
3D 게임 모델링을 위한 Spline 특성 연구

조형익

예원예술대학교

A Study for properties of Spline to 3D game modeling

Hyung-ik Cho

Yewon art university

E-mail : hypocrycy@naver.com

요 약

오늘날 기술의 발전으로 인해 3D는 game graphic의 필수요소가 되었다. 3D를 사용하여 게임을 만들면 게임유저는 2D게임에 비해 실사와 구분이 안 되는, 멋진 그래픽의 game 을 즐길 수 있고, 각종 이펙트 및 특수효과를 효과적으로 구현할 수 있어 게임 개발하는데 있어서도 유리한 점이 많기 때문에 업계에서 3D Game이 자리 잡게 되었다.

본 논문에서는 3D 게임 그래픽을 만드는데 필요한 각종 3D modeling의 기본을 이루는 2D spline의 특징과 각각의 spline 종류의 장단점에 대해 비교 분석하고 그 발전사에 대해 알아보도록 하겠다.

ABSTRACT

Today due to the development of technology, 3D graphics have become an essential element of the game graphic. When game companies create a game using 3D graphics, game users can enjoy a better game graphics like photo-realistic live action than 2D game graphics. And because the game companies have many advantages in creating games which are easy to deal with many basic effects and special effects, in video game business field, 3D game have become the mainstream.

This paper will inquire the characteristics of 2D spline which is basic to various kinds of 3D modeling necessary to making 3D game graphics, compare and analyze the merits and demerits of each kind of spline and find out its development history.

키워드

spline, 3D, Modeling, Game graphic

1. 서 론

컴퓨터에서는 수학적 함수를 이용하여 각종 모델링을 구현한다. 대다수 게임에서 나오는 핵심 3D object들은 대다수 유기체들인데 이 object들은 유기체 특성상 곡선 및 곡면이 들어간다. 하지만 컴퓨터상에서 곡선 및 곡면 표현은 굉장히 어려운 일이다. 왜냐하면 컴퓨터 자체가 곡면과 곡선의 개념을 알지 못하며 다만 어떤 함수를 주고, 여기에 연속되는 x값을 넣어주면 그에 대응되는 y값이 나오므로써 곡선과 곡면을 표현하며, 이렇

게 함수로 표현된 곡선은 실제 곡선이 아니라 점들이 촘촘하게 모여 있는 집합이기 때문이다. [1] 이런 난점을 해결하기 위해 각종 curve인 Linear Spline, Cardinal spline, B-Spline, Bezier curve, NURBS 같은 2D spline들이 탄생하게 되었다. 이 논문에서는 게임에서 효과적인 곡선과 곡면의 표현을 위해 상용화된 각종 기본 2D curve인 Spline 종류들의 특징을 알아보고, 어느 spline이 게임에서 곡면을 표현하는데 최적화 되어있는지 살펴보고 하겠다.

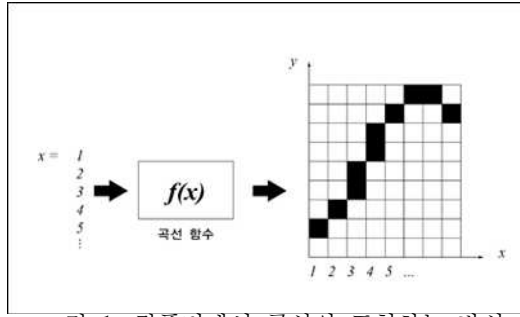


그림 1. 컴퓨터에서 곡선의 표현하는 방식

시작된 방법인 것이다. Linear spline의 장점은 직선을 계속 이어서 만드는 방법이기 때문에 계산이 간단해서 빠르게 곡선을 구현할 수 있다는 것이다. 그 때문에 1초에 수십 장씩, 엄청난 양 그려내야 하는 3차원 그래픽에서는 아직도 이 Polygon 방식을 널리 쓰고 있다. 하지만 확대를 하면 할수록 각진 모양이 뚜렷해지며, 매끈하게 표현하기 위해서는 곡선에서 많은 점을 뽑아서 보관해야 하고 그럴수록 컴퓨터에 부하를 가져온다는 약점을 가지고 있다. [2]

