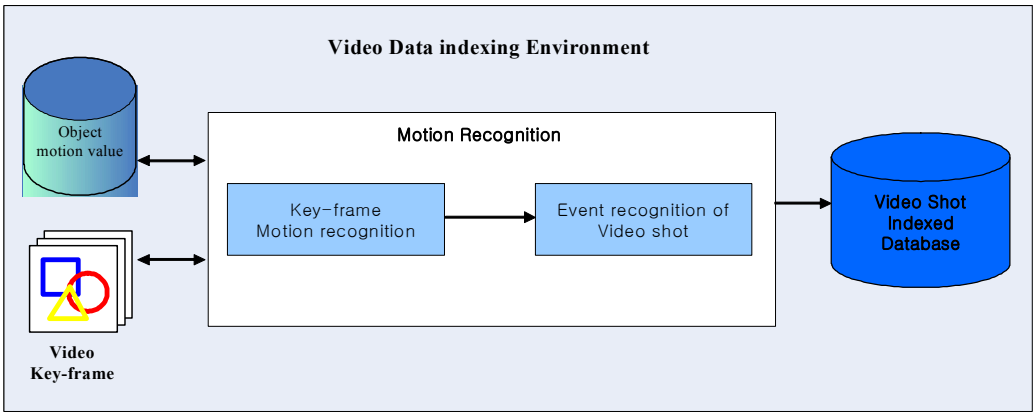


(그림 5.3) 연속프레임에 대한 움직임 인식과 이벤트 검출

2. 움직임 인식에 의한 비디오 데이터 인덱싱

비디오 정보를 검색하고 비디오 데이터의 효율적인 조작을 위해 비디오 데이터의 특징정보를 고려한 적절한 인덱싱 기법이 필요하다. 기존에 연구된 대부분의 기법은 영상내의 패턴 변화나 색상, 질감, 영상에서의 개체 모양 등과 같은 저차원적 특징에 기반을 둔 인덱싱과 고차원적 방법으로 주석기반 비디오 인덱싱을 수행하고 있다. 비디오 인덱싱은 사용자가 원하는 조건을 만족하는 비디오를 신속하고 정확하게 찾는데 목적이 있다. 하지만 저차원의 인덱싱 정보는 사용자의 다양한 질의 패턴에 대응하지 못할 뿐만 아니라 meet, enter 등의 이벤트와 같은 의미적 검색을 수행할 수 없다. 본 논문에서는 의미적 검색이 가능한 인덱싱을 위해 위에서 제시된 움직임 규칙을 적용하여 각각의 프레임에 대한 움직임 인식을 실험한 후 최종적으로 비디오 Shot이 지닌 의미적 이벤트를 인식함으로써 각각의 Shot에 대한 인덱싱을 수행한다.

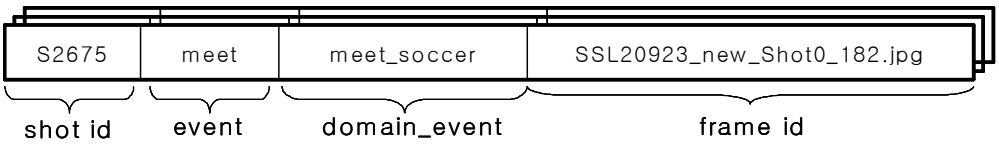
이러한 인덱싱 정보는 온톨로지 접근을 위해 OWL/RDF(S) 형식의 메타데이터로 표현되며 의미적 검색을 수행하는데 사용된다.



(그림 5.4) 비디오 데이터 인덱싱 구조

(그림 5.4)는 비디오 데이터에 대한 인덱싱 과정을 나타내고 있으며, Shot 데이터 파일의 자료 구조를 C언어의 구조체로 표현한 후 실제 데이터 파일에 저장하는 예를 보이면 (그림 5.5)와 같다. 인덱스 파일의 구조는 Shot 식별자, 인식된 이벤트, 도메인 정보, 링크 프레임 식별자로 구성된다.

```
struct_motion{
   int    shot_id;                          // shot 식별자
   char semantic_event [maxname]  //  shot이 지닌 이벤트
   char event_domain                  // 온톨로지 적용을 위한 이벤트 도메인
   RecID frame_id;                    // 링크 프레임 시작번호
} shot_mot;
```



(그림 5.5) 비디오 Shot 데이터의 인덱싱 예

C. 온톨로지 적용 비디오 검색 기법

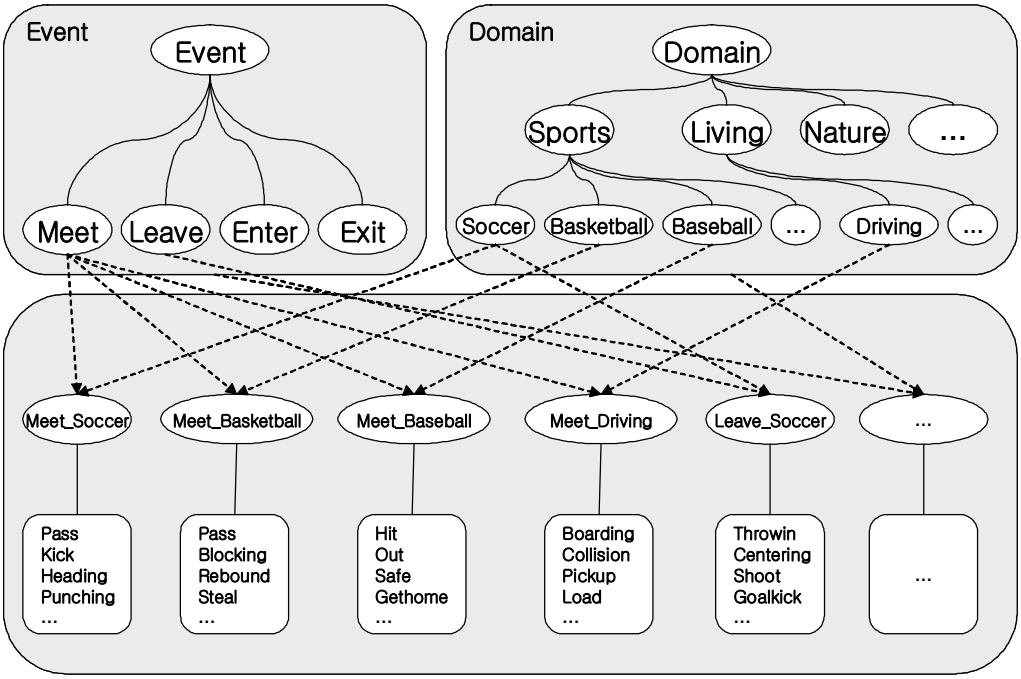
최근 비디오 데이터의 의미적 인식을 위해 특정 시맨틱 웹의 핵심 요소인 온톨로지를 활용하는 방법이 부각되고 있다[37][38]. 온톨로지는 유형 무형의 개념, 객체 등의 자원을 의미적으로 상세하게 기술하는 명세서로서, 자원의 개념을 나타내는 클래스, 클래스들이 갖는 특징 또는 다른 자원과의 관계를 정의하는 속성 그리고 클래스를 이용해 실체를 정의하는 인스턴스로 나뉜다. 이러한 온톨로지 구조로 자원들을 체계적이고 명백하게 정의함으로써 의미적인 정보검색이 가능하고 단어들 사이에 존재하는 의미적 모호성을 해결할 수 있는 강점이 있다.

본 절에서는 비디오 데이터에 대한 의미적 검색을 위해 상위 온톨로지와 하위 온톨로지를 이용한다. 상위 온톨로지는 검색에서 의미적인 접근을 제시하기 위한 것으로 일반적인 이벤트의 기준이 되는 4가지 이벤트(meet, enter, leave, exit)를 상위 개념으로 객체들의 모션에 대한 이벤트를 각 도메인별로 기술하고 있고, 하위 온톨로지는 비디오 Shot의 메타데이터로서 비디오의 도메인과 객체들의 이벤트를 기술한다. 이를 위해, 본 논문에서 제안하는 움직임 규칙을 통해 인식된 움직임에 대한 이벤트들을 파악하고, 비디오의 Shot 데이터들에 대한 인덱싱을 수행한다. 그리고 인덱싱된 비디오 데이터에 대해 온톨로지 구축을 통하여 추론에 의한 의미적 검색을 위한 기반을 제시한다.

1. 이벤트 온톨로지 구축

비디오 데이터 내에 존재하는 객체들의 움직임을 기술하기 위해 여러 응용 영역에 걸쳐서 실제 사용되는 이벤트들을 조사하였다. 먼저 도메인을 스포츠, 생활, 자연으로 크게 나누고, 이들 도메인을 구성할 수 있는 클래스로는 스포츠 종목, 사람의 일상생활에서 행해지는 것들 그리고 야생 동물이나 자연현상 등으로 각각 기술하고 있다. 예를 들면, 스포츠 종목 중의 하나인 축구에서 객체들 사이에서 발생할 수 있는 이벤트를 4가지(meet, leave, enter, exit) 중 meet(한 객체가 다른 객체에 다가와서 만남)

에 대한 것들을 살펴보면 Pass(같은 편으로부터 공을 넘겨받은 경우), Kick(선수들이 공으로 달려가 발로 차는 경우), Heading(선수들이 머리를 이용하여 공을 치는 경우), Punching(골키퍼가 공을 막기 위해 손으로 공을 치는 경우) 등이 있을 수 있고, leave(한 객체가 다른 객체로부터 분리됨)에 대한 것들은 Centering(선수가 득점을 위해 상대방 골대 근처에 있는 같은 팀 선수에게 공을 차는 경우), Throwin(아웃된 공을 손으로 던지는 경우), Shoot(득점을 위해 슛을 쏘는 경우), Goalkick(골키퍼가 공을 차는 경우) 등이 있다. 이처럼 축구와 같은 스포츠 종목을 스포츠 도메인에 분류하고, 각 종목에서 발생할 수 있는 이벤트들을 조사하여 인스턴스를 둠으로써 (그림 5.6)와 같은 이벤트 온톨로지 구조를 형성할 수 있었다.



