

이동 객체의 효과적 표현을 위한 시맨틱 어노테이션 방법

이진활* 홍명덕** 이기성** 정진국** 조근식***

A Semantic Annotation Method for Efficient Representation of Moving Objects

Jin-hwal Lee * Myung-duk Hong ** Kee-sung Lee ** Jin-guk Jung ** Geun-sik Jo ***

요약

최근 비디오가 대화형 콘텐츠를 위한 타입으로 많은 각광을 받기 시작하면서 비디오 데이터에 포함된 객체들을 의미적으로 표현하고 검색하기 위한 시맨틱 어노테이션 방법에 대한 연구가 활발히 진행되고 있다. 비디오 데이터에 포함된 객체들은 시간의 변화에 따라 공간적 위치가 변화하기 때문에 매 프레임마다 상이한 위치 데이터가 발생한다. 따라서 모든 프레임의 객체에 대한 위치 데이터들을 저장하는 것은 매우 비효율적이므로 이를 부적절한 오차가 발생하지 않는 범위 내에서, 효과적으로 압축하여 표현할 필요가 있다. 본 논문은 컴퓨터 또는 에이전트가 직관적으로 객체에 대한 정보를 이해할 수 있도록 표현하기 위해 비디오 데이터가 포함하는 객체에 대하여 의미적 정보를 부여하기 위한 온톨로지 모델링 방법과 이동 객체의 위치 데이터를 압축하기 위해 3차 스플라인 보간법을 적용하여 의미적 정보와 함께 어노테이션하는 방법을 제안한다. 제안한 어노테이션 방법의 효율을 검증하기 위한 대화형 비디오 시스템을 구현하고, 다양한 특징을 가지는 객체가 나타나는 비디오 데이터 셋을 이용하여 샘플링 간격에 따른 오차율과 데이터량을 비교하였다. 그 결과, 샘플링 간격이 15프레임 이하 일 때, 최대 80%의 데이터 저장 공간을 절약할 수 있을 뿐만 아니라 객체의 실제 좌표 대비 최대 31픽셀, 평균 4픽셀 미만의 오차 편차를 얻을 수 있었다.

▶ Keyword : 시맨틱 어노테이션, 온톨로지, 이동 객체, 스플라인 보간법, 비디오

Abstract

Recently, researches for semantic annotation methods which represent and search objects included in video data, have been briskly activated since video starts to be popularized as types for interactive contents. Different location data occurs at each frame because coordinates of moving objects are changed with the course of time. Saving the location data for objects of every frame is too ineffective. Thus, it is needed to compress and represent effectively. This paper suggests two

• 제1저자 : 이진활 • 교신저자 : 이진활
• 투고일 : 2011.03.22, 심사일 : 2011.03.25, 게재확정일 : 2011.03.30.
• 인하대학교 컴퓨터공학과(Dept. of Computer Information & Engineering, Inha University)

methods; the first, ontology modeling for moving objects to make users intuitively understandable for the information, the second, to reduce the amount of data for annotating moving objects by using cubic spline interpolation. To verify efficiency of the suggested method, we implemented the interactive video system and then compared with each video dataset based on sampling intervals. The result follows : when we got samples of coordinate less than every 15 frame, it showed that could save up to 80% amount of data storage; moreover, maximum of error deviation was under 31 pixels and the average was less than 4 pixels.

▶ Keyword : Semantic Annotation, Ontology, Moving object, Spline interpolation, Video.

I. 서론

다양한 영상기술의 발전과 저장 매체의 대용량화로 인해 대규모 영상 정보의 용이한 저장, 관리 및 전송이 가능해 졌다. 이에 따라, 대규모의 영상 정보들에 관하여 효과적으로 질의를 하고 이를 처리해야하는 문제가 새롭게 대두되고 있다. 이를 위해 비디오 데이터를 효율적으로 저장하고 검색하기 위한 많은 연구[1,2,3,4,5,6,7]들이 진행되고 있으며 특히 비디오 데이터에 포함된 객체, 사건과 같은 의미적 내용(Semantic Content)을 표현하고 검색하기 위한 연구[3,4,5]가 활발하다. 또한, 사용자의 함축적인 질의를 정확하게 해석하고 이해하여, 사용자의 의도에 가장 적절한 결과를 제공하는 비디오 검색을 위해 온톨로지를 이용하여 비디오의 각 프레임에 존재하는 객체에 대해 시맨틱 어노테이션(Semantic Annotation)하는 연구[7,8]들이 진행되고 있다. 아울러, 최근 대화형 서비스를 위한 타입으로 인터넷TV, 스마트TV와 같은 양방향 커뮤니케이션 그리고 모바일을 이용한 광고, 엔터테인먼트, 교육 등의 많은 분야에서 현실세계의 주어진 상황에 기인하여 직감적인 사용자 인터페이스 환경을 제공하는 증강현실과[9,10] 인터랙티브 비디오가[11] 주목을 받고 있다. 증강현실에서 카메라 또는 객체의 움직임이 있어도 인식된 객체는 트래킹을 통해 사용자가 직관적으로 알 수 있도록 객체 정보를 객체 주변에 표현해 준다. 이러한 점에 착안하여 본 논문은 증강현실을 위해 사용되는 기술을 비디오에 접목시켜, 비디오 내에 나타나는 객체들에 대해서 의미적으로 질의하고 이를 처리하기 위한 정보를 RDF나 OWL과 같이 기계가 이해할 수 있도록 기술하여 어노테이션하는 방법을 제안한다.

또한 비디오 데이터는 시간의 변화에 따라 공간적인 위치 변화를 나타내는 이동 객체를 포함하고 있다. 비디오 데이터에 포함된 이동 객체를 표현하기 위해서는 시간의 변화에 따른 객체의 위치 변화를 데이터베이스에 저장할 효율적인 방법

이 필요하다. 일반적으로 데이터베이스를 이용하여 연속적인 데이터를 관리할 경우, 매 프레임마다 변경된 모든 위치 데이터를 저장하는 것은 매우 비효율적이므로 일정한 비율의 시간 주기를 결정하여 객체의 위치 데이터를 저장하게 된다. 이 때 데이터베이스에 저장되지 않은 임의의 시점에 대해서는 위치 데이터가 존재 하지 않으므로 불확실한 위치 데이터를 적절히 표현할 수 있는 방법이 필요하다. 전통적으로 이동 객체의 위치를 추정하기 위한 방법으로는 두 시점 이상의 위치 데이터를 사용하여, 저장되지 않은 시점의 이동 객체의 위치를 추정하는 방법들이 연구되었다[12,13,14,15].

