

Attribute Based ModelingTM surface 디자인을 위한 인터페이스 설계 및 구현

김정화*, 박화진**

*숙명여자대학교 정보과학과

**숙명여자대학교 정보과학과

e-mail:bbpoint@sookmyung.ac.kr

Interface design and implementation for Attribute Based ModelingTM surface design

Jeong-Hwa Kim*, Hwa-Jin Park**

*Dept of Computer Science, Sook-myung University

**Dept of Computer Science, Sook-myung University

요 약

모형 설계와 도형의 생성은 점, 곡선, 기울기과 같은 특정 속성들의 정보로 만들어 진다. 도형을 생성할 때 곡면 조각, 움직이는 곡선과 점, 일반적인 곡선, 연속 혹은 닫힌 곡선, 빠른 계산들은 중요하게 생각 되는 모델링 개념이다. 이 논문에서는 Attribute Based ModelingTM(A-B ModelingTM) surface을 사용하여 디자인을 디자인할 때 필요한 Multi-sided patch의 설계와 인터페이스를 구현하고자 한다.

1. 서론

컴퓨터 그래픽스에서 곡면 모델링에 대한 연구는 수십 년간 계속 되어온 과제이다. 실제로 곡면 모델링을 위해 Bezier, B-spline, Nurbs 등과 같은 여러 가지 방법이 제공되었으며, 또한 널리 사용되고 있다.

현재 사용되고 있는 모델링 방법은 수학적이고 기술적인 부분에서 전문적이고 복잡하기에 디자인된 형상을 기하학적인 형태로 변화시키는데 종종 한계를 보인다. 예를 들어, Nurbs는 곡면조각의 사각형의 배열법(rectangular arrangement)을 필요로 한다. 하지만 이것은 물체의 자연적인 형상을 항상 적합하게 표현하지는 못한다.

이 논문에서 제시되는 A-B ModelingTM은 형상을 디자인하는 방법으로 좀더 자유로운 디자인 기술을 가능하게 하게 한다.

본 논문에서는 Attribute Based ModelingTM surface 디자인을 위한 인터페이스를 구현하고자 한다. 2장에서는 관련 연구로서 기존의 다양한 surface 모델링 방법에 대해 간략하게 소개하고, 3장에서는 Attribute Based ModelingTM 방법에 대해

기술한다. 4장에서는 surface를 이루는 Multi-sided patch를 설계 및 구현 하였고, 마지막 5장에서는 결론과 앞으로의 향후 과제를 제시한다.

2. 관련연구

본 절에서는 surface를 만들기 위해 사용된 기존의 모델링 방법을 기술한다.

2.1 Bezier surface

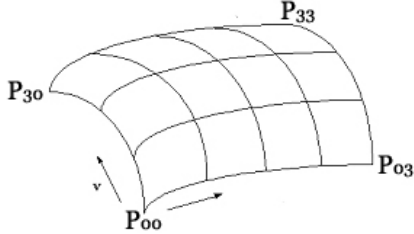
Bezier 곡면은 배합 함수를 통하여 생성할 수 있다. 베지어 곡선이 조정 다각형에 의해 정의되듯이 베지어 곡면은 조정 다면체 (control polygon, characteristic polygon, charateristic mesh)에 의해 정의된다.

베지어 곡면과 각각의 조정 다면체의 예를 그림 1에 보인다. 베지어 곡면은 많은 면에서 베지어 곡선과 유사한 특성을 갖는데, 특히 곡면이 조정다면체의 네 개의 구석점만을 통과하며, 곡면의 모서리 곡선은 구석점에서 조정다면체의 모서리에 접하게 된다.

베지어 곡면 P가 4*4의 제어점 배열이라면 P[p_{ij}]에 대응하는 다음과 같이 표기할 수 있다.

$$f(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 B_i(u) B_j(v) p_i \quad (1)$$

베지어 곡면은 전역 조정만이 가능하고, 서로 인접한 패치를 구성할 때에는 매끄럽게 연결되는 데는 제약을 보인다.



(그림 1) Bezier 곡면 단편

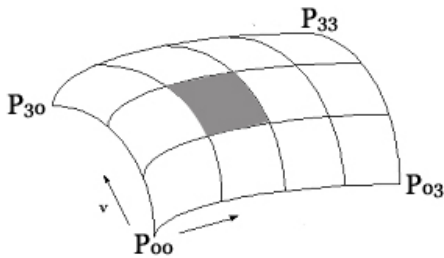
2.2 B-spline surface

베지어 곡면의 제약은 B-스플라인 곡면을 이용하면 대부분 해결된다. B-스플라인 곡면은 우선 곡선에서 베지어 곡선과는 달리 선형 블렌딩함수가 사용될 때를 제외하고는 시작점과 끝점을 통과하지 않는다.

