

## 토러스 패치 기반의 정밀 근사를 이용한 자유곡면의 기하학적 처리

박영진<sup>o</sup>      홍규연      김명수<sup>\*</sup>

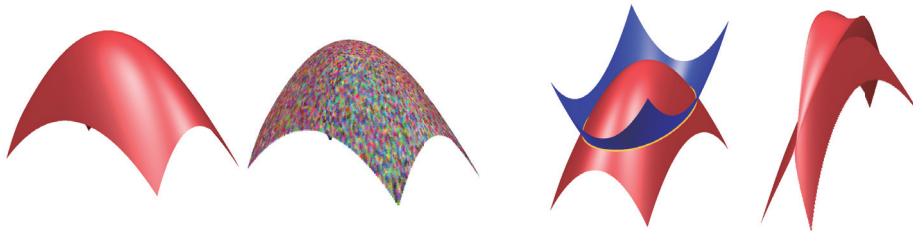
서울대학교 컴퓨터공학부

yjinpark@snu.ac.kr, qhong@3map.snu.ac.kr, mskim@snu.ac.kr

### Geometric Processing for Freeform Surfaces Based on High-Precision Torus Patch Approximation

Youngjin Park<sup>o</sup>      Q Youn Hong      Myung-Soo Kim<sup>\*</sup>

Department of Computer Science and Engineering, Seoul National University



**Figure 1:** A Freeform surface and a set of torus patches. Intersection curve and offset surface using a set of torus patches.

#### 요약

3차원상의 자유곡면에 대한 효율적인 기하 연산을 지원하기 위한 새로운 공간자료구조로서 토러스 패치 기반 정밀 근사를 이용한 자유곡면의 기하학적 처리 기법을 소개한다. 토러스는 곡률이 양이나 음인 경우뿐만 아니라, 0인 부분도 있으므로 자유곡면의 볼록, 오목 여부에 상관없이 곡면을 정밀하게 근사할 수 있다. 전통적인 기법과 달리 토러스 패치는 자유곡면의 법벡터 방향까지 쉽게 모델링할 수 있고, 토러스의 오프셋은 다시 토러스가 되므로 다양한 기하 연산의 가속화를 지원할 수 있다. 자유곡면과 이를 근사하는 토러스 패치 집합 사이의 양방향 하우스도르프 거리의 상한을 계산하여 토러스 패치를 이용하여 자유곡면을 높은 정밀도로 근사할 수 있음을 보였다. 이 기법을 이용하여 두 자유곡면의 교차곡선 계산과 자유곡면의 오프셋 곡면 생성을 쉽게 처리할 수 있음을 보였다.

#### Abstract

We introduce a geometric processing method for freeform surfaces based on high-precision torus patch approximation, a new spatial data structure for efficient geometric operations on freeform surfaces. A torus patch fits the freeform surface with flexibility: it can handle not only positive and negative curvature but also a zero curvature. It is possible to precisely approximate the surface regardless of the convexity/concavity of the surface. Unlike the traditional method, a torus patch easily bounds the surface normal, and the offset of the torus becomes a torus again, thus helps the acceleration of various geometric operations. We have shown that the torus patch's approximation accuracy of the freeform surface is high by measuring the upper bound of the two-sided Hausdorff distance between the freeform surface and set of torus patches. Using the method, it can be easily processed to detect an intersection curve between two freeform surfaces and find the offset surface of the freeform surface.

**키워드:** 토러스 패치, 자유곡면, 정밀 근사, 하우스도르프 거리

**Keywords:** Torus Patch, Freeform Surface, Approximation with High Precision, Hausdorff Distance

\*corresponding author: Myung-Soo Kim/Seoul National University(mskim@snu.ac.kr)

## 1. 서론 및 관련연구

기하학적 물체에 대한 기하 연산의 가속화를 위해 연구되고 있는 다양한 자료구조 중, 복잡한 물체를 간단한 도형의 집합들로 감싸 계층적 트리 구조를 만드는 바운딩 볼륨 계층구조(Bounding Volume Hierarchy, BVH)는 컴퓨터 그래픽스, 애니메이션, 로보틱스에서 가장 많이 사용되고 있는 자료구조 중 하나이다.

