

# 비디오 감시 시스템으로부터 객체 동선과 관계 패턴의 점진적 추상화에 의한 실시간 지식의 추출 및 복원 방법론

김세종, 김태호, 이문근  
{kimsj,thkim,moonkun}@chonbuk.ac.kr

A Methodology for Extraction and Retrieval of Real-time Knowledge from Video Surveillance Systems by Incremental Abstraction of Trajectory and Relation Patterns

Sejong Kim Taeho Kim Moonkun Lee  
Chonbuk National University

## 요 약

멀티미디어의 비중이 커짐에 따라 컴퓨터 과학 각 분야에서 독자적인 기술들을 이용하여 실제 응용 및 시스템을 구축하고 있다. 하지만 멀티미디어 동영상 내에서 객체의 행위 단독적인 움직임을 수치로만 표현하여 자료를 처리함에 따라 의미를 해석하는 것이 부자연스럽고 정확한 숫자에 부합하는 행동의 검출이 어렵다. 본 논문에서는 멀티미디어 동영상의 기본적인 행위를 추출하고 이를 추상화, 정형화하여 보다 상위단계로 접근을 유도하여 멀티미디어 데이터에 대한 접근을 용이하게 하기위한 방법에 대하여 논의하였다.

## 1. Introduction

멀티미디어 데이터의 비중이 커짐에 따라 이를 해석하고, 가공함에 대한 중요성은 이전 연구에서 끊임없이 재거되어왔다. 이러한 활발한 움직임들은 컴퓨터 과학 전반에 걸쳐 각 분야에서 독자적인 시각으로 이루어지고 있으며 각각의 분야에 기술들을 활용하여 실제 응용 및 시스템을 구축하고 있다. 실제로 데이터베이스에서는 멀티미디어 데이터의 자료 구축과 검색에서[Chang(장재우)03] 인공지능에서는 로봇의 행동 패턴 학습 및 인지 등에서 논의가 이루어지고 있으며[Thrun02], 또한, 컴퓨터 비전 분야에서는 움직임을 추출하고 이를 패턴, 수치화하는 것이 연구되어지고 있다[Culter99]. 하지만, 이러한 각 분야의 노력에도 불구하고 아직까지는 만족할만한 결과를 얻고 있지 못하고 있다. 이러한 문제점의 주된 요인으로는 객체의 행위 단독적인 움직임만을 가지고 데이터화 하였으며, 움직임을 수치로만 표현하여 자료를 처리한다는 것이다. 이는 의미를 해석하는 것도 용이하지 못할뿐더러 정확한 숫자에 부합하는 행동이 검출이 어렵다는 단점이 있다.

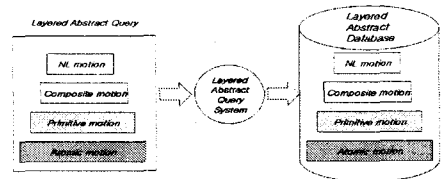


Figure 1. Layered Abstract DB and Query System

이에 따라 본 논문에서는 움직임의 기본적인 행위를 추출하고 이를 추상화, 정형화함으로써 보다 상위단계로 접근을 유도하며 이러한 행위에 의미를 부여함으로써 자연어적 처리를 가능케 하는 장점을 가지고 있다.

- Atomic Motion: Move와 Stop으로 객체의 가장 기본적인 움직임을 표현한다.
- Primitive Motion: 벡터로 표현되는 객체의 기본적인 움직임을 Primitive Motion이라 정의하며 네 가지로 표현될 수 있다: Straight, Back, Curve, Stop
- Composite Motion: Primitive Motion과 Composite Motion의 조합을 통해 객체의 움직임을 의미화하여 표현한다.
- Relation: 둘 이상의 객체들 사이의 관계를 정의하며 다음의 네 가지로 표현한다: Hold, Connect, Away, Close

- Natural Language Motion: 객체들의 움직임을 자연어로 재영시하는 것으로 Composite Motion의 결과를 자연어로 표현한다.

