

플레이어 적응형 GMM 기반 동적 게임 레벨 디자인

이상경⁰, 정기철

숭실대학교, 미디어학과, HCI Lab.

monask@ssu.ac.kr⁰, kcjung@ssu.ac.kr

Player Adaptive GMM-based Dynamic Game Level Design

Sangkyung Lee⁰, Keechul Jung

HCI Lab., School of Media, Soongsil University

요약

게임에서 레벨 디자인 (Level Design)과 캐릭터간의 밸런스는 게임의 흥미를 결정하는 매우 중요한 요소이며, 레벨 디자이너에 의해 결정 된다. 기존의 게임에서는 플레이어가 가장 큰 재미를 느낄 수 있는 캐릭터의 공격 패턴과 속성은 정적으로 정해졌으며 스크립트 형식으로 표현됐다. 이와 같이 정적으로 정해진 레벨에 따라 진행되면 플레이어가 쉽게 적응하게 되고, 플레이어의 학습능력에 따라 레벨 디자이너가 의도했던 밸런스가 깨질 수 있었다. 이런 문제점을 해결하기 위해 본 논문에서는 게임 도중에 플레이어의 대응 패턴을 GMM (Gaussian Mixture Model)으로 모델링하고 분석하여 레벨 디자이너가 의도했던 레벨과 재미를 느낄 수 있는 환경을 제공하는 방법을 제안한다. 제안한 방법을 실제 2D 슈팅게임에 적용하여 플레이어의 패턴을 분석한 결과와 동적 레벨 디자인의 결과를 보인다.

Abstract

In computer games, the level design and balance of characters are the key features for developing interesting games. Level designers make decision to change the parameters and opponent behaviors in order to avoid the player getting extremely frustrated with the improper level. Generally, opponent behavior is defined by static script, this causes the games to have static difficulty level and static environment. Therefore, it is difficult to keep track of the user playing interest, because a player can easily adapt to changeless repetition. In this paper, we propose a dynamic scripting method that able to maintain the level designers' intention where user enjoys the game by adjusting the opponent behavior while playing the game. The player's countermeasure pattern for dynamic level design is modeled using a Gaussian Mixture Model (GMM). The proposed method is applied to a shooting game, and the experimental results maintain the degree of interest intended by the level designer.

Keyword : Level Design, Dynamic Scripting, GMM, EM

1. 서론

컴퓨터 게임을 하는 인구가 증가하고 연령층이 다양해지면서 플레이어의 게임 실력과 욕구도 다양해졌다. 이렇게 다양한 플레이어들의 수준에 맞고 욕구를 만족시키는 게임

개발을 위해서 레벨 디자인 (Level Design)은 중요한 요소로 인식되고 있다. 게임 레벨 디자인은 플레이어의 지루함을 없애기 위해서 적 공격 패턴이나 게임 캐릭터 속성값을 결정하는 것이다. 기존의 고정된 난이도나 미리 정해진 캐릭터 속성값들이 변하지 않는 게임들은 레벨 디자이너가

의도한 흥미 있는 게임을 유지할 수 없다 [1-2].

컴퓨터 게임 개발자들은 플레이어에게 재미를 줄 수 있는 인공지능 기법을 연구해왔다. 게임 캐릭터가 사실적이고 생명체와 비슷한 행동을 하도록 사용된 초기 인공지능 기법은 유한 상태 기계 (Finite State Machine), 퍼지 상태 기계 (Fuzzy State Machine), 의사결정 트리 (Decision Tree), 규칙 기반 시스템 (Rule-base System) 등과 같다. 유한 상태 기계는 상태들과 전이들의 유한한 집합으로 정의하여, 주어진 시점에서 오직 하나의 상태만 활성화한 후 현재 활성화된 행동을 자신의 과제로 삼아서 수행하는 방식이다. 퍼지 상태 기계는 유한 상태 기계에 퍼지 집합의 개념을 접목시킨 것으로 여러 개의 상태들이 현재 상태가 되며 각각의 상태들이 정도가 서로 다른 값을 가지게 해서 현재 상태를 결정짓는 방식이다 [3-4]. 이런 기법들은 단조로운 패턴을 제공하기 때문에 플레이어는 재미있는 게임을 진행할 수 없었다. 최근에는 초기 인공지능 기법의 문제점인 단조로운 패턴을 해결하기 위해서 기계학습 (Machine Learning) 기법인 신경망 (Neural Network), 유전자 알고리즘 (Genetic Algorithms), 강화학습 (Reinforcement Learning) 등을 적용하였다 [5-7]. 이 기법들은 초기에 캐릭터 목적을 학습시키거나, 게임 수행 시 캐릭터를 진화시켜 지능적인 역할을 수행하거나, 플레이어와 상호정보교환을 통하여 게임내의 캐릭터의 행동을 개선해 나가는 학습방법이다. 하지만 이러한 기법들은 레벨 디자이너가 요구하는 게임의 난이도를 조절하기 보다는 게임 내의 사실성을 위해서 사용되었다.

