

모션 패치

최명걸^o, 이강훈, 이제희

서울대학교 컴퓨터공학부

{mingle,zoi,jehee}@mri.ac.kr

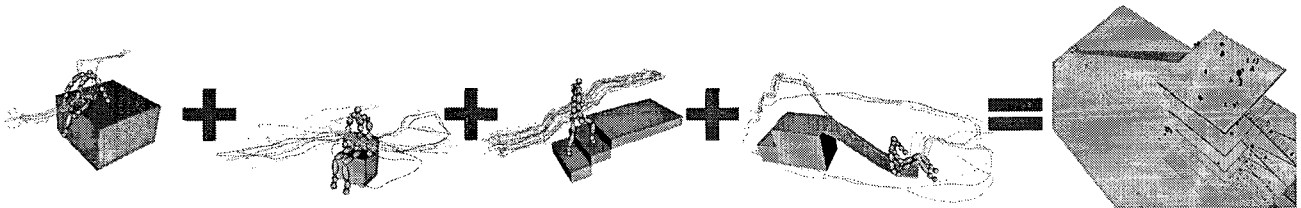
Motion Patch

Myung Geol Choi^o Kang Hoon Lee and Jehee Lee

School of Computer Science and Engineering, Seoul National University

요약

가상환경에서의 실시간 캐릭터 애니메이션은 컴퓨터 게임 및 가상현실 분야에서 중요한 문제이다. 최근에는 캐릭터 애니메이션의 사실성을 높이기 위해 대규모 동작 캡처 데이터를 활용하고 있다. 그러나 거대하고 복잡한 가상환경을 만들기 위해서는 현실 환경에서 동작을 캡처하고 이를 넓은 가상환경에 적용해야 하는 어려움이 따른다. 본 논문에서는 애니메이션 캐릭터가 몇 가지 빌딩 블록(building block)들로 건설된 넓은 가상공간에서 자유롭게 활동하게 하는 새로운 방법을 제안한다. 빌딩 블록들은 반복적인 형태로 배열되어 더 큰 환경을 생성할 수 있다. 우리는 이러한 각각의 블록들을 모션 패치(motion patch)라고 부른다. 모션 패치는 해당 블록 내부에서 캐릭터가 수행할 수 있는 동작들에 대한 정보를 가지고 있다. 우리는 다수의 캐릭터가 방대하고 복잡한 가상환경 속에서 실시간으로 애니메이션되고 제어되는 예제들을 통해 이 논문에서 제시하는 방법이 다양한 목적에 유용하게 사용될 수 있음을 보인다.



1. 서론

복잡한 가상환경에서의 실시간 캐릭터 애니메이션과 제어는 컴퓨터 그래픽스 분야에서 중요한 문제이다. 애니메이션 합성을 위해서는 다양한 기술들이 개발되어왔다. 근래에는 키프레임 보간, 물리 기반 시뮬레이션, 절차적 애니메이션, 동작 캡처 등이 많이 사용되고 있다. 우리는 특히 이 중에서 마지막 방법인 캡처된 동작 데이터베이스를 사용하는 데이터 기반 기법에 초점을 맞추고 있다. 데이터 기반 방식을 사용하면 실제 연기자의 동작을 캡처한 다량의 데이터를 활용하여 풍부하고 다양한 인체 동작을 만들어 낼 수 있다. 그러나 데이터 기반 방식을 복잡한 환경 내부의 캐릭터 애니메이션에 적용하기 위해서는 주어진 상황과 환경에 맞는 동작을 다량의 동작 데이터로부터 실시간에 검색해야 하고, 선택된 동작을 주어진 환경에 맞도록 변형해서 적용해야 하는 어려움이 따른다.

우리가 원하는 가상환경을 동작 캡처 스튜디오 안에 실물과 같은 형태와 크기로 설치하기에는 어려운 경우가 많다. 이처럼 큰 규모의 환경에 대한 동작 캡처는 아주 어렵거나 혹

은 불가능하다. 한 가지 방안은 하나의 대형 환경을 작은 조각들로 나눈 다음, 각각의 조각과 상호 작용하는 동작을 부분적으로 캡처하고, 이 동작 데이터들을 하나로 모아서 전체 환경을 포괄하는 커다란 구조로 만드는 것이다. 이 같은 방식으로 대형 환경에 대한 캐릭터의 동작 데이터를 얻을 수 있다. 그러나 이 방법은 각각의 부분 환경마다 많은 양의 동작 데이터가 필요하기 때문에 캡처 과정에 많은 노력이 필요하다. 또한 서로 다른 부분 환경에 동일한 동작들이 존재하기 때문에 전체 데이터에는 심각하게 많은 중복 정보가 포함된다. 그러나 이러한 중복성은 특정 장소에서 캡처된 데이터를 다른 장소에 대한 캐릭터 애니메이션에서 재사용함으로써 피할 수 있다.

이 논문에서는 애니메이션 캐릭터가 거대한 가상환경 내부에서 애니메이션을 수행하도록 하는 새로운 기술을 제안한다. 가상환경은 유한한 개수의 빌딩 블록들로 이루어져 있다고 가정한다. 이들 각각의 빌딩 블록을 모션 패치(motion patch)라고 부른다. 모션 패치는 해당 블록 내부에서 캐릭터가 수행할 수 있는 동작들에 대한 정보를 가지고 있다. 모션 패치들은 빌딩 블록에 해당하는 동작 세그먼트(motion seg-

ment)들로 구성되어 있으며 서로 부드럽게 연결될 수 있다. 빌딩 블록의 모양은 정사각형 타일처럼 규칙적일 수도 있고, 놀이터 기구의 일부처럼 불규칙적일 수도 있다. 규칙적인 구조물의 경우 반복적인 형태로 배치하면 더 큰 환경을 생성할 수 있다.