II. 본 론

1. LINEAR SPLINE

Linear spline은 말 그대로 직선이며, 함수 좌표값을 간단히 이은 간단한 구조로써 Bezier curve 등의 곡률을 갖는 방법이 개발되기 전에 쓰였던 방식이다. 이를 다른 말로 하면 그 유명한 폴리곤 polygon이라고도 하며, 곡선 위에 있는 몇 개의 점을 집어내고, 이 점을 그냥 직선으로 이어서 곡선을 표현하는 방식이다. 실제로는 직선이지만 지금도 3차원 그래픽 프로그램이나 CAD에서는 이 방법을 많이 쓰고 있으며, 외곽선 글꼴의 초기 형식에서도 이 방식을 사용했다.

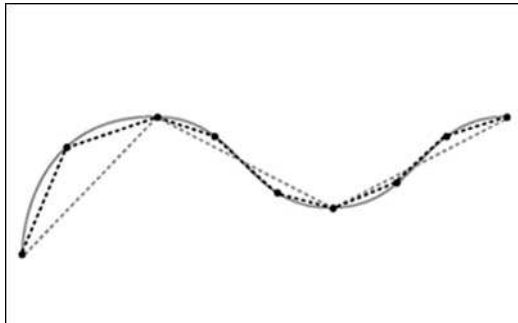


그림 2. Linear spline의 곡선 표시 방식

회색 실선으로 된 곡선을 표현할 때 Polygon 방식은 이 곡선 위에 있는 몇 개의 점에 대한 정보를 갖고 있다가, 곡선을 그려야 할 때에는 그냥 이 점들을 이어서 표현한다. 그림-1에서 알 수 있는 바와 같이, 4개의 점을 사용한 회색 점선보다는 9개의 점을 사용한 검은색 점선이 원래 곡선에 더 가까운 모습을 가지게 된다. 그렇다면 곡선처럼 보이게 하려면 함수 좌표값, 즉 점의 수를 늘리면 되는 방식이다. 만약 수 백개의 점을 촘촘히 배열해서 이것을 연결하면 비록 완전한 곡선은 아닐지라도 곡선처럼 보이게 된다. 이를 응용한 것이 바로 3D에서 Polygon Geometry 방식이며 단위 면적당 면수를 많이 줘서 곡면처럼 보이게 하는 Mesh smooth 명령도 다 여기서 응용되고

2. BEZIER CURVE

Linear spline은 곡선이 아니므로 진정한 곡선을 구현하기 위해서 3차 곡선식의 해를 구하는 공식인 Hermite interpolation이 사용된다.

$$p=p(u)= p_0(1-3u^2+2u^3)+p_1(3u^2-2u^3)+p_0'(u-2u^2+u^3)+p_1'(-u^2+u^3)$$

이 공식은 두 점과 그 점에서의 기울기를 만족하는 삼차곡선에 관한 해법이며, 여기에서 u의 값은 0과 1 사이이다. 여기에서 p₀, p₁, p₀' , p₁' 값이 어떻게 되는가에 따라서 이 공식으로 어떤 곡선이 만들어지는가가 결정된다. 이러한 함수를 Blending function이라고 하고 이것을 응용하여 어떤 곡선도 표현이 가능해졌다. [3]

하지만 이런 Hermite Interpolation은 사실 사용자가 대화형 실체를 진행하기에는 적합하지 않은데, 원하는 형상을 자유자재로 생성하는 데는 부적절하며, P₀' , P₁'의 크기변화에 의한 곡선형상의 변화가 일반인의 직관과 완전히 일치하지는 않기 때문이다. 이런 단점을 해결하고자 60년대 초 프랑스 르노 자동차의 Bezier가 제안한 것이 Bezier curve이다.