이 논문에서는 이러한 불확실한 위치 데이터를 표현하기 위해 3차 스플라인 보간법(Cubic Spline Interpolation)을 사용하였으며, 시맨틱 어노테이션을 위해 비디오 데이터에 나타난 객체들과 객체들간의 관계를 온톨로지로 표현하였다.

본 논문의 구성은 다음과 같다. 먼저, 2장에서는 본 연구와 관련된 기존 연구를 소개한다. 3장에서는 객체 표현을 위한 온톨로지 모델링 방법과 전체적인 동작 프레임워크를 제시하고, 객체의 이동 궤적을 어노테이션하는 방법을 다룬다. 4장에서는 실험 및 성능 평가 결과를 제시하고, 마지막으로 5장에서는 결론 및 향후 연구 방향을 기술한다.

II. 관련 연구

비디오 내에 이동 객체를 표현하기 위해서는 시간에 따라 연속적으로 변하는 공간적 속성을 표현할 수 있는 표현력이 중요하다. 본 연구에서는 크기나 모양에는 상관없이 시간에 따라 공간적 위치가 연속적으로 변하는 객체 표현을 다룬다. 시간에 따라 연속적으로 위치가 변함으로 이동객체의 위치는 시간에 대한 함수형태로 나타내고, [그림1]의 (a)와 같이 (x, y, t) 의 3차원 공간상의 연속적인 무한 점의 집합인 추상적 데이터 모델로 표현할 수 있다. 그러나 실제 데이터베이스에 무한개의 집합을 저장할 수 없고, 이동 객체의 데이터를 수집하는 면에서도 일정시간단위로 객체의 위치 값을 이산적으로

얻어온다. 따라서, 이동 객체를 유한 집합으로 표현할 수 있는 이산적 데이터 모델이 필요하고, 이산적 데이터 모델로 표현된 이동 객체는 데이터베이스에 효율적으로 저장되어질 수 있다. 위의 추상적 모델로 표현된 이동 객체의 위치를 이산적 모델로 대응하게 되면 [그림1]의 (b)와 같은 3차원상의 유한 점의 집합으로 구성되는 다각선으로 표현된다.

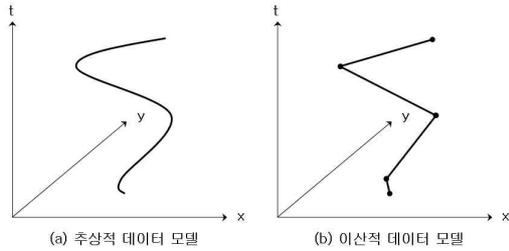


그림 1. 데이터 모델
Fig. 1. Data model

이러한 이산적 모델은 추상적 모델과의 대응에 있어서 각 이산 구간 사이의 값을 보간하는 방법과 함께 표현된다. [그림1]의 (b)의 경우에는 간단하게 선형 보간법을 사용하여 이산적 데이터 모델을 표현하였다. 추상적 데이터 모델의 위치 데이터를 일정 주기로 샘플링하여 데이터량을 줄이고 샘플링 간격 사이의 불확실한 시점의 위치 데이터를 추정하기 위한 방법으로 간단한 직선기반 보간법(Linear Interpolation)과 곡선 기반인 라그랑제 보간법(Lagrange Polynomial Interpolation), 뉴턴 보간법(Newton Interpolation), 스플라인 보간법(Spline Interpolation)[16]등이 이용되고 있다.

비디오 내에서 연속적으로 위치가 변하는 이동 객체의 움직임을 어노테이션해야 할 때 이동 객체의 전체 또는 부분적인 움직임을 표현할 필요가 있다. 이동 객체의 움직임을 모두 처리하기 위해서는 가장 단순한 방법은 모든 시간에 대해 연속적으로 변하는 이동 객체의 위치를 저장하는 것이다. 그러나, 이 방법은 이산적인 데이터일지라도, 데이터 베이스에 저장되는 데이터의 양을 엄청나게 증가시키게 되어 효율적이라고 판단하기 어렵다. 이와 같이 데이터 저장 공간이 커지는 문제점을 해결하기 위해 지금까지 연구된 이동 객체의 위치 관리 방법에서는 정규적인 시간의 간격에서 정렬된 순서대로 객체의 위치를 획득하여 선형 보간법을 적용하였다. 하지만 비디오 내 이동 객체는 선형 보간법인 직선으로 표현하기 보다는 불규칙적인 움직임까지 표현할 수 있는 방법이 필요하다. 이러한 방법으로 [12]는 스트림 형태로 발생하는 위치 데이터에 대하여 제한된 메모리를 사용하여 임의의 과거시점 위

치를 추정하는 방법을 제안하였다. 데이터베이스에 저장되지 않은 이동 객체의 불확실한 시점의 위치 데이터를 추정하기 위해 라그랑제 다항식 보간법을 사용하였다. 비디오는 수많은 프레임들로 구성되어 있으며, 보통 디지털 비디오는 1초에 15~30 프레임들이 포함되어 있다. 따라서, 매 프레임마다 변화하는 객체의 움직임을 라그랑제 다항식 보간법 또는 뉴턴의 보간법으로 해결하기에는 적절하지 않다. 왜냐하면, 주어진 데이터들을 라그랑제 다항식 보간법이나 뉴턴의 방법으로 보간한다면 차수가 늘어남에 따라 계산 복잡성이 커지고 더욱 큰 오차가 발생하게 될 것이다. [그림2]에서 차수가 작은 (a)와는 다르게 차수가 큰 (b)의 경우, x축의 첫 번째 데이터 주변 0과 2사이와 마지막 데이터 주변 근처 8과 10사이에서 진동이 발생하여 그 구간 사이에서 보간 할 경우, 큰 오차가 예상된다.

