한국컴퓨터그래픽스학회 Vol. 15, No. 2, P. 9~17

마르코프 게임 학습에 기초한 다수 캐릭터의 경쟁적 상호작용 애니메이션 합성

이강훈

광운대학교 컴퓨터소프트웨어학과

kang@kw.ac.kr

Learning Multi-Character Competition in Markov Games

Kang Hoon Lee

Dept. of Computer Science, Kwangwoon University

요 약

다수 캐릭터가 경쟁적으로 상호작용하는 애니메이션의 합성은 컴퓨터 게임, 애니메이션 등의 응용분야에서 종종 요구되는 중요한 문제이다. 하지만 상대의 예측하기 어려운 행동에 효과적으로 대응하는 전략적 경쟁 양상을 모사하는 것은 어려운 문제로 남아있다. 본 논문은 다수 에이전트 학습 분야에서 제안된 마르코프 게임 강화학습 알고리즘을 촬영된 동작 데이터로부터 생성된 행위 모델에 적용하여 사실적인 경쟁 애니메이션을 합성하는 방식을 제안한다. 추격-회피, 간격 유지, 총격전 등의 다양한 경쟁적 상황에 대하여 효과적인 전략을 학습하여 흥미로운 애니메이션을 합성하는 예제들을 통하여 본 논문이 제안하는 방법의 효용성을 보인다.

Abstract

Animating multiple characters to compete with each other is an important problem in computer games and animation films. However, it remains difficult to simulate strategic competition among characters because of its inherent complex decision process that should be able to cope with often unpredictable behavior of opponents. We adopt a reinforcement learning method in Markov games to action models built from captured motion data. This enables two characters to perform globally optimal counter-strategies with respect to each other. We also extend this method to simulate competition between two teams, each of which can consist of an arbitrary number of characters. We demonstrate the usefulness of our approach through various competitive scenarios, including playing-tag, keeping-distance, and shooting.

키워드: 캐릭터 애니메이션, 모션 캡처, 마르코프 게임, 강화학습

Keywords: Character Animation, Motion Capture, Markov Games, Reinforcement Learning

1. 서론

사실적이고 섬세한 동작 데이터의 확보가 용이해짐에 따라 이를 활용한 캐릭터 애니메이션 합성 및 제어에 대한 연구는 많은 주목을 받아왔다. 특히 최근에는 기계학습

분야의 탐색 및 학습 알고리즘을 적용함으로써 주어진 목표 및 주위 상황에 대응하여 적절한 동작을 합성하는 자율적 캐릭터에 대한 관심이 증가하고 있다. 하지만 아직까지 컴퓨터 게임, 애니메이션 등에서 흔히 요구되는 다수 캐릭터의 경쟁적 상황을 다룬 연구는 많지 않다.

본 논문은 다수 캐릭터의 경쟁적 상호작용 장면을 효과적으로 연출하기 위하여, 자신이 선택한 동작에 따라 보상을 받는 게임 상황을 설정하고 각 캐릭터가 매번 자신의 보상을 최대화하는 동작을 선택하도록 하는 방법을 제안한다. 일종의 카드 게임을 연상하면 된다. 매 순간 각 캐릭터는 한 장 이상의 카드를 손에 쥐고 있고, 각각의 카드는 자신이 수행할 수 있는 동작에 해당된다. 서로 어떤 카드를 쥐고 있는지는 공개되어 있다고 가정한다. 게임에 참여한 모든 캐릭터는 동시에 한 장씩의 카드를 선택하여 내고, 선택한 카드에 따라 동작을 수행한 후 그에 따른 보상을 돌려받는다. 이때 선택된 카드의 조합에 따라 상대적인 보상을 제공하여야 경쟁적 게임이 성립한다. 예를 들어 두 캐릭터가 추격-회피 게임을 할 경우, 동작 수행 후 서로 간의 거리가 짧아졌다면 추격하는 캐릭터는 높은 보상을 받는데 반하여 회피하는 캐릭터는 낮은 보상을 받아야 할 것이다.

보상을 최대화하는 카드를 선택하는 것은 쉽지 않은 문제이다. 우선 모두가 동시에 카드를 공개하므로 상대가 어떤 카드를 선택할지 알 수 없고, 또한 현재의 선택이 당장의 보상뿐 아니라 이후에 얻게 될 보상에도 영향을 미치기 때문이다. 게임이론 분야에서는 전자의 특성을 가진 게임을 동시 게임이라고 부르고, 그 중 보상의 합이 0 이 되는 제로섬 게임에 대하여 선형 프로그래밍 방식으로 최적의 확률적 전략을 구하는 방법을 연구하였다. 한편, 기계학습 분야에서는 후자의 특성을 가진 마르코프 의사결정 과정에서 즉각적인 보상과 미래의 기대보상을 합한 총 보상을 최대화하는 전략을 학습하는 강화학습 알고리즘을 개발하였다. 나아가서, 기계학습 분야의 연구자 Littman 은 이 두 가지 특성이 결합된 형태의 제로섬 마르코프 게임에 대하여 총 보상을 최대화하는 최적의 확률적 전략을 학습하는 알고리즘을 제안하였다 [1]. 단, 이 알고리즘은 두 명이 참여한 게임에 대하여서만 적용 가능하고, 그 이상의 참여자가 개입된 게임에 대한 학습 알고리즘은 아직 알려진 바가 없다.