본 연구는 관계 카메라의 관측지점은 CCTV와 같이 항상 고정된 지정만을 바라보고 있고 3차원 정보의 영상을 2차원 정보로 변환함에 있어서 배경에 대한 정보는 대부분 제거되고 감지되는 객체들의 관계만 추출된다는 가정 아래에서 진행되었다.

## 2. Related Works

동영상 데이터 정보 추출에 대한 접근 방법은 크게 두 가지로 볼 수 있는데,

첫째, 움직임 전체를 수치화 하여 이를 기존 동영상에서 비교하여 판별하는 접근법이다. 이러한 방법의 문제점으로는 기존 데이터베이스에 저장된 동영상 정보를 새로운 질의와 비교하는 것인데, 이때 매년 새로운 움직임과 비교 판단해야한다는 점과, 기존 동영상에 내제된 세부적인 정보를 의미화 한다는 것이 불가능하다는 것이다. 더욱이 기존 질의 시스템에 대한 전반적인 개선을 요한다는 점을 들 수 있다[Chang(장재우)][Cutler99].

둘째, 움직임의 반복적인 행위를 수치로 패턴화하여 이에 의미를 부여하는 접근 방법이다. 기존 움직임 전체를 수치화 하는 것보다는 내제된 행위를 추출할 수 있고 동일한 패턴에 대해서는 어느 정도 추상화 한다는 점은 기존 접근법에 대한 장점으로 볼 수 있으나, 객체의 움직임에 대한 패턴이 일정하지 않다는 점이다. 또 여전히 이전과 마찬가지로 이를 수치화한다는 문제는 해결되지 않았다[Thrun02].

## 3. Recognition of Abstract Patterns of Trajectory

본 논문은 다음의 두 가정을 따른다: 1) 카메라의 관측지점은 CCTV와 같이 항상 고정된 배경만을 비추고 2) 3차원 영상의 정보를 2차원 정보로 전환함에 있어서 배경에 대한 정보는 대부분 제거되고 감지되는 객체들의 관계와 움직임만 추출된다는 가정 하에 진행되었다.

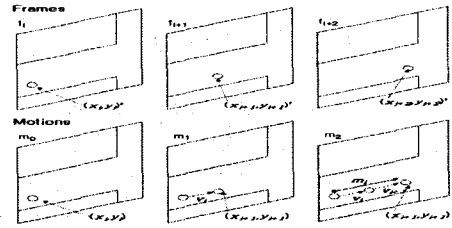


Figure 2. Motions between frames.

### 3.1 Primitive Motion

동영상 정보를 추출함에 있어서 가장 기본이 되는 데이터는 각각의 프레임이라고 할 수 있다. 하나하나의 프레임은 객체의 위치 정보를 2차원 데이터로 가지고 있으며 (Figure 2-\$m\_0\$) 두 개의 프레임을 이용하면 이전의 위치와 현재의 위치를 비교하여 객체의 이동 여부를 판단하고 사이의 거리를 알 수 있다(Figure 2-\$m\_1\$). 여기서 더 나아가 세 개의 다른 프레임을 비교하게 되면 각 모션사이의 각각 거리를 비교하여 객체의 기본적인 모션을 추출할 수 있는데(Figure 2-\$m\_2\$) 이를 추상화하면 크게 Straight, Back, Curve, Stop으로 나타낼 수 있고 본 연구에서는 이를 Primitive Motion이라고 정의하였다.