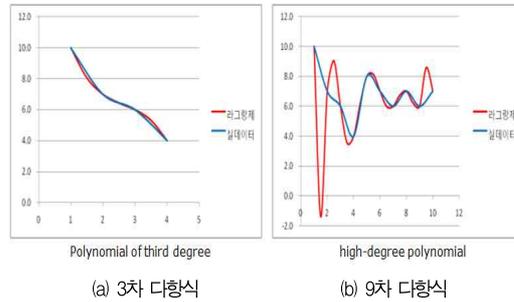


그림 2. 라그랑제 다항식 보간
Fig. 2. Lagrange Polynomial Interpolation

이동 객체의 불확실한 위치 정보로 인해 데이터 모델링 질의 처리, 인덱싱, 질의 결과의 부정확성 등이 발생된다. 보간법의 정확도는 사용하는 다항식의 차수가 커질수록 높아진다고 말할 수 없으며, 그렇다고 해서 점과 점 사이를 직선으로 보간한다면 많은 오차를 발생시키므로, 좀 더 정확한 위치 추정 방법이 필요하다. 따라서, 다항식의 차수가 큰 경우에는 [그림3]과 같이 주어진 구간을 여러 개의 소구간으로 나누고, 소구간별로 보다 낮은 차수의 다항식을 사용하여 데이터 점들을 지나는 매끈한 곡선을 얻는 것이 바람직하다.

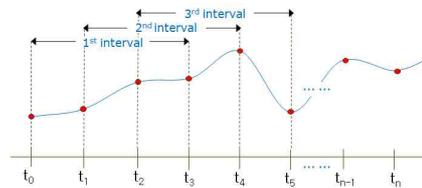


그림 3. 스플라인 보간법
Fig. 3. Spline Interpolation

이를 위해 스플라인 보간법(Spline Interpolation)을 사용하여 이동 객체의 위치 변화를 표현하는 것이 적절하다. 이 방법에서 소구간 사이를 연결하는 곡선으로는 직선, 2차 다항식, 3차 다항식이 있으나, 그 중에서도 3차 다항식이 가장 많이 사용되고 있다. 3차 스플라인 보간법(Cubic Spline Interpolation)[16]은 데이터 점들의 집합을 다항식 곡선으로 나타낸 것으로 주어진 점들을 지나면서 오차에 영향을 받지 않는 유연한 이동 객체의 궤적을 그릴 수 있다.

III. 이동 객체에 대한 시맨틱 어노테이션

본 장에서는 비디오 데이터에 나타나는 장면 및 이동 객체 표현을 위한 온톨로지 기반 어노테이션 동작 프레임워크를 3.1절에서 기술하고, 3.2절에서는 이동 객체를 효과적으로 어노테이션하기 위해, 인식된 객체를 트래킹하는 과정에서 발생하는 다량의 위치 데이터를 샘플링하여 3차 스플라인 보간법을 적용한 방법에 대해 기술한다. 3.3절에서는 지식베이스로부터 추출된 객체 정보들간의 의미적 차이를 줄이기 위해 객체의 본질과 객체간의 관계를 나타내는 온톨로지를 이용한 모델링 방법을 기술하였다

3.1. 시스템 구조

제안된 시스템은 비디오 내 지속적으로 변화하는 객체에 대한 위치 데이터를 샘플링하고 지식베이스로부터 추출된 객체 관련 정보와 함께 샘플링된 위치 정보를 온톨로지를 이용하여 어노테이션하는 시스템이다.

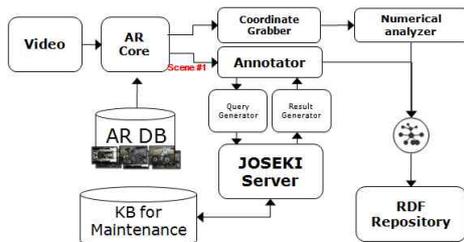


그림 4. 시스템 구조도
Fig. 4. System architecture

[그림4]는 비디오 내 각각의 객체들에 대해 어노테이션하기 위한 구조를 나타낸다. 그림을 보면, 우선 AR Core가 비디오 프레임으로부터 객체를 인식하고 각 객체의 위치 데이터를 추출한다. 이때 객체의 위치 데이터는 비디오의 매 프레임마다 발생하기 때문에 Numerical Analyzer가 일정한 프레

임의 주기로 n개의 위치를 샘플링을 하여 이동 객체의 궤적을 나타낼 수 있는 3차 다항식의 미정계수를 생성한다. 그리고 지식베이스로부터 추출된 객체 관련 정보와 생성된 미정 계수는 온톨로지 스키마를 통해 정제된 후 최종적으로 RDF 저장소로 이동된다.

3.2. 이동 객체 궤적 추정

일정한 시간 주기를 결정한 후, 이동 객체의 위치를 샘플링하여 각 시간 구간마다 변화된 위치 정보를 계산할 수 있는 다항식을 생성한다. 그리고 다항식에 있는 각 계수들의 값을 데이터베이스에 저장한다. 이를 위해, 3차 스플라인 보간법을 사용하여 이동 객체의 위치 변화를 표현한다. 3차 스플라인 보간법은 데이터 점들의 집합을 다항식 곡선으로 나타낸 것으로 주어진 점들을 지나면서 오차에 영향을 받지 않는 유연한 곡선을 그릴 수 있다.

비디오 내 이동 객체의 특정 시점에 해당하는 위치는 시점 t 와 좌표 값 x, y 를 (시간, 위치 데이터)형태인 2차원 데이터로 표현할 수 있다. 이동 객체를 O 라 하고, O 의 2차원 위치 데이터를 (x, y) 로 표현한다. 객체 O 는 시간에 따라서 위치 데이터가 변화하므로, 시간 값인 t 를 추가하여 특정 시점 t_k 에 해당하는 이동 객체 O 의 위치 데이터 (x_k, y_k) 를 구할 수 있다. 이동 객체 O 의 x 축 및 y 축의 위치 데이터를 각각 시간 축을 기준으로 표현하면 x 축의 경우 (t, x) , y 축의 경우 (t, y) 로 나타낼 수 있다. 따라서 실제 위치 좌표를 구하기 위해서는 x 와 y 의 값을 구하기 위한 개별적인 두 개의 3차 다항식이 필요하다.