(그림 5.6) 이벤트 온톨로지 구조

(그림 5.6)에서 상위개념 계층과 하위개념 계층으로 나눠지는데, 상위개념 계층에는 4가지 이벤트와 이들 이벤트의 도메인으로 구성하고, 하위개념 계층으로는 각 도메인에서 발생할 수 있는 이벤트들을 4가지로 구분하여 기술하였다.

이렇게 분류된 비디오의 도메인과 이벤트에 대해 구체적으로 (표 5.6)에서 보이고 있다.

(표 5.6) 이벤트에 대한 도메인별 동사 수집

event	class	class event
meet	meet_soccer	pass, kick, heading, punching, ...
	meet_basketball	pass, blocking, rebound, steal, ...
	meet_baseball	hit, out, safe, gethome, catch, ...
	meet_cctv	collision, accident, fight, ...
	meet_driving	boarding, collision, pickup, load, ...
	meet_animal	shooting, hunting, landing, ...
leave	leave_soccer	throwin, shoot, centering, goalkick, ...
	leave_basketball	no goal, shoot...
	leave_baseball	home run, pitch, hit, ...
	leave_driving	passing, ...
	leave_animal	takeoff, runaway, escape...
enter	enter_soccer	shoot, goal, heading, goal, offside, ...
	enter_basketball	shoot, goal, ...
	enter_cctv	building_in, car_in, elevator_in, ...
exit	exit_cctv	building_out, car_out, elevator_out, ...

이와 같이 수집된 각 도메인에 대한 이벤트는 사용자의 질의어를 의미적으로 처리할 수 있는 기반이 될 수 있고, 비디오의 저차원 처리를 통해 확보된 공간관계와 이들 공간관계를 이용하여 4가지 이벤트를 추출하는 움직임 규칙을 적용하여 고차원의 의미적 검색을 적용할 수 있다. 본 논문은 이처럼 저차원의 가공되지 않은 데이터를 이용하여 자동으로 이벤트를 추출하고, 이를 이용하여 의미적 검색까지 적용하는 것으로 저차원과 고차원의 의미적 차이(semantic gap)을 줄이는데 의미가 있다.

2. 비디오 의미적 검색을 위한 OWL/RDF(S) 표현

OWL/RDF(S)는 시맨틱 웹을 실현하기 위해 W3C(World Wide Web Consortium)에서 2004년에 표준으로 제정한 웹 온톨로지 언어이다[37]. RDF(Resource Description Framework)는 정보를 구조적으로 표현하는 XML과 달리, 객체의 관계를 지향하는 모델을 사용하여 표현할 정보들을 객체단위로 나누어 표현하고, RDF 스키마(scheme)를 이용하여 설정된 객체 관계의 의미적 해석이 가능하다.

객체로는 웹페이지, 웹페이지 내에 표현되는 개념들 등으로 모든 것이 가능하고, 웹페이지를 객체로 보고 웹페이지의 내용을 RDF로 표현하기 위해서는 기본적으로 제목, 저자, 최종 수정일, 저작권, 주제 등과 같은 웹 문서에 관한 메타데이터를 간단하게 표현할 수 있다. OWL(web ontology language) 은 사람을 위해 정보를 표현하는 것에 그치지 않고 정보의 내용을 사람의 지식체계와 흡사하게 묘사하기 위한 언어로 RDF의 확장판이라 할 수 있다.

OWL은 RDF에서 자원들을 기술하기 위해 정의한 어휘들을 포함하여 논리적 관계(교집합, 합집합, 보집합) 및 구체적인 속성 정의를 할 수 있도록 어휘를 확장하였으며 OWL과 RDF(S) 어휘는 보통 함께 쓰인다. OWL과 RDF(S)를 이용하여 표현된 온톨로지는 컴퓨터가 처리하기 위한 구조적인 형태로 표현되며 이를 트리플(triple)이라 한다. 트리플의 구조는 사람과 사람이 대화할 때 의미전달을 위해 기본적으로 필요한 주어, 서술어, 목적어 형태를 따라 주어-서술어-목적어로 표현된다. 표현된 서술어에 따라서 주어와 목적어의 관계가 표현되고 또한 다양한 추론이 가능하다.

이러한 특징들을 갖는 OWL/RDF(S)를 이용하여, 본 논문에서는 (그림 5.6)의 온톨로지 계층구조를 의미적 검색에 활용하기 위해 온톨로지로 표현한다. (표 5.7)은 이벤트에 대한 온톨로지를 OWL/RDF(S)로 표현한 일부분을 나타낸다.

(표 5.7) 이벤트에 대한 OWL/RDF(S) 표현

```
<rdf:RDF
 xml:base= "http://vector.chosun.ac.kr/ontology/Event#"
 xmlns:owl= "http://www.w3.org/2002/07/owl#"
 xmlns:rdf= "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs= "http://www.w3.org/2000/01/rdf-schema#"
 xmlns:xsd= "http://www.w3.org/2001/XMLSchema#">

...

<owl:Class rdf:ID="Event"/>
<owl:Class rdf:ID="Domain"/>
<owl:Class rdf:ID="Meet">  <rdfs:subClassOf rdf:resource="#Event" />
</owl:Class>

<owl:Class rdf:ID="Leave">  <rdfs:subClassOf rdf:resource="#Event" />
</owl:Class>

<owl:Class rdf:ID="Enter">  <rdfs:subClassOf rdf:resource="#Event" />
</owl:Class>

<owl:Class rdf:ID="Exit">  <rdfs:subClassOf rdf:resource="#Event" />
</owl:Class>

<owl:Class rdf:ID="Sports">  <rdfs:subClassOf rdf:resource="#Domain" />
</owl:Class>

<owl:Class rdf:ID="Soccer">  <rdfs:subClassOf rdf:resource="#Sports" />
</owl:Class>

<owl:Class rdf:ID="Meet_Soccer">
     <owl:intersectionOf rdf:parseType="Collection">
       <owl:Class rdf:about="#Meet" />
       <owl:Class rdf:about="#Soccer" />
   </owl:intersectionOf>
</owl:Class>

<Meet_Soccer rdf:ID="Pass"/>
<Meet_Soccer rdf:ID="Heading"/>
<Meet_Soccer rdf:ID="Punching"/>
...
```

(표 5.7)은 이벤트 온톨로지의 앞부분만을 보이고 있으며, 객체들의 상/하위 관계를 위해 'rdfs:subClassOf'를 이용하였고 두 객체의 교집합을 이용하여 하나의 새로운 객체를 생성하기 위해 'owl:intersectionOf'를 사용하였다. OWL/RDF(S) 어휘들은 내부적으로 추론을 위한 규칙들을 정의하고 있어 온톨로지에서 표현하고 있지 않는 객체들의 관계를 추론을 통하여 파악할 수 있는 강점이 있다.

또한, 앞 절에서 각 프레임을 분석하여 비디오 내에 객체들 사이의 이벤트를 Shot 단위로 인식할 수 있었다. 인식된 Shot의 이벤트들을 이용하여 의미적 검색에 적용하기 위해 각 Shot에 대한 온톨로지를 작성한다. 각 Shot에 대한 정보, 객체들 사이의 이벤트 정보를 온톨로지로 표현하기 위해, (표 5.8)과 같이 클래스 'Shot'과 Shot에서 발생한 이벤트를 기술할 수 있는 'hasShotEvent'와 이벤트 추출 어플리케이션을 통해 이벤트 추출 시 설정한 비디오의 응용 영역을 기술할 수 있는 'hasShotDomain'을 온톨로지에 작성하였다.

(표 5.8) 비디오 Shot에 대한 클래스 정보

이 름	특 징	
Shot	rdf:type	owl:Class
	owl:onProperty	hasShotEvent
	owl:cardinality	1
	owl:allValuesFrom	Event:Event
	owl:onProperty	hasShotDomain
	owl:cardinality	1
	owl:allValuesFrom	Event:Event
hasShotEvent	rdf:type	owl:ObjectProperty
	rdfs:domain	Shot
	rdfs:range	Event:Event
hasShotDomain	rdf:type	owl:ObjectProperty
	rdfs:domain	Shot
	rdf:range	Event:Domain

(표 5.8)에서 클래스 'Shot'은 'owl:Class'의 유형으로 속성 'hasShotEvent'과

'hasShotDomain'을 갖고 값으로는 'Event:Event'내의 인스턴스를 각 1개씩 취한다. 이들 어휘를 이용하여 축구 비디오에서 인식한 'meet'에 대한 이벤트를 표현하면 (표 5.4)와 같다.

(표 5.9) 비디오 Shot에 대한 이벤트를 위한 OWL/RDF(S)

```
<rdf:RDF
 xml:base= "http://vector.chosun.ac.kr/ontology/Shot#"
 xmlns:owl= "http://www.w3.org/2002/07/owl#"
 xmlns:rdf= "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs= "http://www.w3.org/2000/01/rdf-schema#"
 xml:Event= "http://vector.chosun.ac.kr/ontology/Event#"
 xmlns:xsd= "http://www.w3.org/2001/XMLSchema#">

<owl:Ontology rdf:about="">
  <rdfs:label>Event Ontology</rdfs:label>
</owl:Ontology>

<Event:Shot rdf:ID="ID000513">
    <Event:hasShotEvent rdf:resource="#Event:Meet" />
    <Event:hasShotDomain rdf:resource="#Event:Soccer" />
</Event:Shot>

</rdf:RDF>
```

Ⅵ. 실험 및 결과 분석

본 장에서는 비디오 데이터의 객체 움직임 인식을 위한 제안된 규칙의 유용성을 보이기 위해 정지객체와 움직임 객체로 구성된 비디오 영상을 실험데이터로 선정하여 성능 평가를 수행했다. 성능 평가를 위해 본 논문에서 제안된 모델링을 이용한 움직임 인식에 대한 실험을 통해 정확율을 평가한다. 실험을 통해 인식된 움직임에 대한 최종 결과로 이벤트 검출에 대한 평가를 수행하고, 온톨로지를 이용한 의미적 검색을 수행한다. 이를 위해 먼저, 실험 환경과 성능 평가를 위해 사용된 실험 데이터를 기술한다.