2. 배경

지난 10년 동안 컴퓨터 그래픽스 분야에서는 가상환경에서의 캐릭터 애니메이션과 제어에 대한 많은 연구가 이루어져왔다 [1, 2, 3, 4, 5]. 그러나 인체의 동작은 고차원적이며, 자유도가 높고, 충돌을 회피해야 하기 때문에 이는 쉽지 않은 문제이다.

캐릭터 애니메이션 분야에서는 자연스러운 인간의 동작을 만들기 위해서 동작 캡처 데이터가 널리 사용되어왔다. 캐릭터의 애니메이션과 제어를 위해 최근 몇몇 연구자들은 방대한 양의 동작 데이터를 분석하고 표현하는 방법을 연구하고 있다 [6, 7, 8, 9, 10]. 특히 이제희 등[8]은 동작 데이터로부터 미로, 거친 지형, 놀이터 등의 복잡한 환경에서의 캐릭터 애니메이션을 합성하고 제어하는 것을 보여주었다. 나아가 이 방법은 더 나은 제어 능력[11, 12, 13], 효율적인 탐색[14], 동작의 혼합과 매개변수화[15, 16], 음악과의 동기화[17], 매개변수 조절[18], 특정 환경에서의 효율성 평가[19] 등 다양한 연구에 접목되었다.

이 논문에서 다루는 문제는 장애물이 있는 환경에서의 캐릭터 경로 계획(path planning)과 관련이 있다. 경로 계획 문제는 여러 분야에서 연구되어온 고전적인 문제이기 때문에 많은 연구결과[20]들이 있다. 그러나 그 중 캐릭터에 관련된 결과는 극히 일부이다. 일부 연구자들은 고차원 공간에서의 경로 계획 문제를 해결하기 위해 무작위 기법(randomized techniques)을 사용하였다 [21, 22, 23]. 이 방법은 걷기, 살금살금 걷기, 오르기 등의 캐릭터 애니메이션에 성공적으로 적용되었다 [24, 25, 26, 27, 28]. 또한 물체를 잡거나 조작하는 동작에도 효과적으로 사용되었다[29, 30]. 대부분의 최적화 경로 계획 알고리즘은 모든 상태 공간을 빠짐없이 탐색하는 계산 집약적 알고리즘이거나 상태 공간을 이산화하여 저장하는 메모리 집약적 알고리즘이다. 모션 패치는 적은 메모리를 사용하여 상태 공간을 이산화하면서도 캡처된 데이터의 미세함과 다양성을 유지하는 방법이다. 이 방법은 성만규 등 [31]이 제안했던 캐릭터 동작의 목록을 포함하고 있는 환경을 이용한 군중 시뮬레이션과 관련 있다.

컴퓨터 그래픽스 분야에서 텍스처나 지형의 타일링(tiling)에 대한 연구는 광범위하게 이루어져왔다. 특히 정사각형 타일은 타일링하기에 편하고, 표현과 조작도 쉽기 때문에 많이 사용되어왔다. 스타만(Stam) [32]과 코헨(Cohen) 등[33]은 비대칭적 텍스처를 만들기 위해 왕-타일(Wang-tile)을 사용하였다. 체니(Cheney)[34]는 속도 정보로 구성된 정사각형 타일을 만들고, 이것이 도심 거리의 군중 시뮬레이션, 강의 흐름, 안개의 소용돌이 등의 예제에 유용하게 사용되는 것을 보였다. 레이츠마(Reitsma)와 폴라드(Polard) [19]는 거대한 환경에 동작 데이터를 내장하기 위해 정사각형의 타일의 사용을 제안하였다. 그리고 이러한 정사각형이 반복적으로 배치된

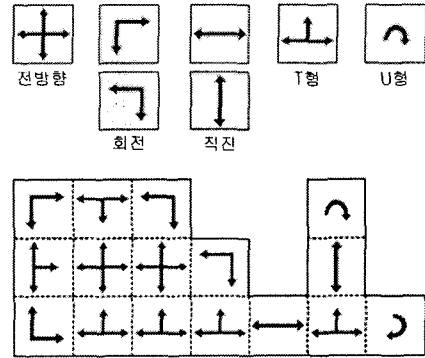


그림 1: (상)여러 형태의 타일들. 필요에 따라 더 많은 타일들이 추가될 수 있다. (하)정사각형의 타일은 넓은 평지나 도로망을 건설하는데 쓰일 수 있다.

환경 위를 걸어가는 캐릭터를 보여주었다. 모션 타일(즉, 정사각형의 모션 패치)은 이들 연구결과로부터 영향을 받았다. 본 논문에서는 경계 조건과 안전성을 충족하면서 동작 데이터를 정사각형 타일에 포함시키는 문제를 다룬다. 또한 비규칙적인 3차원 모션 패치를 생성하는 문제와 이것을 정사각형 타일과 연결하는 문제를 다룰 것이다.

