컴퓨터 게임을 위한 물리 엔진의 성능 향상 및 이를 적용한 지능적인 게임 캐릭터에 관한 연구

최 종 화·신 동 규·신 동 일

요 약

이 논문은 컴퓨터 게임을 위한 물리 엔진의 성능 향상 및 이를 적용한 지능적인 게임 캐릭터에 관한 연구를 시도한다. 물리적 상황을 자동으로 인식하는 알고리즘으로는 Momentum back-propagation를 적용하였다. 또한 우리는 각 상황에 따른 적당한 방식의 실험 결과를 제시한다. 실험을 위하여 Euler Method, Improved Euler Method, 및 Runge-kutta Method의 세 가지의 적당 방식을 적용하였다. 각 방식의 실험 결과에서 중점을 두지 않는 상황에서는 Euler Method가 최적의 성능을 보였지만, 중복 상황에서는 세 가지 방식이 모두 비슷한 성능을 보여주었다. 또한 중복 상황에서는 중복 방식 모두 비슷한 성능을 보여주었고, Runge-kutta Method가 최적의 성능을 보여주었다. 물리 상황인 이해한 실험결과에서는 입력 출력 높이 고정된 상황에서 중복 총이 3일 때 가장 좋은 성능을 보여주었고, 입력 높이 30000일 때 최적의 성능을 보여주었다. 앞으로 우리는 다양한 장르의 게임에 이러한 물리적 컨텍스트(context)를 인식하는 연구를 진행할 것이며 이를 통해 게임의 성공을 증가할 수 있도록 M-BP의 이의 안식 알고리즘을 적용할 것이다.

키워드 : 물리엔진, 인공지능, 인공지능 게임 캐릭터

Research on Intelligent Game Character through Performance Enhancements of Physics Engine in Computer Games

Jonghwa Choi* · Dongkyoo Shin** · Dongil Shin***

ABSTRACT

This paper describes research on intelligent game character through performance enhancements of physics engine in computer games. The algorithm that recognizes the physics situation uses momentum back-propagation neural networks. Also, we present an experiment and its results, integration methods that display optimum performance based on the physics situation. In this experiment, we divided the methods, the Euler method was shown to produce the best results in terms of fps in a simulation environment with collision detection. Simulation with collision detection was shown similar fps for all three methods and the Runge-kutta method was shown the greatest accuracy. In the experiment on physics situation recognition, a physics situation recognition algorithm where the number of input layers (number of physical parameters) and output layers (destruction value for the master car) is fixed has shown the best performance when the number of hidden layers is 3 and the learning count number is 30,000. Since we tested with rigid bodies only, we are currently studying efficient physics situation recognition for soft body objects.

Key Words : Physics Engine, Artificial Intelligence, Intelligence Game Character

1. 서 론

현재 3차원 게임에 있어서 가장 중요한 이슈는 게임 속 세계에 존재하는 모델 캐릭터들이 사실감과 생동감을 유지하면서 프로그램의 속도 감소 문제를 발생시키지 않는 방안을 찾는 것이다. 게임 속 캐릭터의 사실감이란 캐릭터의 움직임이 현실의 물리현상과 같은 현상을 나타내는 것을 말하되며 게임 물리엔진의 적용이 그 해결책으로 제시되고 있다[1]. 그러나 물리엔진이 적용된 게임은 캐릭터의 사실적인 움직임은 표현할 뿐 캐릭터에게 생명력을 심어주지는 못한다. 이 논문에서는 물리엔진이 적용된 게임에서 주인공 캐릭터가 행동할 때 발생하는 물리적 요소를 지능적 캐릭터가 인식하여 자신의 행동을 스스로 결정하는 상태인 캐릭터 생성을 위한 연구를 제시하고자 한다. 이러한 연구는 게임 인공지능의 한 분야라고 할 수 있다. 하지만 지
금까지의 게임 인공지능의 대부분 연구는 캐릭터가 게임 속에서 어떻게 행동해야 하는지에 대한 자신의 규칙을 가지고 행동하는 양식으로 진행되었다[2].

이 논문에서 각각의 캐릭터가 주인공 캐릭터의 물리적 요소를 인식하기 위한 알고리즘으로 momentum back-propagation NNNs[3]이 사용되었다. 하지만 게임 속에서의 물리현상에 대한 페턴 인식 알고리즘의 적용은 게임의 성능에 저항을 가졌으며, 이에 대한 해결방안으로 물리엔진에서의 상황에 따른 측정하는 적응기를 사용하여 물리 엔진의 성능을 향상하는 방식으로 게임에서의 반응속도를 최적화한다[4].

