

## 베이지안 네트워크를 이용한 자동차 시뮬레이션

김태현<sup>○</sup>, 손민우, 신동규, 신동일  
세종대학교

{crowlid82, minwoo15}@gce.sejong.ac.kr {shindk, dshin}@sejong.ac.kr

### Simulation of Automobile Model Using Bayesian Network

Taehyun Kim<sup>○</sup>, Minwoo Son, Dongkyoo Shin, Dongil Shin  
Sejong University

#### 요 약

본 논문은 물리엔진을 기반으로 구현한 자동차 시뮬레이터 프로그램에서 베이지안 네트워크를 이용해서 최적화된 이동방식을 계산하여 제공하는 기능을 구현한 결과를 보여준다. 자동차 시뮬레이터로부터 입력 받은 각 코스별 통과시간과 이동위치 및 회전각을 토대로 수집된 정보에 베이지안 네트워크를 적용하여 가장 빠른 시간 내에 완주한 코스의 이동위치에 따른 회전각을 산출해 낸 다음 각 위치마다 확률적으로 가장 적합한 핸들 조작법을 화면에 제공함으로써 사용자가 현 위치에 가장 최적화 된 조작법을 알 수 있게 한다. 또한 반복적인 레이스 트랙 완주에 따라서 더욱 최적화 된 각도를 피드백 함으로서 좀 더 빠른 완주가 가능해지도록 하는 것이 이 연구의 목적이다.

#### 1. 서 론

최근 들어 하드웨어 및 소프트웨어 기술의 발달로 인해 실감나는 게임을 많이 선보이고 있으며, 특히 과거와 달리 대부분의 게임은 흔히 게임엔진이라고 불리는 물리엔진과 렌더링 엔진이 결합 된 프로그램을 이용하여 제작되고 있다. 게임 엔진을 사용해서 많은 게임들이 실생활에서는 느낄 수 없는 것을 가상공간을 통해 구성하는 것이 가능해졌고, 그로 인해 게임의 매력을 느끼고 즐기는 유저들이 점점 증가하는 추세이다. 그러나 게임의 높은 난이도나 조작인터페이스의 사용은 게임에 익숙치 않은 사람들에게는 여전히 높은 진입장벽을 가지고 있다. 본 논문에서는 베이지안 네트워크를 이용하여 게임 안의 환경을 인지하고 수집한 수치를 이용, 학습하여 유저가 컨트롤하는 게임 내의 물체가 최적화 된 움직임을 보일 수 있도록 하는 데 목표를 두고 있다. 이를 위해 기본적으로 레이싱 게임을 구현하여 유저가 방향 컨트롤에 어려움을 겪을 수 있는 위치에서 제반 서비스를 기반으로 좀 더 쉽게 사용자가 박진감 넘치는 운전 조작을 할 수 있도록 도움을 주는 실험을 하였다.

#### 2. 관련 연구

##### 2.1 물리엔진

물리엔진이란 게임 내의 오브젝트를 대상으로, 질량 ·

속도 · 마찰 · 공기 저항 등의 수치를 이용하여 뉴턴 역학 모델을 시뮬레이트하는 프로그램을 통칭한다[1]. 물리 엔진이 내세우는 가장 핵심적인 기능은 강체(剛體 rigid body)의 동역학(dynamics)과 충돌 검출(collision detection)이다. 이외에 부수적으로 입자(particle)나 파동(wave), 옷감(cloth), 래그돌(ragdoll) 등의 시뮬레이션을 지원하기도 한다. 이제부터 이들 각각에 대하여 알아보자. 강체의 동역학은 물리 엔진의 근간을 이루는 가장 중요한 분야이다. 강체란 외부에서 어떠한 힘을 가해도 모양과 크기가 절대로 변하지 않는 물체를 이르는 말로, 현실에서는 존재하지 않는 이상적인 물체이지만 물리 엔진 상의 시뮬레이션에서는 연산의 편이를 위해 각종 탈것이나 인체 등의 오브젝트를 모두 강체로 가정한다. 동역학이란 힘과 운동 상태와의 관계를 다루는 역학으로, 이를 응용하여 게임 내에서 등장하는 물체에 힘이 가해지는 현상이나 충돌 등을 시뮬레이트하여 다시 게임 내의 물체들의 상태에 반영하게 된다. 강체의 동역학을 시뮬레이트하기 위해, 물리 엔진은 강체 · 조인트 · 충돌 · 마찰 등을 구현하여 게임 제작자들이 사용하기 쉬운 형태로 제공한다. 게임 제작자는 물리 엔진이 지원하는 이러한 기능을 이용하여 흔들리는 다리, 밧줄, 로봇 팔, 탈것들, 인체 등의 다양한 오브젝트가 가진 물리적인 특성을 게임 내에서 효과적으로 구현할 수 있는 것이다. 충돌 검출은 모든 게임이 필요로 하는 기본적인 기능이다. 물리 엔진이 논리적으로 공간을 구성하고 그 공간 안에서 일어나는 현상을 시뮬레이트하는 역할을 담당하기 때문에, 충돌 검출은 물리 엔진에서 도맡게 된