BVH를 사용할 때, BVH를 이루는 요소인 바운딩 볼륨(Bounding Volume, BV)으로 어떤 구조를 쓸 것인지 선택하는 것은 중요한 문제이다. 구(sphere), 박스 등 간단한 BV는 물체의 위치 정보만으로 만들 수 있어 생성 시간은 빠르지만, 물체를 잘 묘사하지 못해 근사정밀도가 떨어지는 단점이 있다. 반면 곡선, 또는 곡면의 기하학적 정보를 이용하여 생성한 BV의 생성시간은 느리지만, 근사정밀도가 높다. BVH는 한번 생성하면 물체의 변형이 있지 않는 이상 재사용이 가능하므로 최근 연구들에서는 간단한 도형으로 빠르게 BVH를 생성하는 것보다 근사정밀도가 높은 도형을 BV로 채택하여 효율적인 BVH를 설계하는 방향에 초점을 맞추고 있다.

2차원의 경우 자유 곡선에 대한 BVH를 생성하기 위해 가장 단순한 원으로 BV를 구성하는 것부터 AABB(Axis Aligned Bounding Box), OBB(Oriented Bounding Box)[1], LSS(Line Swept Sphere)[2], RSS(Rectangle Swept Sphere)[3] 등 다양한 BVH가 제안되었다. Meek 등[4]은 곡률 정보를 활용하여 만든 원호(arc)를 기반으로 한 평면곡선을 근사하는 방법을 제안하였으며, 이후 Barton 등[5], Kim 등 [6, 7], Lee 등[8]은 이를 BVH 생성에 활용하여 더욱 근사정밀도를 높인 알고리즘을 제안하였다. 원호 기반 접근 방법의 도입으로, 이전까지 완벽한 해결이 불가능한 것으로 여겨졌던 고정밀도( $10^8 \times 10^8$ ) 환경에서의 자유 곡선의 오프셋이나 보로노이 다이어그램 계산 문제들이 해결되었고[9, 10, 11], 최근 연구에서는 BCA Tree[12]로 계산속도와 안정성이 대폭 개선되었다. 이 결과로 2차원상에서의 자유곡선에 대한 기하 문제는 더이상 개선의 여지가 없이 완벽하게 해결된 것으로 인식되고 있다.

2차원 문제들에서 효과적으로 사용된 원호 기반 BV를 3차원 기하 문제의 해결을 위한 새로운 BV로 아이디어를 확장해보면, 2차원의 원호에 대응되는 3차원의 물체는 토러스이다. 흔히 구면이라고 생각하기 쉽고, 과거 구면을 이용한 BVH 연구가 진행되었던 적도 있다[13, 14]. 하지만 구면은 자유곡면의 오목한 부분을 제대로 근사하지 못하며 효율성이 떨어진다. 또한 자유곡면이 볼록하더라도 구면은 모든 방향으로 하나의 고정된 곡률만을 가지므로 일반적으로 서로 직교하는 두 방향으로 크기가 다른 곡률을 가지는 자유곡면을 구면으로 정밀하게 근사하기에는 한계가 있다.

이 논문에서는 3차원상의 자유곡면에 대한 효율적인 기하 연산을 지원하기 위한 새로운 공간자료구조인, 토러스 패치 기반의 정밀 근사를 이용한 자유곡면의 기하학적 처리 기법(이하 토러스 패치 근사 기법)을 소개한다. 토러스는 곡률이 양이나 음인

경우뿐만 아니라, 0인 부분도 있으므로 곡면의 볼록, 오목 여부에 영향을 받지 않고 근사할 수 있다. 또한 곡면에서 직교하는 두 방향의 곡률이 서로 다르더라도 토러스의 반지름을 조정하여 곡면을 정밀하게 근사할 수 있다. 이에 대한 근거로, Liu 등[15]은 3차원상의 임의의 점에서 곡면까지의 수선을 내리는 문제를 해결하기 위해 토러스를 이용하여 곡면을 근사하였다. 이는 Hu 등[16]의 방법에 비해 높은 정밀도를 보장하면서도 더 효율적으로 수선을 구하는 방법으로 알려져 있다.