2장에서는 게임에서의 물리엔진에 관한 연구와 게임 인공지능에 관한 내용에 따라 지금까지 진행된 연구를 살펴보고, 3장에서는 이 논문에서 제시하는 컴포넌트 아키텍처를 제시한다. 4장에서는 컴퓨터 아키텍처에서 제시할 각각의 컴퓨터에 대한 세부설명을 한다. 5장에서는 상황에 따라 변화하는 물리요소를 입력 받아서 현재의 물리현상을 결정하는 알고리즘의 성능에 대한 평가를 제시한다. 6장에서는 이 논문의 결론을 요약한다.

2. 관련 연구


3. 게임에서의 자동적인 물리적 상황 인식을 위한 컴포넌트 아키텍처

3.1 프로그램 시나리오

이 논문에서는 자동적인 물리적 상황 인식 를 자동차를 게임에 대해서 아키텍처를 제시하고 그에 대한 설명을 한다. 자동차 게임에서는 하나의 주인공 자동차와 주인공 자동차를 공격하는 많은 수의 적 자동차들이 존재한다. 주인공 자동차는 움직일 때마다 변화하는 물리적 요소가 발생한다. 적 자동차는 주인공 자동차의 변화된 물리적 요소를 인식하여 주인공 자동차에게 많은 손을 줄 수 있는 순간을 파악하고 주인공 자동차를 공격한다. 현재까지 게임용으로 적 자동차들이 악수된 행동 및 주인공 자동차의 위치에 따라 행동을 할 경우 주인공 자동차의 변화하는 물리적 요소를 파악하여 지능적으로 행동하는 적자동차에 관련 연구는 상당히 미흡한 실정이다.

3.2 컴포넌트 아키텍처

가능을 포함하면 게임전체의 성공 저하를 초래하는데 Numerical Integration Component는 이에 대한 성능감소 부분을 보완하는 역할을 담당한다.

Game Character Manager에서는 적 자동차 리스트 중에 주인공 자동차와 가장 근거리에 있는 적 자동차를 선택하여 적 자동차의 현재의 물리적 요소를 Physics Context-aware Component에게 전송하는 역할을 담당하고 그 결과값(주인공 자동차의 피해수치)을 적 자동차에게 다시 전달해 주는 역할을 담당한다.

4. 내부 컴퓨터 구조

4.1 Physics Context-aware Component

4.1.1 물리 컨텍스트 정의

Physics Context-aware Component는 적 자동차가 주인공 자동차의 움직임을 파악하여 주인공 자동차에게 최대의 피해를 줄 수 있는 상황을 판단하여야 하는 역할을 담당한다. 이 논문에서는 상황 판단을 위하여 M-BP 알고리즘을 적용하여 상황을 학습하고 판단한다. <표 1>은 상황판단을 위한 주인공과 적 자동차에서 추출되는 물리적 컨텍스트를 정의한다. 타이틀1에 정의된 물리적 컨텍스트는 물리상황식 알고리즘의 입력 값으로 적용한다.

자동차 사망편도에 자동차가 음직일 때마다 변화하는 물리적 요소는 Position, Linear Velocity, Variable Velocity가 있고 자동차간의 충돌시 충돌반응에 영향을 미치는 요소는 Mass, Center of Mass가 있다. <표 1>에서 제시된 5가지의 컨텍스트는 자동차의 충돌에 있어서 영향을 미치는 요소이며, 물리엔진에서 추출 가능한 요소이기도 하다. 우리는 <표 1>에서 제시된 5가지의 충돌에 관련하는 요소를 기반으로 두 차의 충돌이 발생하는 충돌 값을 기반으로 최대의 충돌이 일어날 수 있는 상황을 예측하는 실험을 제시한다.

그리고, 모든 컨텍스트는 0.1-0.9사이의 값으로 정규화(normalization value)하였다. 컨텍스트의 정규화하는 이유는 모든 컨텍스트를 우리가 실험에서 사용하는 M-BP의 입력값(input value)으로 적용하여 출력값(output value)의 정확도 높이기 위한 것이다. Position은 Game World로 9sector 설정하여 9등분 하였고 Linear Velocity는 그 범위를 0-0으로, Center of Mass는 개개의 영역 각각 9등분 하였다. Mass는 현재 값을 45까지 기준으로 하였고 Variable Velocity는 45를 기준으로 9등분하여 실험 데이터를 설정하였다.