본 연구는 보건복지부 보건의료기술진흥사업의 지원에 의하여 이루어짐. (0412-M101-0416-0002)

다. 예를 들어 레이싱 게임에서 게이머의 차량이 경쟁자의 차량을 들이받아 차량이 파손되거나, 비행 시뮬레이션 게임에서 게이머의 기체가 지면에 추락하는 등의 상황에서는 강체의 기본적인 충돌 외에 다른 종류의 시뮬레이션이 필요하다. 이런 경우에 적절한 처리를 하기 위해 충돌 검출이 필요하게 된다[1].

### 2.2 렌더링 엔진

우선 PC 기반의 3차원 시뮬레이션을 제작하기 위해서는 보통 다이렉트X나 오픈GL과 같은 3D 그래픽라이브러리(graphic library)를 이용해 그래픽 가속 하드웨어의 기능을 활용하는 것이 일반적이다. 이러한 그래픽 라이브러리를 이용할 경우 다각형으로 구성된 시뮬레이션 환경을 주어진 광원 조건과 카메라 조건, 물체 표면 속성에 따라 화면에 출력할 수 있으므로 3차원 시뮬레이션 제작에 요구되는 기본적인 3차원 객체 출력 기능을 수행할 수 있어 이를 통한 시뮬레이션 화면구현이 가능하다. 하지만 시뮬레이션 제작과정에는 그래픽 라이브러리가 제공하는 단순한 기능 이외에 보다 많은 렌더링 기능이 요구된다. 3차원 렌더링 엔진은 그래픽 디자이너에 의해 작성된 각종 시뮬레이션 그래픽 데이터를 진행 상황에 따라 시뮬레이션 환경 내에서 적절히 변형, 이동시킨 후 빠르게 화면에 출력해주는 엔진을 말한다.

### 2.3 베이지안 네트워크

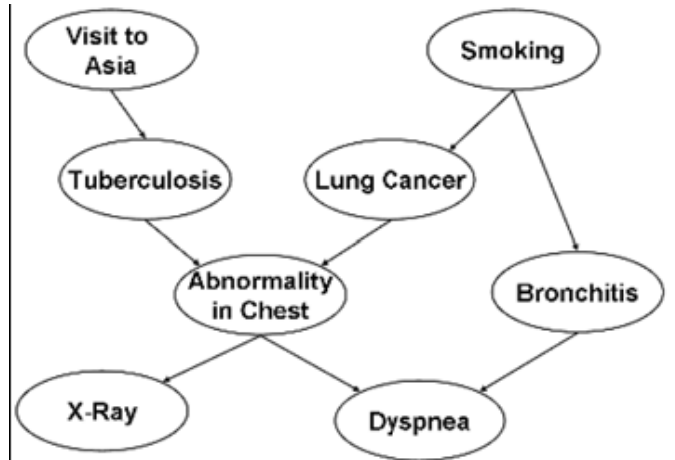
베이지안 네트워크는 베이즈 규칙을 기반으로 노드로 표현되는 각 변수의 의존 관계를 통해 확률 값을 추론하여 신뢰 값을 구하기 위한 DAG (Directed Acyclic Graph) 모델이다[2]. 노드들 간의 관계는 방향을 가진 에지를 통해 표현되며, 원인이 되는 부모노드와 결과가 되는 자식노드로 나뉘어진다. 각 노드는 여러 개의 속성을 가질 수 있고 이들의 합은 1이 된다. 부모를 가진 자식노드들은 의존 관계를 나타내는 확률 테이블을 가지고 있고 부모가 없는 노드들은 초기 확률 값을 갖는다. 베이지안 네트워크는 부분적인 증거만으로도 추론이 가능하기 때문에 불확실한 조건에서 부분적인 신뢰도를 얻는데 사용될 수 있다[3]. 부모가 한 개일 때 자식 노드의 확률 값은 다음의 식을 통해서 얻어진다.

$$P(A) = \sum_i P(A|Bi)P(Bi) \quad (1)$$

위 식에서 B는 A노드의 부모 노드이고 i는 부모 노드의 상태 수이다. A노드의 확률 값은 부모 노드가 가진 상태의 확률 값과 자식 노드가 가지고 있는 조건부 확률 값의 곱으로 표현된다. 일반적인 노드들 사이의 확률 분포  $P(x_1, x_2, \dots, x_n)$ 은 다음과 같이 나타난다.

$$P(x_1, x_2, \dots, x_n) = \prod_i P(x_i | Parents(X_i)) \quad (2)$$

그림 1은 베이지안 네트워크의 예를 보여준다.