A. 실험 환경

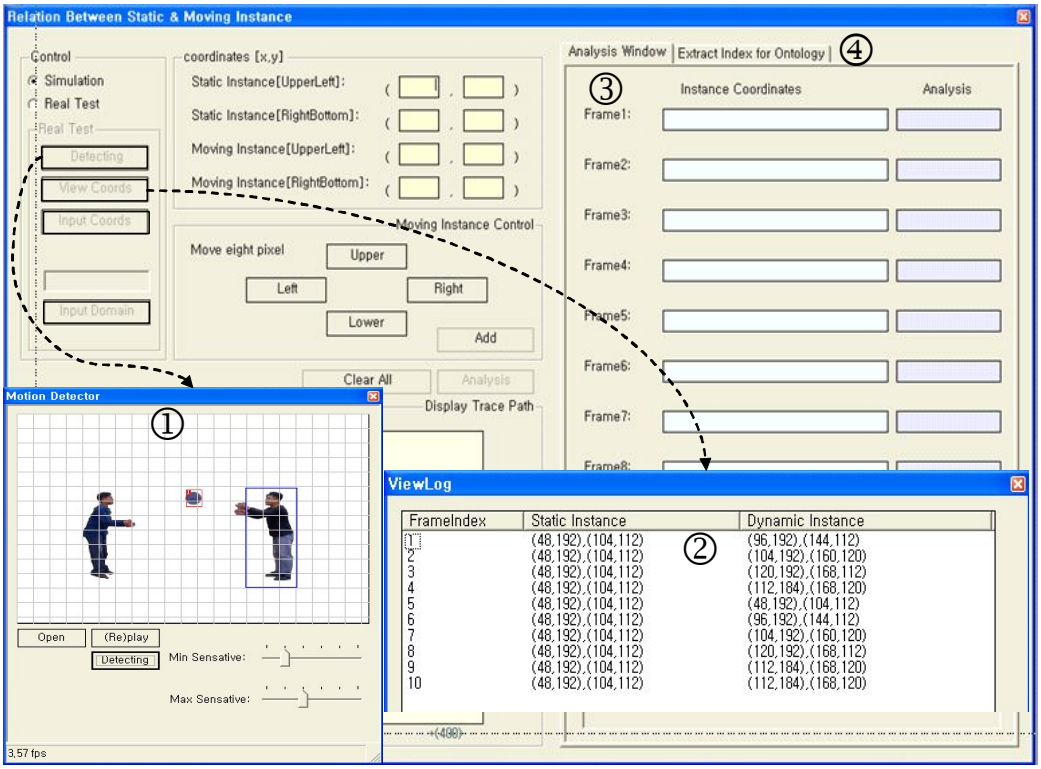
본 논문의 실험 및 성능 평가를 위한 실험 데이터는 스포츠, 생활, 자연과 관련된 비디오 영상으로 실제 영상과 직접 촬영한 영상을 사용한다. 실험에 사용된 비디오 영상은 사용자 관점에서 인지되는 meet, leave, enter, exit 상태를 포함하는 비디오 영상이다. 이벤트 별로 각각 10개의 영상을 사용하였으며 Shot당 10~15개의 움직임으로 구성된다. (그림 6.1)은 실험을 위해 사용된 인터페이스로 각 기능별로 살펴보면 다음과 같다.

①은 객체의 좌표값 추출과 로그 생성 부분으로 객체 움직임을 추적하여 프레임단위로 절대 위치값을 취득하여 로그를 생성한다.

②는 생성된 로그를 볼 수 있는 뷰로 프레임별로 정적객체와 동적객체의 위치값이 기록됨을 알 수 있다.

③의 경우 Input Coords 버튼을 통해 기록된 로그에 대한 좌표값을 이용하여 각각의 프레임에 대한 움직임 상태가 분석된다. 1차적으로 프레임에 대한 공간 관계 분석을 수행하고 최종 Analysis 버튼을 통해 연속된 프레임의 패턴을 적용하여 세부 움직임과 이벤트를 인식한다.

④의 경우 최종 분석된 이벤트에 대해 비디오 Shot단위로 인덱싱을 수행한다. 이는 의미적 검색 시 사용된다.



(그림 6.1) 실험 인터페이스

B. 제안된 모델링을 이용한 이벤트 검출

본 절에서는 제안된 움직임 모델링 기법에 대한 실험을 위해 실험 환경과 사용한 실험 데이터에 대한 내용을 기술하고 제안된 규칙을 적용하여 움직임 인식에 대한 실험과 성능 평가를 수행한다. 또한 인식된 비디오 Shot에 대한 인덱싱을 통해 온톨로지를 이용한 의미적 검색을 수행한다.

1. 비디오 프레임별 움직임 인식

본 논문에서 제안하는 객체 움직임 규칙을 적용하여 준비된 각 영상에 대해 실험한 결과 높은 정확율을 얻을 수 있었다. (표 6.1)은 각 Shot의 프레임들이 포함하고 있는 객체들의 움직임을 인식한 결과를 표현하고 있으며, 각각의 프레임에 대한 움직임은 인식되었으나 Shot3, Shot6의 경우 최종 비디오 Shot에 대한 이벤트 검출은 실패된 상태임을 나타낸다.

(표 6.1) 프레임별 움직임 인식 실험 결과

shot	shot1	shot2	shot3	shot4	shot5	shot6	shot7	shot8
frame1	come toward	come toward	overlap	come toward	overlap	touch	come toward	come toward
frame2	approach	approach	overlap	approach	overlap	separate	approach	approach
frame3	approach	go away	overlap	approach	overlap	approach	approach	approach
frame4	approach	go away	contain	approach	overlap	approach	approach	overlap
frame5	approach	approach	contain	approach	separate	approach	overlap	contain
frame6	touch	approach	overlap	approach	go away	approach	overlap	overlap
frame7	overlap	approach	contain	approach	go away	approach	overlap	separate
frame8	overlap	touch	overlap	go away	go away	go away	overlap	go away
frame9	overlap	contain	overlap	touch	go away	go away	overlap	go away
frame10	contain	contain	overlap	touch	out	out	overlap	out
Event	enter	enter	실패	실패	leave	실패	meet	enter

(표 6.1)에 대한 각 Shot에 대한 실험과정과 움직임 인식 상태를 요약해 보면 다음과 같다.

[Shot#1] shot1의 경우 사람이 자동차 문을 열고 들어가는 장면을 나타낸다.

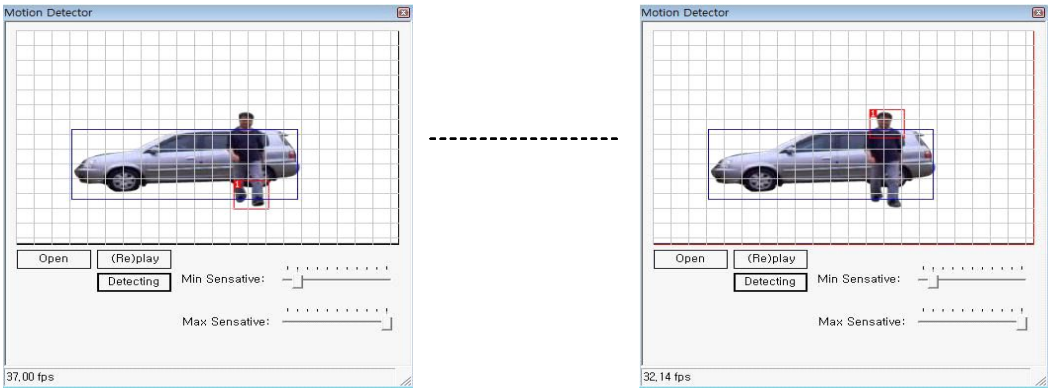
분석된 움직임 중 frame#2~frame#10의 움직임을 초기단계(IS), 중간단계(MS), 마지막 단계(FS)로 표현하면 approach-touch(overlap)-contain 관계가 성립된다. 이와 같은 연속된 움직임에 대한 최종 이벤트는 enter임을 알 수 있다. (그림 6.2)는 Shot#1에 대한 10개의 대표프레임을 나타낸다.

(그림 6.2) Enter 이벤트를 포함한 Shot내의 대표 프레임

[Shot#2] shot2의 경우 또한 사람이 자동차 문을 열고 들어가는 장면을 나타낸다. frame#3~frame#4는 출발지점으로부터 자동차 쪽으로 가다가 다시 되돌아오는 상태를 나타낸다. 분석된 움직임 중 frame5~frame10의 움직임 approach-touch-contain 관계를 통해 최종적으로 인식된 이벤트는 enter임을 알 수 있다.