#### Definition 3.1 (Preliminary)

- Frame: 관측지점으로부터 저장된 이미지 한 장.  $f$ 로 표기한다.
- Object: 프레임 안에서 움직임이 가능한 객체.  $o$ 로 표기한다.
- Frame space: 프레임에 표현되는 3차원 공간.  $S$ 로 표기한다.
- Frame time: 프레임 사이의 시간.  $t$ 로 표기한다.
- Environment space: 3차원으로부터 전환된 2차원 공간.  $E$ 로 표기한다.
- Motion: 객체의 자리가  $E$ 의 다른 좌표로 이동하는 것.  $m$ 으로 표기한다.
- No motion: motion이 없는 객체.  $m_0$  혹은  $stop$ 으로 표기한다. □

#### Definition 3.2 (Atomic motion)

프레임  $i$ 에서 객체  $x$ 의 atomic motion (AM)은 다음과 같이 정의한다:

$$\bullet v_i^{(r,d)} = f_i(o_x) \Delta f_{i+1}(o_x), \quad (1)$$

where  $f_i(o_x)$ ,  $d$  and  $r$  imply the 2D location

of  $x$  on  $E$ ,  $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ ,  
 $r = \sin^{-1}\left(\frac{(x_2 - x_1)}{d}\right)$  for  $\vec{v} = ((x_1, y_1), (x_2, y_2))$ ,  
 respectively. □

$\Delta$ 는 두 연속적인 프레임에서  $x$ 의 벡터를 계산하는 difference operation이다. atomic motion의 예제는 <Figure 2>의  $m_1$ 으로 표현할 수 있다.

**Definition 3.3 (Primitive motions)**

두 연속적인 atomic motion  $v_i$  and  $v_{i+1}$ 사이의 객체  $x$ 의 Primitive motion (PM)은 다음과 같이 정의한다:

- $m_{(i,i+1)}^{(r',d')} = v_i^{(r,d)} \sqcap v_{i+1}^{(r',d')}$ , (2)

where  $r'_i$ , and  $d'_i$  represent updated radian and relocated 2D distance, respectively.

□

$\sqcap$ 는 <Figure 3>에서 보는 것과 같이 두 연속적인 atomic motion을 더해 primitive motion을 얻기 위한 concatenation operation이다. <Figure 2>의  $m_2$ 가 Primitive motion의 예를 보이고 있다.

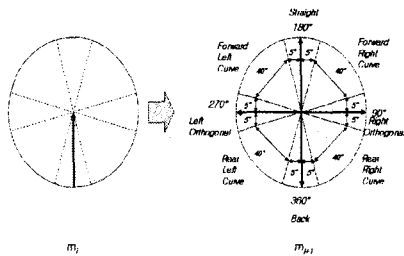


Figure 3. Pictorial definition of primitive motions.

**Definition 3.4 (Types of primitive motions)**

<Figure 3>의  $m_1$ 과  $m_2$ 의 Primitive motion의 종류는 다음과 같이 정의한다.

- $\tau(m) = S|O|C_F|C_R|B$  (3)

where

- Straight(S): 만약  $m_1$ 으로부터  $m_2$ 까지의 움직임이  $180 \pm 5$ 도 이내라면 이 객체의 움직임을 *straight*라 하고 S로 표기한다.
- Reverse(R):  $m_1$ 에서  $m_2$ 로의 움직임  $0 \pm 5$  혹은  $360 \pm 5$ 도 이내라면  $m_1, m_2$ 의 움직임을 *reverse*라

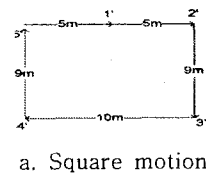
고 부르며 R로 표기한다.

- Orthogonal(O):  $m_1$ 으로부터  $m_2$ 의 움직임이  $90 \pm 5$  혹은  $270 \pm 5$ 도 이내라면  $m_1$ 과  $m_2$ 의 움직임을 *orthogonal*이라 부르고 O로 표기한다. 이는 *left*와 *right orthogonal*로 확장된다.
- Curved(C):  $m_1$ 로부터  $m_2$ 의 움직임이 이전의 각도와 다르다면  $m_1$ 와  $m_2$ 의 움직임을 *curved*라고 부르며 C로 표기한다. 객체가 꺾이는 방향과 각도의 정도에 따라 *curved*는 *forward*와 *backward, left*와 *right*로 분류된다.