B-스플라인 곡면의 식은 다음과 같이 표기할 수 있다.

$$f(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 B_i(u) B_j(v) p_i \quad (1)$$

이 표현은 다른 곡면에 수식 표현과 같은 형태이지만 앞의 그림에서 보듯이 단편의 가운데 부분만 사용하므로, Bezier곡면과 비교할 때 9배의 작업을 해야 한다. 그러나 B-스플라인 곡선으로부터 컨벡스할 특성과 가장자리에서의 부가적인 연속성을 물려받았기 때문에 B-스플라인 곡면은 같은 데이터를 가지고 구성된 Bezier곡면 보다 상당히 부드럽다.



(그림 3) B-spline 곡면 단편

2.3 Nurbs surface

$$P(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m W_{i,j} N_{i,k}(v) N_{j,l}(u) V_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m W_{i,j} N_{i,k}(v) N_{j,l}(u)} \quad (3)$$

Nurbs곡면은 이를 표현하는 곡선의 궤적에 있어 하나의 매개변수를 일정하게 유지하고 다른 하나를

변화시킴으로써 얻어질 수 있다. Nurbs 곡면표현은 (3)과 같이 u와 v 두 변수에 대한 매개변수식으로 정의한다.

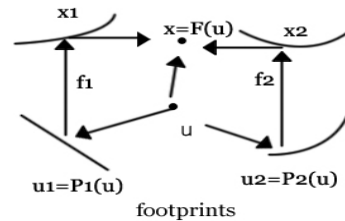
3. A-B ModelingTM

A-B ModelingTM은 가중 이산 최소 제곱법 (weighted discrete least squares solution)에 근거를 둔다.

$$\sum_i [(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2] W_i(x, y, z) \quad (4)$$

이 방법은 최대 가중 $W_i(x, y, z)$ 을 가지는 점들을 제거하고 최소 가중치 $x = (x, y, z)$ 를 구한다. [4]

A-B ModelingTM은 그림 4에서 보이는 것처럼 (4)에서 사용되는 interpolate 점과 곡선의 가중치를 매개변수로 가진다. 간단한 점이나 곡선으로 주어지는 U_i 는 *footprints*라고 한다. R2에서 어떤 점 u 를 위해 $P_i(u)$ 는 각 *footprint* U_i 의 유일한 점 u_i 로 투영된다. 각 *footprint*는 attribute function f_i 이다. 이는 R3에 U_i 로 그려진다. 여기서 $f_i(U_i)$ 는 attributes라고 한다. 어떤 파라미터 u 에 대해 $F(u) = x$ 는 가중최소제곱법으로 구해진다. 이때 가중치는 u 에서부터 투영된 u_i 까지의 상호간의 거리로 주어진다. u_i 와 *footprint* 사이의 거리를 계산하고 *footprint*로 투영하기 위한 다양한 방법이 있다. *footprint*의 거리는 중요한 계산이 된다. 그러므로 점과 직선과 같은 간단한 *footprints*를 유지하기 위한 동기가 된다. [1]

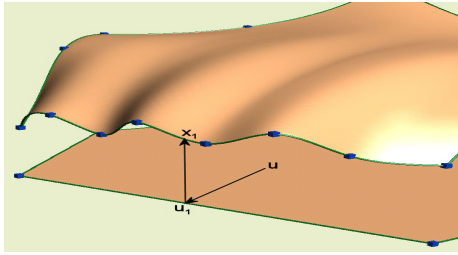


(그림 4) Map u to x via footprints and attributes.[1]

예를 들어 그림 5에서 다각형의 모서리는 *footprints*가 된다. *footprint*에서 점 u 는 u_i 으로 투영한다. 점 $x = F(u)$ 는 (5)에 의해 계산된다.

다음의 식으로 일반화 할 수 있다.

$$F(u) = (x, y, z) = \sum_i f_i(u) W_i(u) / \sum_i W_i(u) \quad (5)$$



(그림 5) Map u to xi via footprint.

그림 4에서 surface를 보면. 가까운 attribute 곡선은 곡면의 생성할 때 더 밀접한 영향을 미치게 된다. 곡면의 중앙은 attribute $f_i(u)$ 의 평균치가 된다.

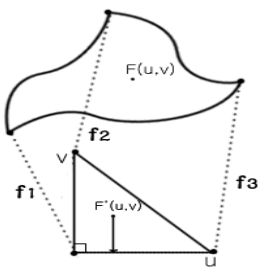
4. A-B Modeling™기반 Multi-sided patch의 footprint 설계 및 interactive interface 구현

모든 물체의 형태에는 특징적인 곡선이 반드시 있다. 하지만 Nurbs나 Bezier와 같은 표준 모델링 기술에서 보이는 것처럼 항상 사각의 형태를 유지하는 것은 아니다. 곡선을 나타내는 기본 형상이 곡선 조각이듯이 패치(patch)는 곡면을 생성하는 기본 요소이다. 따라서 본 절은 A-B Modeling™기반하여 three-sided patch와 four-sided patch에 대한 설계와 인터페이스를 구현하였다.