3. 정사각형 모션 타일

2차원 정사각형 타일은 넓은 땅이나 도시의 도로망 등의 환경을 만드는 빌딩 블록으로 사용될 수 있다. 우리의 목표는 동작 데이터를 포함하는 정사각형 타일을 만드는 것이다. 이 타일을 반복적으로 배치하면 자동으로 자연스럽게 연결된 동작들이 나타나야 한다. 가장자리에서 외부로 벗어나는 일이 없도록 하기 위해서는 그림 1과 같이 다양한 형태의 타일들이 필요하다. 우리는 캐릭터가 모든 방향으로 나가고 들어올 수 있는 전방향(omni-directional) 타일을 만드는 과정을 설명할 것이다. 다른 형태의 타일은 비슷한 방법으로 만들 수 있거나 전방향 타일로부터 불필요한 동작 세그먼트를 제거함으로써 만들 수 있다.

정사각형 모션 타일의 네 개 경계면 위에는 여러 개의 입구와 출구가 존재한다. 이때 모든 입구는 맞은편 경계면 위에 그에 대응되는 출구가 있어야 하고, 모든 출구는 맞은편 경계면 위에 그에 대응되는 입구가 있어야 한다. 이 제약조건을 만족시키면 모션 타일을 연속해서 배치하는 것이 가능하다. 본 연구에서는 주어진 동작 데이터로부터 모션 타일을 만들어내는 과정을 세 단계로 나누었다. 먼저 입구와 출구를 추출하고, 다음으로 입구와 출구를 연결하는 동작 세그먼트들을 찾는다. 마지막으로 막다른 종료 지점(dead-end)의 발생을 피하기 위해 불필요한 동작 세그먼트들을 제거한다. 동작 데이터는 연속된 프레임 정보 외에 다른 정보는 가지고 있지 않다. 하나의 프레임은 캐릭터가 특정 시점에 취하는 자세와 대응된다. 우리는 동작 간의 자연스러운 연결을 확보하기 위해 연기자의 연속된 움직임 비교적 긴 시간 단위로 캡처했다.

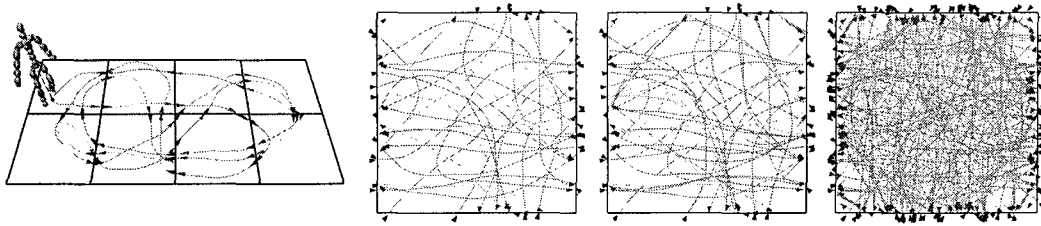


그림 2: 정사각형 모션 타일 만들기. (좌)캡처된 걷기 동작의 중심 궤적 위에 타일 격자를 겹쳐 놓는다. 붉은 화살표와 푸른 화살표는 각각 타일의 출구와 입구를 나타낸다. (중.좌)모든 입구/출구와 연결 동작들을 하나의 타일 위에 모은다. (중.우)모션 데이터에서 새로운 연결 동작들을 찾아 추가한다. 여기서는 그림으로 표현하기 위하여 모션 데이터의 일부분만을 사용하였기 때문에 연결 동작이 충분히 풍부 않다. (우)모든 입구/출구와 연결 동작들을 90도씩 회전하면서 복사한다.

입구/출구 추출. 입구와 출구에는 위치, 수평방위, 캐릭터 자세 등의 정보가 저장된다. 입구와 출구의 추출을 위해서 몇 가지 방법을 생각해 볼 수 있다. 위치와 수평방위는 규칙적으로 분포시킬 수도 있고 무작위로 분포시킬 수도 있을 것이다. 우리는 동작 데이터에 존재하는 자연스러운 이동 경로를 따라 입구와 출구를 추출하는 방법을 사용했다. 먼저 주어진 동작 데이터의 루트 세그먼트를 이용해 2차원 궤적을 만들고 그것을 타일 격자위에 올려놓는다(그림 2). 이 궤적이 격자의 경계선을 지나는 교차점들을 찾아내고 그 지점에서 한 쪽 타일에는 입구를, 다른 쪽 타일에는 출구를 추가한다. 이렇게 만들어진 모든 입구와 출구를 하나의 타일 위로 모은다. 필요한 경우 90도씩 회전함으로써 입구와 출구를 추가적으로 얻을 수 있다.

연결 동작의 생성. 다음 단계는 입구와 출구를 잇는 풍부한 양의 연결 동작들을 찾는 것이다. 동작 데이터를 탐색하여 하나의 입구로부터 시작하여 하나의 출구로 끝날 수 있는 동작을 연결 동작으로 선택한다. 임의의 동작 세그먼트가 입구와 출구를 잇는 연결 동작이 되기 위해서는 먼저 동작 세그먼트의 첫 번째 자세와 마지막 자세가 각각 입구와 출구에 해당하는 자세와 충분히 비슷하여야 한다. 또한 이 동작 세그먼트의 양 끝 지점들은 들어오고 나가는 곳의 위치와 방향에 맞게 적절히 수정될 수 있어야 한다.