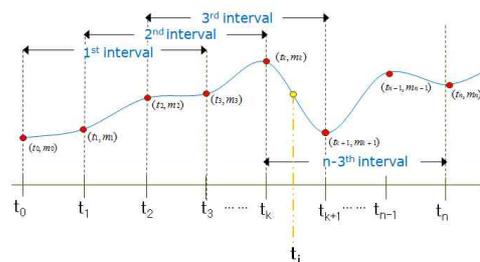


그림 5. 3차 스플라인 보간
Fig. 5. Cubic Spline Interpolation

[그림5]와 같이 $n+1$ 개의 점으로 구성된 집합 $(t_k, m_k)_{k=0}^n$ 이 있을 때, n 개의 부분구간 3차 다항식을 구함으로써, 연속적으로 존재하는 모든 구간에서의 추정된 위치

좌표 값을 구할 수 있다. 이때 m 은 (x, y) 좌표 값을 의미하며, 이동객체가 나타나는 전체 유효 시간 간격이 $T_a = [t_0, t_n]$ 이면, 여기에서 t_0 는 저장된 유효 시간의 시작 시점을 의미하고 t_n 은 마지막 시점을 나타낸다. T_{intv_0} 는 $T_{intv_0} = [t_0, t_1]$, $T_{intv_1} = [t_1, t_2]$, ..., $T_{intv_{n-2}} = [t_{n-2}, t_{n-1}]$, $T_{intv_{n-1}} = [t_{n-1}, t_n]$ 과 같이 n 개의 부분 시간 구간 T_{intv_k} 로 분할할 수 있으며, 이때 $\{k=0, 1, \dots, n-1\}$ 이 된다. 임의의 시점에 대한 객체 위치는 질의 시점 t_j 에 대해, $t_j \in T_{intv_j}$, $T_{intv_j} = [t_j, t_{j+1}]$ 의 조건을 만족할 때 t_j 에서의 위치 좌표 값 m_j 를 구하는 부분구간 3차 스플라인 보간 다항식은 (t_{j-1}, m_{j-1}) , (t_j, m_j) , (t_{j+1}, m_{j+1}) , (t_{j+2}, m_{j+2}) 의 네 점을 이용하여 구한다. 실제 객체의 위치 좌표는 x 와 y 의 값을 구하기 위한 개별적인 두 개의 3차 다항식으로, x 좌표 (t_{j-1}, x_{j-1}) , (t_j, x_j) , (t_{j+1}, x_{j+1}) , (t_{j+2}, x_{j+2}) , y 좌표 (t_{j-1}, y_{j-1}) , (t_j, y_j) , (t_{j+1}, y_{j+1}) , (t_{j+2}, y_{j+2}) 를 이용하여 계산한다.

3.2.1. 추정 위치 연산

추정 위치 연산을 위한 각 부분 구간 3차 다항식으로서 보간식 $S(t)$ 은 식 (1)과 같이 나타낼 수 있다[16].

$$S(t) = a_i(t-t_i)^3 + b_i(t-t_i)^2 + c_i(t-t_i) + d_i \quad (1)$$

for, $t = [t_i, t_{i+1}]$, $i = 0, (N-1)$

식 (1)은 x 좌표 값을 구하기 위한 함수이고, $X_i(t_p)$ 와 y 좌표 값을 구하기 위한 함수 $Y_i(t_p)$ 는 식 (2)와 식 (3)으로 나타낸다.

$$X_i(t_p) = a_i(t_p - t_i)^3 + b_i(t_p - t_i)^2 + c_i(t_p - t_i) + \quad (2)$$

$$Y_i(t_p) = a_i(t_p - t_i)^3 + b_i(t_p - t_i)^2 + c_i(t_p - t_i) + \quad (3)$$

각 부분 구간에 대해 4개의 미정 계수 a_i, b_i, c_i, d_i 를 결정해야 하므로 n 개의 부분 구간에 대하여 $4n$ 개의 조건식[16]을 필요로 한다. 이러한 미정 계수들을 결정하기 위해서 알 수 있는 조건은 다음과 같다. 3차 다항식은 부분 구간 양 끝점을 지나야 한다. (2n개의 조건)

조건1. $S_i(t_i) = y_i, \{i = 1, 2, \dots, n-1\}$

조건2. $S_i(t_{i+1}) = y_{i+1}, \{i = 1, 2, \dots, n-1\}$

각 부분 구간의 3차 다항식들은 그들이 만나는 양 끝점에서 1계 도함수(기울기)와 2계 도함수(곡률)는 같아야 한다. (2n-2개의 조건)

조건3. $S_{i-1}'(t_i) = S_i'(t_i), \{i = 1, 2, \dots, n-1\}$

조건4. $S_{i-1}''(t_i) = S_i''(t_i), \{i = 1, 2, \dots, n-1\}$

식 (1)은 두 점을 지나게 되므로 두 점을 대입한 결과식은 식(4)와 식 (5)와 같이 표현된다.

$$S_i(t_i) = a_i(t_i - t_i)^3 + b_i(t_i - t_i)^2 + c_i(t_i - t_i) + d_i = d_i \quad (4)$$

$$S_{i+1}(t_{i+1}) = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i, \quad h_i \triangleq t_{i+1} - t_i \quad (5)$$

식 (4)와 식 (5)의 1차 미분과 2차 미분을 구해보면 식 (6)과 식 (7)과 같이 표현된다.

$$S' = 3a_i(t-t_i)^2 + 2b_i(t-t_i) + c_i \quad (6)$$

$$S'' = 6a_i(t-t_i) + 2b_i \quad (7)$$

위의 식들로부터 미정 계수 a_i, b_i, c_i, d_i 는 식 (2)와 (3)을 이용하여 각각 구한다.

