3D 그래픽의 연결된 Joint 구조에서 비틀림이 없는 Up-Vector 연구

이상원O
O청강문화산업대학교 게임콘텐츠스쿨
e-mail: one@ck.ac.krO

Twist-free Up-Vector study in 3D graphic connected joint structure

SangWon LeeO
OSchool of Game, ChungKang College of Cultural Industries

● 요 약 ●

이 연구는 3D 그래픽의 로프 시뮬레이션이나 3D 캐릭터의 포니테일 등 Joint들이 복잡하게 꼬이는 상황에서Joint들의 Up-Vector에 대해 이상적으로 꼬이지 않는(Twist-free) 회전을 계산하는 연구이다. 꼬이지 않는 방향은 이상적인 기준점이 될 수 있어서 Twist 연출에 다양한 응용을 할 수 있는 시발점이 된다.

키워드: 로프 시뮬레이션 (Rope Simulation), 스플라인 IK (Spline IK)

I. Introduction

3D 그래픽의 연결된 Joint 구조는 다양한 경우에 사용되고 있다. 예를 들어 3D 게임에서 움직이는 캐릭터나 3D 로프 시뮬레이션(rope simulation) 결과를 시각적으로 표현하는 등의 경우가 있다. 좀 더 구체적으로는 캐릭터의 팔과 다리, 포니테일 머리카락, 시뮬레이션에 의해 흔들리는 밧줄 등이다. 팔이나 다리의 Joint 회전은 미리 결정되어 있는 경우가 대부분이다. 하지만 포니테일이나 로프 시뮬레이션은 Joint의 Twist 회전에 문제가 되는 경우가 많다.

Twist 회전 문제는 3ds-Max나 Maya 등의 DCC 툴에서도 위치에 의해 Spline의 형태가 결정되는 Spline-IK 방식의 Joint 구성에서 같은 문제가 발생한다.

본 연구에서는 부모 조인트(parent joint)의 Twist 방향에 의해 자식 조인트(child joint)의 Twist 방향을 계산하는 방법을 제시하며, 이 방법은 자식 조인트가 바라보는 Look-At 방향이 공간상의 모든 방향에 대해서 유효함을 보여준다.

II. Preliminaries

1. Related works

1.1 Unity-3D의 Character Joint

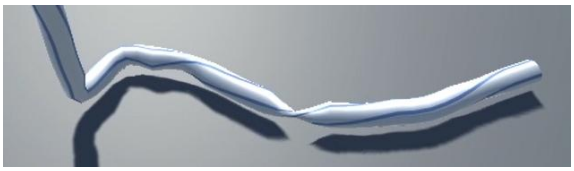

Fig. 1. Unity-3D의 Character Joint

그림 1은 Unity-3D의 Joint 시뮬레이션을 사용하여 로프 시뮬레이션을 수행한 결과이다. Twist 회전의 허용 한도를 최소로 설정하였음에도 Twist 현상을 피할 수 없다.

1.2 DCC 툴에서의 Spline-IK

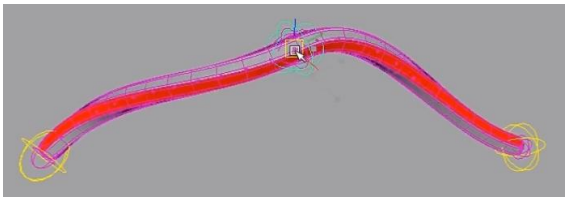

Fig. 2. Maya에서의 Spline-IK

그림 2는 Maya의 Spline-IK에서 수동으로 Twist 제어를 하는 모습이다.[1] 3ds Max나 Maya 등의 DCC 툴에서 Spline-IK의 Twist 처리는 대부분 애니메이터에 의해 수동으로 방향이 제어된다.

DCC 툴에서 Spline-IK의 Twist는 매듭처럼 복잡하게 꼬이는 형태를 표현할 때 각 Joint의 Twist 회전을 결정짓는 Up-Vector의 기준이 모호해지고 애니메이터가 수동으로 방향을 제어하는 기능을 추가하는 것이 일반적이다.