존재하는 모든 연결 동작들을 찾아내기 위해서는 동작 데이터 내에 있는 모든 동작 세그먼트를 조사해야 한다. 이 과정은 간단하지만 많은 계산이 요구된다. 우리는 효율적인 계산을 위해 동작 데이터의 각 프레임에 대해서 그와 비슷한 프레임들을 미리 찾아두었다. 두 프레임간의 비슷한 정도를 측정하기 위해서는 이제희 등[8]이 사용했던 각 관절의 각도와 속도를 비교하는 방법을 사용하였다. 미리 계산된 비슷한 프레임들에 대한 정보를 이용하면 입구/출구에 대응되는 자세를 쉽게 확인할 수 있기 때문에 효율적으로 연결 동작들을 찾아낼 수 있다.

타일내의 연결 동작은 다양한 동작을 충분히 만들어낼 수 있을 만큼 풍부해야 한다. 연결의 다양성은 동작 데이터의 크기뿐 아니라 동작 세그먼트를 수정할 수 있는 한계치의 크기에도 많은 영향을 받는다. 우리는 동작 경로의 수정을 위해 레이츠마(Reitsma)와 폴라드(Pollard) [19]가 사용했던 간단한 선형 모델을 사용하였다. 이 선형 모델에서는 중점의

위치와 방위에 대한 변동 범위가 동작 세그먼트의 시작에서 마지막으로 갈수록 선형적으로 변한다. 마지막 부분에서의 총 변동 가능 범위는 동작 세그먼트 전체의 길이에 비례한다.

가지치기. 모션 타일은 다음 두 조건이 만족될 때 **안전성**을 유지할 수 있다. 첫 번째 조건은 모든 입구가 네 방향 각각에 대해서 최소 하나 이상의 연결을 가져야 한다는 것이다. 우리는 많은 양의 동작 데이터를 사용했지만 타일의 모서리로 향하고 있는 입구에 대해서는 충분한 연결 동작을 찾을 수 없었다. 캐릭터의 유연한 조작용을 위해 이와 같은 입구와 그에 대응되는 출구는 미리 제거하였다. 두 번째 조건은 모션 타일들이 서로 이어질 때 막다른 종료 지점이 발생하지 않아야 한다는 것이다. 모션 타일에서 마주보는 두 변들을 서로 붙이면 도넛 형태가 된다. 이 도넛 모양 위에 놓인 동작 세그먼트들은 말단 노드(leaf-node)가 없는 유향 그래프 형태가 된다. 이 그래프 내에서 임의의 두 노드가 서로 도달 가능하면 모션 타일은 막다른 종료 지점을 만들지 않는다. 우리는 스트롱리 커넥티드 컴포넌트(strongly connected component) 알고리즘 [35]을 적용하여 이 조건을 만족하는 가장 큰 부분 그래프를 찾아냈다. 이와 같은 두 조건을 동시에 만족시키기 위해서 더 이상 제거되는 연결 동작이 없을 때까지 두 조건에 대한 검사를 번갈아가며 시행하였다.

타일의 크기. 타일의 크기를 결정할 수 있는 간단한 규칙은 없다. 하지만 경험을 통해 몇 가지 규칙을 발견하였다. 타일의 크기는 캡처된 동작 데이터의 내용에 따라 결정되어야 한다. 모션 타일은 동작 데이터 내의 하나의 인지 가능한 행위를 담기에 충분할 만큼 커야 한다. 예를 들어, 걷기 모션 데이터를 이용하여 만든 모션 타일을 만들 때 우리는 가로, 세로 길이를 두 걸음 정도의 크기와 같게 했다. 작은 타일과 큰 타일에는 각각의 장단점이 있다. 크기가 작은 타일은 타일 내에 풍부한 연결 동작을 만들기 어렵지만, 큰 타일은 다양한 가상환경을 구성하기가 어렵다.

서로 다른 동작 데이터의 타일링. 서로 다른 동작 데이터를 사용하면 “걷기” 타일, “달리기” 타일, “발끝으로 걷기” 타일 등의 다양한 타일을 만들 수 있다. 이와 같이 여러 종류의 타일을 사용하면 가상환경의 위치에 따라 다양한 패턴의 동작이 나타나게 할 수 있다. 일반적으로 이들 타일은 서로 다른 형태의 입구와 출구를 가지기 때문에 경계면을 그대로 이어 붙이는 것은 불가능하다. 서로 다른 종류의 타일을 연결하기