$$a_i = \frac{S_{i+1} - S_i}{6h_i}, \quad b_i = \frac{S_i}{2}, \quad (8)$$

$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_i s_i + h_i S_{i+1}}{6},$$

$$d_i = y_i$$

이제, (t_{i-1}, m_{i-1}) , (t_i, m_i) 와 (t_i, m_i) , (t_{i+1}, m_{i+1}) 의 관계에서 공통으로 갖는 한 점 (t_i, m_i) 의 미분한 값이 동일하다는 조건을 이용하여 식 (6)에 (t_i, m_i) 의 값을 대입해 보면 식 (9)와 같이 표현된다.

$$S' = 3a_i(t_i - t_i)^2 + 2b_i(t_i - t_i) + c_i = c_i \quad (9)$$

(t_{i-1}, m_{i-1}) 와 (t_i, m_i) 의 관계에서 주어진 1차 미분 값에 (t_i, m_i) 값을 대입하면 식 (10)와 같이 표현된다.

$$S' = 3a_{i-1}(t_i - t_{i-1})^2 + 2b_i(t_i - t_{i-1}) + c_{i-1} \quad (10)$$

$$= 3a_{i-1}h_{i-1}^2 + 2b_{i-1}h_{i-1} + c_{i-1}$$

이를 간단히 하면 식 (11)과 같이 된다.

$$h_{i-1}S_{i-1} + (2h_{i-1} + 2h_i)S_i + h_iS_{i+1}$$

$$= 6 \left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}} \right) \quad (11)$$

4n개의 미정 계수를 결정하기 위해서는 4n개의 조건이 필요하지만, 조건1, 2, 3, 4를 통해서 얻을 수 있는 조건의 개수는 4n-2개로 식 (11)은 i=1부터 i=n-1까지의 점들에 대해서 이용될 수 있다. 따라서 S_0 과 S_n 의 값은 알지 못하므로 2개의 조건이 부족하다. 이를 해결하기 위한 조건을 끝점 조건이라고 한다[16]. 이 끝점 조건을 구하는 방법에 따라 조금씩 형태가 다른데 본 논문에서는 포물선으로 끝나는 끝점 조건을 다음과 같이 정한다.

$$S_1 = S_2, S_n = S_{n-1}$$

이제 S_1, S_2, \dots, S_n 의 값을 모두 알 수 있으므로, 식 (8)에 대입하여 4개의 계수를 구한 후 각 계수를 식 (2)와 (3)에 대입하면 시점 t 에서의 위치 좌표 x, y 값을 구할 수 있다.

표 1. 예제 데이터
Table 1. Example data

t	1s	2s	3s	4s	5s	6s	7s	8s	9s	10s
x	2	1	2	2	1	3	2	1	3	2
y	4	3	2	4	4	3	2	3	4	3

예를 들어 10초의 재생 시간을 갖는 비디오에서, 1초의 간격을 두고 저장된 이동 객체의 위치 데이터 <표 1>을 알고 있다고 가정하자. 이 때 저장되지 않은 시점 t_p 에 대한 위치 데이터 x, y 는 $t_p \in T_{intv_k}, T_{intv_k} = [t_k, t_{k+1}]$ 을 만족하는 추정 함수는 식 (2), (3)을 이용하여 구한다. 먼저 식 (2)를 이용하여 $X_i(t_p)$ 를 구할 때, 각 구간에 대한 곡선을 구하기 위한 미정 계수들은 식 (8)을 이용하여 <표 2>와 같이 구할 수 있다. 임의의 시점 $t_p = 1.2$ 에 대한 $X(1.2)$ 는 $[t_1, t_2]$ 구간에 해당하므로 값 1.79를 식 (2)로부터 얻을 수 있다.

표 2. 예제 결과
Table 2. The result of example

t \ 계수	a	b	c	d
if $x \in [1, 2]$	0.569219	0	-1.56922	2
if $x \in [2, 3]$	-0.8461	1.707658	0.138439	1
if $x \in [3, 4]$	-0.18483	-0.83063	1.015464	2

if $x \in [4, 5]$	1.585424	-1.38513	-1.2003	2
if $x \in [5, 6]$	-2.15686	3.371143	0.78572	1
if $x \in [6, 7]$	1.042027	-3.03945	1.057418	3
if $x \in [7, 8]$	0.988753	0.026637	-2.01539	2
if $x \in [8, 9]$	-1.99704	2.992897	1.004144	1
if $x \in [9, 10]$	0.999408	-2.99822	0.998816	3

t_p 에서 y 좌표 값을 구하는 함수 $Y_i(t_p)$ 는 식 (3)을 이용하여 구하며 $t_p \in T_{intv_k}, T_{intv_k} = [t_k, t_{k+1}]$ 을 만족할 때, 4개의 미정 계수를 구하는 방법은 $X_i(t_p)$ 를 구할 때와 동일한 식 (8)을 이용하여 구한다.

3.3. 시맨틱 어노테이션

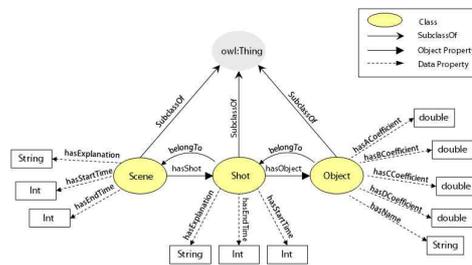


그림 6. 비디오 내의 움직이는 객체를 위한 온톨로지
Fig. 6. Ontology for Moving Object in Video

사용자의 함축적인 질의를 정확하게 해석하고 이해하여, 사용자의 의도에 가장 적절한 결과를 제공하는 효과적이고 효율적인 비디오 검색을 위해 [그림6]과 같은 구조를 갖는 온톨로지를 이용하여 비디오의 각 프레임에 존재하는 객체에 대해 시맨틱 어노테이션(Semantic Annotation)을 한다. [그림6]의 온톨로지는 3개의 클래스를 가지고 있다. 객체에 대한 정보(이름, 움직임)를 표현하기 위해 Object 클래스를 정의하였고, Object 클래스는 hasName, hasAcoefficient, hasBcoefficient, hasCcoefficient, hasDcoefficient의 속성을 갖는다. 그리고 비디오를 구성하는 단위인 Shot과 Scene 클래스를 정의하였다. Shot과 Scene 클래스는 각각 hasStartTime, hasEndTime의 속성을 갖는다. <표 3>은 각각의 클래스가 가지는 프로퍼티들에 대해서 기술한다. 각각의 표는 속성이름과 속성이 갖는 Range, 속성의 개수 즉, Cardinality를 기술하였다(N은 0개 이상의 Cardinality를 갖는다).