(그림 1) 아시아 네트워크, Lauritzen, 1988

베이지안 네트워크에서 추론은 크게 예측 방향과 진단 방향으로 나뉘어져 확률 값의 전파로 이루어진다. (그림 1)에서 결핵(Tuberculosis)이나 폐암이 있을 때 흉부에 이상이 올 것을 추론하여 X-레이 결과나 호흡 곤란(Dyspnea) 증상을 예측하는 것이 전자에 속하고, X-레이 결과나 호흡 곤란 증상을 관측함으로써 결핵이나 폐암을 원인으로 추론하는 것이 후자에 속한다. 베이지안 네트워크에서 추론 속도는 얼마나 노드들이 연결되어 있는지, 루프의 수, 증거와 질의 노드 간의 위치 등과 같은 구조적인 요인에 영향을 받는데, 노드 간의 경로가 여러 개 존재할 경우엔 계산이 어려워지기 때문에 근사 값을 사용하는 알고리즘을 적용한다[3]. 단일 경로 구조를 갖는 베이지안 네트워크에서는 조건부 독립 조건과 베이즈 규칙을 적용하여 추론이 이루어진다. 두 노드 X, Y의 관계가  $X \rightarrow Y$ 와 같을 때 X가 증거노드이면 Y의 확률 값 추론은 확률 테이블을 통해서 이루어지고 Y가 증거노드인 경우엔 X의 확률 값 추론이 베이즈 규칙을 적용해서 이루어지는데 베이즈 규칙은 다음과 같다.

$$Bel(X) = P(X|Y) = \frac{P(X|Y)P(X)}{P(Y)} = \alpha P(x)\lambda(x) \quad (3)$$

이 때  $\alpha = \frac{1}{P(Y)}$ ,  $\lambda(x) = P(Y|X)$

위 식을 바탕으로 X의 모든 상태 값의 합이 1이 된다는 사실을 적용하여 값을 구하고 Bel(X)의 확률 값을 계산한다. 본 논문에서 제안하는 자동차 시뮬레이션을 처리하는 모듈은 이러한 과정을 반복적으로 거치는 알고리즘을 적용하여 레이싱 트랙에 최적화 된 회전각도를 제공하도록 한다.

### 3. 베이지안 네트워크를 적용한 자동차 시뮬레이션의 구현

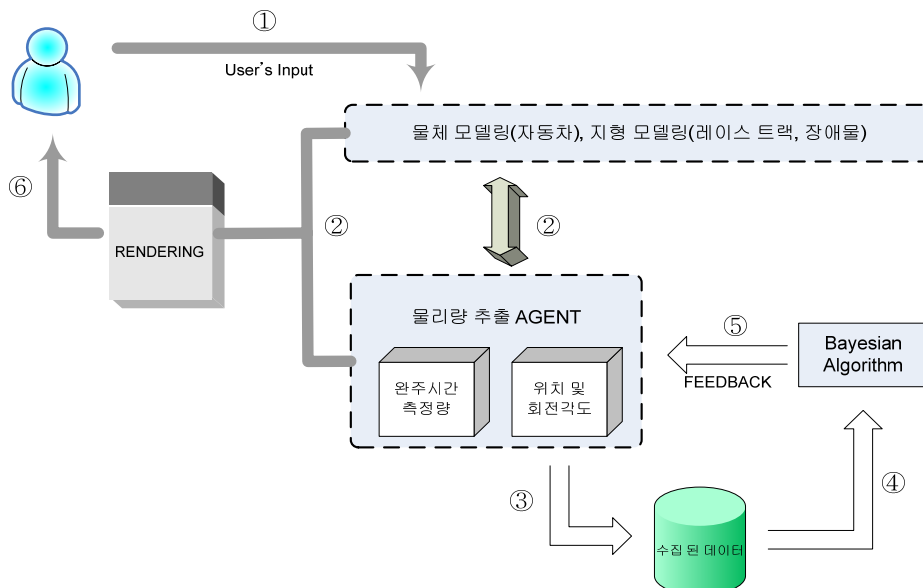
이 장에서는, 자동차의 물리엔진 구현정보와 베이지안 네트워크를 바탕으로 자동차의 트랙에서의 회전 시 각도의 수치 자료를 저장하여 급커브 시 가장 효율적인 회전각을 구한다. 그리고 자동차에 영향을 주는 환경적인 힘을 합하여 알고리즘을 적용했을 시에 계산과 실제적인 시뮬레이션 상황에서의 차이를 정정하도록 학습하는데 그 목표를 두고 있다. 아래에 제시된 모델은 ODE물리엔진의 구성에 따라 구현된 것이며, 자동차 시뮬레이팅을 위한 단순화 된 시뮬레이션 모델이다.