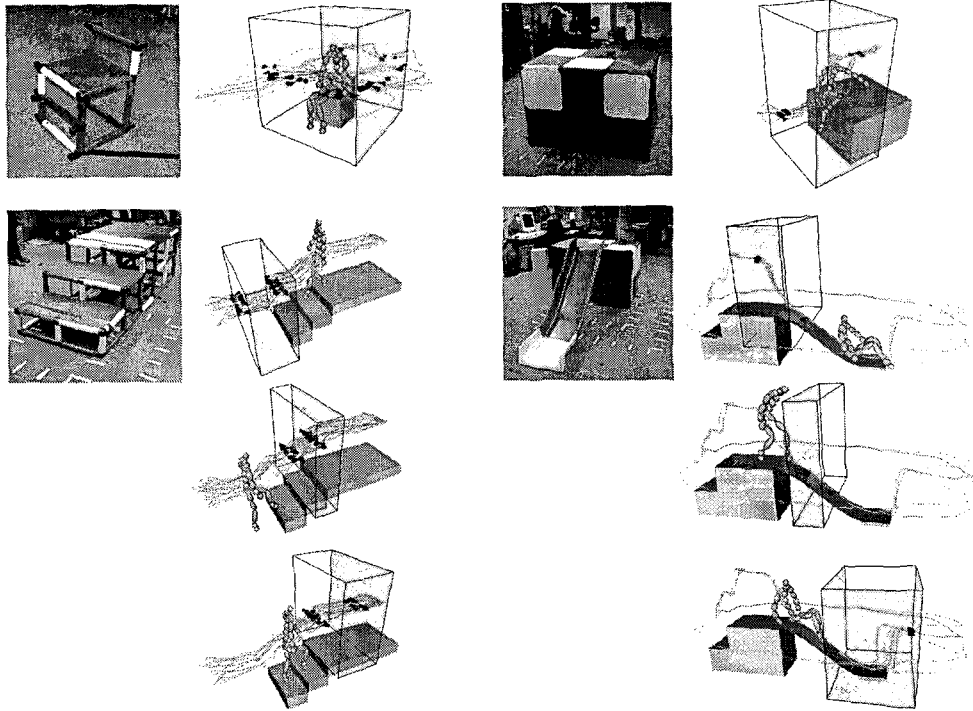


그림 3: 3차원 모션 패치

위해서 두 타일의 경계면을 이어주는 전이(transition) 타일을 만든다. 전이 타일 내의 연결 동작은 두 개의 동작 데이터에서 선택된 동작 세그먼트들을 혼합(blending)하여 생성하였다. 예를 들어 “걷기” 타일과 “달리기” 타일을 연결하는 전이 타일의 한 쪽 면에는 “걷기” 타일의 입구/출구가 있고 다른 면에는 “달리기” 타일의 입구/출구가 있다. 각각의 입구-출구 쌍에 대해 “걷기”와 “달리기” 동작 세그먼트를 하나씩 혼합하여 연결을 만들면 “걷기” 입구에서 “달리기” 출구로의 자연스러운 이동이 이루어진다. “달리기”에서 “걷기”로의 이동도 같은 방법으로 만들 수 있다.

4. 3차원 모션 패치

4장에서는 3차원 구조물에 대한 동작 데이터를 추가하는 방법에 대해서 논의한다. 우리는 3차원 구조물을 한 방향으로 확장 가능한 구조물과 확장이 불가능한 구조물로 분류하였다. 사다리, 계단, 미끄럼틀과 같이 한 방향으로 확장 가능한 구조물에서는 캐릭터가 동작이 일정한 패턴을 가지고 반복된다. 따라서 동작을 일반화시킨 다음 한 방향으로 늘리거나 줄이는 것이 가능하다. 확장 불가능한 구조물의 경우에는 이러한 일반화가 불가능하다.

확장 불가능 패치. 모션 타일로 이루어진 바닥 위에 의자나 책상과 같은 확장 불가능 구조물이 놓여있는 사무실을 예로 들어보자. 캐릭터는 바닥 타일에 포함되어 있는 동작 데이터를 이용해서 사무실 내부를 돌아다닌다. 그러던 중 캐릭터가 의자가 놓인 블록으로 들어가면 캐릭터는 상태를 바꾸

어 의자 모션 패치에 포함된 동작 데이터를 애니메이션하게 된다.

우리가 데이터를 캡처한 동작 캡처 스튜디오는 가로 5미터, 세로 5미터, 높이 2.5미터의 영역만 캡처 가능하다. 따라서 본 논문의 실험에 사용된 모든 확장 불가능 구조물은 이 영역 내에 설치할 수 있을 정도의 크기를 갖는다. 또한 이 영역 안에는 구조물과 상호 작용하는 동작이 충분히 이루어 질수 있을 만큼의 여유 공간도 포함되어야 한다. 이보다 더 큰 구조물의 경우에는 작은 부분으로 나누어서 캡처해야 한다. 동작 데이터를 캡처하기 위해서 먼저 스튜디오 내부에 구조물을 설치하였다. 그리고 연기자가 해당 구조물에 다가가서 상호작용을 한 다음 다시 빠져나오는 동작을 여러 번 반복 수행하였다. 이렇게 캡처된 동작 데이터를 이용하여 모션 패치를 만든다. 모션 패치는 구조물을 둘러싸는 박스(box)에 포함되는 형태이다. 우리는 동작 데이터를 프레임 단위로 검사하면서 캐릭터의 중심 궤적이 박스의 경계면을 통과하는 지점을 찾아 그 곳에 입구와 출구를 추가하였고, 새로운 입구-출구 쌍이 추가되는 지점마다 그 사이를 지나는 동작 세그먼트를 가져와 모션 패치에 포함시켰다.

한 방향 확장가능 패치. 한 방향으로 반복적인 모양을 하고 있는 구조물은 양 끝 부분과 반복 가능한 가운데 부분으로 나눌 수 있다. 한 칸씩 오르는 계단이 이러한 구조의 전형적인 예이다(그림 3). 계단을 반복적으로 오르내리는 동작 데이터가 있을 때 다음의 과정을 통해 모션 패치를 만들 수 있다. 먼저 양 외곽 부분은 확장 불가능 구조물과 같은 방법으

표 1: 프레임 수는 각 실험에서 사용된 초기 동작 데이터의 크기이다. 입구 수는 출구의 수와 동일하다. 연결 동작의 총 수는 불안정한 동작을 제거한 후의 수이다. 계단과 미끄럼틀의 입구 수와 연결 수는 외부 패치와 가운데 패치를 따로 계산하였다.