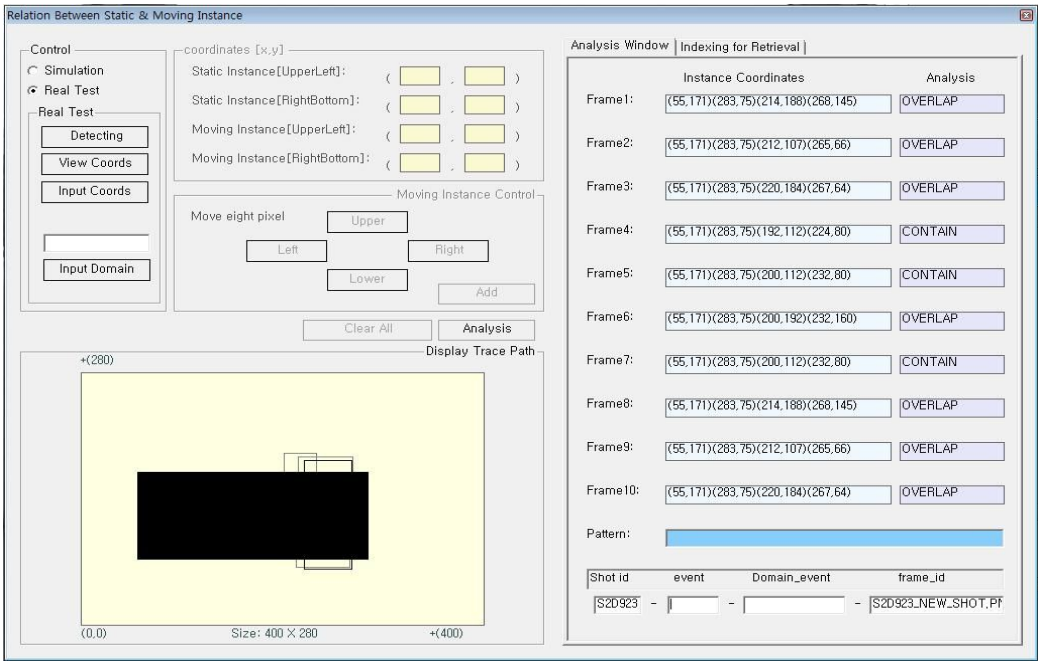
[Shot#3] shot3의 경우 사람이 자동차로부터 나와서 멀어지는 장면을 나타낸다. 프레임에 대한 실험 결과 overlap과 contain 상태만이 인식됨을 알 수 있다. 사용자관점의 움직임은 overlap-separator-go away 상태로 인지되지만 본 제안 규칙에서는 카메라의 위치를 고려하지 않았기 때문에 사용자와는 전혀 다르게 인식됨을 알 수 있다.

(그림 6.3)은 대표프레임에 대한 움직임 위치값 추출 과정을 나타내며, (그림 6.4)와 (그림 6.5)는 각각 단일 프레임상의 공간관계 분석과 연속 프레임에 대한 움직임 인식 상태를 나타낸다.

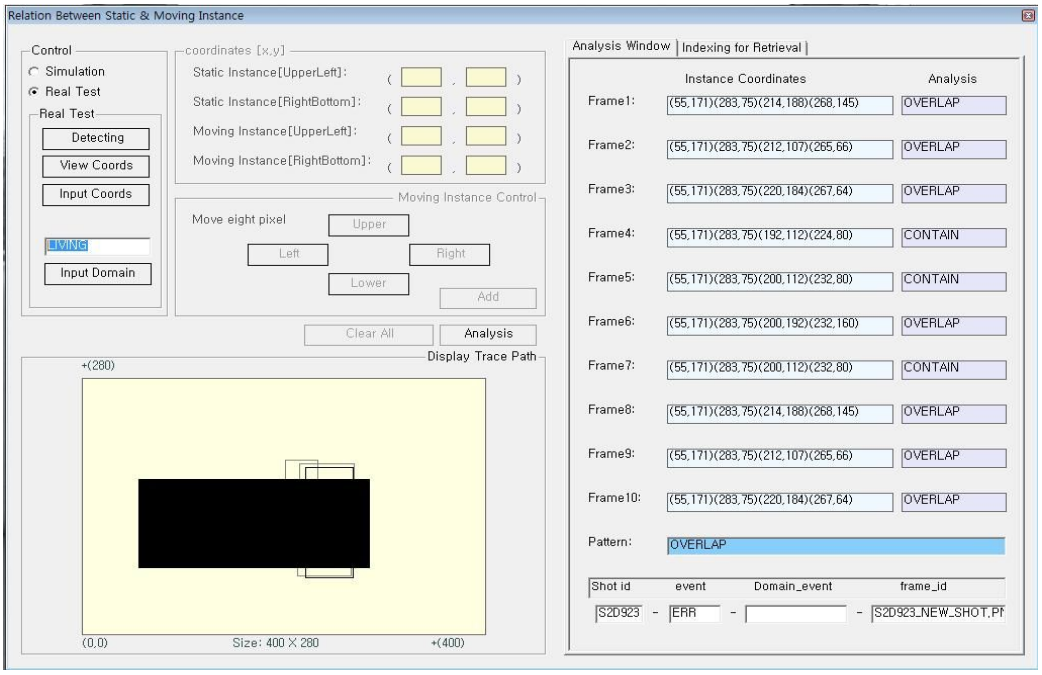


첫 프레임 **마지막 프레임**

(그림 6.3) Shot3 에 대한 객체 움직임 위치값 추출 과정



(그림 6.4) Shot3 에 대한 단일 프레임 움직임 인식



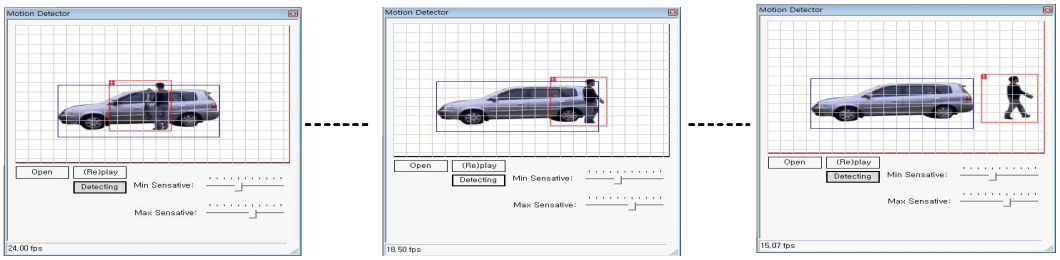
(그림 6.5) Shot3 에 대한 연속 프레임 움직임 인식

[Shot#4] shot4의 경우 사람이 자동차 쪽으로 다가가는 장면을 나타낸다.

분석된 움직임 중 frame8에서 go away 상태로 추출된 객체의 영역값이 한순간 커진 상태로 최종적으로 touch로 인식된다.

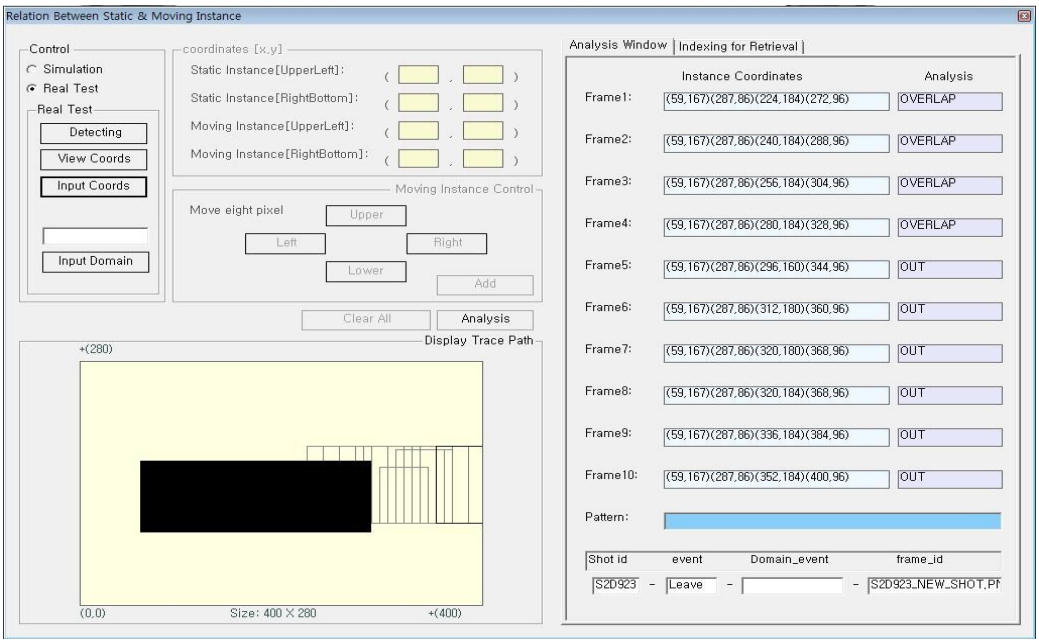
[Shot#5] shot5의 경우 사람이 자동차 문을 열고 나가는 장면을 나타낸다.

분석된 움직임에서 overlap-separate-go away 상태가 연속됨으로써 leave 이벤트로 인식됨을 알 수 있다. (그림 6.6)은 객체 움직임 위치값 추출 과정을 나타내며, (그림 6.7)과 (그림 6.8)은 각각 단일 프레임의 공간관계 분석과 연속된 프레임에 대한 움직임 인식 상태를 나타낸다.