본 논문은 Littman 이 제안한 방법을 캡처된 동작 데이터로 애니메이션 되는 두 캐릭터 간의 경쟁적 애니메이션에 적용하여 그 효용성을 확인하고, 나아가서 간단한 알고리즘을 동원하여 셋 이상의 다수 캐릭터 간의 경쟁적 애니메이션 합성이 가능함을 보인다. 본 논문은

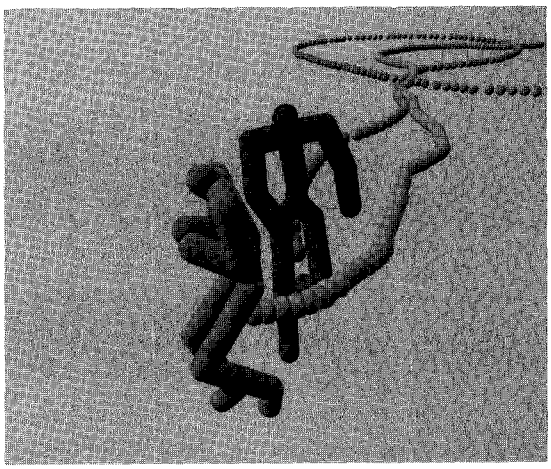

그림 1. 두 캐릭터의 줓고 쫓기기 게임. 파란 색 캐릭터는 거리를 최소화, 빨간 색 캐릭터는 최대화하고자 하는 목표에 따라 최적의 전략을 수행한다. 일련의 작은 공으로 연결된 곡선은 이동 궤적을 나타낸다.

게임이론과 결합된 강화학습 알고리즘을 캐릭터 애니메이션에 적용한 첫 번째 연구결과로서, 아직 시작 단계에 속한 다수 캐릭터 애니메이션 연구의 새로운 가능성을 모색한다.

2. 관련 연구

본 논문은 촬영된 동작 데이터를 활용하여 캐릭터의 동작 시퀀스를 합성한다. 주어진 동작 데이터로부터 새로운 동작을 합성하는 방법으로 제약조건 기반 편집[2,3], 물리 기반 편집[4], 혼합[5], 그래프 기반 재배열[6,7] 등 다양한 방법이 제안되어 왔는데, 이 중 본 연구에서 사용하는 합성 방식은 동작 그래프를 이용하여 부드럽게 연속된 새로운 시퀀스를 생성하는 마지막 접근방법에 가깝다. 단, 학습을 위한 계산 및 저장공간 비용이 지나치게 증가하는 것을 방지하기 위하여 하나의 노드와 여러 에지로 구성된 비교적 작고 단순한 그래프 구조를 손으로 기술하여 사용한다. 이와 같은 그래프 구조는 Gleicher 등이 제안한 그래프 구축 알고리즘을 적용하여 자동적으로 생성하는 것도 가능하다 [8].

기계학습 분야의 강화학습 알고리즘은 환경에 대응하는 능력을 갖춘 지능형 캐릭터를 제작하기 위한 목적으로 최근 캐릭터 애니메이션 분야에서 효과적으로 활용되고 있다. Lee 등은 권투선수의 동작 데이터로부터 목표물에

접근하여 가격하는 행동을 학습시켜 효과적인 대화형 제어 및 다수 캐릭터 애니메이션 합성이 가능함을 보였다 [9]. Treuille 등은 상태의 차원증가에 따른 저장공간 증가를 완화할 수 있는 함수근사에 기초한 강화학습 알고리즘을 소개하였다 [10]. 최근에는 Lo 등이 매개화된 동작 데이터와 트리 기반 회귀 알고리즘을 이용함으로써 좁은 문을 통과하거나 물체를 집는 등의 정확도가 요구되는 작업에도 강화학습을 활용할 수 있음을 보였다 [11]. 이들 연구에서 학습된 제어기를 활용하여 다수 캐릭터 애니메이션에 적용한 사례는 찾을 수 있지만, 경쟁관계에 놓인 상대를 고려하여 학습을 수행하고 이를 다수 캐릭터의 경쟁적 애니메이션 합성에 적용한 경우는 없다.

다수 캐릭터의 상호작용 애니메이션 합성은 비교적 최근에 들어서 연구가 진행되고 있다. Zordan 등은 캡처된 동작 데이터를 이용하여 물리적 충격에 따른 반응 동작을 합성하는 알고리즘을 제안함으로써 두 캐릭터 간의 사실적인 격투 애니메이션을 합성할 수 있음을 보였다 [12]. Liu 등은 상호작용 제약조건에 따른 시공간 최적화를 적용함으로써 단일 연기자의 동작 데이터로부터 두 캐릭터가 손을 잡고 다니는 등의 상호 결합된 애니메이션을 생성하는 방법을 제안하였다 [13].

이들 방법이 공통적으로 동작 데이터의 물리적 변형에 기초하는데 반하여, 본 논문과 유사하게 동작 데이터를 재배열함으로써 긴 시간의 상호작용 동작 시퀀스를 생성하고자 하는 연구도 있었다. 특히 Shum 등이 제안한 방법은 제로섬 게임을 정의하고 각 캐릭터가 기대보상을 최대화하는 동작을 선택하게 함으로써 경쟁적 애니메이션을 합성한다는 점에서 본 연구와 가장 유사하다 [14,15]. 가장 큰 차이점은 본 연구에서 모든 캐릭터는 동시에 의사결정을 수행하기 때문에 확률적 전략이 최적인데 반하여, Shum 등의 연구에서는 캐릭터들이 순차적으로 의사결정을 수행하기 때문에 최적의 결정적 전략을 찾는다는 점이다. 이는 캐릭터들이 동일한 상황에 놓일 경우 Shum 등의 방법에서는 이후의 시퀀스가 계속 동일하게 나타날 수밖에 없음을 의미한다. 또한 Shum 등이 일정 깊이까지의 최소-최대 트리 탐색에 기초하여 지역적 최적 전략을 찾는데 반하여, 본 연구는 동적 프로그래밍에 기초하여 전역적 최적 전략을 찾는 것도 중요한 차이점이다. Shum 등은 보다 최근에 상호작용 패치를 시공간적으로 결합하여 다수 캐릭터의 애니메이션을 합성하는 별도의 방법을 제안하기도 하였다 [16]. 한편 앞서 기술한 모든 방법은 단일 연기자의 동작 데이터를 이용하고 있지만, Kwon 등은 두 연기자의 동작 데이터로부터 학습한 상호작용의 통계적 모델을 이용하여 동작을 재배열하는 방법을 제안하였다 [17].