4.1 A-B Modeling™기반 Multi-sided patch 설계

4.1.1. three-sided patch

그림 6에서 보이는 것처럼 three-sided patch에 대한 도메인은 직각 삼각형으로 둔다.



(그림 6) Map 3-sided patch

$F'(u,v)$ 에 대한 속성 함수(attribute function)는 $f_i(6)$ 로 주어진다. 점 $F'(u,v)$ 와 투영된 점 $F'_i(u,v)$ 사이의 거리는 $d_i(7)$ 로 나타낸다.

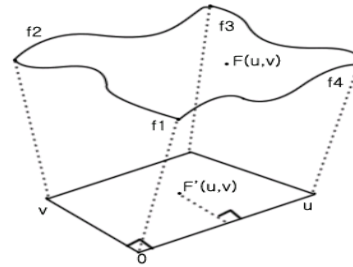
$$\begin{cases} f1 = v \\ f2 = \frac{u-v+1}{2} \\ f3 = u \end{cases} \quad (6) \quad \begin{cases} d1 = u \\ d2 = \sqrt{\frac{(1-u-v)^2}{2}} \\ d3 = v \end{cases} \quad (7)$$

3절의 (5)을 기반으로 하여 f_i, d_i 를 대입시키면 다음과 같은 식이 나온다. 이다.

$$F(u,v) = \frac{\sum_{i=1}^3 f_i(u,v) \times \frac{1}{d_i}(u,v)}{\sum_{i=1}^3 \frac{1}{d_i}} \quad \left(\frac{1}{d_i} = W_i\right)$$

4.1.2. four-sided patch

그림 7에서 보이는 것처럼 four-sided patch에 대한 도메인은 사각형으로 둔다.



(그림 7) Map 4-sided patch

$F'(u,v)$ 에 대한 속성 함수(attribute function)는 $f_i(8)$ 로 주어진다.

점 $F'(u,v)$ 와 투영된 점 $F'_i(u,v)$ 사이의 거리는 $d_i(9)$ 으로 나타낸다.

$$\begin{cases} f1 = v \\ f2 = u \\ f3 = 1-v \\ f4 = 1-u \end{cases} \quad (8) \quad \begin{cases} d1 = u \\ d2 = 1-v \\ d3 = 1-u \\ d4 = v \end{cases} \quad (9)$$

3절의 (5)을 기반으로 하여 f_i, d_i 를 대입시키면 다음과 같은 식이 나온다. 이다.

$$F(u,v) = \frac{\sum_{i=1}^4 f_i(u,v) \times \frac{1}{d_i}(u,v)}{\sum_{i=1}^4 \frac{1}{d_i}} \quad \left(\frac{1}{d_i} = W_i\right)$$

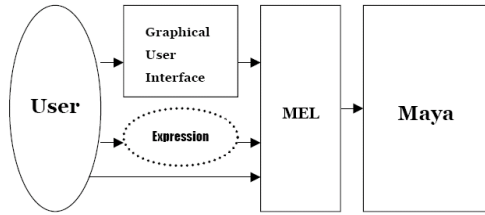
4.2 Interactive interface구현

본 절에서는 3절에서 설계된 공식을 기반으로 하여 Multi-sided patch의 인터페이스 구현을 Windows XP를 기반으로 Maya7.0과 mel을 이용하여 구현하였다.

구현에 사용된 MEL(Maya Embedded Language)은 마야의 기본적인 동작을 수행하는 스크립트 언어이다. Maya는 사용자가 좀 더 유연하게 사용할 수 있게 하기 위하여 자신의 모든 동작을 MEL로 수행되게 하였다.

따라서, 사용자는 MEL을 사용하여 GUI를 통해 제

공되지 않는 기능을 추가하거나, GUI를 자신의 취향에 맞게 customize하거나 새로운 매크로를 설정할 수 있다.