4.1.2 Physics Context-aware Component의 구조도


Physics Context-aware Component는 학습시에는 주인공 자동차와 적 자동차에서 물리 컨텍스트 및 그 성향에서의 주인공 자동차의 피해수치를 입력 값으로 받고 예측 시에는 주인공 자동차와 적 자동차의 물리 컨텍스트를 입력값으로 받는다. 학습시에는 Learning Process에서 해당 학습을 한 후에 M-BP의 가중치 값을 변경시킨다. 예측은 학

| Table 1 | 주인공차와 적자동차의 물리적 컨텍스트 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | P | L | C | M | V | P | L | C | M | V |
| 0.1 | 1 | 0-10 | 0.0-1.0 | 1 | 1-5 | 1 | 0-10 | 0.0-1.0 | 1 | 1-5 |
| 0.2 | 2 | 11-20 | 1.1-2.0 | 2 | 6-10 | 2 | 11-20 | 1.1-2.0 | 2 | 6-10 |
| 0.3 | 3 | 21-30 | 2.1-3.0 | 3 | 11-15 | 3 | 21-30 | 2.1-3.0 | 3 | 11-15 |
| 0.4 | 4 | 31-40 | 3.1-4.0 | 4 | 16-20 | 4 | 31-40 | 3.1-4.0 | 4 | 16-20 |
| 0.5 | 5 | 41-50 | 4.1-5.0 | 5 | 21-25 | 5 | 41-50 | 4.1-5.0 | 5 | 21-25 |
| 0.6 | 6 | 51-60 | 5.1-6.0 | 6 | 26-30 | 6 | 51-60 | 5.1-6.0 | 6 | 26-30 |
| 0.7 | 7 | 61-70 | 6.1-7.0 | 7 | 31-35 | 7 | 61-70 | 6.1-7.0 | 7 | 31-35 |
| 0.8 | 8 | 71-80 | 7.1-8.0 | 8 | 36-40 | 8 | 71-80 | 7.1-8.0 | 8 | 36-40 |
| 0.9 | 9 | 81-90 | 8.1-9.0 | 9 | 41-45 | 9 | 81-90 | 8.1-9.0 | 9 | 41-45 |
행해야 하기 때문에 게임의 성공에 있어서 많은 장애를 가진다. 이러한 연산 장애를 극복하고자 우리와의 논문에서는 Numerical Integration Component를 제시하였다. 이 논문에서는 우리는 지능적인 캐릭터의 적응 대상으로 자동차 시뮬레이션을 구현하였다. 물리학에서는 개체의 이동에 따른 개체의 방향 및 위치계산에 Euler Method, Improved Euler Method, Runge-Kutta Method를 적용한다. 하지만 각각의 적응 메커니즘은 물리적 상황에 따라 가장 좋은 성능을 나타내는 적응메커니즘을 적용함으로써 물리원리를 이용한 인공지능의 적응에 따른 성능 장애를 보상해 줄 수 있도록 한다. 세 가지 적응 메커니즘의 정의는 다음과 같다.

Euler Method는 초기 조건이 정정직(1)에서 주어졌을 때 태일러 급수에 의해 일차 미분 계수까지 정리된다. Euler Method는 개체의 위치 신장직 정확도는 Runge-Kutta Method보다 떨어지지만 계산 속도에 있어서는 세가지 메커니즘 중에서 가장 좋은 효과를 보인다.

\[
\frac{dy}{dx} = f(x, y), y(x_0) = y_0
\]
\[
\frac{dy}{dx} \approx \frac{y(x_0 + h) - y(x_0)}{h}
\]
\[
\frac{dy}{dx} \approx \frac{y(x_0 + h) - y(x_0)}{h}
\]
\[
\frac{dy}{dx} \approx \frac{y(x_0 + h) - y(x_0)}{h}
\]
\[
\frac{dy}{dx} \approx \frac{y(x_0 + h) - y(x_0)}{h}
\]

Improved Euler method는 아래 방정식 (2)에서 보이는 바와 같이 이차 미분 계수까지 정리된다.

\[
y_{n+1} = y_n + (1 - h)k_1 + kh_2, h = \frac{1}{2}v_1
\]

\[
k_1 = f(x_n, y_n)
\]

\[
k_2 = f(x_n + h, y_n + \frac{k_1}{26})
\]

Runge-Kutta method는 태일러 급수에 의해 4차 미분계수까지 정리되며 Euler Method와 비교하여 정확도에서 좋은 성능을 보인다.

\[
y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\]

\[
k_1 = f(x_n, y_n)
\]

\[
k_2 = f(x_n + \frac{h}{2}, y_n + \frac{k_1}{2})
\]

\[
k_3 = f(x_n + h, y_n + \frac{k_2}{2})
\]

\[
k_4 = f(x_n + h, y_n + k_3)
\]