III. The Proposed Scheme

1. Motive

실제 전선의 경우 전체적으로 아무리 복잡하게 꼬이더라도 짧은 단위로 확대해서 살펴보면 부분적으로는 꼬이지 않은 상태임을 알 수 있다. 마찬가지로 수 많은 Joint들로 이어진 경우도 바로 위 부모와의 관계에서 Twist가 발생하지 않는다면 전체적으로도 Twist가 발생하지 않을 것이다.

2. Calculation

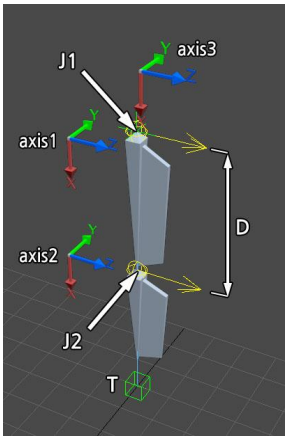

Fig. 3.

부모와 자식으로 연결된 두 Joint가 있다(J1, J2). J1과 J2의 길이는 같지 않다. 각 Joint의 방향은 빨강(X), 초록(Y), 파랑(Z)으로 표시되었다. J1의 방향은 axis1이고, J2의 방향은 axis2이다. D는 J1과 J2의 거리이다. T는 J2의 X 방향으로 D 거리만큼 떨어진 위치이다. axis3은 J1 위치에서 T를 바라보며 J1의 Z 방향과 Up 정렬되어있다.

이제 T의 위치가 변경될 때 J2의 Z축 Up 방향은 axis3의 Z 방향과 Up 정렬함으로써 얻을 수 있다.

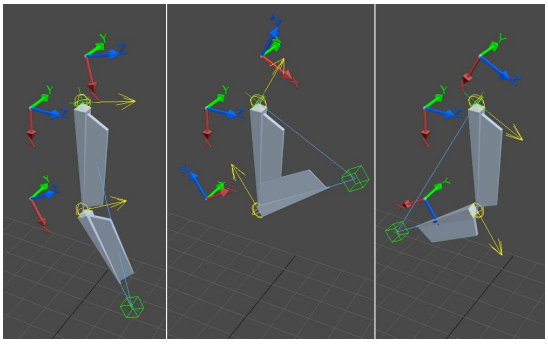

Fig. 4.

3. Example

다음 링크는 Unity 3D 엔진에서 연결된 Joint로 구성된 Rigidbody들을 대상으로 작동하는 코드의 예제이다. 실제로 작동하는 모습을 보여준다.

GitHub : https://github.com/lswsox/JointBinder

YouTube : https://youtu.be/H8opKtMH2QY

참고로 이 예제의 Joint의 방향은 그림 3과는 달리 Unity 3D의 물리 엔진의 축 방향에 최적화되어있기 때문에 축의 방향이 다르다.

IV. Conclusions

이 연구는 수 많은 Joint들이 복잡하게 연결된 상황에서 Joint들의 위치 정보만으로도 모든 Joint들의 올바른 Twist 방향을 알아낼 수 있는 방법을 제시한다.

일단 Twist가 없는 회전을 알게 되면 이를 기준으로 다양한 응용이 가능하다. 예를 들어 복잡하게 얽힌 로프 시뮬레이션에서 일정하게 꼬이는 Twist 연출 역시 Twist가 없다는 엄격한 기준이 존재하면 가능한 계산이다.

하지만 모든 Joint들에 대해 부모로부터 자식까지 순차적으로 계산되어야한다는 특징은 Joint들의 수가 많아질수록 연산량이 많아지는 단점으로 작용할 수 있다. 이런 단점은 부모의 정보를 개별 Joint에 기록하여 병렬 및 비 동기 연산을 통해 해결할 수 있을 것으로 예상한다.

REFERENCES

[1] New Spline IK Tool Features Variable Squash, Stretch and more. https://lesterbanks.com/2017/10/spline-ik-tool-variable-squash-stretch/