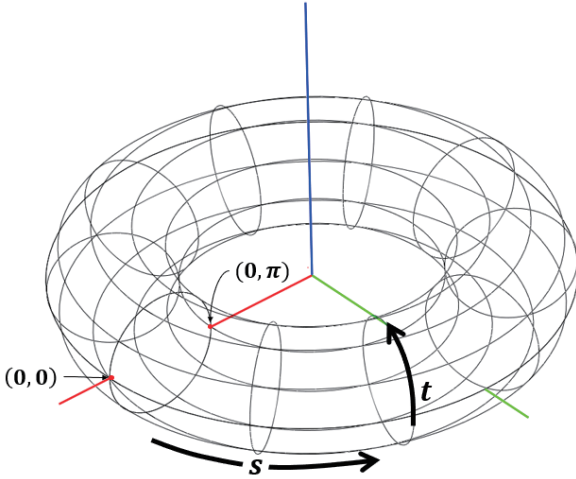
자유곡면의 법벡터 정보는 컴퓨터 그래픽스에서 렌더링을 위해서 뿐만 아니라, 충돌감지나 거리계산 등의 기하 연산을 위해서 필수적으로 요구되는 기하학적 정보이다. 법벡터가 필요하지 않은 계산일지라도 법벡터를 이용하면 불필요한 계산을 수행하지 않아도 되는 경우가 많아, 대부분의 경우 기하 연산의 가속화에 큰 도움을 준다[6]. 주어진 곡면이 가지는 모든 법벡터의 범위를 포함하는 방법으로는 Normal Cone[17], Convex Pyramid[18] 등이 제안되었다.

토러스 패치를 이용하여 자유곡면을 근사하면 위치 정보뿐만 아니라 법벡터도 거의 유사하게 변화하므로 복잡한 자유곡면의 기하학적 정보를 간단하게 표현할 수 있다. 기존의 전통적인 BVH들이 위치 정보만 포함할 뿐, 법벡터를 저장하는 경우는 아주 예외적이므로 이는 토러스 패치 근사 기법의 중요한 장점들 중 하나이다.

오프셋 곡면은 곡면상의 각 점을 곡면의 법벡터 방향으로 일정한 거리만큼 이동시켰을 때 구해지는 곡면을 말한다. 자유곡면에 대한 오프셋 곡면은 일반적으로 입력 곡면보다 훨씬 더 높은 차수의 대수곡면이 되므로[19], 오프셋 곡면의 근사곡면(일반적으로 오프셋 곡면보다 낮은 차수를 가지는 B-spline 곡면)을 찾는 많은 이전 연구들이 진행되었다[20, 21, 22].

만약 오프셋 곡면이 입력 곡면과 동일한 종류의 곡면으로 변환된다면 별도의 시간을 들여 근사곡면을 계산하지 않아도 되므로 오프셋 곡면을 쉽게 구할 수 있다. 토러스 패치의 경우 토러스를 구성하는 반지름을 조정하는 것으로 오프셋 곡면을 근사하는 토러스 패치를 추가 연산 없이 곧바로 구할 수 있다. 토러스 이외에 이러한 특징을 가지는 2차원상의 곡선의 예로는 직선, 원 등이 있으며 3차원상의 곡면의 예로는 평면, 구면, cylinder, cone 등이 있다[23, 24, 19].

하우스도르프 거리(Hausdorff distance)는 한 물체에서 다른 물체까지의 최대 편차를 나타낸다. 하우스도르프 거리가 0에 가까울수록 두 물체가 서로 유사함을 나타내며, 이런 성질로 두 기하학적 물체의 유사성을 판단하는 도구로 사용된다[25, 26, 27]. 자유곡선과 자유곡면에 대한 하우스도르프 거리 측정에 대한 연구는 일반적으로 다각형 모델에 대한 하우스도르프 거리 측정[25, 28, 29]보다 어려운 문제로 알려져 있다. Kim 등[27]은 이중원호(biarc)를 이용하여 평면곡선의 하우스도르프 거리를 실시간으로 측정하는 방법을 제안하였으며, 자유곡면 간 하우스도르프 거리를 실시간으로 측정하기 위해 Hanniel 등[30]과 Kirishnamurthy 등[31]은 GPU를 사용하였다. Kim 등[32]은 자유곡면에 BVH를



**Figure 2:** Torus  $\hat{\mathbf{T}}(s, t)$  with point  $(0, 0)$  and point  $(0, \pi)$ .

사용하여 효과적으로 하우스도르프 거리 후보군을 줄여나가는 방법을 사용하여 자유곡면 간 하우스도르프 거리를 측정하였다.