(그림 8) Maya 에서의 Mel 역할
Mel을 사용하여 Object의 Attribute를 정확하게 컨트롤하거나 새로운 매크로를 만들거나 새로운 User Interface를 만들 수 있다.

```

$point0 = `pointOnCurve -pr ($v*$spanu) -p $crv[0]`;
$point1 = `pointOnCurve -pr (($u-$v+1)/2*$spanv) -p $crv[1]`;
$point2 = `pointOnCurve -pr ((1-$u)*$spanu1) -p $crv[2]`;

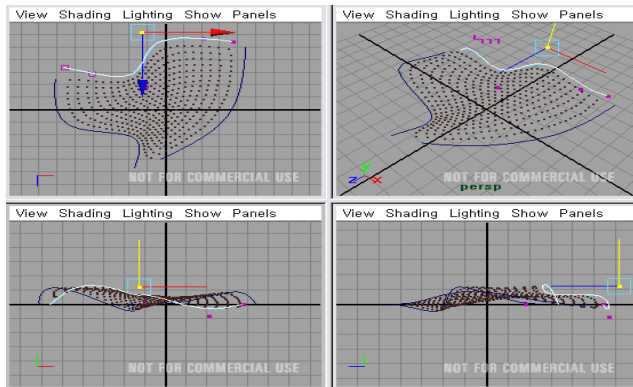
$invdist[0] = $u;
$invdist[1] = sqrt((1-$u-$v)*(1-$u-$v)/2);
$invdist[2] = $v;

$ABsurfacex[$i][$j] = $point0.x*$invdist[0] +
$point1.x*$invdist[1] + $point2.x*$invdist[2];
$ABsurfacex[$i][$j] = $ABsurfacex[$i][$j]/
($invdist[0]+ $invdist[1]+$invdist[2]);

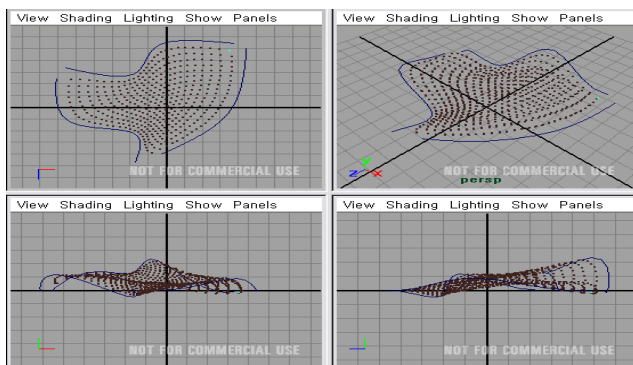
$ABsurfacey[$i][$j] = $point0.y*$invdist[0] +
$point1.y*$invdist[1] + $point2.y*$invdist[2];
$ABsurfacey[$i][$j] = $ABsurfacey[$i][$j]/
($invdist[0]+ $invdist[1]+$invdist[2]);

$ABsurfacez[$i][$j] = $point0.z*$invdist[0] +
$point1.z*$invdist[1] + $point2.z*$invdist[2];
$ABsurfacez[$i][$j] = $ABsurfacez[$i][$j]/
($invdist[0]+ $invdist[1]+$invdist[2]);
    
```

(그림 9) three-sided patch Mel program



(그림 10) three-sided patch implementation



(그림 11) 곡선의 점을 이동시켰을 때의 변화

그림 10은 mel로 구현된 three-sided patch를 보여 준다. 그림 11은 생성된 patch에서 하나의 점 혹은 선을 이동하였을 때 변화하는 모습이다.

5. 결론 및 향후 과제

A-B ModelingTM surface 디자인을 위한 인터페이스를 설계, 구현하였으며 이로 인해 각 점 혹은 선의 위치 이동에 따른 변화를 interactive하게 알 수 있게 하였다. A-B ModelingTM은 최근 모델링 분야에 새로운 연구 방법으로 다양한 주제를 만들 수 있다. 예를 들어 footprint를 생성 시 다양한 도메인의 형태로 하였을 때 발생 할 수 있는 효과를 비교 가능하다. 4절의 three-sided patch에서 적용된 직각 삼각형 도메인을 정삼각형의 형태로 변화 하였을 때의 footprint와 weighted를 설계하고 이에 따른 모델링 효과를 비교 할 수 있다.

또한, footprints를 직선으로 하였을 때와 곡선으로 하였을 때의 차이점을 비교하여 모델링에 좀더 효과적인 방법을 제시 할 수 있다.

참고문헌

[1] Ghostasby, A. 1988. Image Registration by Local Approximation Methods. Image and Vision Computing, Vol 6 (4), Nov., 255-261.

[2] Kun Gao, Hwajin Pack. Alyn Rockwood. 2005. "Attribute Based Interfaces for Geometric Modeling"

[3] Edward Angel, 2002. opengl을 이용한 컴퓨터 그래픽스

[4] Gao, K., Rockwood, A., 2005, Multi-sided attribute based modeling, IMA Mathematics of Surfaces XI, R. Martin (ed.), Springer Verlag, 2005. 219-232

[5] Malraison, P. 1999. Multi sided Surfaces: a survey. *Curve and Surface Design*. Editors Pierre Jean Laurent, Paul Sablonière, Larry Schumaker, 246-256.

[6] Foley. Van, Dam.feiner, Hughes.phillips, 1994, Introduction to computer graphics.