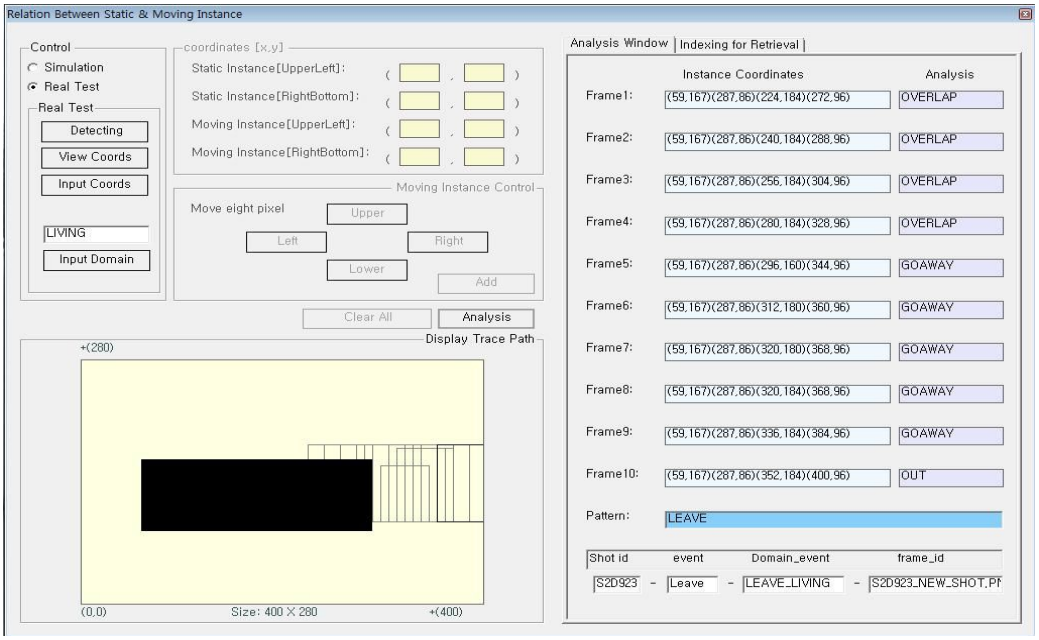


첫 프레임 중간 프레임 마지막 프레임

(그림 6.6) Shot5 에 대한 객체 움직임 위치값 추출 과정



(그림 6.7) Shot5 에 대한 단일 프레임 움직임 인식

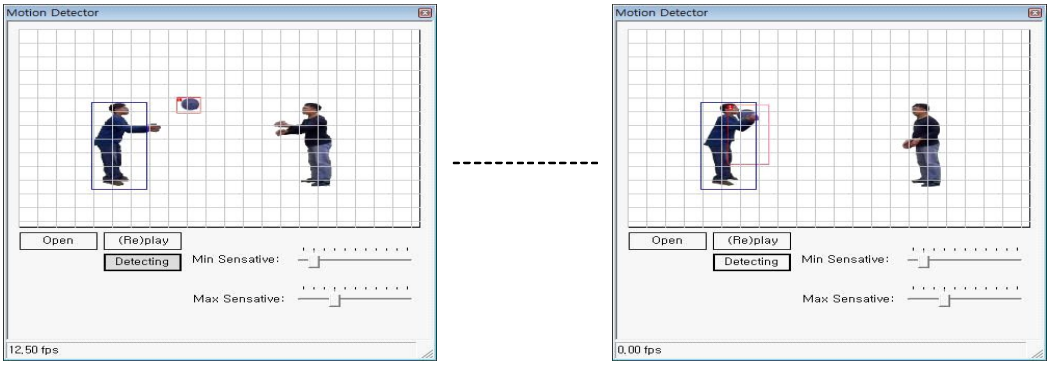


(그림 6.8) Shot5 에 대한 연속 프레임 움직임 인식

[Shot#6] shot6의 경우 자동차로부터 나와서 멀어지다 다시 다가가다가 또다시 멀어지는 장면을 나타낸다. Frame 단위의 움직임 인식은 잘 되었으나 이벤트 검출은 실패했음을 알 수 있다.

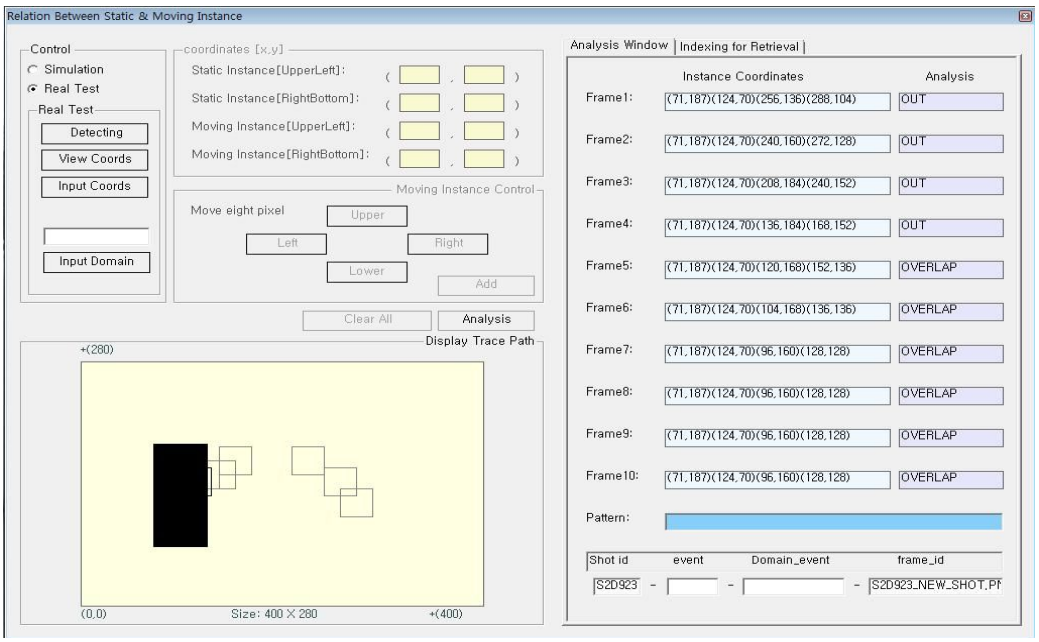
[Shot#7] shot7의 경우 두 사람이 공을 패스하고 있는 장면을 나타낸다.

분석된 움직임으로 come toward-approach-overlap 상태가 연속적으로 인식됨에 따라 최종 meet 이벤트임을 알 수 있다. (그림 6.9)는 객체 움직임 위치값 추출 과정을 나타내며, (그림 6.10)과 (그림 6.11)은 각각 단일 프레임의 공간관계 분석과 연속된 프레임에 대한 움직임 인식 상태를 나타낸다.

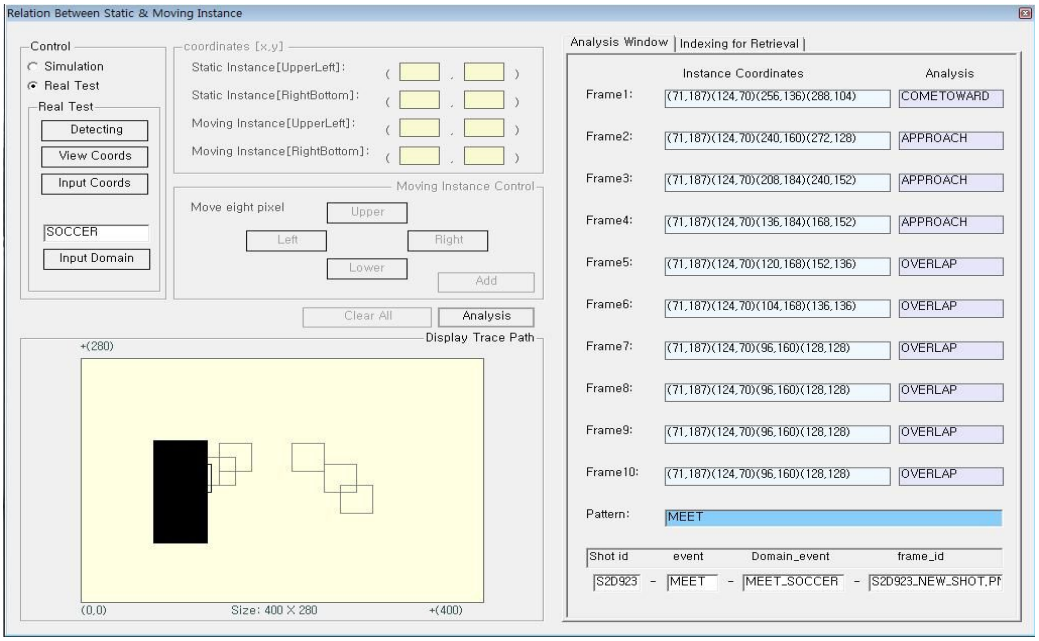


중간 프레임 **마지막 프레임**

(그림 6.9) Shot7 에 대한 객체 움직임 위치값 추출 과정



(그림 6.10) Shot7 에 대한 단일 프레임 움직임 인식



(그림 6.11) Shot7 에 대한 연속 프레임 움직임 인식

[Shot#8] shot8의 경우 사람이 자동차 문을 열고 들어갔다가 다시 나오는 장면을 나타낸다. 분석된 움직임 중 Enter와 Leave 이벤트가 동시에 들어있음에도 최종적으로 인식된 상태는 Enter 이벤트임을 알 수 있다.

(표 6.1)에서 Shot3, Shot4, Shot6에 대한 움직임 인식 상태를 통해 사람이 인지할 수 있는 움직임과 컴퓨터가 인지할 수 있는 움직임 인식의 차이를 알 수 있다.

정확율(Precision) 평가를 위해 각 Shot별로 실험에 사용된 전체 프레임수에 대한 의미 분석이 성공한 프레임수를 사용한다. (표 6.2)는 8개의 비디오 Shot에 대한 정확율을 나타내며, (그림 6.12)는 Shot에 대한 정확율을 그래프로 보이고 있다.

(표 6.2)와 (그림 6.12)는 단지 8개의 비디오 Shot에 대한 실험 결과를 보이고 있으며 평균 정확율은 이벤트 별로 10개의 비디오 Shot에 대한 실험 결과를 나타낸다.

(표 6.2) 비디오 Shot에 대한 프레임 단위의 움직임 인식과 실패

비디오	전체 프레임수	성공 프레임수	실패한 프레임수	정확율(%)	이벤트 검출
shot1	10	10	0	100	enter
shot2	10	10	0	100	enter
shot3	10	4	6	40	실패
shot4	10	10	0	100	실패
shot5	10	10	0	100	leave
shot6	10	8	2	80	실패
shot7	10	10	0	100	meet
shot8	10	10	0	100	enter
Total	80	72	8	90%	60%

(그림 6.12) 비디오 Shot의 프레임별 움직임 인식 정확률 그래프

(표 6.2)는 8개의 비디오 Shot에 대한 실험 결과로 90%의 정확율을 나타낸다. 전체 실험을 통해 산출된 평균 정확율은 약 85%를 보였다. 평균 정확율 측정을 위해 (식 6.1)을 사용한다.

$$\text{정확율} = \frac{\text{움직임인식이성공한프레임수}}{\text{실험에사용된전체프레임수}} \qquad \text{식 (6.1)}$$

실험 결과, 카메라 방향에서 직선으로 움직이는 객체에 대한 상태나 정지객체 방향으로 다가가다 사라지는 경우, 객체 위치에 대한 영역값이 갑자기 커지는 경우 등이 움직임 인식에 실패한 경우임을 알 수 있다. 다음은 실험을 통해 인식된 움직임의 연속된 패턴을 고려하여 이벤트 검출에 관한 실험을 수행한다.