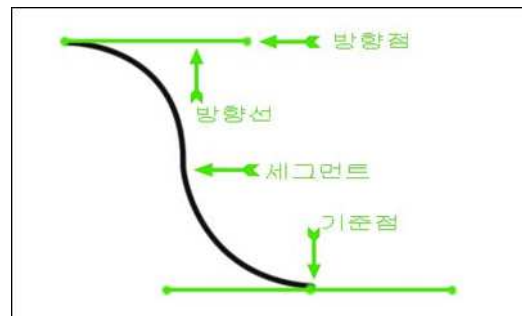


그림 3. Bezier curve의 구성

Bezier curve는 모든 컨트롤 포인트를 통과하지만 각각의 컨트롤 포인트마다 tangent vector라고 하는 방향과 길이를 동시에 표시하는 조절선을 가지며 특정한 컨트롤 포인트에서 그 커브를 스치

고 지나가는 이 접선(tangent)의 방향과 길이(vector)는 각각 그 spline curve의 곡률 방향 및 tangent vector에 의해 영향을 받는 범위를 나타낸다. 이 곡선의 특징은 첫 점과 끝점이 반드시 곡선 안에 있으며, 두 컨트롤 포인트는 언제나 이 곡선을 매끄럽게 해주는 효과를 갖는다는 점입니다. 따라서 사용자가 그래픽 인터페이스를 통하여 쉽게 곡선을 표현할 수 있으며, 항상 매끄럽고 정확한 곡선을 그려내는 장점을 갖게 된다.

하지만 Bezier curve는 처음과 마지막 점을 제외하면 공간상의 점을 통과하지 않고 근사적으로 나타내게 된다. 또한 global modification (전역 조정)만 가능하다는 제약이 있다. 다시 말하면, 블렌딩 함수가 $0 < u < 1$ 의 전 영역에서 0이 아닌 값을 가지게 되므로 패치 하나를 구성하는 점 가운데 어떤 것을 움직여도 곡선이 전체적으로 변한다는 단점이 있다.

3. CARDINAL SPLINE

Cardinal spline은 각 curves들이 합쳐져 큰 curve를 이룬 것이다. 이 Cardinal spline은 각 points와 tension parameter에 의해 특화되는데 선을 구성하는 모든 컨트롤 포인트들을 통과하는 Spline curve인 것으로 Cardinal spline의 장점은 컨트롤 포인트의 위치와 curve가 그리는 궤적의 직접적인 상관관계에 있다는 것이다. 이러한 방식의 장점은 point의 위치에 curve가 만들어지는 직접적인 관계를 가지고 있기 때문에 만약 curve가 위치하고자 원하는 지점이 있다면 그곳에 point를 위치시키면 된다. [4]

반면에 이러한 직접적인 관계의 단점은 그것이 부드럽고 점진적인 곡률의 curve를 만드는데 있어서 문제점을 가지고 있다. 만약 control point의 위치가 조금이라고 잘못 놓인다면 curve는 정확히 잘못 놓인 위치를 통과하면서 그림 4에서 보여지는 것과 같이 부드럽지 못한 모양을 가지게 된다.

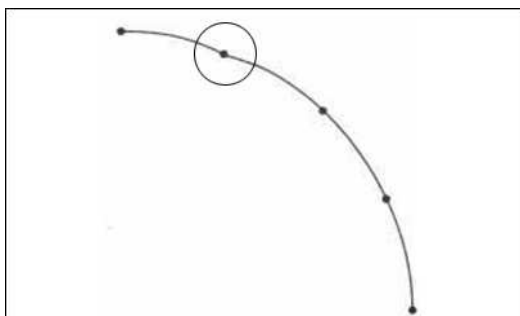


그림 4. Cardinal spline에서 control point가 잘못 놓였을 때 곡선모양이 변형된다.