게임 내에서 캐릭터들의 행동은 스크립트로 작성되어 해당 이벤트 발생 시 작성된 스크립트대로 캐릭터는 행동하게 된다. 이렇게 작성된 스크립트가 게임 실행 중 변하지 않고 초기에 작성된 내용대로 실행하는 것을 정적 스크립트 (Static Script) 방식이라 한다.

기존의 인공지능 기법과 정적 스크립트는 플레이어가 게임을 할수록 패턴에 익숙해지며 게임 약점을 파악하여 쉽게 게임의 미션을 수행할 수 있게 된다. 즉, 게임의 흥미를 떨어뜨리는 문제를 가진다. 이런 문제를 해결하기 위해서 동적 스크립트 (Dynamic Script) 기법이 연구되고 있다. 동적 스크립트는 고정된 난이도를 가지는 정적 스크립트를 사용하지 않고, 동적으로 상황에 맞게 스크립트를 생성하여 게임의 레벨과 캐릭터 AI를 조절하는 장점을 가진다. RPG 게임에서 플레이어와 컴퓨터 AI가 전투할 때 학습을 통해

가중치를 계산해서 효율적으로 전투할 수 있는 스크립트를 선택하는 연구가 있다 [8-9]. 이와 같은 연구에서는 전투 횟수가 제한되어 있으므로 학습시간 단축이 중요한 문제이다.

본 논문에서는 플레이어의 수행 패턴을 실시간으로 분석하여 레벨 디자이너의 의도한 방향으로 게임이 수행할 수 있게 동적으로 스크립트를 생성 및 적용하는 방법을 제안한다. 제안된 동적 스크립트 기법은 레벨 디자이너의 의도대로 게임의 재미를 느낄 수 있는 방법으로 GMM (Gaussian Mixture Model) [10-12]을 사용하며, 실시간으로 플레이어의 게임수행 패턴을 분석하고 레벨 디자이너의 의도대로 플레이어가 재미를 느끼며 미션을 수행하게 함으로써 플레이어의 긴장도를 유지시킨다. 제안된 기법을 실제 2D 게임에 적용해서 플레이어의 이동 패턴을 분석하여 적절한 적의 공격 패턴을 결정 하며, 무기 및 스킬 사용을 분석하여 난이도를 조절할 수 있음을 보인다.

2. GMM을 이용한 플레이어 반응 패턴 모델링

레벨 디자이너는 가장 재미있게 게임을 할 수 있는 적 캐릭터 공격 패턴과 무기 속성 값 등을 고려하여 레벨을 설계한다. 레벨 디자이너의 의도가 게임 플레이 도중에 유지 되도록 적의 공격 패턴에 대한 플레이어의 반응 및 무기 사용 빈도를 분석하여 게임의 난이도를 실시간으로 조정함으로써 플레이어에게 재미있는 게임을 제공하고자 한다. 우선 2.1절에서는 플레이어의 대응 패턴 데이터가 무엇인지 살펴보고, 2.2절에서는 추출된 데이터를 이용하여 어떻게 GMM을 사용하여 모델링 하는가에 대한 설명이다. 마지막으로 2.3절에서는 모델링된 GMM을 이용하여 게임에 적용하는 방법에 대해서 설명한다.

2.1 플레이어의 대응 패턴 데이터

동적으로 게임 레벨 및 환경을 설정하기 위해서는 게임 시스템이 동작 중에 판단할 수 있는 기준이 필요하며, 이러한 기준은 레벨 디자이너가 의도한 바를 수치화 시켜서 표현된다. 플레이어 체감 난이도를 조절할 때 수치화된 기준과 비교되는 데이터가 플레이어 대응 패턴 데이터이다. 즉 게임 플레이 도중에 적들의 움직임에 대한 플레이어의 대

용 동작, 무기 사용 빈도, 플레이어의 라이프 감소 수, 미션 처리 속도 등 플레이어의 대응을 수치화 시킨 것이 대응 패턴 데이터로 사용될 수 있다. 게임 장르에 구분 없이 이런 대응 패턴 데이터는 수치화될 수 있다. 레벨 디자이너는 난이도나 밸런스에 영향을 줄 수 있는 플레이어의 대응 패턴 도메인(예: 플레이어 위치, 무기 사용 빈도수)을 설정해서 시스템 내에서 수집할 수 있도록 해야 한다. 이렇게 정의된 도메인에 따라 추출된 대응 패턴 데이터들은 플레이어의 게임 플레이 수준을 평가하는데 사용될 수 있으며 레벨 디자이너가 설정한 환경을 겪을 수 있도록 하는 기반 데이터가 된다.

레벨 디자이너는 플레이어의 대응 패턴이 어떤 분포를 이룰 때 가장 재미 있는 플레이가 된다는 가정을 하고 이런 분포도를 정하는데 이를 목적 패턴이라고 한다.