2. 이벤트 검출

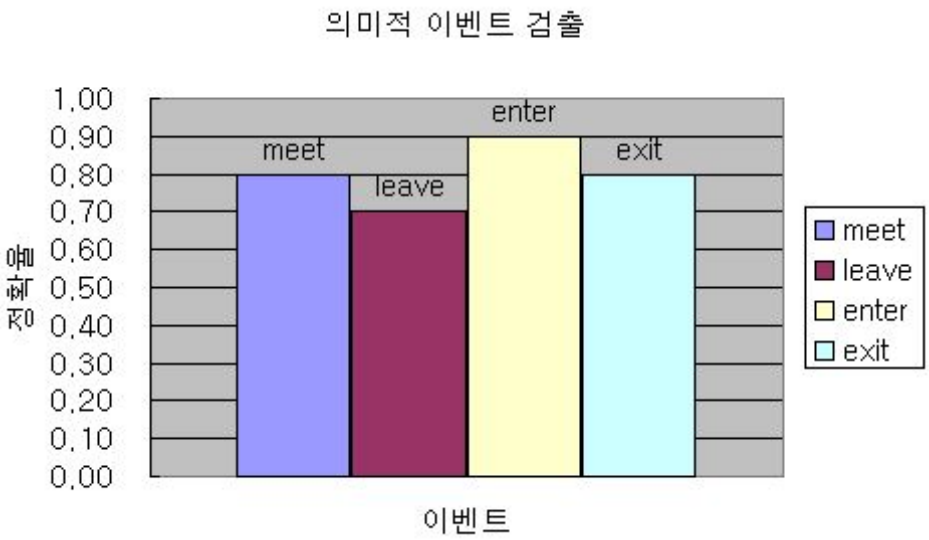
본 논문의 최종 목적은 객체의 움직임 인식을 위해 사용자의 관심대상인 비디오 Shot에 대한 이벤트를 검출하는데 있다. 실험에 사용된 Shot은 meet, leave, enter, exit 등 4개의 이벤트와 관련된 Shot이며 각각의 프레임에 대한 움직임 인식을 통해 연속된 패턴에 의해 최종 이벤트를 검출한다. 검출된 이벤트는 검증단계를 거쳐 해당되는 Shot에 대해 인덱싱을 수행하며, 이러한 인덱싱 정보는 다음 절에서 수행하는 온톨로지를 이용한 의미적 검색에 활용된다. (표 6.3)은 실험에 사용된 각 이벤트에 대한 연속된 움직임 패턴을 Initial State, Mid-State, Final State 상태에 대해 나타낸다. 각 Shot이 가진 프레임에 대한 움직임 분석 후 반복되는 움직임에 대해서는 하나의 움직임으로 본다.

(표 6.3) 이벤트에 대한 연속된 움직임 패턴

Event	Initial State	Mid-State	Final State
meet	come toward	approach	touch
leave	overlap	separate	go away
enter	approach	overlap, touch	contain
exit	contain	overlap, touch	separate

(그림 6.13)은 4개의 이벤트에 대해 정확율 그래프를 보이고 있다. 실험 결과를 통해 알 수 있듯이 이벤트에 대한 정확율은 각각의 프레임에 대한 움직임 인식에 대한 정확율 보다 낮음을 알 수 있다. 그래프에서 enter 이벤트의 경우가 가장 높은 정확율을 보인다.

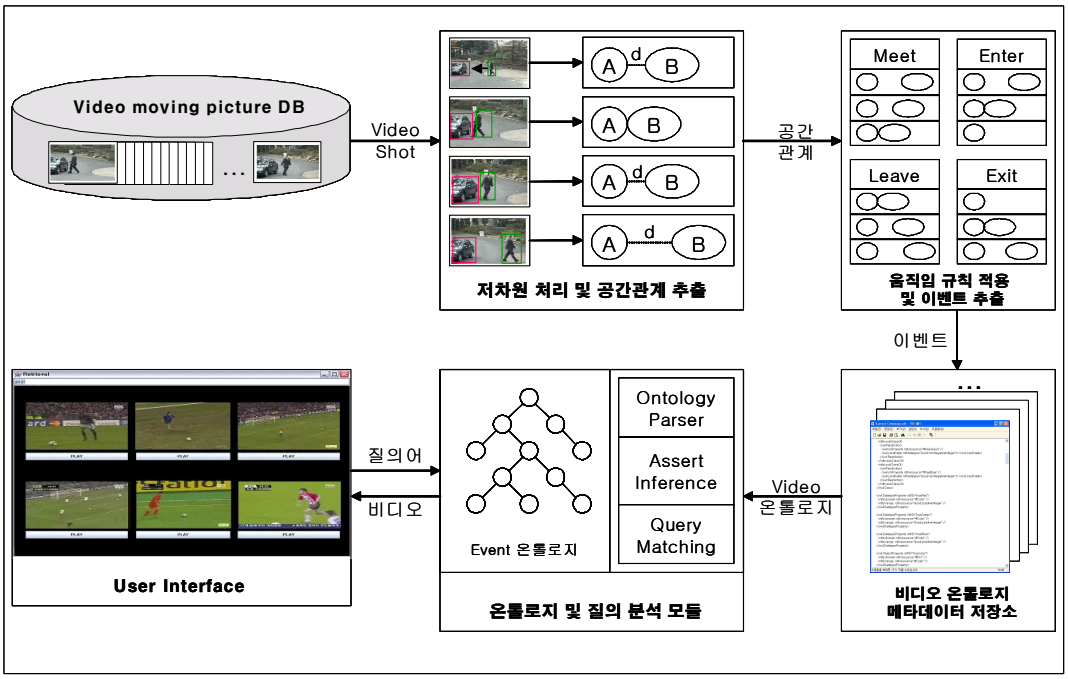
$$\text{이벤트검출정확율} = \frac{\text{이벤트검출이성공한비디오}Shot}{\text{전체비디오}Shot} \qquad \text{식 (6.2)}$$



(그림 6.13) 비디오 Shot에 대한 이벤트 검출 정확률 그래프

C. 온톨로지 적용 이벤트 검색

비디오의 저차원 처리를 통한 객체들의 공관관계 추출, 연속된 공간관계들을 이용한 이벤트 추출 규칙을 이용한 4가지 이벤트로 분류, 추출된 이벤트와 Shot의 간단한 정보를 이용한 OWL/RDF(S) 온톨로지 구축을 통한 메타데이터 작성 그리고 4가지 이벤트와 응용 영역을 이용한 이벤트 온톨로지를 이용하여 비디오의 이벤트 기반 의미적 검색을 수행하기 위한 전체 시스템 구성도는 (그림 6.14)와 같다.



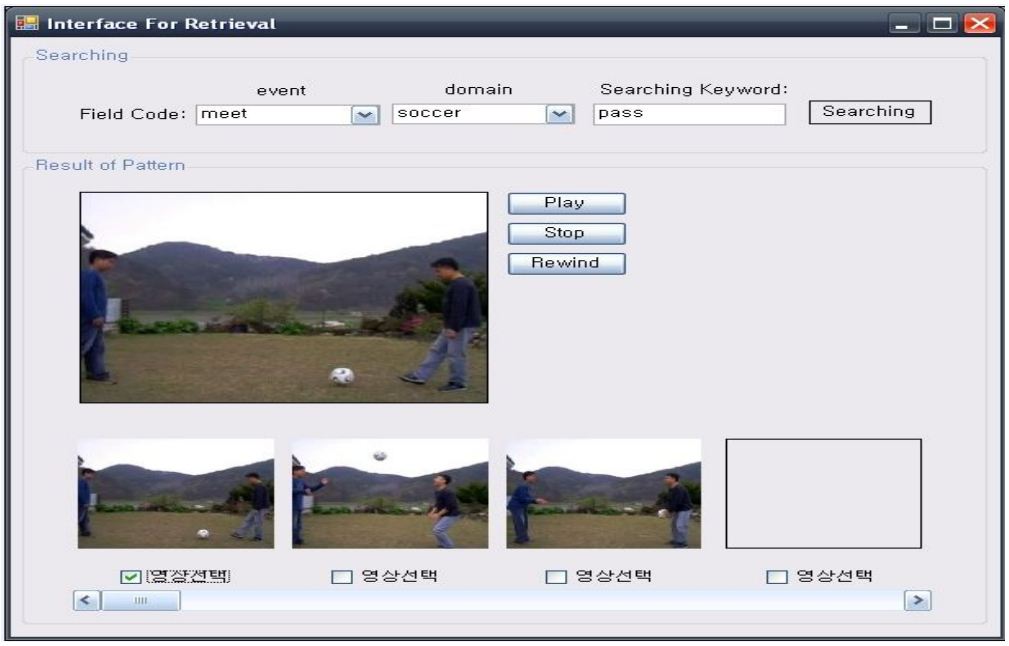
(그림 6.14) 검색 시스템 구성도

(그림 6.14)에서 전체적인 구성은 질의어를 입력하고 검색된 비디오를 보여주는 사용자 인터페이스(좌측 하단)와 이벤트의 전반적인 정보를 포함하는 이벤트 온톨로지(온톨로지 및 질의 분석모듈 내의 온톨로지), 각 Shot에 대한 정보를 온톨로지로 작성한 비디오 메타데이터(우측 하단), 실제 Shot 영상을 저장하고 있는 비디오 데이터베

이스(좌측 상단), 비디오를 입력받아 저차원 처리를 거쳐 Shot을 추출하고 객체들의 공간관계를 파악하는 모듈(중간 상단), 연속된 프레임의 공간관계를 4가지 이벤트 추출 규칙에 적용하는 이벤트 추출 모듈(우측 상단), 그리고 의미적 검색의 핵심 요소로서 온톨로지 분석과 추론 및 질의어와 매칭을 하기 위해 온톨로지 파서(ontology parser), 추론기(inference) 및 질의 처리기를 포함하는 검색 엔진 모듈(중간 하단)이 존재한다. 이러한 구조의 시스템을 통해 의미적 검색이 가능함을 보이기 위해, 앞의 과정에서 추출된 50개의 Shot을 비디오 데이터베이스에 저장하고 각 온톨로지 메타데이터를 메타데이터 저장소에 저장하였다.