토러스 근사 기법의 근사정밀도를 측정하기 위하여 한 물체에 서 다른 물체로의 하우스도르프 거리와 그 반대의 하우스도르프 거리 중 최대값인 양방향(two-sided) 하우스도르프 거리를 이용할 수 있다. 토러스 패치 근사 기법으로 구한 토러스 패치 집합과 자유곡면 간 양방향 하우스도르프 거리의 상한을 Filip 등[33]의 방법을 사용하여 구한 최대근사오차를 이용하여 측정하고, 이 값이 근사오차한계인  $10^{-6}$  보다 작음을 보여 토러스 패치 집합이 자유곡면을 정밀근사함을 입증하였다.

이 논문의 구성은 다음과 같다. 2절에서는 자유곡면에 대한 미분기하 계산을 이용하여 토러스 패치를 근사하는 알고리즘과 하이브리드 BVH 구조를 이용하여 토러스 패치의 근사오차를 최소화하는 방법을 소개한다. 3절에서는 자유곡면 예제와 실험 결과를 소개하고, 토러스 패치들을 이용하여 두 자유곡면의 교차곡선과, 오프셋 곡면을 모델링하였다. 4절에서는 이번 연구를 확장하여 해결할 수 있는 문제들에 대해 소개한다. 마지막으로 5절에서는 본 연구의 결론을 내린다.

## 2. 알고리즘

### 2.1 자유곡면에 대한 미분기하 계산

자유곡면  $\mathbf{S}(u, v)$ 와 이를  $u, v$ 에 대해 1차 미분한  $\mathbf{S}_u, \mathbf{S}_v$ , 2차 미분한  $\mathbf{S}_{uu}, \mathbf{S}_{uv}, \mathbf{S}_{vv}$ 가 주어졌을 때, 제1차 기본형식의 값  $E, F, G$ , 단위법선벡터  $\mathbf{N}$ , 그리고 제2차 기본형식의 값  $L, M, N$ 은 다음과 같이 정의된다.[34].

$$E = \langle \mathbf{S}_u, \mathbf{S}_u \rangle, F = \langle \mathbf{S}_u, \mathbf{S}_v \rangle, G = \langle \mathbf{S}_v, \mathbf{S}_v \rangle$$

$$\mathbf{N} = \frac{\mathbf{S}_u \times \mathbf{S}_v}{|\mathbf{S}_u \times \mathbf{S}_v|}$$

$$L = \langle \mathbf{S}_{uu}, \mathbf{N} \rangle, M = \langle \mathbf{S}_{uv}, \mathbf{N} \rangle, N = \langle \mathbf{S}_{vv}, \mathbf{N} \rangle$$

자유곡면  $\mathbf{S}(u, v)$ 의 법곡률  $\kappa_n$ 은 임의의 방향  $v = a\mathbf{S}_u + b\mathbf{S}_v$ ,  $a^2 + b^2 = 1$ 에 대해 다음과 같이 정의된다.

$$\kappa_n = \frac{La^2 + 2Mab + Nb^2}{Ea^2 + 2Fab + Gb^2}$$

여기서 법곡률  $\kappa_n$ 의 최소값  $\kappa_1$ , 최대값  $\kappa_2$ 은 주곡률이 되며, 주곡률이 될 때의 방향  $v$ 는 주방향  $\mathbf{e}_1$ 과  $\mathbf{e}_2$ 가 된다. 가우스 곡률 (Gaussian curvature)  $\kappa$ 는 두 주곡률의 곱으로 정의된다.

$$\kappa = \kappa_1 \kappa_2$$

### 2.2 자유곡면의 근사 토러스 패치 생성

자유곡면을 근사하는 토러스 패치를 생성하기 위해 다음 단계를 따른다.

1. 자유곡면을 근사하는 토러스를 생성한다.
2. 생성한 토러스 중 실제 알고리즘에서 사용할 영역을 잘라 토러스 패치를 생성한다.