	가위	바위	보
가위	(0, 0)	(-1,+1)	(+1,-1)
바위	(+1,-1)	(0, 0)	(-1,+1)
보	(-1,+1)	(+1,-1)	(0, 0)

표 1. "가위-바위-보" 게임의 행렬 표현.

3. 제로섬 게임

게임 이론 분야에서는 다수의 참여자로 구성된 집단에서 각 개인의 의사결정이 다른 참여자의 의사결정에 영향을 미치는 전략적 상황을 여러 종류의 게임으로 분류하고 이에 따른 수학적 분석을 수행하여 왔다. 게임을 분류하는 한 가지 기준은 참여자의 의사결정이 순차적으로 이루어지는가, 혹은 동시에 이루어지는가에 따른 것이다. 순차 게임에서 다음 참여자는 이전 참여자의 선택을 반영하여 번갈아 의사결정을 수행하는데 반하여, 동시 게임에서 모든 참여자는 다른 참여자가 어떤 선택을 할지 모르는 상황에서 동시에 의사결정을 수행한다.

참여자가 선택할 수 있는 가능성이 유한할 경우 동시 게임의 결과는 다차원 표 형태로 표현 가능하다. 표 1 은 간단한 동시 게임의 예로서 두 사람의 "가위-바위-보" 대결에 해당하는 표를 보이고 있다. 두 사람 모두 가위, 바위, 보의 세 가지 중 하나만 선택할 수 있으므로 게임에서 등장할 수 있는 모든 조합은 가로 3, 세로 3 크기의 표 안에 표현되고, 각 원소는 해당 조합에 따라 두 참여자에게 주어지는 보상을 기록한다. 예를 들어 2 행, 3 열의 원소 (-1,+1)은 첫 번째 참여자가 '바위'를 내고 두 번째 참여자가 '보'를 낼 경우 각 참여자에게 -1 과 +1 의 점수가 주어짐을 의미한다.

표 1 에서 모든 경우에 점수의 합은 항상 0 으로 귀결되는데, 게임 이론에서는 이와 같은 게임을 제로섬 게임으로 분류한다. 두 명이 참여하는 제로섬 게임에서는 최악의 경우에 확보 가능한 보상을 최대화하는 확률적 전략이 존재한다. "가위-바위-보" 게임의 예를 통하여 최적의 전략이 확률 분포가 되는 이유를 생각하여볼 수 있다. 항상 '바위'를 선택하는 결정적 전략을 사용한다면 상대가 항상 '보'를 선택하는 최악의 경우에 확보 가능한 기대 보상은 -1 이다. 반면 '가위', '바위', '보'를 각각

1/3 의 확률로 선택하면 상대의 전략과 상관없이 확보 가능한 기대 보상은 0 이 된다. 이것은 "가위-바위-보" 게임에서 최소 보상을 최대화하는 최적의 전략에 해당한다.

첫 번째 참여자가 선택 가능한 행위 집합을 A, 두 번째 참여자가 선택 가능한 행위 집합을 O, 두 참여자가 각각 $a \in A$ 와 $o \in O$ 의 행위를 선택하였을 때 첫 번째 참여자가 얻을 수 있는 보상을 $R_{a,o}$ 라고 하면, 첫 번째 참여자가 얻을 수 있는 최소 보상의 최대값 V 는 다음의 식으로 표현할 수 있다.

$$V = \max_{\pi \in PD(A)} \min_{o \in O} \sum_{a \in A} R_{a,o} \pi_a \tag{1}$$

여기에서 $PD(A)$ 는 집합 A 에 대한 이산 확률 분포의 집합을 나타낸다. 즉, $PD(A)$ 에 속한 임의의 원소 $\pi = (\pi_1, \cdots, \pi_n)$ 는 행위 $(a_1, \cdots, a_n)$ 을 선택하는 확률 분포에 해당한다. 식 (1)을 풀어서 쓰면, V 는 다음과 같은 일련의 식을 만족하는 최대의 값이고, 이에 해당하는 π 는 첫 번째 참여자가 취할 수 있는 최적의 확률적 전략이다.

$$\begin{aligned} R_{1,1}\pi_1 + \cdots + R_{n,1}\pi_1 &\geq V \\ &\vdots \\ R_{1,m}\pi_1 + \cdots + R_{n,m}\pi_n &\geq V \\ \pi_1 + \cdots + \pi_n &= 1 \\ \pi_1, \cdots, \pi_n &\geq 0 \end{aligned} \tag{2}$$

선형 프로그래밍 기법을 이용하면 이와 같이 선형 등식과 부등식이 포함된 최적화 문제를 효율적으로 해결할 수 있다. 본 연구의 실험에서는 오픈소스 라이브러리인 lpsolve 를 이용하여 선형 프로그래밍을 해결하였다.

4. 마르코프 게임

앞서 살펴본 제로섬 게임에서는 참여자가 처한 상황을 고려하지 않은 채, 단지 각 참여자가 선택한 행동에 따라서만 보상이 결정된다고 가정하였다. 하지만 애니메이션 되는 캐릭터에게 주어지는 보상은 행동뿐 아니라 상태까지 함께 고려하여 주어져야 한다. 예를 들어 캐릭터 간의 격투 게임에서는, 하나의 캐릭터가 공격 동작을 선택하였을 때 다른 캐릭터의 상대적 위치와 자세가 가격범위에 속할 경우에만 보상이 주어져야 할 것이다. 따라서 게임에서 발생 가능한 모든 상태 및 상태 간의 전이 과정을 기술한 후, 상태와 행동을 함께 고려한 보상 기준을 결정하여야 한다.