	크기 (미터)	프레임 수	입구 수	연결 수
타일 (걸기)	1.6 × 1.6	21084	516	18988
(달리기)	1.6 × 1.6	12933	608	6716
계단	1.0 × 2.4 × 2.4	9503	31/62/31	31/326/31
미끄럼틀	0.4 × 0.8 × 1.6	5994	7/14/7	7/144/7
의자	4.5 × 0.8 × 4.5	5836	20	20
육면체	1.6 × 0.8 × 1.6	2505	11	11

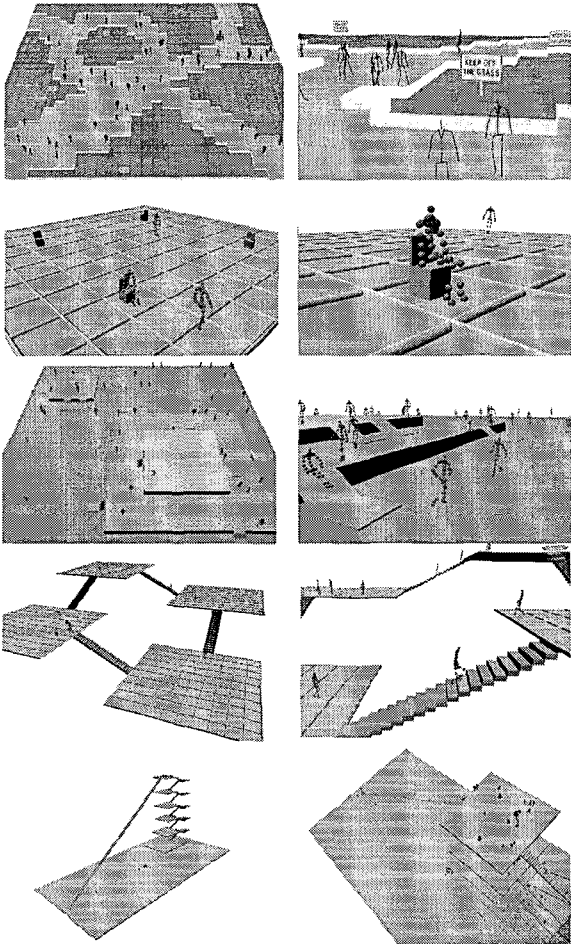


그림 4: 모션 패치로 만들어진 가상환경들. 왼쪽 열은 멀리서 본 장면이고 오른쪽 열은 가까이서 본 장면이다. (위에서부터) 잔디 광장, 의자, 놀이터, 바닥과 계단, 거대한 미끄럼틀

로 패치를 만든다. 가운데 부분은 두 개의 경계면만을 가지는 3차원 타일로 볼 수 있다. 따라서 2차원 모션 타일을 만들 때 사용한 알고리즘을 일반화하여 적용하면 이 부분에 해당하는 모션 패치를 만들 수 있다. 주의해야 할 것은 이렇게 만든 두 외곽 부분의 패치와 가운데 부분의 패치는 일반적으로 경계면에서의 입구/출구의 구조가 서로 다르다는 점이다. 따라서 부드러운 타일링을 위해서 가운데 부분의 패치에 추가적인 입구/출구를 만들어 주어야 하고 막다른 종료 지점을 피하기 위해 일부의 입구/출구를 제거해야 한다. 우리는 앞에서 설명한 전이 타일과 유사한 전이 패치를 만들어 외부 패치와 가운데 패치를 부드럽게 연결하였다.

타일과 패치의 연결. 모션 타일과 패치는 서로 어긋난 방향으로 연결할 수 있다. 타일과 패치 사이의 연결 동작을 만들기 위해 먼저 서로 보이는 위치에 모든 입구와 출구의 쌍을 찾는다. 그리고 3장에서 설명한 것과 같은 방법으로 입구-출구 쌍의 연결 동작을 찾아낸다. 이 과정은 빠르게 이루어지기 때문에 사용자는 실시간으로 모션 패치를 임의의 타일 위에 올려놓을 수 있다.

연결 동작 사이의 이동. 입구와 출구 사이의 연결 동작이 풍부할수록 캐릭터의 움직임은 부드러워진다. 각 패치의 출구에서 캐릭터가 선택할 수 있는 동작은 다양하다. 그러나 동작을 선택한 캐릭터가 패치의 내부로 들어가면 일시적으로 캐릭터에 대한 제어권을 잃게 된다. 일단 정해진 동작은 캐릭터가 출구에 도착할 때까지 변화시킬 수 없다. 이제희 등[8]은 위치, 방향, 자세가 유사한 프레임들을 연결하여 모션 그래프(motion graph)를 생성하는 알고리즘을 소개하였다. 이 알고리즘을 이용하여 패치에 포함되어있는 동작들 상호간에 연결을 만들면 캐릭터가 출구에 도착하기 전에도 방향을 변화시키는 것이 가능하다.

5. 실험 결과

실험에 사용된 모든 동작 데이터는 바이콘 광학 시스템(Vicon optical system)에서 초당 120프레임으로 캡처하였다. 그리고 실시간 출력을 위해 초당 30프레임으로 다운샘

플링(down-sampling)하여 사용하였다. 동작 캡처 데이터는 중심점(골반)의 위치와 방향, 인체 각 관절의 상대적인 각도(joint angle)를 가지고 있다.

정사각형 타일. 우리는 가로 3미터, 세로 5미터의 캡처 장소에서 12분 정도의 걷는 동작을 캡처하였다. 그리고 캡처 영역을 가로 2칸, 세로 8칸의 타일 격자 위에 올려놓고 입구와 출구를 추출하였다. “걷기” 타일은 경계면 위에 각각 516개의 입구와 출구를 가지고 있고 18988개의 연결 동작들을 포함한다. 또한 같은 장소에서 7분 가량의 달리는 동작을 캡처하여 “달리기” 타일을 만들었다. “달리기” 타일에는 12933프레임의 캡처 동작이 사용되었고 608개의 입구와 출구가 있으며 6716개의 연결 동작들이 있다.