표 3. 비디오 내의 움직이는 객체를 위한 온톨로지 속성
Table 3. Ontology Properties for Moving Object

Class	Properties	Range	Cardinality
Scene	hasStartTime	Int	1
	hasEndTime	Int	1
	hasExplanation	String	1
	hasShot	Shot Class	N
Shot	hasStartTime	Int	1
	hasEndTime	Int	1
	hasExplanation	String	1
	belongTo	Scene	1
	hasObject	Object Class	N
Object	hasName	String	1
	hasAcoefficient	Double	1
	hasBcoefficient	Double	1
	hasCcoefficient	Double	1
	hasDcoefficient	Double	1
	belongTo	Shot Class	1

각 프레임에서 추출된 객체들에 대한 정보는 [그림7]과 같은 RDF 형태로 표현이 된다.

```

<Scene rdf:ID="Scene_01">
  <hasStartTime rdf:datatype="&xsdtint">"Value"</hasStartTime>
  <hasEndTime rdf:datatype="&xsdtint">"Value"</hasEndTime>
  <hasShot rdf:resource="#Shot_001"/>
  ...
</Scene>
...
<Shot rdf:ID="Shot_001">
  <hasStartTime rdf:datatype="&xsdtint">"Value"</hasStartTime>
  <hasEndTime rdf:datatype="&xsdtint">"Value"</hasEndTime>
  <hasObject rdf:resource="#Object_001"/>
</Shot>
...
<Object rdf:ID="Object_001">
  <hasName rdf:datatype="&xsdtstring">"name of object"</hasName>
  <hasAcoefficient rdf:datatype="&xsdtdouble">"value of coefficient"/>
  <hasBcoefficient rdf:datatype="&xsdtdouble">"value of coefficient"/>
  <hasCcoefficient rdf:datatype="&xsdtdouble">"value of coefficient"/>
  <hasDcoefficient rdf:datatype="&xsdtdouble">"value of coefficient"/>
</Object >
...
    
```

그림 7. RDF 기반 메타데이터
Fig. 7. RDF based metadata

IV. 실험

제한한 어노테이션 방법의 효용성을 검증하기 위해 비디오 내 이동 객체에 대한 모든 프레임의 위치 데이터를 이용하여 3차 스플라인 보간법을 적용하여 데이터 압축률과 오차율을 비교 분석한다. 본 논문에서는 임의의 시점에서 실제위치와 추정된 위치 간의 거리 차이를 오차로 정의하였으며, 비디오의 해상도와 객체의 크기에 따른 상대적 오차 거리는 고려하지 않았다. 실험을 위한 비디오 데이터 셋을 이동 객체가 빠르게 움직이는 경우, 완만하게 움직이는 경우, 느리게 움직이는 경우로 구분하여 실험을 하였으며, 위치의 편차를 구하기 위해서 실험 대상이 되는 이동 객체의 시간에 따른 완전한 이력 위치 좌표 값을 저장하고 이를 일정한 간격으로 균등하게 샘플링한 후, 이에 따른 추정된 위치 좌표 값과 실제 위치 좌표 값의 오차율, 그리고 어노테이션 데이터량의 차이를 각각 비교 분석하였다. 실험 대상이 되는 이동 객체는 자동차, 항공기 정비 과정, 특정 상품의 광고를 포함하는 3개의 비디오 내 10개의 이동 객체를 이용하였다. 각 비디오에서 적용된 샘플링 간격은 동일하게 4, 7, 15, 30, 45 프레임이며, 이는 위치 좌표를 획득하는 일정한 시간의 주기를 의미한다. 샘플링 간격이 4일 때 데이터베이스에 저장되는 위치 데이터량이 가장 많고 45일 때 가장 적은 위치 데이터를 가진다.

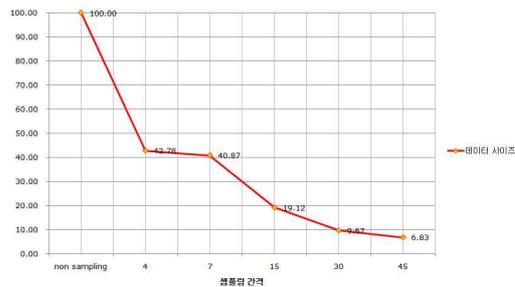


그림 8. 데이터 압축률
Fig. 8. Reduction rate of data

[그림8]의 그래프에서 알 수 있듯이 샘플링의 주기가 짧을 수록 저장되는 데이터 크기가 줄어든다. 이는 3개의 비디오 내 객체들을 각 샘플링 주기에 따라 어노테이션하고, 각 샘플링 간격별 어노테이션 된 데이터 크기의 평균으로 나타내었다. 모든 위치 데이터를 저장했을 경우의 데이터 크기 대비, 각 샘플링 간격에 따라 42%, 40%, 19%, 10%, 7%의 저장 공간을 차지함으로써 데이터 크기가 크게 감소시킬 수 있음을 확인하였다.

표 4. 샘플링 간격별 최대 오차(Pixel)
Table 4. The biggest error by sampling intervals

Object Frame	4	7	15	30	45
Static Object	2.87	3.25	3.33	5.60	12.28
Gradual Object	5.65	10.71	22.68	22.51	31.19
Fast Object	12.09	24.67	30.68	78.71	124.63

이에 대한 각 구간별 최대 오차는 <표 4>에 나타내며, 각 비디오에서 나타나는 객체의 위치 편차는 [그림 9-11]과 같다.