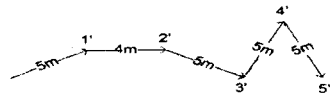
예를 들어 <Figure 4-a>는 다음과 같이 연속적으로 움직이는 PM의 예제를 보이고 있다: 9미터 직진 후 세 번 오른쪽 직각회전을 반복적으로 한다. 이 행위와 사각형과 같은 모양을 가진 움직임을 볼 수 있으며 다음과 같이 표기 한다:

$S_{(0,2)}^{9,0} \cdot O_{(1,3)}^{9,1} \cdot O_{(2,4)}^{10,1} \cdot O_{(3,5)}^{9,1}$ . <Figure 4 -b> 역시 다음과 같은 PM의 움직임을 보이고 있다: 오른쪽으로 CF를 두 번 행하고 O를 왼쪽, 오른쪽 순으로 움직인다. 이 행위는 특정의 의미가 없는 자유로운 움직임을 보이고 있으며 다음과 같이 표기 한다:

$FC_{(0,2)}^{9,1} \cdot FC_{(1,3)}^{5,1} \cdot RC_{(2,4)}^{5,-1} \cdot RC_{(3,5)}^{5,1}$ .



a. Square motion



b. Free curve motion

Figure 4. Examples of primitive motions.

**3.2 Composite Motion**

Primitive Motion이 두 벡터간의 움직임만으로 기본적인 움직임을 정의한 것이라면 Composite Motion은 Primitive Motion과 Composite Motion의 조합으로 더욱 많은 움직임을 추상화한 것이다.

**Definition 3.5 (Composite motions)**

두 연속적인 atomic motion  $v_i$ 와  $v_{i+1}$  사이의 객체  $x$ 의 composite motion (CM)은 다음과 같이 정의한다.

$$\bullet M_{(i,i+1)}^{(r_i,d_i)} = m_{(i,i+1)}^{(r_i,d_i)} \quad (4)$$

모든 PM은 CM이다.

$$\bullet M_{(i,i+2)}^{(r_i,d_i)} = m_{(i,i+1)}^{(r_i,d_i)} \cap m_{(i+1,i+2)}^{(r_{i+1},d_{i+1})} \quad (5)$$

두 연속적인 PM의 조합은 CM이다.

$$\bullet M_{(i,j)}^{(r_i,d_i)} = M_{(i,j-1)}^{(r_i,d_i)} \cap m_{(j,j+2)}^{(r_j,d_j)} \quad (6)$$

CM과 그 뒤의 연속적인 PM의 조합은 CM이다.

$$\bullet M_{(i,k)}^{(r_i,d_i)} = M_{(i,j-1)}^{(r_i,d_i)} \cap M_{(j,k)}^{(r_j,d_j)} \quad (7)$$

두 연속적인 CM의 조합은 CM이다.  $\square$

$\cap$ 은 Definition 3.7과 <Table 1>에서 정의된 composition operation이다.

**Definition 3.5 (Types of composite motions)**

두 모션 사이의 Composite Motion의 종류에는 다음과 같은 것들이 있다. (Figure 3):