자유곡면을 근사하는 토러스는 Liu 등[15]의 방법을 이용하여 생성한다. 자유곡면  $\mathbf{S}(u, v)$ 와 토러스 근사에 사용할 자유곡면 위의 점  $(u_0, v_0)$ 가 주어졌을 때 곡면 위의 점  $\mathbf{p} = \mathbf{S}(u_0, v_0)$ , 단위법선벡터  $\mathbf{N}$ , 주곡률  $\kappa_1, \kappa_2$ , 그리고 주방향  $\mathbf{e}_1, \mathbf{e}_2$ 를 구한다. 이 과정을 통해 중심이  $\mathbf{p} + \mathbf{N}/\kappa_1$ , 축의 방향은  $\mathbf{e}_2$ , 큰 반지름  $R = -1/\kappa_1 + 1/\kappa_2$ , 작은 반지름  $r = 1/|\kappa_2|$ 를 가지는, 곡면을 근사하는 토러스를 구할 수 있다.

회전축은 Z축이고, XY평면 위 원의 중심점까지의 거리를 큰 반지름  $R$ , 원 중심에서 토러스 곡면까지 거리인 작은 반지름을  $r$ 이라 할 때, 일반적인 토러스( $\hat{\mathbf{T}}$ )의 식은 다음과 같다.

$$\hat{\mathbf{T}}(s, t) = ((R + r \cos t) \cos s, (R + r \cos t) \sin s, r \sin t), \\ (s, t) \in [0, 2\pi] \times [0, 2\pi]$$

Figure 2은 직교 좌표계와 임의의 반지름  $R, r (R > r)$ 에 대한  $\hat{\mathbf{T}}(s, t)$ 를 나타내고 있다. 회전축의 방향을  $(0, 0, 1)$ 에서  $\mathbf{e}_2$ 로 변환하는 회전형렬  $\mathbf{r}$ 을 곱하고 중심을  $(0, 0, 0)$ 에서  $\mathbf{p} + \mathbf{N}/\kappa_1$ 로 이동하도록 이동행렬  $\mathbf{t}$ 을 더하면  $\mathbf{S}(u_0, v_0)$ 의 성분들로 근사한 최종적인 토러스  $\mathbf{T}(s, t)$ 를 구할 수 있다.

$$\mathbf{T}(s, t) = \mathbf{r}\hat{\mathbf{T}}(s, t) + \mathbf{t}, (s, t) \in [0, 2\pi] \times [0, 2\pi]$$

앞서 구한 토러스에서 알고리즘에 사용될 토러스 패치의 영역을 결정하기 위해 토러스 패치의 중심점과 토러스 패치의 범위를 결정해야 한다. 자유곡면의  $\mathbf{S}(u_0, v_0)$ 에 대응되는 토러스 패치의 중심점 좌표는  $(u_0, v_0)$ 에서의 가우스 곡률  $\kappa$ 에 따라 결정된다. 토러스의 내부는 가우스 곡률이 음수이고, 토러스의 외부의 가우스



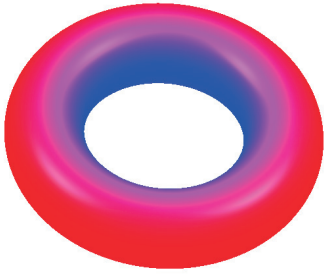
---

**Algorithm 1** Torus Patch Approximation

---

**Require:** Freeform surface  $\mathbf{S}(u, v)$ , and point  $(u_0, v_0)$ 

- 1:  $R \leftarrow -1/\kappa_1 + 1/\kappa_2$
  - 2:  $r \leftarrow 1/|\kappa_2|$
  - 3:  $\hat{\mathbf{T}} = \text{MakeGeneralTorus}(R, r)$  ▷ It makes a torus which center is  $(0,0,0)$ , the axis of rotation is  $(0,0,1)$ , the major radius is  $R$ , and the minor radius is  $r$ .
  - 4:  $\mathbf{c} \leftarrow \mathbf{S}(u_0, v_0) + \mathbf{N}/\kappa_1$
  - 5:  $\mathbf{r} \leftarrow$  Rotation matrix  $(0,0,1)$  to  $\mathbf{e}_2$
  - 6:  $\mathbf{t} \leftarrow$  Translation matrix  $(0,0,0)$  to  $\mathbf{c}$
  - 7:  $\mathbf{T} = \mathbf{r}\hat{\mathbf{T}} + \mathbf{t}$
- 