본 논문에서 설정한 플레이어의 대응 패턴 데이터는 슈팅 게임 내에서 움직임 위치 분포도와 무기 사용 빈도수로 설정하였다. 움직임 위치 분포도는 플레이어의 움직임의 활발함을 나타내고 무기 사용 빈도수는 적을 격파하는데 걸린 시간이나 난이도를 뜻하며 보유하고 있는 필살무기 수는 앞으로 나올 적들에 대한 긴장도를 뜻한다고 볼 수 있다. 특히 플레이어의 움직임은 슈팅게임에서 플레이어의 활발함 정도를 나타낸다. 즉 적의 공격을 피하거나 공격하는 대응 패턴이 다른 장르의 게임 보다 슈팅장르에서는 플레이어의 움직임으로 표현 된다. 즉 이런 움직임 분포도, 혹은 시간에 따른 활발함 정도는 게임 난이도 혹은 흥미도로 볼 수 있다. 단순로운 움직임은 쉬운 조작을 의미하며 활발한 움직임은 플레이어가 어렵다고 느낀다고 볼 수 있다.

2.2 Gaussian Mixture Model 과 EM Algorithm

GMM은 데이터를 모델링 하는 방법의 일종으로 가우시안 분포를 이용하여 데이터를 모델링하는데 이때 2개 이상의 가우시안 분포를 사용하여 더욱 정확히 데이터를 모델링 하게 된다. 이렇게 데이터를 모델링 하는 이유는 데이터를 대표하는 파라미터 값으로 표현하여 비교, 분석을 쉽게 하기 위해서다. 우리는 게임에서 발생하는 플레이어의 대응 데이터들을 모델링 해서 비교, 분석하는 것이 목적이므로 비교적 간단한 가우시안 분포를 이용하는 GMM를 사용했다.

플레이어 대응 패턴 및 레벨 디자이너의 목적 패턴을

GMM로 표현했다. 일반적으로 데이터의 GMM분포를 구하는 것은 데이터의 GMM 파라미터 $(\pi_k, \mu_k, \Sigma_k, k = 1, \dots, K)$ 를 구하는 것이다. 이들 중 컴포넌트의 수 k 는 적절한 값으로 미리 주어질 수도 있지만, 경우에 따라서는 최적화 되어야 하는 대상이 될 수도 있다. 본 논문에서는 적절한 상수 값($k = 3$)으로 설정했다. k 개의 컴포넌트를 가지는 GMM의 확률 밀도 함수는 식(1)과 같다.

$$G(X|\Theta) = \sum_{i=1}^k p(i) g(X|\theta_i) \quad \text{where,} \tag{1}$$

$$\sum_{i=1}^k p(i) = 1, \quad p(i) \geq 0$$

이 식에서 x 는 특징 벡터이고 Θ 는 $\{p(i), \theta_i\}_{i=1}^k$ 이며, θ_i 는 계수 벡터로써 각 컴포넌트의 평균과 공분산이다. $p(i)$ 는 i 번째 컴포넌트가 발생할 사전확률을 나타낸다.

g 는 다변량 가우시안 분포 함수이다. GMM에 대한 최대 우도 계수 (maximum likelihood parameter)를 결정하기 위해 EM (Expectation-Maximization) 알고리즘을[13] 사용하였다. EM 알고리즘은 주어진 계수들을 사용하여 평균 우도 함수를 구하는 2번째 단계와 데이터들에 대한 최대 우도 함수를 구하는 3번째 단계로 구성되어 있으며, 우도 함수 값이 수렴할 때까지 다음과 같은 단계를 반복하는 알고리즘이다[그림 1].

1. Each parameter is set initial value. $p(i)$ is set $1/k$.
2. The $p(x)$ of each component is calculated in E-step (Eq. 2).

$$P_i(x) = \frac{p(i)g(x|\theta_i)}{\sum_{j=1}^k p(j)g(x|\theta_j)} \tag{2}$$
3. Each parameter is maximized (Eq. 3).

$$p^{(i)} = \frac{\sum_{j=1}^n P_i(x_j)}{n}, \quad \mu_i = \frac{\sum_{j=1}^n P_i(x_j)x_j}{\sum_{j=1}^n P_i(x_j)}$$

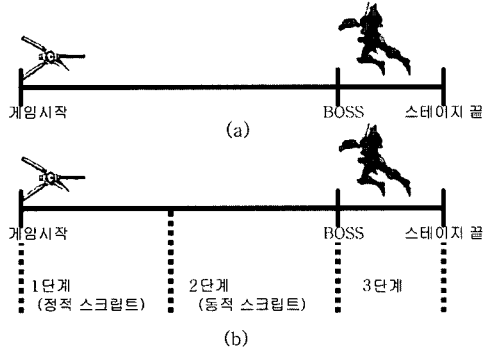
$$\Sigma_i = \frac{\sum_{j=1}^n P_i(x_j)(x_j - \mu_i)(x_j - \mu_i)^T}{\sum_{j=1}^n P_i(x_j)} \tag{3}$$
4. These two steps are reiterated until converging (Eq. 4).

$$L(\Theta) = \sum_{j=1}^n \log G(x_j|\Theta) \tag{4}$$

[그림 1] EM 알고리즘

2.3 슈팅 게임에 적용

본 논문에서는 위에서 말한 GMM을 2D 슈팅게임에 적용해 플레이어의 대응 패턴을 모델링했으며 대응 패턴을 분석해 레벨 디자이너가 원했던 레벨이나 재미를 느낄 수 있도록 동적으로 적의 공격 패턴을 조절한다.