#### 3.1 Bayesian Network 사용한 개인화 서비스 제공을 위한 자동차 시뮬레이션 구조

본 논문에서는, 실제 물리엔진 연구를 위해 사용되는 ODE (Open Dynamic Engine)라는 오픈 소스 기반의 강체 역학 라이브러리를 사용한다. ODE는 차량, 가상 현실 환경의 객체나 가상 생물을 시뮬레이트 하는 것에 도움이 된다. 또한 여러 종류의 컴퓨터 게임, 3차원 편집 지원 툴과 시뮬레이션 도구에서 일반적으로 사용된다[4]. 실제 게임을 제작하는 데는 물리엔진뿐만 아니라 화면에 그려주기 위한 렌더링엔진이 필요하고, 물리엔진과 렌더링엔진과의 연동 부분이 필요하다. 즉, 게임엔진은 물체의 위치 및 속도 등의 정보를 물리엔진에 전해줘야 하며, 이를 바탕으로 물리엔진은 그 물체의 다음 위치를 계산하게 된다. 이렇게 구해진 물체의 새로운 위치는 렌더링엔진에 전달되며 렌더링엔진은 새로운 위치에 물체를 다시 그려주게 되는 것이다[2]. 또한 앞서 살펴본 바와 같이 물리엔진에서는 역학의 해석뿐만 아니라 물체들간의 충돌을 검출하고 그 반응을 구해내는 부분이 필요하다. 이러한 기능을 바탕으로 실험 환경을 구현하였으며, 이 실험에서는 자동차와 같은 물체 모델

링과 레이스 트랙 및 장애물 같은 지형 모델링을 구현하여 각각의 성질에 맞는 속성을 제공한다. 또한 게임 실행 내에서 모듈 별로 측정된 정보를 데이터베이스화하여 베이지안 네트워크를 적용시켜 가장 최적화된 회전각 등의 이동정보를 사용자에게 제공한다. 위에 언급한 시뮬레이션 구조를 적용하기 위해선 기본적으로 고려해야 하는 사항이 2가지가 있다. 첫째, 차량의 이동 위치 및 속도, 각도의 입력 값을 알아야 하는 것이고 둘째, 각 코스에 따른 완주시간을 측정해서 결과값을 분석해야 하는 것이다. 그림 2는 다음에 설명하는 실험 과정에 따라 자동차 시뮬레이션 내의 측정된 정보가 학습 과정을 거쳐 사용자에게 전달되는 것을 표현하고 있다.

- ① 먼저 응용 프로그램을 통하여 사용자가 조작한 핸들의 회전값을 받는다. 이 때 게임에 설정된 가상 환경에 따라 장애물 등의 물체와 지형의 물리량을 측정하여 현재 자동차의 위치 및 회전각 등의 다양한 값들이 생성된다.
- ② 생성된 값을 토대로 물체와 지형모델은 실시간으로 변화된 위치와 힘을 렌더링 엔진에 제공하는 한편, 물리량 추출 하는 모듈에서 요구하는 완주시간 및 위치와 회전각도를 전달한다.
- ③ 물리량 추출 모듈에선 수집된 정보를 데이터베이스에 넘겨준다.
- ④ 데이터베이스에 수집된 개인 사용자 별 완주시간과 각 위치에 따른 회전각도 정보를 베이지안 네트워크를 적용시켜 현재 사용자의 위치에서 가장 최적화된 회전각도를 구한다.
- ⑤ 현 위치에 최적화된 회전각도의 값을 물리량 추출 모듈에 전달한다.
- ⑥ 물리량 추출 모듈에 전달된 최적화된 값을 렌더링 엔진을 거쳐서 화면 상에 현재각도와 최적화된 각도를 동시에 표시하여 사용자에게 제공한다.