4. B - SPLINE

Spline의 곡률을 조정하는 방법은 크게 두 가지로 나눌 수 있는데 control point가 curve의 위에 존재하는 interpolating splines 방식과(Bezier curve, Cardinal spline) 두 번째 형태는 curve의 근처에 놓이기는 하지만 직접적으로 control point가 curve의 위에는 놓이지 않는 approximating spline의 형태이다.[5]

Control point가 곡선 위에 있는 Bezier curve, Cardinal spline의 경우 패치 하나를 구성하는 점 가운데 어떤 것을 움직여도 곡선이 전체적으로 변하는 단점이 있다. 반면 approximating spline의 경우는 어느 정도 잘못된 위치의 point 에도 부드러운 curve를 계산할 수 있어 각 점이 일정한 범위에만 영향을 미치기 때문에 전체적으로 모양을 바꾸지 않고 원하는 부분만 조절해 줄 수 있는 장점이 있다. 이러한 approximating spline 형태를 B-spline이라고 한다.

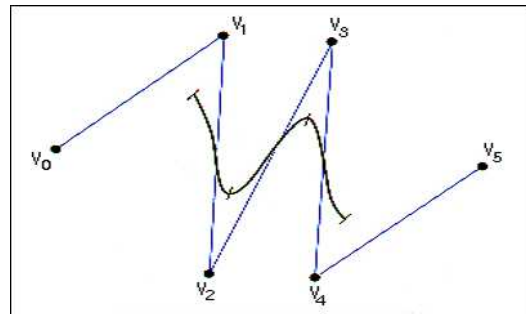


그림 5. Approximating spline인 B-spline

5. NURBS

NURBS는 Non-Uniform Rational B-Spline, 즉 NURBS 곡선이며 비균일 매듭 vector를 사용한다. NURBS 곡선은 Non-Rational B-Spline과 Bezier 곡선뿐 아니라 직선과 2차의 해석적 곡선을 하나의 형태로 나타낼 수 있으며, 근사적인 모드나 보간적인 형태로도 사용할 수 있다. 즉 NURBS는 모든 곡선을 포용하는 특성이 있어 곡선의 표준으로 설명할 수 있으며 Linear, Bezier, Cardinal, B-Splines 곡선은 NURBS의 하위 개념이라고 할 수 있다. 다른 곡선으로 구현할 수 있는 것이면 모두 NURBS로도 구현할 수 있다는 장점을 가지고 있는 것이다. 이를 거꾸로 생각하면, NURBS curve는 위에서 언급된 spline curve 들의 장점을 고루 갖고 있으며 Cardinal spline 처럼 curve의 시작점과 끝점에서는 control point를 지나면서도 중간점들에서는 Approximating spline처럼 control point의 주변을 지나기 때문에 부드러운 곡률을 가진다. 또한 NURBS는 control point외에 커브위에 존재하는 edit point라고 하는 별도의 조절점들

을 갖기 때문에 상황에 따라 정확도가 필요한 경우에는 Cardinal spline처럼 직접 curve가 지나는 위치를 조절을 할 수도 있다. NURBS가 가지는 또 하나의 중요한 특징은, 각각의 컨트롤 포인트마다 weight를 조정할 수 있다는 점인데 이 weight값이 클수록 curve는 그 control point 쪽으로 치우치게 되므로 control point의 위치이동 없이 커브의 미세한 곡률까지 조정할 수 있다는 장점을 가지고 있다. 이런 weight를 조정할 수 있다는 점 때문에 NURBS는 다른 여타 곡선에 비해 많은 자유도를 가지고 있으며 이런 특징 때문에 다른 곡선에서 만들 수 없는 불규칙 곡선까지도 만들 수 있다.(그래서 Non-Rational임)

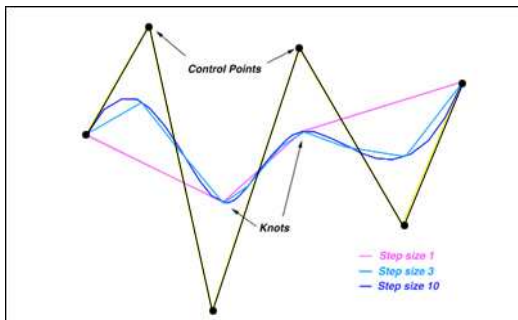


그림 6. Spline의 완성형인 NURBS

그렇기 때문에 현재 CG에서 다른 Spline을 제치고 이 NURBS 곡선이 채택되고 추앙받는 이유이기도 하다.