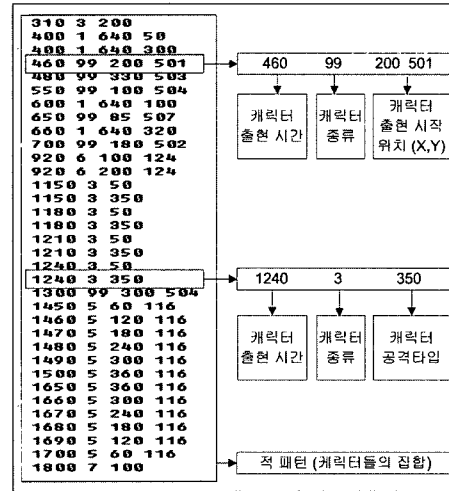


[그림 2] 슈팅게임의 기본 구조: (a) 기존의 슈팅 게임 구조 (b) 동적 스크립트 슈팅게임 구조 (대응 패턴 수집 단계, 동적 스크립트 적용 단계, 보스 출현 단계)

일반적인 슈팅게임에서는 그림 2(a)와 같이 스크롤 형식으로 게임이 진행되며 아이템 습득 및 적 캐릭터와 보스를 격파하게 되면 전체 스테이지가 클리어 된다. 기존의 정적 스크립트를 사용하는 슈팅게임에서는 정해진 스크립트를 따라서 진행되며 적 캐릭터는 정해진 시간에 정해진 패턴으로 등장하고 이동, 공격한다. 이런 방식은 플레이어가 쉽게 적의 공격 패턴에 적응해서 레벨 디자이너가 의도했던 레벨을 보장할 수 없으며 적용된 적의 움직임은 플레이어를 지루하게 만든다. 그래서 그림 2(b)와 같이 진행 과정을 3단계로 나누고 논문에서 제안하는 방법을 적용하고자 한다. 1단계에서는 기존의 정적 스크립트 방식으로 진행되고 플레이어의 대응 패턴을 추출한다. 2단계에서는 1단계에서 추출된 대응 패턴을 분석해서 레벨 디자이너가 의도했던 레벨이나 게임 환경을 만들 수 있는 스크립트를 동적으로 선택한 뒤 이를 구성해서 게임을 진행한다. 이렇게 동적으로 구성된 스크립트에 의해서 플레이어의 동작을 더욱 활발하게 움직이게 한다거나 플레이어가 사용한 무기 수를 조절해서 적절한 무기소지 비율을 유지할 수 있게 된다. 이렇게 함으로써 전체적으로 플레이어의 긴장도를 유지하며 3단계에서 보스를 대적할 때 적절한 무기 개수를 유지해서 레벨 디자이너가 원했던 게임 밸런스를 유지하도록 한다.

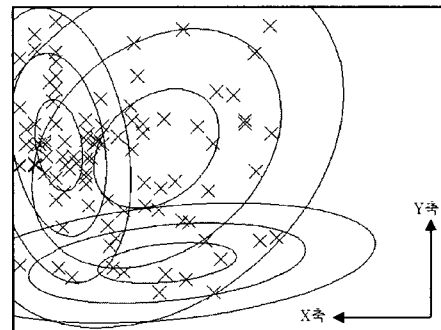
레벨 디자이너는 1단계에서 추출하게 될 플레이어의 대응 패턴 중 어떤 요소를 사용할지 정해야 한다. 본 논문에서

는 게임 플레이어의 위치(x, y 좌표 값)를 도메인으로 정했으며 이렇게 추출된 데이터를 GMM으로 모델링 한다. 이렇게 모델링 된 데이터와 레벨 디자이너가 의도했던 분포도의 차이를 비교해서 레벨 디자이너가 의도했던 게임 레벨이나 환경을 만들어줄 적의 공격 패턴을 동적으로 설정해서 2단계에 사용될 스크립트 내용을 정하게 된다.



[그림 3] 게임에서 사용된 스크립트: 한 라인은 하나의 캐릭터 속성을 나타내고 라인들의 집합은 적의 공격 패턴으로 정의

[그림 3]은 게임에서 사용되는 적 캐릭터의 공격 패턴 스크립트이다. 스크립트 안의 한 라인은 한 캐릭터의 속성값을 나타내며 이런 라인들의 집합을 하나의 패턴으로 정의한다. 속성값은 시작위치 및 캐릭터 하나의 공격 패턴 루틴을 나타낸다. 본 논문에서는 이렇게 정의된 하나의 패턴에 대한 플레이어의 반응 데이터를 추출하게 된다. 위에서 정의된 적의 패턴에 플레이어가 어떤 움직임으로 대응했는지를 나타낸다 [그림 4].