본 연구는 게임의 상태가 마르코프 의사결정 과정에 따라 전이된다고 가정한다. 게임에서 발생 가능한 상태의 집합을 S , k 개의 캐릭터가 참여자가 취할 수 있는 행동의 집합을 $A_1, \cdots, A_k$ 라고 하였을 때, 상태 간의 전이 T 는 다음과 같이 현재 상태 및 각 참여자가 선택한 행동에 의하여 확률적으로 결정된다.

$$T : S \times A_1 \times \cdots \times A_k \rightarrow PD(S) \tag{3}$$

또한 각 참여자에게 주어지는 보상 역시 현재 상태 및 참여자들이 선택한 행동에만 의존한다고 가정한다.

$$R_i : S \times A_1 \times \cdots \times A_k \rightarrow \Re \tag{4}$$

이와 같이 정의된 게임은 마르코프 게임 혹은 확률적 게임으로 분류된다. Littmann 은 기존의 강화학습 알고리즘과 제로섬 게임의 최소-최대 전략을 결합하여 두 명이 참여하는 제로섬 마르코프 게임의 최적 전략을 학습하는 알고리즘을 제안하였다. 이 알고리즘을 간단히 요약하면 다음과 같다. 먼저 임의의 상태 $s \in S$ 로부터 두 참여자가 각각 a 와 o 행동을 선택하였을 때 즉각적으로 주어지는 보상을 $R_{s,a,o}$, 그 결과 새로운 상태 s' 으로 전이할 확률을 $T_{s,a,o,s'}$ 로 정의한다. 이들은 사용자에 의하여 미리 정의된 게임 세계의 규칙에 해당하는 함수이다. 이제 게임의 현재 상태 s 에서 상대가 취할 수 있는 최악의 행동에 대하여 첫 번째 캐릭터의 즉각적인 기대 보상을 최대화하는 전략은 식 (1)에 상태 변수만 추가하여 다음과 같이 기술할 수 있다.

$$V(s) = \max_{\pi \in PD(A)} \min_{o \in O} \sum_{a \in A} R_{s,a,o} \pi_a \tag{5}$$

이 전략은 당장의 보상은 최대화하지만, 이후 상태 전이를 통하여 최종적으로 확보 가능한 보상의 합계를 최대화한다고 보장하지 않는다. 강화학습은 즉각적인 보상 $R_{s,a,o}$ 를 최대화하는 지역 최적 전략 대신, 다음과 같이 향후 확보 가능한 기대 보상의 합계를 최대화하는 전역 최적 전략을 찾고자 하는 방법이다.

$$V(s) = \max_{\pi \in PD(A)} \min_{o \in O} \sum_{a \in A} \left(R_{s,a,o} + \gamma \sum_{s'} T_{s,a,o,s'} V(s') \right) \pi_a \quad (6)$$

여기에서 γ 는 미래 기대 보상을 매 전이마다 감소시키는 비율을 나타낸다. 이와 같이 부분 문제의 최적값이 전체 문제의 최적값을 구하는데 사용되는 재귀적 관계의 경우, 동적 프로그래밍 기법을 이용하여 전역 최적값을 구하는 것이 가능하다. 단, 연속적인 상태 공간이 주어질 경우 먼저 유한한 크기의 이산적인 상태 공간으로 변환할 필요가 있다. 가장 간단한 방법은 각 상태 차원을 일정 범위 안에서 규칙적으로 추출하여 다차원 표를 구성하는 것이다. 일단 이산적 상태 공간이 주어지면, 반복적으로 $V(s)$ 를 갱신하는 방식을 적용하여 최적값에 수렴하도록 할 수 있다. 먼저 모든 상태에 대하여 $V(s)$ 를 임의의 값으로 초기화한 후, 매 회마다 모든 상태 s 를 순차적으로 선택하여 식 (6)의 오른쪽 부분을 식 (2)와 같은 형태로 전환한 후 선형 프로그래밍으로 해결하여 그 결과를 다시 $V(s)$ 에 대입하는 방식이다. 매 회가 끝날 때마다 $V(s)$ 의 변화 합계를 확인하고, 그 값이 충분히 작으면 최적의 값에 수렴하였다고 판단하고 반복을 멈춘다 (그림 2).

5. 상태-행위 모델

마르코프 게임의 정의는 게임의 상태 공간, 행위 모델, 행위에 따른 상태 전이 및 보상 함수 등을 포함한다. 4 장에서 소개한 강화학습 알고리즘은 상태 공간의 차원이 증가함에 따라서 다차원 상태 표를 저장하기 위한 공간 및 수렴에 이르는 계산 시간이 급격하게 증가하기 때문에 현실적인 응용을 가능케 하려면 비교적 낮은 차원의 상태 공간을 정의할 필요가 있다. 본 연구에서는 다음과 같이 두 캐릭터 간의 상대적인 배치와 관련된 변수만으로 3 차원 상태 공간을 정의한다.

$$S = \{(x, y, \theta) \mid x, y, \theta \in \Re, 0 \le \theta \le 2\pi\} \quad (7)$$

여기에서 x, y, θ 는 첫 번째 캐릭터의 지역 좌표계를 기준으로 한 두 번째 캐릭터의 상대적 위치와 방향을 나타낸다. 값-반복 알고리즘을 적용하기 위하여 이산적 공간으로 분할할 때에는 $-r \le x, y \le r$ 의 범위로 한정한다. r 이 클수록 더 넓은 공간적 범위에 대한 학습이

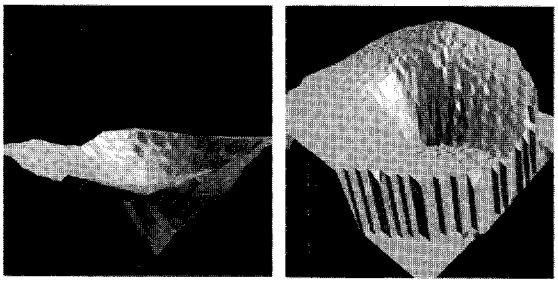

그림 2. 마르코프 게임 강화학습에 의하여 계산된 값-함수의 예. 방향 변수 θ 를 고정하고 단지 x, y 의 변화에 따른 값의 변화를 표현하였다. 왼쪽은 'MARCH 대 HOP', 오른쪽은 'HOP 대 SLOW'의 학습 결과에 해당한다.