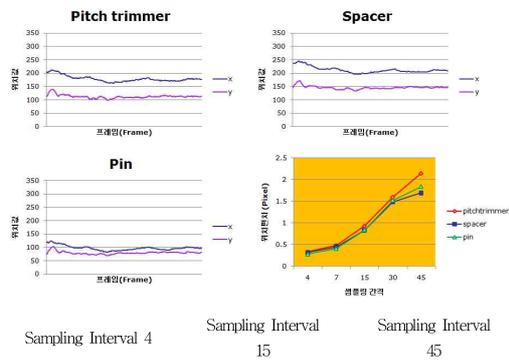


그림 9. 정적 객체 (랜딩기어 부품)
Fig. 9. Static Objects (Landing gear parts of aircraft)

[그림9]의 우측 하단의 결과는 정적인 객체에 대한 추정 위치 연산을 하였을 때 나타나는 각 샘플링 간격별 추정 위치 오차 값을 보여준다. 오차 변화 추이는 샘플링 간격이 커질수록 추정 위치 연산 결과의 오차가 커지는 현상을 보이고 있다. 이는 [그림 9-11]의 모든 실험에서 동일하게 나타나는 결과이다. 이것은 카메라의 흔들림으로 비디오 해상도 내 객체의 위치 좌표가 바뀌거나, 실험 대상인 객체들이 불규칙하게 움직이는 특성을 가지고 있기 때문이다. [그림9]의 그래프에서 x와 y좌표가 프레임에 따라 비교적 평행하게 보이는 것은 객체가 움직이는 특성을 지니기 보다는 카메라의 흔들림으로 위치 편차가 발생함을 알 수 있다. 따라서 정적인 객체에 추정 위치 연산을 하였을 때 샘플링 간격을 45로 했을 경우, 최대 12픽셀, 평균 3픽셀 미만의 위치 편차를 보여주었으며 또한 카메라의 흔들림에도 어노테이션된 가상 객체의 흔들림은 감소됨을 확인하였다.

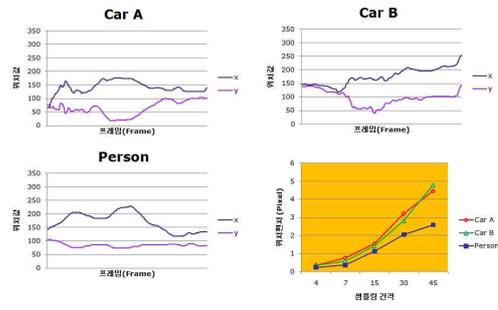


그림 10. 완만하게 움직이는 객체 (주차장)
Fig. 10. Gradual Moving Objects (Parking Lot)

[그림10]의 결과는 주차장의 감시카메라(Surveillance Camera)에 나타나는 3가지의 객체의 추정 위치 오차 값을 보여준다. 샘플링 간격을 45로 했을 때, 주차장 공간 내 비교적 완만하게 움직이는 객체들은 정적 객체보다 높은 최대 31 픽셀, 평균 5픽셀 이하의 오차 값을 얻을 수 있었다.

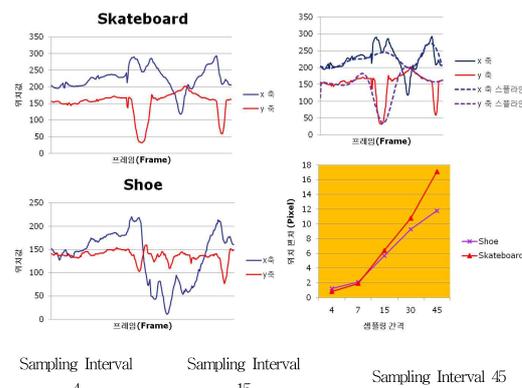


그림 11. 빠르게 움직이는 객체 (스케이트보드)
Fig. 11. Fast Moving Objects (Skateboard)

[그림11]은 스케이트보드, 신발과 같은 제품을 홍보하는 비디오를 이용하여 실험한 결과를 보여준다. 샘플링 간격 45로 하였을 때, 최대 오차 값은 124픽셀, 평균 18픽셀 이하의

비교적 큰 오차가 발생하였다.(가로 37cm, 세로 25cm의 모니터의 해상도가 1024x768 일 때, 124 픽셀 \approx 4.3cm) 이는 객체가 급격히 움직이는 구간에 대한 샘플링을 하지 못했기 때문이다. [그림11]의 우측 상단의 그래프는 45프레임의 주기로 스플라인 보간법을 적용한 위치와 실제 이동 객체의 위치를 보여준다. 이동 객체가 급격한 움직임을 보이는 구간에 대한 샘플링을 하지 않고 보간하였기 때문에 큰 오차가 발생하였다.

V. 결론

본 논문은 객체를 의미적으로 표현하고 검색하기 위해 시맨틱 기반 어노테이션 방법을 적용하고, 불규칙적으로 움직이는 객체에 적합한 곡선 기반 보간법인 3차 스플라인 보간법을 이용하여 어노테이션을 위한 데이터량을 크게 압축하는 것과 동시에 실제 이동객체의 궤적과 근사한 위치 정보를 표현 할 수 있는 방법을 제안하였다.

추정 위치 실험 결과를 토대로 세 개의 비디오에서 발생한 이동 객체의 위치 편차 평균을 각 샘플링 간격에 따라 비교하였다. 샘플링 간격이 30일 때, 최대 오차는 약 80픽셀, 45일 때, 최대 오차는 124픽셀의 오차 값을 가지므로 이는 이동 객체의 크기가 작을 시에 상대적으로 큰 오차 거리를 가지므로 샘플링 간격을 30프레임 이상으로 할 경우, 적절치 않음을 알 수 있다. 따라서, 본 실험에서는 샘플링 임계값을 15프레임 이하의 간격을 두고 하는 것이 어노테이션 데이터량을 최대 80% 감소시킬 뿐만 아니라, 최대 31픽셀의 오차 거리를 보여주므로 이동 객체의 속도에 크게 영향을 받지 않고 객체의 위치를 적절히 표현할 수 있음을 확인하였다.

본 논문에서 제안된 방법은 두 가지의 장점을 갖는다. 첫 번째는 시맨틱 어노테이션 정보를 생성함으로써 비디오 내 객체가 가지는 개념, 그리고 객체와 객체의 관계 등을 표현하여 의미적인 정보를 전달 할 수 있다는 것이다. 두 번째는 3차 스플라인 보간법을 이용하여 데이터 저장 공간을 절약할 수 있을 뿐만 아니라, 동시에 이동 객체의 궤적을 효율적으로 표현하기 위한 방법을 제시하였다는 것이다.