본 시스템을 통해 온톨로지로 구축된 각 응용 영역의 4가지 이벤트와 각 이벤트 하부에 구축된 응용영역의 구체적 이벤트를 질의어로 검색하였다. 예를 들어, 'Kick'으로 검색을 하게 되면 'Kick'은 이벤트 'Enter'와 응용영역 'Soccer'의 'owl:intersectionOf'로 정의된 'Meet_Soccer' 개념의 인스턴스로 작성되어 있어 'Soccer' 영역에 포함된 이벤트 중에 'Enter'로 기술된 비디오 Shot들이 (그림 6.15)과 같이 검색된다. 또한 'Heading'으로 검색을 하게 되어도 이는 'Soccer' 영역과 'Meet'의 'owl:intersectionOf'로서 해당하는 비디오 Shot이 검색됨을 볼 수 있다.

또한 4가지 대표 이벤트(meet, leave, enter, exit)를 이용하여 검색을 하게 되면, 모든 응용영역에서 질의어 이벤트와 교집합으로 기술된 비디오는 모두 검색되며, 인스턴스가 중복되는 'Pass'를 질의로 검색하게 되면 'Soccer'와 'Basketball'의 응용영역에서 Shot들이 검색되는 결과를 보인다. 실제 Shot들은 4가지 이벤트중 하나를 이용하여 기술되지만, 4가지 이벤트와 각 응용영역의 이벤트들을 표현하는 이벤트 온톨로지를 통해 각 응용영역에서 실제로 사용되는 이벤트 용어(Ex. Pass, Kick, Heading, Throwin, Hunting 등)를 이용하여 의미적인 검색이 가능함을 보인다.



(그림 6.15) 온톨로지 기반의 비디오 Shot 검색 결과

D. 모델링 기법들의 비교 분석

제안하는 움직임 객체 모델링을 기존의 모델링 기법들과 비교 분석을 수행하였다. 비교분석을 위해 Zaher Aghbari[39]의 VST 모델, M.Petkovic와W.Jonker[10]의 COBRA 모델, Miyoung Cho[40] 모델링 기법과 비교한다.

(표 6.4) 기존 연구와의 비교 분석

비교대상 / 비교항목	VST 모델	COBRA 모델	Mi-young Cho	제안하는 모델링
움직임 객체 의미적 인식	표현하지 않음	저차원 인식	고차원 인식	고차원 인식
객체 관계 요소	위상관계	위상관계	방향, 속도	공간관계, 거리, 방향, 프레임의 전후 패턴 고려
의미기반 검색	제공하지 않음	내용기반 검색 지원	지원하지 않음	의미적 이벤트 검색
인덱싱	지원하지 않음	지원하지 않음	지원하지 않음	Shot 단위의 인덱싱 수행
이벤트 검출	제공하지 않음	저차원 속성 정보	제공하지 않음	이벤트에 대한 Shot 검출
검색 기법	유사도 기반	내용기반	유사도 기반	움직임 동사 온톨로지 적용 의미적 검색

(표 6.4)와 같이 제안하는 객체 움직임을 위한 모델링은 기존의 방법에 비해 객체 움직임을 효과적으로 표현할 수 있을 뿐만 아니라 움직임 객체에 대한 의미적 검색을 지원한다. 이를 위해 제안하는 모델링에서는 비디오의 객체 움직임에 대한 고차원적 의미 표현 기법이며, 객체관계 표현을 위해 공간관계, 거리, 방향, 프레임의 전후 패턴을 고려하였다. 의미기반 검색을 위해 비디오 Shot 단위의 인덱싱을 수행하여 이벤트에 대한 Shot 검출을 지원한다. 최종 검색기법으로 움직임 동사 온톨로지를 적용하여 사용자 관심의 이벤트에 대한 의미적 검색 기법을 제공한다.

이와 같이, 본 연구에서 제시하는 비디오의 저차원적 처리를 통해 고차원의 개념과 매칭함으로써 본 논문은 자동으로 비디오의 의미적 검색을 위한 기반을 마련했다는 데 큰 의미가 있다고 할 수 있다.

Ⅶ. 결론 및 향후 연구

본 논문에서는 비디오 내의 객체 움직임 인식을 위한 의미적 이벤트 검출에 대한 고차원적 접근 방법을 제안했다. 비디오의 이벤트는 비디오 검색에 있어서 사용자의 관심도가 매우 높다. 예를 들어, 축구 비디오의 경우 킥장면, 드리블 장면, 패스 장면 등은 사용자의 관심도가 높은 대표 이벤트라고 할 수 있다. 따라서, 본 논문에서는 비디오내의 이벤트 검출을 위한 고차원 모델을 제시하고 이를 저차원 특징과 매핑할 수 있는 규칙 정의를 통해 온톨로지를 이용한 의미적 검색 방법을 제안하고 성능 평가를 수행했다.

첫째, 이벤트 검출을 위한 움직임 모델링을 위해 비디오 영상에 대한 움직임 객체와 정지객체를 중심으로 인간이 인지할 수 있는 공간 관계를 connect, disconnect, overlap, contain 상태로 정의하였다. 정의된 공간관계 중 disconnect 상태에 대해서는 거리, 방향 등의 요소에 따라 가다, 오다, 나가다, 들어오다와 같은 자세한 움직임 인식이 가능하다. 따라서 이러한 움직임 표현을 위해 인간의 대화 수단인 언어 중 영어의 동사를 이용하여 이벤트 표현을 위한 개념구조를 제시했다.

둘째, 고차원 모델과 저차원 특징을 매핑할 수 있는 방안으로 객체의 위상관계를 이용한 움직임 인식 규칙을 제안했다. 제안하는 움직임 인식 규칙은 먼저 단일 프레임에서 인식할 수 있는 connect, overlap, contain에 대한 규칙을 정의하고 disconnect에 대한 자세한 움직임 인식을 위해 연속된 프레임에 대한 전·후 패턴을 고려하였다. 최종적으로 비디오 Shot에 대한 의미 분석을 위해 프레임 단위로 분석된 움직임 패턴을 통해 최종 이벤트 검출을 수행함으로써 객체 움직임 인식이 가능함을 보였다.

제안하는 규칙에 대한 움직임 인식의 정확율 평가를 위해 축구, 생활, 자연과 관련된 비디오 영상에 대해 저차원 처리를 수행하고 움직임 객체에 대한 절대 위치값을 추출하였다. meet, leave, enter, exit 이벤트를 포함하는 이벤트에 대해 각각 10개의 영상을 제시된 알고리즘에 적용하여 실험한 결과 프레임 단위의 움직임 인식에 대한 정확율은 약 85%, Shot내의 이벤트 검출에 대한 정확율은 약 80%로 분석되었다.

셋째, 이벤트에 대한 도메인 온톨로지와 비디오 Shot에 대한 온톨로지 구축을 통해 비디오 데이터의 의미적 검색 방법을 제안했다. 제안하는 온톨로지 방법은 스포츠, 자연, 생활 등에 대한 각종 이벤트에 대한 도메인 온톨로지를 구축하였으며, 의미적 검색을 위해 OWL/RDF(S)로 표현하고 메타데이터를 생성하였다. 시맨틱 웹의 표준인 OWL/RDF(S)로 표현함으로써 웹을 통한 비디오 검색 서비스나 TV 방송국, 비디오 처리 하드웨어 제공업체 등 다양한 분야에서 기술적, 사회 경제적으로 활용될 것으로 기대된다.

본 논문의 향후 연구 방향으로는 객체 움직임에 대한 세부적 요소를 파악하여 방향, 거리, 공간적, 시간적 관계를 통해 인식되지 못한 움직임을 인식할 수 있도록 본 논문에서 제안하는 규칙을 확장할 수 있는 방법을 연구할 필요가 있다. 또한 온톨로지를 이용한 의미적 검색을 위해 응용분야에 대한 이벤트를 추가 분석하여 관련 어휘를 추가하고 시맨틱 웹을 통한 비디오 검색이 가능할 수 있는 기법을 연구한다.

참 고 문 헌

[1] M. Swain and d. Ballard, "Color indexing", Int. Jour. Computer Vision, vol. 7, no. 1, pp. 11-32, 1991.

[2] J. Huang, S. R. Kumar, M. Mitra, W. J. Zhu, and R. Zabih, "Image indexing using color correlograms", CVPR, pp. 762-768, 1997.

[3] J. Huang, S. R. Kumar, and M. Mitra, "Combining supervised learning with color correlograms for content-based image retrieval", Proc. 5th ACM Multimedia Conf., pp. 325-334, 1997.

[4] G. Pass and R. Zabih, "Histogram refinement for content-based image retrieval", IEEE Workshop on Applications of Computer Vision, pp. 96-102, 1996.

[5] J.F.Allen. "Maintaining knowledge about temporal intervals", Communications of ACM, vol. 26(11), pp. 832-843, 1983.

[6] M.J.Egenhofer, K.K.Taha, "Theories and Methods of Spatio-Temporal Reasoning in Geographical Space", Theories and Methods of Spatio-Temporal Reasoning Conf., LNCS, no. 693, pp. 196-219, 1992.

[7] J. Yamato, J, Ohya, K. Ishii, "Recognizing Human Action in Time-Seuential Images using Hidden Markov Model", Proc. of IEEE Computer Vision and Pattern Recognition, pp. 379-385, 1992.

[8] S. Jeannin, "MPEG-7 visual part of eXperimentation Model 6.0", ISO/IEC JTC1/SC29/WG11 MPEG2000/N3398, Geneva, CH, June. 2000.

[9] J.R.Smith, S.F.Chang, "VisualSEEK: A Fully Automated Content-Based Image Query System", ACM Multimedia Conference, Boston, pp. 87-98, Nov. 1996.

[10] M.Swain, D.Ballard, "Color Indexing", International Journal of Computer Vision, pp. 11-32, Nov. 1991.

[11] B.Furht, S.W.Smoliar, H.Zhang, "Video and Image Processing in Multimedia Systems", Kluwer Academic Publishers, Norwell, MA, USA, pp. 1-2, 1996.