가능하지만, 이산 분할의 해상도가 동일할 경우 다차원 상태 표의 크기가 급격하게 증가하기 때문에 절충점을 찾을 필요가 있다. 본 연구의 실험에서 r 은 약 20 미터에 해당하는 값을 사용하였고, 다차원 표는 x, y 축에 대하여 각각 40 칸, r 축에 대하여 36 칸으로 균등 분할하여 총 57,600 개의 칸으로 구성하였다.

캐릭터의 행위는 동일 자세에서 시작하고 종료하는 일정 길이의 동작 세그먼트 집합으로 구성된다. 이는 하나의 노드와 고정된 길이의 동작이 기록된 여러 에지를 가지는 특수한 형태의 동작 그래프로 간주할 수 있다. 각각의 동작 세그먼트는 일련의 자세 정보로 구성되고, 이로부터 상태 변화에 영향을 미치는 요소인 동작 시작으로부터 종료까지의 캐릭터의 위치 및 방향 변화를 얻을 수 있으므로 행위 집합은 다음과 같이 기술할 수 있다.

$$A = \{(M, \Delta x, \Delta y, \Delta\theta) \mid M = [\vec{p}_1, \cdots, \vec{p}_l], \Delta x, \Delta y, \Delta\theta \in \Re, 0 \le \Delta\theta \le 2\pi\} \quad (8)$$

여기에서 M 은 l 개의 자세 벡터 $\vec{p}_i$ 로 구성된 동작 세그먼트에 해당하고, $\Delta x, \Delta y, \Delta\theta$ 은 첫 번째 자세의 지역 좌표계를 기준으로 한 마지막 자세의 상대적 위치와 방향을 나타낸다.

상태 $s = (x, y, \theta)$ 로부터 두 캐릭터가 각각 행위 $a = (M_1, \Delta x_1, \Delta y_1, \Delta\theta_1)$ 와 $o = (M_2, \Delta x_2, \Delta y_2, \Delta\theta_2)$ 를 선택하였을 때 전이된 다음 상태 $s' = T(s, a, o)$ 는 다음과 같이 기술할 수 있다.

$$s' = (x - \Delta x_1 + \Delta x_2, y - \Delta y_1 + \Delta y_2, \theta - \Delta\theta_1 + \Delta\theta_2) \quad (9)$$

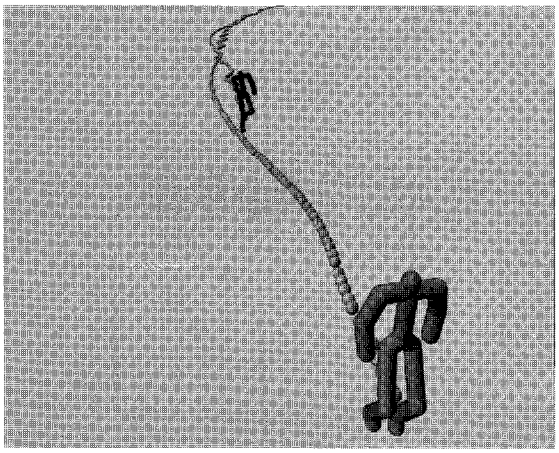

그림 3. 'JOY 대 HOP'의 게임 결과. 두 행위 집합은 유사한 속도와 방향 범위를 가지므로 최적의 전략에 따르면 어느 한 쪽이 압도적으로 승리하기 어렵다.

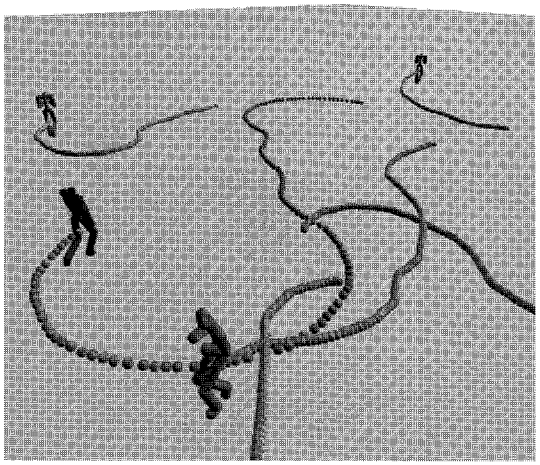

그림 4. 좇는 팀은 JOY 캐릭터 하나, 쫓기는 팀은 SLOW 캐릭터 여럿인 경우, 일대일 상황과 달리 여러 캐릭터를 번갈아 좇는 양상이 나타난다.

보상 함수는 게임의 목표에 따라 상태 및 행위의 속성을 이용하여 가변적으로 정의 가능하다. 상태 변수가 단지 캐릭터 간의 상대적 배치 정보만을 기술하고 있기 때문에, 환경과의 상호작용 등 절대 좌표계에 근거하여 보상을 제공하는 것은 불가능하다. 7 장에서는 본 연구에서 제안하는 비교적 단순한 상태-행위 모델에 기초한 여러 가지 보상 함수를 정의함으로써 폭넓은 종류의 경쟁적 게임을 유도할 수 있음을 보인다.