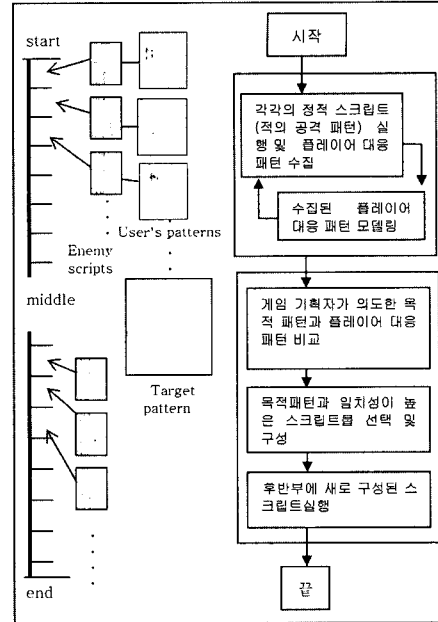


[그림 4] 플레이어의 동작이 GMM으로 표현된 예

[그림 4]에서의 가로축과 세로축은 각각 플레이어의 x, y 위치 좌표 값에 대응되고 샘플링된 데이터 (그림 안의 X 표시)는 적의 공격 패턴 하나가 시작되고 끝날 때까지 샘플링된 플레이어의 대응 동작 위치를 나타낸다. 각각의 타원은 가우시안 컴포넌트로서 타원의 중앙 좌표가 컴포넌트에 속한 데이터들의 평균값이다. 이와 같이 적의 공격 패턴에 대한 플레이어 대응 동작을 추출 및 모델링 함으로써 게임 후반부에 레벨 디자이너가 의도했던 게임 환경이 될 수 있는 패턴을 동적으로 선택할 수 있게 된다.

게임 후반부에 쓰일 동적 스크립트는 다음과 같은 과정으로 만들어 진다 [그림 5]. 중반 부까지 정적 스크립트 방식으로 진행되며 적의 공격 패턴(하나의 독립된 스크립트) 마다 대응되는 플레이어 대응 패턴을 모델링 한다. 이렇게 모델링 된 플레이어 대응 패턴은 기획자가 의도한 목적 패턴과 비교하여 최종적으로 플레이어 움직임 분포도를 목적 패턴과 가깝게 만들어줄 스크립트를 선택해서 후반부에 사용될 동적 스크립트를 구성하게 된다. 이렇게 선택된 스크립트들은 플레이어의 대응을 레벨 디자이너가 의도한 대로 만들어 줄 확률이 더 높은 스크립트들이다. 최종적으로 게임이 끝난 후에 플레이어의 대응 패턴은 레벨 디자이너가 의도했던 분포를 따름을 볼 수 있다.

GMM으로 표현된 데이터를 이용해서 동적으로 적의 공격 패턴을 선택하는 방식으로 두 가지 방법을 사용했다. 첫 번째 방식은 레벨 디자이너가 정한 목적 분포도와 가장 근접한 분포도 [그림 6]를 보이는 적의 공격 패턴을 선택하는 방법이다. 두 번째 방식은 목적 패턴의 각 컴포넌트의 평균점을 기점으로 해서 추출된 분포도의 평균 값과 반대 방향으로 기준점을 정하고 그 평균과 근접하는 패턴을 선택하도록 했다 [그림 7]. 즉, 최종 플레이어 대응 패턴을 목적 패턴에 수렴하는 속도를 증가 시키기 위해 반대 방향으로 기준점을 설정하고 그 기준점과 일치도가 높은 대응 패턴을 보이는 적의 공격 패턴을 선택하는 방식이다. 알고리즘 2는 기준점을 새로 정한 후 알고리즘 1을 실행하는 방법으로 알고리즘 1을 이용해서 선택된 스크립트와 알고리즘 2를 이용해서 선택된 스크립트를 조합하면 목적 패턴과 유사한 스크립트만 선택했을 때 보다 후반부의 적의 공격 패턴을 다양하게 구성할 수 있으며 목적 패턴에 수렴 하는 속도를 증가시킬 수 있다.



[그림 5] 동적 스크립트의 생성과정

1. 레벨 디자이너가 디자인한 GMM에서 각 컴포넌트 평균을 u_k 로 하고 적의 패턴 GMM의 각 컴포넌트 평균을 u'_k 라 하자.
2. u_k 와 u'_k 간의 유클리디안 거리를 비교해서 가장 거리가 가까운 점(u_k, u'_k)끼리 짝을 짓는다.
3. 짝지어진 점끼리 거리 차를 구한다.
4. 3에서 구한 거리 차를 k 개 만큼 구한 뒤 합한다.