가상환경의 대화형 편집. “잔디 보호”(Keep-off-the-grass) 예제에서는 걷기 타일, 달리기 타일, 녹색의 걷기 타일 등 세 가지 종류의 정사각형 타일을 배치했다. 군중 캐릭터들이 타일 위에서 동작하고 있을 때 사용자는 바닥의 종류를 즉각적으로 바꿀 수 있다. 녹색 타일위에 있는 캐릭터들은 강제적으로 녹색 타일로부터 빠져나가도록 되어있다. 군중의 행동은 타일 해상도(resolution) 범위에서 계획과 애니메이션이 이루어진다.

의자에 해당하는 모션 패치를 만들기 위해서 연기자가 의자로 다가가서 앉은 다음 다시 일어서서 걸어 나오는 동작들을 캡처했다. 의자는 사용자에게 의해 타일 바닥 위 어느 곳에나 놓일 수 있다. 의자 패치는 부드럽게 타일과 연결된다. 실험 결과를 통해 타일과 의자 사이의 연결이 비교적 짧은 시간 내에 이루어짐을 확인했다. 펜티엄4 2.4 GHz 컴퓨터에서 의자와 타일의 연결에 필요한 계산 시간은 1~2초 정도이다.

놀이터. 놀이터 예제에서는 놀이터 환경을 구축하기 위해 정육면체의 구조물과 미끄럼틀을 사용하였다. 정육면체 구조물에 대한 모션 패치를 만들기 위해서 기어오르고 뛰어내리는 동작들을 약 90초 정도 캡처하였다. 미끄럼틀 패치는 3분 정도의 동작 데이터를 사용해서 만들었다.

대형 미끄럼틀. 마지막 예제에서는 극도로 높은 미끄럼틀과 계단들을 조합하여 확장 가능한 패치의 사용을 보여주었다. 적은 개수의 모션 패치만으로 거대하고 복잡한 가상 환경을 만들 수 있었고 수백 명의 캐릭터들이 그 위를 실시간으로 움직이도록 했다.

6. 토의

모션 패치는 거대한 가상환경에서 많은 수의 캐릭터를 실시간으로 제어하고 애니메이션해야 하는 응용분야에서 특히 유용할 것이다. 모션 패치는 단순하며 구현이 쉽고 다양한 목적으로 활용될 수 있다. 모션 패치의 가장 중요한 장점은 동작 데이터의 크기와 가상환경의 복잡도 증가에 따라 무리없이 확장할 수 있다는 것이다. 컴퓨터 게임 분야에서는 모션 패치의 다양한 기능과 간결함이 도움이 될 것이다. 많은 컴퓨터 게임 캐릭터들은 오직 네 가지 또는 여덟 가지 방향으로만 움직이는 것을 볼 수 있다. 모션 패치를 활용할 경우 비록 패치들은 격자위에 규칙적으로 배열되지만 내부의 풍

부한 연결을 통해 다양한 동작들이 표현 가능하다.

본 논문에서 소개한 빌딩 블록을 이용하면 폭넓고 다양한 가상환경을 생성할 수 있다. 그러나 실제계를 모델링하는데 유용하게 쓰일 수 있는 다른 종류의 구조물들이 존재한다. 삼각형 모션 패치가 좋은 예이다. 삼각형 메쉬(mesh)는 지형을 만들기 위해 널리 쓰이고 있으므로 삼각형 모션 패치는 기존의 시스템과 병합하기가 쉬울 것이다. 현재의 동작 캡처 장비들은 물에서의 움직임이나 공중을 날아다니는 동작은 캡처할 수 없다. 미래에 그러한 동작 데이터를 만들 수 있게 되면 육면체나 사면체의 패치를 공간적으로 연결하는 방법도 고려할 수 있을 것이다.

본 논문에서 제안한 방법의 중요한 단점은 물리적인 법칙을 무시했다는 점이다. 매우 긴 미끄럼틀을 내려오는 캐릭터의 동작은 짧은 미끄럼틀에서 캡처된 동작을 반복 연결해서 만들어진 것이기 때문에 적절한 가속도가 발생하지 않는다. 데이터 기반 기술과 물리 기반 기술의 결합은 미래의 연구 방향에 많은 가능성을 열어 줄 것이다.

참고 문헌

- [1] S.Bandi and D.Thalmann. A configuration space approach for efficient animation of human figures. In *Proc. of IEEE Non Rigid and Articulated Motion Workshop*, IEEECS Press. 1997.
- [2] R.Bindiganavale, J.P.Granieri, S.Wei, X.Zhao and N.I.Badler. Posture interpolation with collision avoidance. In *Proceedings of Computer Animation '94*, 13-20, 1994.
- [3] M.R.Jung, N.I.Badler and T.Noma. Animated human agents with motion planning capability for 3d-space postural goals. *The Journal of Visualization and Computer Animation* 5, 4, 225-246, 1994.
- [4] H.Noser, O.Renault, D.Thalmann and N.M.Thalmann. Navigation for digital actors based on synthetic vision, memory, and learning. *Comput. & Graphics* 19, 1, 7-19. 1995
- [5] M.Throne, D.Burke and M.van de Panne. Motion Doodles: an interface for sketching character motion. *ACM Transactions on Graphics (SIGGRAPH 2004)* 23, 3, 424-431, 2004.
- [6] O.Arikan and D.A.Forsyth. Interactive motion generation from examples. *ACM Transactions on Graphics(SIGGRAPH 2002)* 21, 3, 483-490, 2002.
- [7] L.Kovar, M.Gleicher and F.Pighin. Motion graphs. *ACM Transactions on Graphics (SIGGRAPH 2002)* 21, 3, 473-482. 2002.
- [8] J.Lee, J.Chai, P.S.A.Reitsma, J.K.Hodgins and N.S.Pollard. Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics (SIGGRAPH 2002)* 21, 3, 491-500, 2002.