우리의 실험은 20개의 이하의 캐릭터에 대한 적응 결과이며 실제 게임에서는 수많은 캐릭터가 존재하기 때문에 적응의 차이가 실제 게임에서는 암시한 성능 차이를 발생시킨다.
(그림 5) 적분 방식에 따른 성능 결과

(그림 5)의 왼쪽은 중돌이 존재하는 환경에서의 개체(가공)에 따른 frame per second(y좌표)에 관한 실험이다. 개체의 개수가 늘어나 낮에 따라서 Euler Method의 성능이 현저하게 좋음을 알 수 있다. (그림 5)의 오른쪽은 중돌이 존재하는 환경에서의 개체(가공)에 따른 frame per second(y좌표)에 관한 실험이다. 중돌이 존재하는 환경에서 는 모든 에이드가 FPS의 성능 비교에 있어서 비슷한 성능을 나타내며 평균도가 높아진 Runge-Kutta Method가 적절의 성능을 보인다고 말씀할 수 있다. Numerical Integration Component는 자동차(주행고 및 각도)의 방향 및 위치 계산에 있어서 중돌이 존재하지 않는 환경에서는 Euler Method를 적용하고 중돌이 존재하는 환경에서는 Runge-Kutta를 적용해주는 역할을 담당한다.

5. 실험 및 평가

이 논문에서 제시하는 실험은 물리엔진이 적용된 게임 속에서 주인공 및 각 자동차가 움직일 때 발생하는 상황에 따른 최대의 충격을 M-BP를 이용하여 정확하게 예측할 수 있는지에 대한 실험이다. 실험은 두 가지 방식으로 정확도를 실험하였다. 첫 번째는 온온 충돌의 변경으로 인한 알고리즘 정확도 실험이고, 두 번째가 각주조수의 동작에 의한 알고리즘 정확도 실험이다. 주인공 자동차의 열차 자동차로부터 5가지의 온블 내Pages 항목을 일괄 같은 장단방식에 입혀 총의 수는 고정되어있으며 실험이되었다. 또한 블록체의 개수도 주인공 자동차의 표적과 100을 기준으로 10등분하여 총 10층의 출력 총이 고정되어 실험이되었다. <표 2>는 실제 두 자동차 간에 발생하는 총충격량을 0.1-0.9로 정규화 한 값이고, 이 값은 M-BP의 학습 시에 5가지의 온블 내Pages 항목의 결과 값으로 얻어졌다.

<표 3>은 온온 충돌의 변경에 따른 예측정확도를 표로 나타낸 것이다. 예측정확도는 M-BP 알고리즘에 의해 예측된 정확도와 실제 물리엔진에 의해서 계산된 두 개체간의 충격량의 정확도가 일치하는 것인가에 대한 테스트이다.

<표 4>는 학습횟수에 따른 정확도 실험이다.

<table>
<thead>
<tr>
<th>Power value of Master Car</th>
<th>0-10</th>
<th>11-20</th>
<th>21-30</th>
<th>31-40</th>
<th>41-50</th>
<th>51-60</th>
<th>61-70</th>
<th>71-80</th>
<th>81-90</th>
<th>91-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Norm Value</td>
<td>0.0-0.1</td>
<td>0.11-0.2</td>
<td>0.21-0.3</td>
<td>0.31-0.4</td>
<td>0.41-0.5</td>
<td>0.51-0.6</td>
<td>0.61-0.7</td>
<td>0.71-0.8</td>
<td>0.81-0.9</td>
<td>0.91-0.99</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hidden Layer</th>
<th>Success Rate(%)</th>
<th>Cross Validation error signal value by hidden layer</th>
<th>Cross validation error signal value by output layer</th>
<th>Test error signal value by output layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>63</td>
<td>31.2332</td>
<td>231.3453</td>
<td>234.1282</td>
</tr>
<tr>
<td>3</td>
<td>92</td>
<td>13.2343</td>
<td>164.2313</td>
<td>166.0923</td>
</tr>
<tr>
<td>5</td>
<td>84</td>
<td>24.1213</td>
<td>128.6409</td>
<td>130.2930</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Learning Count</th>
<th>Success Rate(%)</th>
<th>Cross Validation error signal value by hidden layer</th>
<th>Cross validation error signal value by output layer</th>
<th>Test error signal value by output layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000</td>
<td>71</td>
<td>23.2550</td>
<td>224.2292</td>
<td>225.5520</td>
</tr>
<tr>
<td>20000</td>
<td>81</td>
<td>25.2989</td>
<td>130.9064</td>
<td>131.0934</td>
</tr>
<tr>
<td>30000</td>
<td>92</td>
<td>13.2343</td>
<td>164.2313</td>
<td>166.0623</td>
</tr>
<tr>
<td>40000</td>
<td>85</td>
<td>23.2550</td>
<td>127.9424</td>
<td>128.0259</td>
</tr>
<tr>
<td>50000</td>
<td>85</td>
<td>23.2550</td>
<td>127.8004</td>
<td>128.0123</td>
</tr>
</tbody>
</table>
도 실험 결과를 나타냈다. 테이블 4에서는 학습횟수 30000
일 경우에 가장 좋은 성능을 보여주었다.