[그림 6] 레벨 디자이너가 설정한 목적 패턴과 플레이어 대응 패턴과의 일치성 계산 알고리즘 1

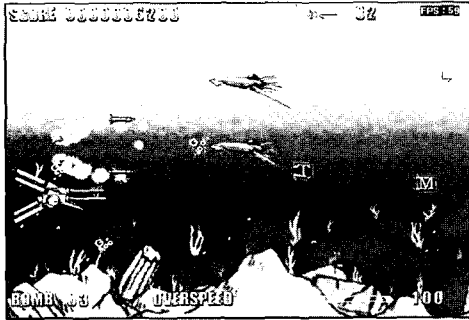
1. 레벨 디자이너가 디자인한 GMM의 컴포넌트의 평균점을 기준점으로 잡고 플레이어의 반응 패턴 컴포넌트의 평균과의 차이벡터 값을 구한다.
2. 1에서 구한 벡터의 반대 방향 벡터로 기준점을 설정한다.
3. 2에서 정한 기준점을 이용해서 알고리즘 1을 수행 함으로써 일치성을 계산한다.

[그림 7] 레벨 디자이너가 설정한 목적 패턴과 플레이어 대응 패턴과의 일치성 계산 알고리즘 2

3. 실험 및 결과

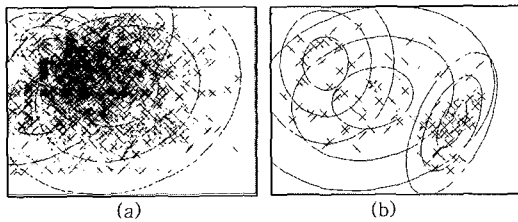
위에 설명된 이론을 실제 개발된 게임에 적용했으며 [그림 8] 실험의 목적은 동적으로 조절된 난이도에 따라 반응

한 플레이어의 대응 패턴이 레벨 디자이너가 의도했던 움직임 패턴과의 일치성을 높일 수 있음을 보이는 것이다. 사전에 레벨 디자이너가 가장 재미를 느끼게 하는 동작 패턴을 정의한 뒤 이를 GMM으로 모델링 하고 게임 초반과 중반부에 사용될 적의 공격 패턴을 스크립트로 구성한다.



[그림 8] 실험에 사용된 게임

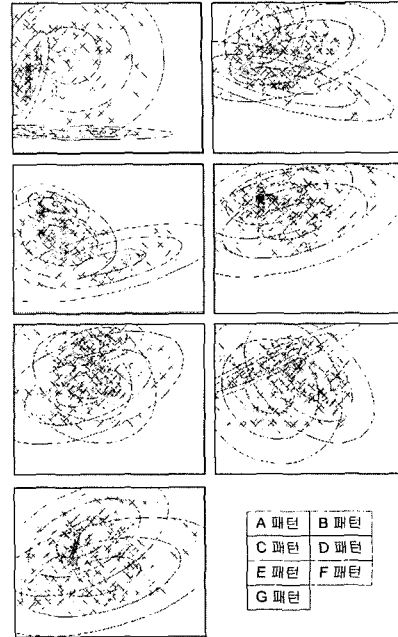
본 실험에서는 적의 공격 패턴을 7개로 정의했으며 하나의 패턴은 적 비행기들의 출현 조합, 시작 위치, 동작들로 구성된다. 각각의 공격 패턴 지속 시간은 1분으로 정했으며 적의 공격 패턴(이를 정의한 스크립트) 하나당 2회 수행되며 정적 스크립트를 사용하는 1단계에서 14개의 스크립트가 수행되며 14분 동안 지속된다. 플레이어의 동작 위치는 초당 6번 샘플링되며 수집된 플레이어 대응 패턴은 4~5회 EM 알고리즘이 반복 수행되면 수렴하므로 실시간으로 모델링 될 수 있다.



[그림 9] (a) 게임 중반까지 플레이어 대응 패턴을 GMM으로 모델링한 그림, (b) 사전에 레벨 디자이너가 의도한 움직임 분포도를 모델링한 그림

[그림 9] (a)는 게임 초반부터 중반부까지 정해진 스크립트대로 게임이 진행됐을 때 플레이어의 반응 데이터를 GMM으로 모델링한 그림이고 그림 9(b)는 레벨 디자이너가 의도한 움직임 분포도를 모델링한 그림이다.

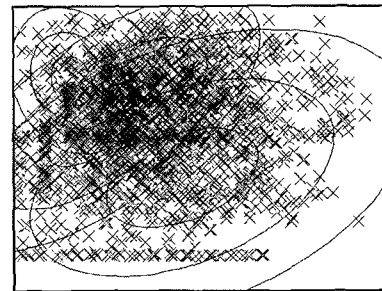
[그림 10]은 게임에 사용된 적의 공격 패턴 별로 플레이어의 대응 동작을 GMM으로 표현한 것이다.