6. 애니메이션 합성

상태-행위 모델, 전이 및 보상 함수가 정의되면 4 장의 방법에 의하여 주어진 상태 공간에 대한 값 함수 $V(s)$를 학습한다. 애니메이션 되는 캐릭터는 학습된 값 함수를 참조하여 실행 시간에 적은 계산 비용만으로 주어진 게임에 대한 최적의 전략에 따라 행동하게 된다. 먼저 두 캐릭터의 애니메이션 합성 과정을 살펴보도록 하자. 사용자는 두 캐릭터를 원하는 위치와 방향에 맞게 배치한 후 게임 시작을 명령한다. 두 캐릭터의 상대적 배치에 따른 현재 상태를 s 라고 하면, 각 캐릭터의 최적 전략은 다음의 식으로 구할 수 있다.

$$\pi_A = \arg\max_{\pi \in PD(A)} \min_{o \in O} \sum_{a \in A} \left(R_{s,a,o} + \gamma \sum_{s'} T_{s,a,o,s'} V(s') \right) \pi_a \quad (10)$$

$$\pi_O = \arg\max_{\pi \in PD(O)} \min_{a \in A} \sum_{o \in A} \left(-R_{s,a,o} - \gamma \sum_{s'} T_{s,a,o,s'} V(s') \right) \pi_o \quad (11)$$

$V(s)$ 는 첫 번째 캐릭터의 기대보상에 해당하므로, 제로섬 게임의 정의에 의하여 두 번째 캐릭터는 $-V(s)$에 대한 최소-최대 전략을 찾음에 주의할 필요가 있다. 식 (10)과 식 (11)은 값-반복을 수행할 때와 마찬가지로 식 (2)의 형태로 전환하여 선형 프로그래밍으로 해결한다.

이와 같이 얻어진 확률 분포 π_A 와 π_O 에 따라 두 캐릭터의 다음 행위 a와 o를 확률적으로 선택하고, 각 행위에 기록된 동작 세그먼트에 따라 해당 시간 동안 두 캐릭터의 자세를 변화시킨다. 동작 세그먼트의 재생이 끝나면 식 (9)에 의하여 상태를 전이하고, 다시 새로운 상태에 대한 최적의 행위를 찾는 과정을 반복한다. 이를 지속하면 주어진 게임에 대한 최적의 경쟁적 상호작용을 수행하는 두 캐릭터 애니메이션을 임의의 시간 동안 합성할 수 있다.

다수 캐릭터 애니메이션으로 확장하기 위한 한 가지 방법으로 본 연구는 두 개의 팀으로 구성된 게임 방식을 제안한다. 각 팀에는 임의의 개수만큼의 캐릭터가 포함될 수 있다. 매 시뮬레이션 단계마다, 첫 번째 팀에 속한 각 캐릭터는 두 번째 팀에 속한 캐릭터 중 일정 반경 내에 포함된 모든 캐릭터에 대하여 식 (6)을 적용하여 그 중 가장 높은 기대보상을 제공하는 캐릭터를 상대 캐릭터로

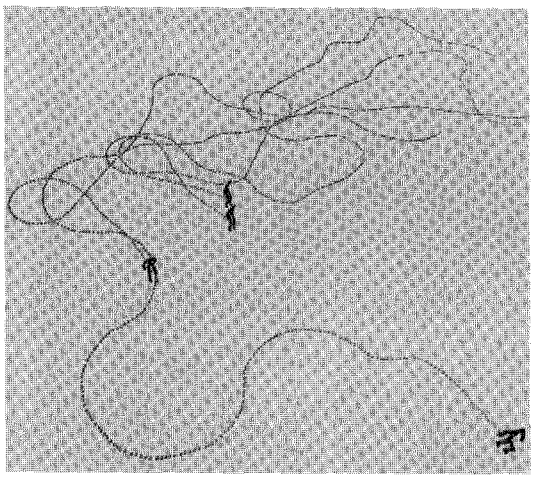

그림 5. 밀고 당기기의 반복에 의하여 생성된 복잡한 이동 궤적.

결정한다. 팀 간 경쟁을 캐릭터 간 경쟁으로 축소시켰으므로, 식 (10)에 의하여 최적의 행위를 구할 수 있다. 마찬가지로, 두 번째 팀에 속한 각 캐릭터 역시 첫 번째 팀에서 최대의 기대보상을 제공하는 캐릭터를 선택한 후 식 (11)을 적용하여 행위를 선택한다. 이와 같은 확장 방식은 실질적으로 두 캐릭터 간의 최적 전략에만 의존하기 때문에 팀 단위의 전략을 이끌어낼 수 없다. 하지만 7 장의 실험 결과에서 확인할 수 있듯이, 비교적 단순한 방법으로 그럴듯한 다수 캐릭터의 애니메이션을 생성하는 것이 가능하다.

7. 실험 결과

캐릭터 간의 다양한 경쟁적 상황을 이끌어내기 위하여, 서로 다른 이동 속도 및 회전 범위를 가진 총 4 개의 행위 집합을 구성하였다: '깡충깡충 뛰기'(HOP), '신나게 걷기'(JOY), '느릿느릿 걷기'(SLOW), '행진하듯 걷기'(MARCH). 각 행위 집합에는 서로 다른 방향으로 회전하는 5 개씩의 이동 동작이 포함되어 있고, 모든 동작은 동일한 자세에서 시작하고 종료되며 약 1 초에 해당하는 동일한 길이를 갖는다. 총 3 가지 종류의 게임에 대하여 이들 행위 집합을 적용하여 본 연구에서 제안하는 방법의 유효성을 확인하였다. 인텔 코어 2 CPU 3.2GHz 가 장착된 PC 에서, 하나의 게임을 학습하는데 소요된 시간은 약 30 분 정도였고, 모든 애니메이션 합성은 실시간에 이루어졌다.