- [9] K.Pullen and C.Bregler. Motion capture assisted animation: Texturing and synthesis. *ACM Transactions on Graphics (SIGGRAPH 2002)* 21, 3, 501-508, 2002.
- [10] A.Schödl, R.Szeliski, D.H.Salesin and I.Essa. Video textures. In *Proceedings of SIGGRAPH 2000*, 489-498, 2000.
- [11] O.Arikan, D.A.Forsyth and J.F.O'Brien. Motion synthesis from annotations. *ACM Transactions on Graphics (SIGGRAPH 2003)* 22, 3, 402-408, 2003
- [12] E.Hsu, S.Gentry, and C.J.Popovi. Example-based control of human motion. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 69-77, 2004.
- [13] M.Stone, D.Decarlo, I.Oh., C.Rodriguez, A. Stere, A. Lees, and C.Bregler. Speaking with hands: creating animated conversational characters from recordings of human performance. *ACM Transactions on Graphics (SIGGRAPH 2004)* 23, 3, 506-513, 2004.
- [14] J.Lee, and K.H.Lee. Precomputing avatar behavior from human motion data. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 79-87, 2004.
- [15] L.Kovar, and M.Gleicher. Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics (SIGGRAPH 2004)* 23, 3, 559-568, 2004.
- [16] S.I.Park, H.J.Shin, T.Kim, and S.Y.Shin. On-line motion blending for real-time locomotion generation. *Computer Animation and Virtual Worlds* 15, 3, 125-138, 2004
- [17] T.Kim, S.I.Park, and S.Y.Shin. Rhythmic-motion synthesis based on motion-beat analysis. *ACM Transactions on Graphics (SIGGRAPH 2003)* 22, 3, 392-401, 2003.
- [18] J.Wang, and B.Bodenheimer. Computing the duration of motion transitions: An empirical approach. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/ Eurographics symposium on Computer animation*, 337-346, 2004.
- [19] P.S.A.Reitsma, and N.S.Pollard. Evaluating motion graphs for character navigation. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/ Eurographics symposium on Computer animation*, 89-98, 2004
- [20] J.C.Latombe. Robot Motion Planning. *Kluwer Academic Publishers*. 1991.
- [21] J.Barraquand and J.C.Latombe. Robot motion planning: A distributed representation approach. *International Journal of Robotics Research* 10, 6, 628-649, 1991.
- [22] L.E.Kavraki, P.Svestka, J.C.L., and M.Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation* 12, 4, 566-580, 1996.
- [23] S.LaValle, and J.Kuffner. Randomized kinodynamic planning. *International Journal of Robotics Research* 20, 5, 378-400, 2001.
- [24] M.G.Choi, J.LEE, and S.Y.Shin. Planning biped locomotion using motion capture data and probabilistic roadmaps. *ACM Transactions on Graphics* 22, 2, 182-203, 2003.
- [25] M.Kalisiak, and M.Van De Panne. A grasp-based motion planning algorithm for character animation. *The Journal of Visualization and Computer Animation* 12, 3, 117-129, 2001.
- [26] J.J.Kuffner, K.Nishiwaki, S.Kagami, M.Inaba, and H.Inoue. Footstep planning among obstacles for biped robots. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'01)*, 2001.
- [27] F.Pette, J.-P.Laumond, and T.Simeon. A 2-stages locomotion planner for digital actors. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/ Eurographics symposium on Computer animation*, 258-264, 2003.
- [28] Z.Shiller, K.Yamane, and Y.Nakamura. Planning motion patterns of human figures using a multi-layered grid and the dynamics filter. In *Proc. of the International Conference on Robotics and Automation 2001*, 2001.
- [29] Y.Koga, K.KondoONDO, J.Kuffer, and J.Latombe. Planning motions with intensions. *Proceedings of SIGGRAPH '94* 28 (July), 395-408, 1994.
- [30] K.Yamane, J.J.Kuffner, and J.K.Hodgins. Synthesizing animations of human manipulation tasks. *ACM Transactions on Graphics (SIGGRAPH 2004)* 23, 3, 532-539, 2004.
- [31] M.Sung, M.Gleicher, and S.Chenney. Scalable behaviors for crowd simulation. *Computer Graphics Forum (Eurographics 2004)* 23, 3, 519-528, 2004.
- [32] J.Stam. Aperiodic texture mapping. *Technical report R046, European Research Consortium for Informatics and Mathematics (ERCIM)*, 1997.
- [33] M.F.Cohen, J.Shade, S.Hiller, and O. Deussen. Wang tiles for image and texture generation. *ACM Transactions on Graphics (SIGGRAPH 2003)* 22, 3, 287-294, 2003.
- [34] S.Chenney. Flow tiles. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 233-242, 2004.
- [35] R.Tarjan. Depth first search and linear graph algorithms. *SIAM Journal of Computing* 1, 146-160, 1972.