[그림 10] 각각의 적 공격 패턴에 따라 반응한 플레이어의 움직임 분포를 GMM으로 모델링한 그림

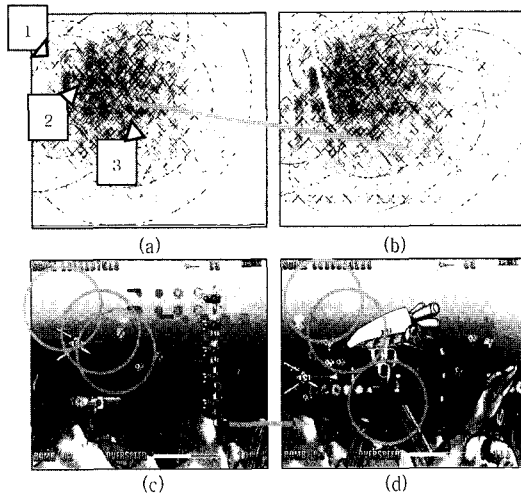
전반적인 플레이어 대응 동작이 레벨 디자이너가 의도했던 대로 분포시키기 위해서는 중반부까지 플레이어의 대응 패턴과 목적 패턴과의 차이를 분석하고, 이 차이를 줄일 수 있는 스크립트(적의 공격 패턴)를 선택한 뒤 이 스크립트를 조합해서 중반부 이후의 스크립트를 구성한다.

2.3절에서 언급한 선택 알고리즘에 따라 A, C, G 패턴의 분포도가 [그림 9] (b)와 가까운 패턴이라 할 수 있다. 게임의 중반부 이후는 이 패턴을 중점으로 게임이 구성되며 게임이 끝난 후 전체적인 플레이어의 패턴은 다음 그림 11과 같고 각각의 컴포넌트들의 평균 이동을 보면 레벨 디자이너가 의도했던 분포도를 따름을 볼 수 있다.



[그림 11] 동적 스크립트가 적용된 후 플레이어의 움직임 분포도

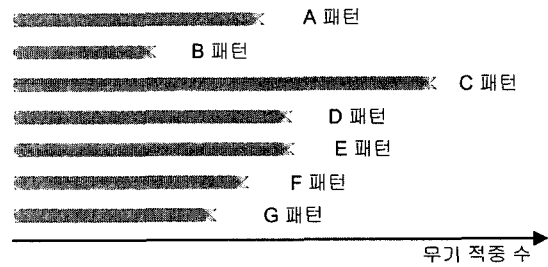
1) 2D 슈팅게임(게임명: FINAL MISSION)이며 2001년에 오픈소스로 공개한 버전 (개발자: 임윤호)



[그림 12] 움직임 분포도의 변화 비교: (a)게임 중반 부까지 플레이어의 움직임 분포도, (b)후반 부에 동적 스크립트가 적용된 후 플레이어의 움직임 분포도, (c)세 개의 타원은 중반 부까지 플레이어의 주된 움직임 분포를 나타낸다, (d)게임 실행 후 끝날 때까지 플레이어 움직임 분포

[그림 12]에서 동적 스크립트를 적용 후 움직임 분포의 변화가 실제 게임상에서 어떻게 나타나는지 보여준다. [그림 12] (b)는 게임 실행 후 끝날 때까지 플레이어가 보인 움직임 분포를 모델링한 그림이다. 세 번째 하단의 가우시안 컴포넌트(그림 12 (a)번호 3)가 더욱 분포가 커지고 평균이 아래로 내려 갔음을 볼 수 있다. [그림 12] (a)와 비교해서 보면 [그림 12] (b)의 하단 부분의 컴포넌트를 반영했음을 볼 수 있다. [그림 12] (a)의 1번 컴포넌트의 분산의 변환 역시 반영됨을 볼 수 있다. [그림 12]의 두 번째 열은 각 컴포넌트들이 게임상에서 어디에 위치하는지 나타내고 타원은 움직임의 주요 분포 위치를 나타낸다. 중반까지 왼쪽 위에 치우친 플레이어의 움직임 분포를 [그림 12(c)] 레벨 디자이너가 의도한 움직임 분포도로 변화하는 것을 볼 수 있다(그림 12(d)).

추가적으로 플레이어가 각 패턴마다 소비한 무기, 적중한 무기수를 분포를 보고 난이도를 동적으로 조절 할 수 있다. [그림 13]은 각 적의 공격 패턴 마다 플레이어가 쓴 무기가 적에게 적중한 수를 나타낸다. 예를 들어 레벨 디자이너가 어느 단계에서 특정 무기수를 유지하고 싶다면 이런 무기 사용 빈도수 자료를 추출해서 특정 무기 소비량이 높은 적의 공격 패턴을 중심으로 스크립트를 구성하는 전략을 취할 수 있다. 또한 어떤 스크립트를 선택할 때 가중치로 사용될 수 있다.