■ **쫓고 쫓기기.** 한 캐릭터는 거리를 최소화, 상대 캐릭터는 거리를 최대화 하는 것을 목표로 하는 경쟁적 상황을 연출하기 위하여 다음과 같이 보상 함수를 정의하였다.

$$R_{s,a,o}^{CHASE} = \sqrt{x^2 + y^2} - \sqrt{x'^2 + y'^2} \tag{12}$$

여기에서 현재 상태는 (x, y, θ) 이고, 두 캐릭터의 동작에 의하여 전이된 다음 상태는 (x', y', θ') 이다. 총 4 가지 서로 다른 행위 집합의 조합인 'HOP 대 JOY', 'JOY 대 HOP', 'JOY 대 SLOW', 'SLOW 대 MARCH'에 대하여 학습을 수행하였다. 학습된 전략에 따라 시뮬레이션을 수행한 결과, 쫓기는 캐릭터의 이동 속도가 쫓는 캐릭터에 비하여 빠른 경우에는 추격에 성공하지 못하는 경우가 많았고 (그림 3), 'JOY 대 SLOW'의 조합에서는 대부분의 경우 추격에 성공함을 확인할 수 있었다 (그림 1). 또한 두 캐릭터 간의 경쟁을 두 팀 간의 경쟁으로 확장하여 다수 캐릭터 애니메이션을 합성하였다. 'JOY 대 SLOW'의 경우 쫓기는 팀의 캐릭터를 증가시킴으로써 추격의 어려움을 증가시켰고(그림 4), 'SLOW 대 MARCH'의 경우 쫓는 팀의 캐릭터를 증가시킴으로써 보다 빠른 시간 안에 추격이 이루어지는 장면을 생성할 수 있었다.

■ **밀고 당기기.** 쫓고 쫓기기의 보상 함수를 다음과 같이 약간 변형하여, 한 캐릭터는 특정 거리로부터 벗어나기 위하여 노력하고 상대 캐릭터는 해당 거리를 유지하기 위하여 노력하는 상황을 연출하였다. 이는 첫 번째 캐릭터가 다가서면 상대 캐릭터가 밀려나가고, 반대로 멀어지면 상대 캐릭터가 당겨지는 결과를 낳게 된다.

$$R_{s,a,o}^{SPRING} = \left|\sqrt{x'^2 + y'^2} - d\right| - \left|\sqrt{x^2 + y^2} - d\right| \tag{12}$$

여기에서 d 는 두 캐릭터가 벗어나거나 유지하고자 하는 서로 간의 거리에 해당한다. 이 보상 함수는 두 캐릭터 간의 애니메이션 시에는 쫓고 쫓기기와 크게 다르지 않은 양상을 유도하지만, 한 캐릭터가 벗어나려고 하고 다수의 캐릭터가 유지하고자 하는 경우 서로 간의 밀고 당기는 효과가 확연히 드러난다 (그림 5). 이는 지도자와 수행원으로 구성된 군중 장면 합성이나 일정 간격을 두고 공격을 수행하는 궁사와 같은 게임 캐릭터 제작 등에 활용 가능할 것으로 보인다.

■ **총 쏘기.** 컴퓨터 게임과 같이 명확한 경쟁관계가 드러나는 응용 분야에 대한 활용 예로서 시점 방향에 따라 주기적으로 자동사격을 하는 간단한 슈팅 게임을 상정하고, 이에 적합한 전략을 유도할 수 있는 보상 함수를 다음과 같이 정의하였다.

$$R_{s,a,o}^{SHOOT} = \frac{1}{\|v\|}\left(\frac{v \cdot v_1}{\|v \cdot v_1\|} + \frac{v \cdot v_2}{\|v \cdot v_2\|}\right) \quad (12)$$

전이된 다음 상태를 $s'=(x',y',\theta')$ 라고 하면, 여기에서 $v=(x',y')$ 는 첫 번째 캐릭터의 위치로부터 두 번째 캐릭터의 위치로 향하는 벡터, $v_1=(0,1)$ 은 첫 번째 캐릭터의 시점 벡터, $v_2=(\cos\theta',\sin\theta')$ 은 두 번째 캐릭터의 시점 벡터에 해당한다. 두 캐릭터 간의 거리가 동일할 경우, 이 함수의 값은 첫 번째 캐릭터가 두 번째 캐릭터를 정확히 조준할수록 커지고, 반대로 두 번째 캐릭터가 첫 번째 캐릭터를 정확히 조준할수록 작아진다. 따라서 첫 번째 캐릭터가 정확히 조준하고 두 번째 캐릭터가 완전히 등을 돌리고 있을 때 가장 큰 보상을, 정반대의 상황일 때는 그 음수에 해당하는 보상을 받게 된다. 이는 상대의 사정권에서 최대한 벗어나는 동시에 상대가 자신의 사정권 안에 들어오도록 이동하는 전략을 창출한다. 실제 슈팅 게임과 유사한 장면을 합성하기 위하여 탄환의 움직임 시뮬레이션 및 충돌 검사를 추가하여 다양한 장면을 연출할 수 있었다 (그림 6).

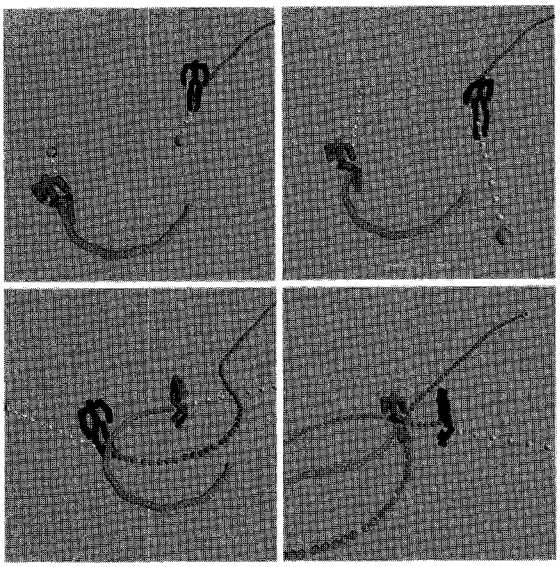

그림 6. 두 캐릭터의 총 쏘기 게임 시퀀스. 큰 회색 공이 탄환을 해당되고, 그 뒤를 잇는 작은 공들은 지나온 궤적을 나타낸다. 마지막 장면에서 빨간 색 캐릭터는 탄환에 맞아 동작을 멈춘 상태이다.