[12] E.Ardizzone, M.L.Cascia, D.Molinelli, "Motion and Color Based Video Indexing and Retrieval", ICPR, pp. 135, 1996.

[13] M.L.Cascia, E.Ardizzone, "JACOB:Just a content-based query system for video databases", ICASSP, Atlanta, GA, USA, vol. 2, pp. 1216-1219, 1996.

[14] S.F.Chang, W.Chen, H.J.Meng, H.Sundaram, D.Zhong, "VideoQ:An Automated Content Based Video Search System Using Visual Cues", ACM Multimedia Conf. Seattle, WA, Nov, 1997.

[15] S.Dagtas, W.Al-Khatib, A.Ghafoor, A.Khokhar, "Trail-based approach for video data indexing and retrieval", ICMCS'99, Florence, Italy, Jun. 1999.

[16] M.Flickner, H.Sawhney, W.Niblack, J.Ashley, Q. Huang, B.Dom, M.Gorkani, J.Hafner, D, Lee, D.Perkovic, D.Steele, P.Yanker, "Query by Image and Video Content:The QBIC System", IEEE, Sept. 1995.

[17] Z.Aghbari, K.Kaneko, A.Makinouchi, "A Motion-Location Based Indexing Method for Retrieving MPEG Videos", 9th Int'l Workshop on Database and Expert System Applications(DEXA'98), vienna, Austria, Aug. 1998.

[18] M. Andrea Rodriguez, Max J. Egenhofer, Andress D.Blaser, "Query based with Neighborhood-Based Approach", SSTD 2003, LNCS 2750, pp. 362-279, 2003.

[19] M.Petkovic, W.Jonker, "content-based retrieval of spatio-temporal video events" University of Twente, 2003.

[20] *http://www.codeproject.com/cs/media/Motion_Detection.asp*

[21] Juhyun Shin, Pankoo Kim, "Semantic Representation of Moving Object in Video Data Using Motion Ontology", Journal of korea multimedia society, vol. 10, no. 3, pp. 117-127, 2007.

[22] Gellbaum, Christianne, "A semantic network of English verbs", In WordNet: An Electronic Lexical Database, C. Fellbaum(Ed.), Cambridge, MA:MIT Press, pp. 278 - 301, 1990.

[23] Miyoung Cho, Dan Song, Pankoo Kim, "Human Action Understanding Using

Motion Verbs in WordNet", Advances in Artificial Intelligence AI, vol. 3809, pp. 1209-1212, 2005.

[24] Miyoung Cho, Dan Song, Pankoo Kim, "Interval Relations in Lexical Semantics of Verbs" Artificial Intelligence Review pp. 293-316, June. 2004.

[25] Dan Song, Miyoung Cho, Chang Choi, Juhyun, Shin, Pankoo Kim, "Knowledge Representation for Video Assisted by Domain-Specific Ontology", PKAW, pp. 144-155, 2006.

[26] Dorr, B. J. & Jones, D. "Acquisition of Semantic Lexicons:using word sense disambiguation to improve precision", In Evelyne Viegas(ed.), Breadth and Depth of Semantic Lexicons, MA:Kluwer Academic Publishers, pp. 9-98, 1999.

[27] Minhua ma & Paul mc kevitt. "Interval Relations in Lexical Semantics of Verbs", Kluwer Academic Publishers. Artificial Intelligence Review, pp. 293-316, 2004.

[28] Stephen W. Smoliar and HongJiang Zhang, "Content-Based Video Indexing and Retrieval," IEEE Multimedia, pp. 62-72, Summer. 1994.

[29] John S. Boreczky and Lawrence A. Rowe, "Comparison of Video Shot Boundary Detection Techniques," Storage and Retrieval Image and Video Database IV, Proc. of IS&T/SPIE 1996 Symp. on Elec. Imaging: Science and Technology, February, 1996.

[30] Andrea Cavallaro , Olivier Steiger , Touradj Ebrahimi, "Multimedia content description interface-part 8: extraction and use of Mpeg-7 descriptors", ISO/IEC TR 15938-8, 2002.

[31] Jian Feng, Kwok-Tung Lo and Hassan Mehrpour, "Scene Change Detection Algorithm for MPEG Video Sequence," IEEE, pp. 821-824, 1996.

[32] R. V. Baru, K. R. Ramakrishnan, and S.H. Srinivasan, "Video object segmentation: A compressed domain approach", IEEE Trans, Circuits System Video Technology, vol. 14, pp. 462-474, April 2004.

[33] Hariharakrishnan, K. Schonfeld, D. Raffy, P. Yassa, F. "Video tracking using block matching", Image Processing, 2003.

[34] Thomas Meier and King N. Ngan, "Video Segmentation for Content-Based

Coding", IEEE Trans. Circuits System Video Technology, vol. 9, no. 8, Dec. 2004.

[35] Dan Song, Hai Tao Liu, Miyoung Cho, Hanil Kim, Pankoo Kim, "Domain Knowledge Ontology Building for Semantic Video Event Description", international conference on image and video retrieval, pp. 267-275. Singapore 2005.

[36] Kompatsiaris, V. Mezaris, and M. G. Strintzis, "Multimedia content indexing and retrieval using an object ontology", Multimedia Content and Semantic Web-Methods, Standards and Tools Editor G.Stamou, Wiley, New York, NY, 2004.

[37] Christian Halaschek-Wiener, Nikolaos Simou and VassilisTzouvaras, "Image A nnotation on the Semantic Web," W3C Candidate Recommendation, W3C Working Draft 22 March, 2006.

[38] Matthew Horridge Holger Knublauch and Alan Rector, "A Practical Guide To Building OWL Ontologies Using The Protege-OWL Plugin and CO-ODE Tools Edition 1.0," W3C Candidate Recommendation, August, 2004.

[39] Zaher Aghbari, Kunihiko Kaneko and AkifumiMakinouchi, "VST-Model: A Uniform Topological Modeling of the Visual-Spatio-Temporal Video Features", ICMCS, Vol. 2, pp. 163-168, 1999.

[40] Miyoung Cho, Dan Song, Chang Choi, Pankoo Kim, "Comparison Between Motion Verbs Using Similarity Measure for the Semantic Representation of Moving Object", CIVR, LNCS 4071, pp. 281-290, 2006.

[41] Schreiber, G., Dubbeldam, B., Wielemaker, J., and Wielinga, B, "Ontology-Based Photo Annotation", IEEE Intelligent Systems, vol. 16, pp. 66-74, 2001.

[42] B. L. Yeo and B. Liu, "'Rapid scene change detection on compressed video", IEEE Trans. Circuits Syst. Video Technol. vol. 5, pp. 533-544, 1995.

[43] M. G. Strintzis,S. Bloehdorn,S. Handschuh,S. Staab,N. Simou,V. Tzouvaras,K. Petridis, I. Kompatsiaris and Y. Avrithis, "Knowledge representation for semantic multimedia content analysis and reasoning," in European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology,

November. 2004.

[44] Dan Song, Mi Young Cho, Pan Koo Kim, "A Novel Video Data Model for Moving Object Description Based on Spatio-temporal Relations", ISPACS 2004.

[45] A.G. Hauptmann, “Towards a large scale concept ontology for broadcast video", In Proc. of the 3rd int conf on Image and Video Retrieval(CIVR), 2004.

[46] T.Adanck, N.O’Connor, and N.Murphy, "Region-based Segmentation of Image Using Syntactic Visual Features", In Pro. Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS, Montreux, Swizerland, April. 2005.

[47] Thomas Drakengren and Peter Jonsson, "Maximal tractable subclasses of Allen’s interval algebra: Preliminary report", In AAAI-96, pp. 389-394. 1996.

[48] Kautz, H. A. and Ladkin, P, "Integrating metric and qualitative temporal reasoning", In Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91), pp. 241-246, Anaheim, California, USA, July. 1991.

[49] Hollink, L., Schreiber, G., Wielemaker J., and Wielinga B., "Semantic Annotation of Image Collections", In Proceedings of Knowledge Capture-Knowledge Markup and Semantic Annotation Workshop 2003.

[50] S.F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, "A fully automated content-based video search engine supporting spatiotemporalqueries", IEEE Trans. Circuits Syst. Video Technol., vol. 8, pp. 602-615, Sept. 1998.

[51] M. R. Naphade, I. V. Kozintsev, and T. S. Huang, "A factor graph framework for semantic video indexing," IEEE Trans. Circuits Syst. Video Technol., vol. 12, pp. 4052, Jan. 2002.

[52] E. Izquierdo, J. R. Casas, R. Leonardi, P. Migliorati, N. O’Connor, I. Kompatsiaris, and M. G. Strintzis, "Advanced content-based semantic scene analysis and information retrieval: The schema project," in Proc. Workshop on Image Analysis for Multimedia Interactive Services,London, U.K., pp. 519-528, Apr. 2003.

[53] Shu-Ching Chen, Mei-Ling Shyu, Na Zhao, "An Enhanced Query Model for Soccer Video Retrieval Using Temporal Relationships", ICDE 2005.

[54] Y.F. Day, A. Khokhar, and A. Ghafoor, “A framework for semantic modeling of video data for content-based indexing and retrieval“ ACM-Springer Multimedia Systems Journal, vol. 7(5), pp. 409--423, 1999.

[55] Miyoung Cho, Pankoo Kim, "Semantic Image Analysis Based on the Representation of the Spatial Relations Between Objects in Images", ICIAR, LNCS 3211, pp. 471-478, 2004.

[56] ISO/IEC Committee Draft 15938-5, "Information Technology Multimedia Content Description Interface: Multimedia Description Schemes", ISO/IEC/JTC1/SC29/WG11/N3966, March. 2001.

[57] Gennari, S. and Poeppel, D., "Events versus states: empirical correlates of lexical classes", In Proceedings of CogSci, pp. 351-356, USA, August. 2002.

[58] A.G. Hauptmann, "Towards a large scale concept ontology for broadcast video", In Proc. of the 3rd int conf on Image and Video Retrieval(CIVR), 2004.