(그림 2) 자동차 시뮬레이터 실험 과정

### 3.2 자동차 시뮬레이션 시스템

위 그림에서 표현된 자동차 시뮬레이션에는 크게 4가지 요소가 필요하다. 자동차의 위치 및 회전각과 가상공간에 대한 계산함수를 담고 있는 물리엔진과 계산 결과를 화면에 표현하기 위한 렌더링 엔진과 산출된 정보를 수집하는 모듈 및 모듈의 수치를 베이지안 네트워크를 이용하여 분석해서 되돌려주는 데이터베이스 시스템 등이 필요하다. 그림 3은 물리량을 추출해 모듈에 저장하는 한편 화면에 표시하는 시뮬레이션 환경을 구축한 것을 보여주는 화면이다.



화면에서 보여진 바와 같이 모듈에서 받아들이는 가장 중요한 변수는 위치와 회전각 그리고 완주시간이다. 베이지안 인과 네트워크에선 각 정보간의 영향성이 매우 중요시 여겨진다. 자동차 운전에서도 현재 위치에 따른 회전각 또는 속도의 가장 등의 정보가 완주시간에 영향을 미친다.

### 4. 베이지안 네트워크를 적용한 실험결과 예측

모듈을 통해 데이터베이스에 축적된 정보를 바탕으로 베이지안 네트워크를 적용하여 각 트랙 별로 산출된 가장 최적화 된 각도를 제공함으로써 유저는 더 빠른 시간 내에 자동차 코스를 완주할 수 있게 될 것이다. 특히 베이지안 네트워크에 의한 학습결과와 이동코스의 제공으로 인해 가장 적합한 레이싱 컨트롤을 학습하게 되고 레이싱 트랙 별로 저장된 가장 빠른 완주코스를 자동주행 하는 기능을 추가해서 본 게임시뮬레이션에 들어가기 전에 유저가 학습모드를 체험할 수 있도록 할 것이다.

### 5. 결론

본 논문에서는 유저가 게임 환경에 쉽게 익숙해질 수 있도록 하기 위해 게임 내에서 최적화 된 학습정보를 제공하는 방법을 제안하였다. 또한 물리엔진이 무엇이고 기본적인 물리 반응을 수행하기 위한 이론을 살펴보았다. 그리고 이를 화면에 표시하기 위한 렌더링 엔진의 개요를 살펴 보았고 다음으로 베이지안 네트워크가 어떤 알고리즘을 기반으로 정보를 분석하여 최적화 된 환경을 제공할 수 있는지 살펴보았다. 이를 위해 게임엔진을 이용해 가상공간과 자동차 및 지형정보를 구성하고 각 사용자의 위치 별 레이싱트랙 완주정보를 데이터베이스에 입력하는 모듈을 설계하였다. 물론 이러한 데이터베이스 정보와 충돌 검출과 충돌 반응 등 기본적인 물리 반응의 수행을 범용 물리엔진을 사용하여 게임 시뮬레이션 환경을 충족시킨다 하더라도 실제 레이싱 게임을 위해서는 경주 트랙이나 차량의 특성, 비나 눈이 오거나 바람이 부는 등의 환경적인 요인 등 고려해야 할 사항은 훨씬 많다. 이러한 환경들을 포함한 게임의 여러 가지 변수 요소들을 학습하고 반영하는 콘텐츠를 제공하여 게임을 즐기는 사용자로 하여금 쉽게 지루함을 느끼지 못하며 오랫동안 재미를 느낄 수 있게 하는 것이 이 연구의 향후 목표이다.

### 참고문헌

- [1] D. H. Eberly, 유채곤 / 차미리 역, "Game physics (게임 물리 바이블)," 사이텍미디어, Nov. 2004, pp. 223-444.
- [2] Wikipedia, "Bayesian Network", available in [http://en.wikipedia.org/wiki/Belief\\_networks](http://en.wikipedia.org/wiki/Belief_networks)
- [3] 김현국, "A Design and Implementation of Game Physics Engine for Mobile 3D Racing Game, 서울시립대학교 대학원, 2006. 2
- [4] Russell Smith. "Open Dynamic Engine Manual 0.5", available in <http://ode.org>
- [5] I.H Witten & Elbe Frank, "DATA MINING", MORGAN KAUFMANN PUBLISHERS, 2005
- [6] C. Hecker, "Rigid Body Dynamics Part 1-4," available in <http://www.d6.com/users/checker/index.htm>.
- [7] Steve Rabin, 류광 역, "AI GAME PROGRAMMING WISDOM 2", 정보문화사, 2005. 3. 10