$$\bullet \tau(M) = L|A|C_2|C_C|nP|C_L \quad (8)$$

where

- Line(L):  $m_1$ 과  $m_2$  straight라면 이 둘을 조합한 것은 line라고 부르며 L로 표기한다.
- Alternate(A):  $m_1$ 과  $m_2$ 가 반복적으로 reverse를 행하고 있다면 이 둘의 조합은 alternate를 행하고 있다고 하고 A로 표기한다.
- Z-Curve( $C_2$ ):  $m_1$ 과  $m_2$ 가 반복적으로 backward curves (혹은 orthogonals)를 반대 방향으로 번갈아 움직인다면 이들의 조합은 Z-curve라고 불리고 ZC로 표기한다.
- C-Curve( $C_C$ ):  $m_1$ 과  $m_2$  모두 forward 혹은 backward curves를 동일한 방향으로 행한다면 이들의 조합은 C-curve로 불리고 CC로 표기한다.
- S-Curve( $C_S$ ):  $m_1$ 과  $m_2$ 가 반복적으로 forward curves (혹은 orthogonals)를 반대 방향으로 번갈아 행한다면, 이들의 조합은 S-curve라고 불리며 SC로 표기한다.
- n-Polygon(nP):  $m_1$ 과  $m_2$ 가 반복적으로 backward curves (혹은 orthogonals)를 동일한 방향으로 행할 때 이들 조합이 완료가 되지 않았더라도 3P로 표기한다. 만약 동일한 조합이 계속 일어난다면 숫자를 증가시킨다.
- Circle( $C_L$ ): nP when  $n \geq 6$ .  $\square$

**Definition 3.6 (Composition rules)**

Composition rule is defined in Table 1.

Pre \ Next	$S_{(m-1, n)^j, 0}$	$O_{(m-1, n)^j, 1}$	$O_{(m-1, n)^j, -1}$
$S_{(l, m)^i, 0}$	$S_{(l, n)^{i+j}, 0}$	$O_{(l, n)^{i+j}, 1}$	$O_{(l, n)^{i+j}, -1}$
$O_{(l, m)^i, 1}$		$O_{(l, n)^{i+j}, 1}$	$Z_{(l, n)^{i+j}, -1}$
$O_{(l, m)^i, -1}$		$Z_{(l, n)^{i+j}, 1}$	$O_{(l, n)^{i+j}, -1}$
$FC_{(l, m)^i, 1}$			$Z_{(l, n)^{i+j}, -1}$
$FC_{(l, m)^i, -1}$		$Z_{(l, n)^{i+j}, 1}$	
$RC_{(l, m)^i, 1}$		$T_{(l, n)^{i+j}, 1}$	$Z_{(l, n)^{i+j}, -1}$
$RC_{(l, m)^i, -1}$		$Z_{(l, n)^{i+j}, 1}$	$T_{(l, n)^{i+j}, -1}$
$B_{(l, m)^i, 0}$			

Pre \ Next	$FC_{(m-1, n)^j, 1}$	$FC_{(m-1, n)^j, -1}$
$S_{(l, m)^i, 0}$	$FC_{(l, n)^{i+j}, 1}$	$FC_{(l, n)^{i+j}, -1}$
$O_{(l, m)^i, 1}$		$Z_{(l, n)^{i+j}, -1}$
$O_{(l, m)^i, -1}$	$Z_{(l, n)^{i+j}, 1}$	
$FC_{(l, m)^i, 1}$	$FC_{(l, n)^{i+j}, 1}$	$Z_{(l, n)^{i+j}, -1}$
$FC_{(l, m)^i, -1}$	$Z_{(l, n)^{i+j}, 1}$	$FC_{(l, n)^{i+j}, -1}$
$RC_{(l, m)^i, 1}$	$T_{(l, n)^{i+j}, 1}$	
$RC_{(l, m)^i, -1}$		$T_{(l, n)^{i+j}, -1}$
$B_{(l, m)^i, 0}$		

Pre \ Next	$RC_{(m-1, n)^j, 1}$	$RC_{(m-1, n)^j, -1}$	$B_{(m-1, n)^j, 0}$
$S_{(l, m)^i, 0}$	$RC_{(l, n)^{i+j}, 1}$	$RC_{(l, n)^{i+j}, -1}$	$B_{d1+d2}$
$O_{(l, m)^i, 1}$	$T_{(l, n)^{i+j}, 1}$	$Z_{(l, n)^{i+j}, -1}$	
$O_{(l, m)^i, -1}$	$Z_{(l, n)^{i+j}, 1}$	$T_{(l, n)^{i+j}, -1}$	
$FC_{(l, m)^i, 1}$	$T_{(l, n)^{i+j}, 1}$		
$FC_{(l, m)^i, -1}$		$T_{(l, n)^{i+j}, -1}$	
$RC_{(l, m)^i, 1}$	$T_{(l, n)^{i+j}, 1}$	$Z_{(l, n)^{i+j}, -1}$	
$RC_{(l, m)^i, -1}$	$Z_{(l, n)^{i+j}, 1}$	$T_{(l, n)^{i+j}, -1}$	
$B_{(l, m)^i, 0}$			