향후에는 추정 위치 오차를 줄이기 위한 방법과 더불어서, 이동 객체의 속도에 따라 샘플링 간격을 미리 예측하고 이동 객체의 크기, 회전 등과 같은 속성들을 고려하여 효과적으로 어노테이션 하기 위한 연구가 필요하다. 아울러 그러한 속성들을 의미적으로 정의하여 명확하게 객체를 표현하기 위한 방법에 대한 연구도 함께 수행되어야 할 필요가 있다.

참고문헌

- [1] J.H. Lee, S.-B. Park, Y.W. Kim, and G.S. Jo, "Automatic Video Annotation using Extraction of Character Name," KCC, Vol. 36, No. 1, pp. 525-530, 2009.
- [2] M. Everingham, J. Sivic, and A. Zisserman, "Taking the bite out of automated naming of characters in TV video," The 17th British Machine Vision Conf. Image and Vision Computing, Vol. 27, pp. 545-559. April 2009.
- [3] S.-B. Park, Y.W. Kim and G.S. Jo, "Major Character Extraction using Character-Net," KSII, Vol. 11, No. 1, pp 85-102, 2010.
- [4] L. Chen, G.C. Chen, C.Z. Xu, J. March, and S. Benford, "EmoPlayer: A media player for video clips with affective annotations," Interacting with Computers, Vol. 20, No. 1, pp. 17-28, Jan. 2008.
- [5] J.R. Cózar, N. Guil, J.M. González-Linares, E.L. Zapata, and E. Izquierdo, "Logotype detection to support semantic-based video annotation," Signal Processing: Image Communication, Vol. 22, No. 7-8, pp. 669-679, Sep. 2007.
- [6] Y. Li, Y. Zhang, J. Lu, R. Lim, and J. Wang, "Video Analysis and Trajectory Based Video Annotation System," Wearable Computing Systems (APWCS), Asia-Pacific Conference on, pp. 307-310, April 2010.
- [7] A.R.J. Francois, R. Nevatia, J. Hobbs, R.C. Bolles, and J.R. Smith, "VERL: an ontology framework for representing and annotating video events," Multimedia, IEEE, Vol. 12, No. 4, pp. 76- 86, Oct.-Dec. 2005.
- [8] V.S. Tseng, Su. Ja-Hwang, H. Jhih-Hong, and C. Chih-Jen, "Integrated Mining of Visual Features, Speech Features, and Frequent Patterns for Semantic Video Annotation," Multimedia, IEEE Transactions on , Vol. 10, No. 2, pp. 260-267, Feb. 2008.
- [9] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, and M. Ivkovic, "Augmented reality technologies, systems and applications," Multimedia Tools and Applications, Vol. 51 pp. 341-377, 2011.
- [10] S. You, and U. Neumann, "Mobile Augmented

Reality for Enhancing E-Learning and E-Business," Internet Technology and Applications, International Conference on, pp. 1-4, Aug. 2010.

[11] D.B. Goldman, C. Gonterman, B. Curless, D. Sale sin, and S.M. Seitz, "Video object annotation, navigation, and composition," In Proceedings of the 21st annual ACM symposium on User interface software and technology, ACM, pp. 3-12, 2008.

[12] W.C. Lee, Y.S. Moon, S.M. Lee, and H.Y. Noh, "Efficient Management of Moving Object Trajectories in the Stream Environment," KCC, pp. 166-170, 2006.

[13] Y.A. Ahn, J.S. Park, and G.H. Ryu, "Location Prediction of Mobile Objects using the Cubic Spline Interpolation," KCC, Vol. 31, No. 5, pp. 479-491, 2004.

[14] O. Wolfson, S. Chamberlain, Son Dao, Liqin Jiang, and G. Mendez, "Cost and imprecision in modeling the position of moving objects," Data Engineering, Proceedings, 14th International Conference on, pp. 588-596, Feb. 1998.

[15] O. Wolfson, S. Chamberlain, and L. Jiang, "Moving objects databases: issues and solutions," Scientific and Statistical Database Management, Proceedings. Tenth International Conference on, pp. 111-122, Jul. 1998.

[16] Y.J. Ji, H.J. Kim, and J.K. Heo, "Numerical Analysis with C Programming Language," NopiGipi, pp. 279-290, 2010.



이 기 성
 2005 : 인하대학교
 컴퓨터정보공학과 석사
 2005~2008 : 삼우통신(주) 연구원
 2005~현재 : 인하대학교
 컴퓨터정보공학과 박사과정
 관심분야 : 시맨틱웹, 시맨틱 이미지 검색, 증강 현실
 Email : lks@eslab.inha.ac.kr



정 진 국
 2003 : 인하대학교
 컴퓨터정보공학과 석사
 2009 : 인하대학교
 컴퓨터정보공학과 박사
 관심분야 : CSP, 지능형 전자상거래 시스템, 데이터마이닝, 패턴인식
 Email : gj4024@eslab.inha.ac.kr



조 근 식
 1982 : 인하대학교
 전자계산학과 학사
 1985 : 미국 뉴욕 CUNY
 전자계산 석사
 1991 : 미국 뉴욕 CUNY
 전자계산 박사
 1991 ~ 현재 : 인하대학교
 컴퓨터정보공학과 교수
 관심분야 : 인공지능, 시맨틱웹, 지능형 에이전트 시스템
 Email : gsjo@inha.ac.kr

저 자 소개



이 진 활
 2007~현재 : 인하대학교
 컴퓨터정보공학과 석사과정
 관심분야 : 시맨틱웹, 멀티미디어
 Email : hwari09@eslab.inha.ac.kr



홍 명 덕
 2011 : 인하대학교
 컴퓨터정보공학과 석사
 현재 : 인하대학교
 컴퓨터정보공학과 박사과정
 관심분야 : 인공지능, 온톨로지, VR/PTW
 Email : hmdgo@eslab.inha.ac.kr