실험에서 나타난 바와 같이, Physics Context-aware Component는 입력 측이 5로 고정되어있고 출력 측이 10으
로 정정된 상태에서 실행되었고 은닉 측이 3계층이고 학습
횟수가 30000일로 일 때 가장 좋은 성능(정확도 92%)을 보
여주었다.

6. 결 론

이 논문에서 제안되고 실험된 컴퓨터 게임을 위한 물리
엔진의 성능 항상 및 이를 적용한 자공적인 게임 캐릭터에
관한 연구는 물리엔진의 적용으로 인해 게임의 사실성을 높
여주고, 물리엔진에 기반한 게임 속 캐릭터들의 물리 수치를
인식함에 따라 게임 속 캐릭터의 생명력을 느끼게 해 주는
역할을 한다. 우리는 본 연구를 위한 전자 경기장 시각"를 제안
하였고 그에 대한 실험을 하였다. 제안된 기법은 Physics
Context-aware Component, Numerical Integration Component,
그리고 Game Character Manager로 구성되었다. Physics
Context-aware Component에서는 물리 컨텍스트를 기반으로 그에
대한 학습과 예측을 수행한다. Numerical Integration Component는 물리엔진 기반에서의 자공적 캐릭
터 적용이 게임에서의 많은 수도 감소를 유발하는데, 이를 보
상하기 위한 최적의 성능을 발휘하는 적절 메시드 선택 함
으로써 게임의 성능저해를 극복하는 역할을 한다.

현재 많은 게임에서 물리엔진이 사용되고 있으므로, 학습
에는 이러한 물리엔진이 적용된 게임에서의 물리적 상황인
식이 게임에서의 중요한 요소로 자리잡을 것이다. 이 논문에
서의 연구는 자공적 게임에 한정되어 있다. 앞으로 우리는 다
른 장르의 게임에서 이러한 물리 컨텍스트를 인식하는 연구
을 진행할 것이며 또한 전체 게임의 성능을 증가시킬 수 있도록
M-BP이외의 인식 알고리즘을 적용할 것이다.

참 고 문 현

solving real world physics problems," Proceedings of the Sixth
Conference on Artificial Intelligence for Applications,

alternative current permanent magnet servomotor using
neural network", Proceedings of the Fifth International
Conference on Electrical Machines and Systems,Vol.2,

[3] Chen Zhifei, An Yuejun, Jia Kehng, Sun Changzhi, "Intelligent
control of alternative current permanent magnet servomotor
ICEMS 2001. Proceedings of the Fifth International Conferen-

manifolds," Journal of Applied, Number Math, pp.115-127,
1999.


[9] Lawlor, O.S., Kalee, L.V., "A Voxel-based Parallel Collision
Detection Algorithm," Proceedings of the 6th international

Proceedings of the ACM symposium on Virtual reality
software and technology, pp.121-128, November, 2002.

and Cybernetics," Proceedings of International Conference

인 동 규

e-mail : shindk@sejong.ac.kr
1986년 서울대학교 계산통계학과(학사)
1992년 M.S. in Computer Science,
Illinois Institute of Technology
1997년 Ph.D in Computer Science,
Texas A&M University
1986년 2~1991년 8월 한국국방연구원 연구원
1997년 8월~1998년 2월 현대전자 멀티미디어연구소 차장
( 책임연구원)
1998년 3월~현재 세종대학교 컴퓨터공학과 부교수
관심분야: XML보안, 전자상거래, MPEG

신 동 일

e-mail : dshin@sejong.ac.kr
1988년 인제대학교 전산학과(학사)
1993년 M.S. in Computer Science,
Washington State University
1997년 Ph.D in Computer Science,
University of North Texas
1997년 9월~현재 컴퓨터공학부 산업연구원
1998년 3월~현재 세종대학교 컴퓨터공학과 조교수
관심분야: 무선인터넷, HCL, 게임엔진, CSGW