6. NURBS의 단점

제어가 쉬우며, 완벽한 곡률을 만들 수 있으며 높은 퀄리티의 결과물을 만들 수 있는 NURBS이지만 단점이 존재한다. 바로 NURBS 자체는 3D software상에서 렌더링이 불가능하다는 점이다. 렌더링 시에는 NURBS를 Polygon으로 전환시켜야 한다. NURBS를 Polygon으로 전환시키면 NURBS의 곡률을 표현하기 위해 엄청난 수의 Polygon들이 쓰이게 되며, 이로 인해 NURBS 모델링은 일반 Polygon 모델링 보다 데이터 용량이 엄청나게 많이 나갈 수밖에 없다.

그러기에 NURBS 모델링의 경우, 정확한 데이터 처리를 해야 하는 3D 설계나, 고가의 장비가 투여되는 포스트프로덕션이나 영상, 3D 애니메이션의 경우에는 무리 없이 사용할 수 있으나 대규모 Scene을 실시간 렌더링 하면서 제어해야 하는 게임의 경우에는 용량의 최적화 문제가 최우선되어야 하기 때문에 NURBS를 이용한 고품질의 결과물을 사용할 수가 없다.

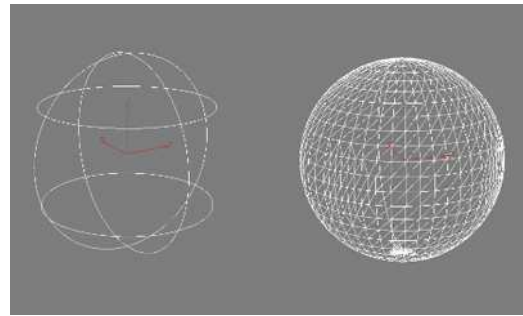


그림 7. NURBS 구가 Polygon으로 전환 시 mesh수가 엄청 늘어나는 것을 알 수 있다.

III. 결 론

본 논문에서는 각 Spine의 특성을 알아보고 어떤 Spine이 게임의 효과적인 그래픽을 구현하는데 효과적인지 살펴보려고 하였다.

곡면을 가장 잘 표현하여 최상의 결과물을 만들기 위해서는 NURBS를 사용하는 것이 좋으나 NURBS의 특성상 대규모 데이터를 실시간으로 처리하면서 플레이어가 제어해야 하는 게임에서는 사용하기 무리이다. 오히려 모델의 곡면 표현은 떨어지지만 이런 게임의 특성과 연계하여 가장 효과적으로 사용할 수 있는 Spline은 용량이 적게 나가는 Linear spline을 이용한 모델링이며 대규모 데이터를 실시간으로 처리하는데 가장 최적인 Spline인 것을 확인하였다.

참고문헌

- [1] Bojanov, B. D. /Hakopian, H. a. /Sahakian, A. A, *Spline Functions and Multivariate Interpolations*, Springer, 1993
- [2] Schumaker, Larry L, *Spline Functions Basic Theory 3rd eds.*, Cambridge University Press, 2007
- [3] Prenter, P. M., *Splines and Variational Methods*, Dover Publications, 2008
- [4]<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/gdicpp/GDIPlus/aboutGDIPlus/linescurvesandshapes/cardinalsplines.asp>
- [5] Lai, Ming-Jun /Schumaker, Larry L., *Spline Functions on Triangulations*, Cambridge University Press, 2007