**4 Recognition of Relation Patterns**

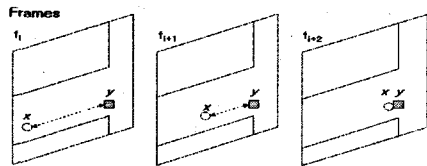


Figure 6 An example of relation between two objects and its transition in frame time.

**4.1 Relation**

프레임이 변함에 따라 그 안의 객체들의 정보 또한 바뀌는데 그 중 객체들 간의 관계에 가장 큰 영향을 미치는 요소가 바로 거리이다. 우리는 이 거리 정보에 중심을 두어 No Relation, Hold, Connect, Away, Close라는 5가지 관계를 정의하였다.

**Definition 4.1: (No Relation)**

화면상에 객체 하나만이 존재할 때를 나타낸다. No Relation이라 부르고 NR로 표기한다.

**Definition 4.2 (Hold)**

움직이거나 움직이지 않는 두 객체 사이의 거리변화가 없을 때의 관계를 표현한다. Hold로 표기한다.

**Definition 4.3 (Connect)**

두 객체가 맞닿아 있는 상황을 나타낸다. Connect로 부르고 Con으로 표기한다.

**Definition 4.4 (Away)**

두 객체가 서로 멀어지는 상황을 나타낸다. Away로 표기한다.

**Definition 4.5 (Close)**

두 객체가 서로 가까워지는 상황을 나타낸다. Close로 표기한다.

**4 Representation of Trajectory and Relation with Natural Language**

앞에서 정의된 객체들의 움직임과 관계를 자연어로 재영시하는 것으로 Composite Motion의 결과로 추출된 움직임 7가지와 각 객체들 사이의 관계 5가지를 실제 세계에서 사람들이 자주 사용하는 언어로 바꾸어 표현한다.

**4.1 Trajectory**

	표현
Stop	정지하다, 멈추다, 중단하다, 그만하다, 세우다
Straight	일직선, 직선, 전진, 앞으로 이동
Back	후진, 뒤로 이동, 돌아가다
Orthogonal	직각이동
Forward Curve	완만한 곡선으로 이동, 원형의 움직임, 전방 회전
Rear Curve	뒤로 꺾다, 후방 회전
ZigZag	지그재그 움직임, 좌우로 왔다갔다 움직인다, 방향을 계속 바꾸다
Triangle	삼각형 움직임

**4.2 Relation**

Relation의 경우 Motion에 보다 애매성이 존재하는데 그것은 실제계의 모든 관계를 언어로 표현하기 힘들기 때문이다. 다음은 앞에서 정의한 관계들을 자연어로 해석한 표현이다.

	표현
No Relation	혼자, 홀로, 주위에 아무도 없이
Hold	변화가 없다, 차이를 유지하다
Connect	붙어있다, 접촉하다, 맞닿아있다, 만지다, 잡다
Away	멀어지다, 떨어지다
Close	가까워지다, 다가오다

이렇게 정의된 표현들을 기반으로 위에서 정의한 Composite Motion과 Relation을 변환하게 되는데, 그 과정에서 객체의 움직임에 순서를 주고, 그 하나하나를 언어로 표현한다.