8. 결론 및 향후 연구

본 논문은 마르코프 게임 강화학습 알고리즘을 데이터 기반 캐릭터 애니메이션 기술에 적용함으로써 기존의 방법으로 생성하기 어려운 수준의 복잡하고 지능적인 다수 캐릭터 간 경쟁 애니메이션의 합성 방법을 제안하였다. 동일한 상태-행위 모델에 대하여 서로 다른 보상 함수를 정의함으로써 다양한 종류의 경쟁적 애니메이션을 합성할 수 있었다.

현재 구현된 시스템에서는 다수 캐릭터 애니메이션 합성 시 서로 간의 충돌이 발생할 수 있다. 이는 향후 캐릭터 간의 충돌 회피 계획이나 충돌 반응 시뮬레이션 등의 방법을 통합하여 해결 가능할 것으로 기대한다.

행위 모델의 단순성도 본 연구가 가진 제약에 속한다. 이는 상태 차원의 증가와 관련되어 있기 때문에 단순히 다수의 노드를 가진 그래프를 사용하여 해결 가능할 것으로 보이지는 않는다. 상태 공간의 크기 증가를 최소화하며 보다 다양한 동작의 합성을 가능케 하는 것은 향후 중요한 연구 과제이다. 매개화된 행위 모델을 사용하거나 동작 계획 알고리즘을 결합하는 방법은 이를 위한 출발점이 될 수 있을 것이다.

참고 문헌

[1] Littman, M. L., "Markov Games as a Framework for Multi-Agent Reinforcement Learning", In Proceedings of the 11th International Conference on Machine Learning, pp. 157-163, 1994.

[2] Gleicher, M., "Motion Editing with Spacetime Constraints", In Proceedings of the 1997 Symposium on Interactive 3D Graphics, pp. 139-ff, 1997.

[3] Lee, J., Shin, S. Y., "A Hierarchical Approach to Interactive Motion Editing for Human-like Figures", Proceedings of the 26th annual conference on Computer graphics and Interactive techniques, pp. 39-48, 1999.

[4] Popovic, Z., Witkin, A., "Physically Based Motion Transformation", Proceedings of the 26th annual conference on Computer graphics and Interactive techniques, pp. 11-20, 1999.

[5] Kovar, L., Gleicher, M., "Automated Extraction and Parameterization of Motions in Large Data Sets", ACM Transactions on Graphics, 23(3):559-568, 2004.

[6] Kovar, L., Gleicher, M., Pighin, F., "Motion Graphcs", ACM Transactions on Graphics, 21(3):473-482, 2002.

[7] Lee, J., Chai, J., Reitsma, P.S.A., Hodgins, J.K., Pollard, N.S., "Interactive Control of Avatars Animated with Human Motion Data", ACM Transactions on Graphics, 21(3):491-500, 2002.

[8] Gleicher, M., Shin, H.J., Kovar, L., Jepsen, A., "Snap-Together Motion: Assembling Run-Time Animations", In Proceedings of the 2003 Symposium on Interactive 3D graphics, pp. 181-188, 2003.

[9] Lee, J., Lee, K.H., "Precomputing Avatar Behavior from Human Motion Data", In Proceedings of the 2004 Symposium on Computer Animation, pp. 79-87, 2004.

[10] Treuille, A., Lee, Y., Popovic, Z., "Near-Optimal Character Animation with Continuous Control", ACM Transactions on Graphics, 26(3):7, 2007.

[11] Lo, W.-Y., Zwicker, M., "Real-Time Planning for Parameterized Human Motion", In Proceedings of the 2008 Symposium on Computer Animation, 2008.

[12] Zordan, V., Majkowska, A., Chiu, B., Fast, M., "Dynamic Response for Motion Capture Animation", ACM Transactions on Graphics, 24(3):697-701, 2005.

[13] Liu, C.K., Hertzmann, A., Popovic, Z., "Composition of Complex Optimal Multi-Character Motions", Proceedings of the 2006 Symposium on Computer Animation, pp. 215-222, 2006.

[14] Shum, H., Komura, T., Yamazaki, S., "Simulating Competitive Interactions Using Singly Captured Motions", In Proceedings of the 2007 ACM Symposium on Virtual Reality Software and Technology, pp. 65-72, 2007.

[15] Shum, H., Komura, T., Yamazaki, S., "Simulating Interactions of Avatars in High-Dimensional State Space", In Proceedings of the 2008 Symposium on Interactive 3D graphics and games, pp. 131-138, 2008.

[16] Shum, H., Komura, T., Shiraishi, M., Yamazaki, S., "Interaction Patches for Multi-Character Animation", ACM Transactions on Graphics, 27(5):114, 2008.

[17] Kwon, T., Cho, Y.-S., Park, S.I., Shin, S.Y., "Two-Character Motion Analysis and Synthesis", IEEE Transactions on Visualization and Computer Graphics, 14(3):707-720, 2008.

〈저자소개〉

이강훈

- 2000년 2월 서울대학교 컴퓨터공학과(학사)
- 2002년 2월 서울대학교 컴퓨터공학부(석사)
- 2007년 8월 서울대학교 컴퓨터공학부(박사)
- 2007년 10월~2008년 2월 서울대학교 컴퓨터공학부 박사후연구원
- 2008년 3월~현재 광운대학교 컴퓨터소프트웨어학과 조교수
- 〈관심분야〉 컴퓨터 그래픽스, 캐릭터 애니메이션, 컴퓨터 게임, 인간-컴퓨터 상호작용 등