[그림 13] 각 패턴마다 무기 적중 개수: X축은 적중한 개수 Y축은 각 패턴을 뜻한다

4. 결론

각각의 정의된 대응 패턴 도메인들은 게임의 재미에 영향을 끼칠 수 있는 변수임을 의미를 가진다. 예를 들어 레벨 디자이너가 플레이어의 활발한 움직임을 게임의 재미 요소로 정했을 때 레벨 디자이너는 대응 패턴 도메인을 x, y축 좌표로 잡고 플레이어의 움직임의 분포도를 분석하면 되고, 적절한 무기 조합과 사용에 재미를 두었다면 대응 패턴 도메인을 무기 사용 적중수로 놓고 적의 공격 패턴당 플레이어의 무기 사용 패턴 분포도를 분석하면 게임 안에서 의도했던 적절한 긴장도를 유지할 수 있게 된다. 이렇게 도메인을 정하고 수집된 플레이어 대응 패턴은 GMM을 사용해서 파라미터 값으로 표현할 수 있다. 이와 같이 레벨 디자이너의 의도와 플레이어의 대응 패턴을 파라메트릭하게 표현할 수 있게 되면, 게임 도중에 시스템이 대응 패턴을 비교하기가 용이하고 목적 패턴과 대응 패턴간의 일치성을 수치화시킬 수 있다. 즉 '활발하게 움직인다', '움직임이 단조롭고 쉬운 난이도'와 같은 추상적인 표현을 수치로 표현이 가능해지면 레벨 디자이너가 스크립트를 설계할 때 데이터를 기반으로 설계가 가능하다. 또한 베타 테스트할 때 수집되는 플레이어의 대응 데이터를 모델링해서 분석이 용이해지는 장점이 있다.

본 논문에서는 이렇게 각각의 대응 패턴 도메인을 정하고 레벨 디자이너가 의도한 체감 난이도를 유지하기 위한 방법을 제시했다. 향후 발전 방향은 게임 내에서 발생하는 예외적인 상황들을 고려한 대응 데이터 수집 방법과 더 정확한 GMM 간의 비교 및 적절한 패턴 추출에 대해 연구할 예정이다.

“본 연구는 숭실대학교 교내 연구비 지원으로 이루어졌습니다.”

참고문헌

- [1] Noah Falstein, "Game Developer Magazine," The Flow Channel, 2004.
- [2] Raph Koster, "Theory of Fun for Game Design," Paraglyph Press, Phoenix, 2004.
- [3] Daniel Johnson, Janet Wiles, "Effective Affective User Interface Design in Games," International Conference on Affective Human Factors Design, Singapore, 2003.
- [4] John E. Laird, "Using a Computer Game to Develop Advanced AI," IEEE Computer Society Press, pp.70-75, July, 2001.
- [5] Bernd Freisleben, "A Neural Network that Learns to Play Five-in-a-Row," International Conference on Artificial Neural Networks and Expert Systems, pp.87-90, 1995.
- [6] I. Faybish, "Applying the Genetic Algorithm to the Game of Othello," Master's thesis, Vrije Universiteit Brussel, Computer Science Department, Brussels, Belgium, 1999.
- [7] Imran Ghory, "Reinforcement learning in board games," Technical Report CSTR-04-004, Department of Computer Science, University of Bristol, May 2004.
- [8] Pieter Spronck, Ida Sprinkhuizen-Kuyper and Eric Postma, "Difficulty Scaling of Game AI," International Conference on Intelligent Games and Simulation, Belgium, pp. 33-37, 2004.
- [9] Pieter Spronck, Ida Sprinkhuizen-kuyper, and Eric Pstma, "Enhancing the Performance of Dynamic Scripting in Computer Games," Lecture Notes in Computer Science, Vol. 3166, pp. 296-307, 2004.
- [10] McLachlan, G. j., Ng, S. K., Galloway, G. J. and Wang, D., "Clustering of Magnetic Resonance Image," Technical Report, Department of Mathematics, University of Queensland, Australia, 1998.
- [11] Todd K. Moon, "The Expectation-Maximization Algorithm," IEEE Signal Processing, Vol. 13, pp.47-60, 1996.
- [12] Richard O. Duda, Peter E. Hart, David G. Stork, "Pattern Classification," 2nd Edition, pp. 55, John Wiley & Sons Inc, 2001.
- [13] Chad Carson, Serge Belongie, Hayit Greenspan, Jitendra Malik, "Blobword: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying," IEEE Trans. On Pattern Recognition and Machine Analysis, Vol. 24, No. 8, pp. 1026-1038, Aug. 2002.
- [14] 이상경, 경동욱, 정기철, "GMM을 이용한 동적 게임 레벨 디자인," 한국 게임학회, 2006 동계한국게임학회 학술발표대회, pp. 51-55, Jan. 2006.



이상경

2005년 승실대학교 미디어학부 (학사)
 2006년 ~ 현재 승실대학교 대학원 미디어학과 (석사과정)
 관심분야 : 컴퓨터 게임 프로그래밍, 컴퓨터 그래픽스, 컴퓨터 비전



정기철

1994년 경북대학교 컴퓨터 공학과(공학사)
 1996년 경북대학교 대학원 컴퓨터공학과(공학석사)
 2000년 경북대학교 대학원 컴퓨터공학과(공학박사)
 1999년 방문 연구원, Intelligent User Interfaces group,
 DFKI(The German Research Center for
 Artificial Intelligence, GmbH), Germany.
 2001년 PRIP Lab., Michigan State University, U.S. 박사후
 연구원
 2003년 ~ 현재 승실대학교 IT대학 미디어학부 교수
 관심분야 : HCI, Interactive Contents, 영상처리, 패턴인식, Augmented
 Reality, Mobile Vision

논문투고일 - 2006년 2월 7일
 심사완료일 - 2006년 3월 7일