**4.3 Query**

데이터베이스에 저장된 정보들을 검색하기 위해서는 질의어가 반드시 요구된다. 하지만 지금까지의 멀티미디어 검색에 있어서 질의어는 객체의 움직임을 뜻하는 tuple이나 trajectory를 그대로 가져다 써야했기 때문에 사용자가 질의하기에는 힘든 부분이 많았다. 하지만 본 연구에서는 객체들의 움직임을 추상화하여 자연어로 표현하기 때문에 자연어 처리가 가능하다는 전제하에서 일반 텍스트 검색과 같이 자연어로 질의를 할 수가 있다.

‘오른쪽으로 완만하게 회전하다가 지그재그 움직임을 보이는 물체’(질의 1)라고 질의를 하게 되면 그와 비슷한 움직임을 보이는 모든 영상을 찾을 것이다. 여기서 좀 더 정확한 영상을 원한다면 시간과 거리, 방향을 질의에 포함시켜 ‘0초에서 3초까지 전방회전으로 14미터를 오른쪽으로 움직이고 5초까지 지그재그 움직임으로 10미터를 왼쪽, 오른쪽의 순서로 움직이는 물체’(질의 2)라고 질의를 하게 되면 (질의 1)에서 나온 결과물보다 좀 더 정확한 정보를 표현하는 영상을 찾을 것이다. 이보다 정확한 영상을 원한다면 각 움직임 사이의 벡터정보와 객체의 움직임이 변하는 시간들을 질의하게 되면 거의 정확한 영상이 나올 것이다.

(질의 1)과 같은 질의를 Natural language Interpretation이라고 하고 (질의 2)를 Composite Interpretation, 그리고 시간, 각에 대한 정보까지 포함한 질의를 Primitive Interpretation이라고 정의하여 최상위의 자연어 계층에서 원하는 영상을 찾지 못하면 정형화된 수식으로 표현된 Composite Motion을 비교하고 이보다 더 정확한 정보를 원할 때는 벡터정보를 포함하는 Primitive Motion을 찾는 것이다. 이를 통해 사용자 스스로 정확율과 재현율의 조절이 어느 정도 가능해진다.

본 논문의 가정에서도 언급하였듯 실제 동영상 데이터

근거 하에 실측된 예제를 통하여 본 논문에서 제안한 방식의 동영상 자료의 정보 추출과정을 보인다. 이하 예제는 주차장이라는 다소 움직임에 제약이 있는 곳에서의 예제이다. 다음 그림은 차( $O_c$ )와 사람( $O_p$ )의 움직임과 시간, 거리등을 기입하였다. 다양한 주차 방식의 처리와 사용자의 패턴을 추출하여 의미화 하는 과정을 보인다.

6. Example

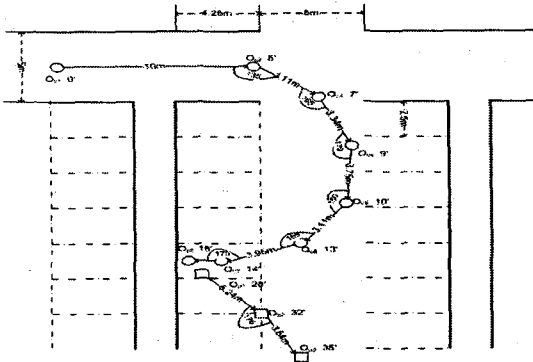


Figure 4 Parking lot Example

<Figure 4>는 차가 오른쪽에서 진입하여 오른쪽으로 회전한 뒤 주차하는 모습을 도식화 한 그림이다. 이 예제는 위에서 보인 Composite Motion의 Rule에 의해 0초에서 6초의 움직임은  $S_{0,5}^{10,0} \cdot FC_{5,6}^{3,12,+1} = FC_{0,6}^{11,12,+1}$ 로 합성된다. 이러한 Rule에 의해  $O_c$ 의 0초에서 15초의 움직임을  $(FC_{0,15}^{27,26,+1})_6$ 으로 계산 가능하다. 또한  $O_p$ 가 나타나기 전까진 어떠한 객체와도 관계를 가지지 않으므로  $NR(O_c((FC_{0,15}^{27,26,+1})_6 \cdot N_{15,28}^{0,+1}))$ 로 표현 할 수 있다. 28초 일때 차에서 운전자가 내리면 이때  $O_p$ 가 나타나는데  $Con(O_c(N_{28,28}^{0,0}), O_p(N_{28,28}^{0,0}))$ 의 관계가 성립하고 이어 운전자는 이때의 관계는  $Away(O_c(N_{28,35}^{0,0}), O_p(S_{28,35}^{88,0}))$ 가 이어진다. 결국 주차를 하는 영상은 다음의 식으로 유도된다:

NR(

$$O_c((FC_{0,15}^{27,26,+1})_6 \cdot N_{15,28}^{0,+1});$$

$$Con(O_c(N_{28,28}^{0,0}), O_p(N_{28,28}^{0,0}));$$

$$Away(O_c(N_{28,35}^{0,0}), O_p(S_{28,35}^{88,0}))$$

6. Conclusion & Future Works

본 논문에서는 동영상 안에서의 객체들의 움직임과 관계를 정형화하고 그것을 추상화하여 자연어로 표현함으로써 멀티미디어 정보를 검색함에 있어서 사용자에게 자

연어 질의를 제공하였다. 또한 모션과 관계를 추상화하는 과정에서 기본적인 움직임을 정의하고 그것들의 조합으로 복합적인 움직임을 정의하였다.

가장 기본적인 움직임을 가지는 Primitive Motion과 그것을 조합하여 생성되는 Composite Motion, 그리고 이를 사용자에게 가까운 자연어로 표현함으로써 세 계층의 데이터베이스를 구축하고 정보를 검색하는데 최상위 NL 계층에서는 객체들의 움직임과 관계를 자연어로 표현하고 이 움직임이 발생한 순서를 정의함으로써 영상에서 일어난 일을 순차적으로 표현한다. 두 번째 Composite 단계에서는 객체가 움직임에 따라 나타나는 거리와 방향성을 정의하고 있어서 NL단계에서 표현하지 못한 정보들을 가지고 있기 때문에 NL단계에서 만족할만한 성과를 얻지 못했을 때 Composite 단계의 정보들을 활용하게 된다. Primitive 단계에서는 Composite단계보다 더욱 자세한 정보들 즉, 정확한 이동거리, 속도, 각도와 같은 벡터를 저장하고 있기 때문에 객체의 전체적인 움직임이 아닌 개별적인 움직임을 비교하여 정확한 검색이 가능하다.

향후 연구에서는 객체의 움직임과 관계를 더욱 명확하게 표현하기 위해 다음과 같은 연구가 필요할 것이다. 현재의 Primitive Motion만으로는 2차원적 선의 움직임만이 표현 가능하다. 따라서 심리학과 통계학을 이용해 사람들이 일반적으로 받아들이는 기본적인 움직임을 재정의하고 3차원의 움직임에 대한 정의가 필요하다. 또한 기하학의 연구를 통해 이미 정의된 Composite Motion 이외에 객체의 움직임에 대한 추상화 과정과 두 객체 사이의 관계의 변화를 표현해야 할 것이다. 예를 들어 현재에는 사각형과 같은 움직임은 판별할 수 있지만 그것이 어떤 모양의 사각형이라고 표현해주지는 않는다. 기하학을 통해 이와 같은 움직임을 정의하고 추상화시키는 한편 두 객체 사이의 변화에 대해서도 기본적인 관계의 조합으로 복합된 관계를 정의하는 등의 연구가 필요하다.

Reference

[1] Ross Culer and Larry Davis, "Real-Time Periodic Motion Detection, Analysis, and Application", IEEE 1999  
 [2] Maren Bennewitz, Wolfram Burgard, and Sebastian Thrun, "Learning Motion Patterns of Persons for Mobile Service Robot", CiteSeer, 2002