**Figure 3:** Rendering torus with Gaussian curvature. Red means positive curvature and blue means negative curvature.

곡률은 양수이다. Figure 3에서 가우스 곡률이 양인 영역은 붉은 색으로, 음인 영역은 푸른색으로 표시하였다. 따라서  $\mathbf{S}(u_0, v_0)$ 의 가우스 곡률이 양인 경우, 토러스의 외부(볼록한 면, Figure 6의 왼쪽 토러스 패치)에 대응되므로  $\mathbf{S}(u_0, v_0)$ 의 토러스 패치의 중심점 좌표  $(s, t) = (0, 0)$ 이 된다. 가우스 곡률이 음인 경우, 토러스 내부(오목한 면, Figure 6의 오른쪽 토러스 패치)에 대응되므로  $(s, t) = (0, \pi)$ 가 된다.

토러스 패치는  $\mathbf{S}(u_0, v_0)$ 의 성분들을 이용하여 생성하였으므로,  $\mathbf{S}(u_0, v_0)$ 의 주변 국소적인 부위에서는 토러스 패치가 자유곡면을 잘 근사한다. 하지만  $\mathbf{S}(u_0, v_0)$ 에서 멀어질수록 토러스 패치와 자유곡면 사이의 오차는 일반적으로 커진다. 따라서 자유곡면 전체를 하나의 토러스 패치로 근사하면 근사오차가 커지므로 토러스 패치를 사용하는 장점을 상쇄한다. 따라서 자유곡면을 매개변수 공간에서 균일한 간격으로 분할하여 자유곡면조각들을 생성하고, 자유곡면조각들을 근사하는 토러스 패치를 생성한다.

토러스 패치에서 사용하는 범위는 토러스 근사에 이용되는 자유곡면조각의 중심점  $(u_0, v_0)$ 을 중심으로 자유곡면조각의 위치정보와 법벡터정보를 충분히 포함하면서도, 자유곡면조각보다 지나치게 크지 않도록 조정한다. Figure 4은 자유곡면(파란색 wireframe)중 자유곡면조각(붉은색 면)의 성분들로 근사한 토러스 패치(붉은색 wireframe)를 나타낸다. Figure 5는 사용하는  $\mathbf{S}(u_0, v_0)$ 에 따라 생성된 토러스의 일부와 알고리즘에서 사용하는 토러스 패치(흰색 선)를 나타내고 있다. Figure 4와 Figure 5에서 보이는 것처럼, 생성되는 토러스의 전체 크기에 비해 실제 알고리즘에 사용하는 토러스 패치의 영역은 아주 국소적이다.

## 2.3 하이브리드 BVH

이 논문에서는 다양한 기하 연산을 지원하기 위한 BVH로서 사면체 BV와 토러스 패치 BV를 동시에 사용하는 하이브리드 BVH 구조를 사용한다. 앞서 자유곡면을 균일한 간격으로 분할한 자유곡면조각을 기준으로, 자유곡면조각에서 전체 자유곡면까지는 bottom-up 방식으로 사면체[35, 36]를 이용하여 BVH를 생성한다. 자유곡면조각들은 개별적인 토러스 패치로 근사시키고, 최대 근사오차가 근사오차한계인  $10^{-6}$ 보다 클 경우 그 자유곡면조각을 매개변수공간  $(u, v)$ 에서 각 축으로 이등분하여 4개의 자유곡면조각으로 분할한다. 분할한 4개의 자유곡면조각들을 다시 토러스 패치로 근사시키고, top-down 방식으로 이 토러스 패치들이 분할하기 전 토러스 패치와 계층적 구조를 가지게 한다. 이 과정을 모든 단말(leaf) 토러스 패치의 근사오차가 근사오차한계보다 작아질 때 까지 반복한다.

생성된 토러스 패치들은 자유곡면과 서로 매개변수를 알 수 있으므로 1:1로 쌍을 이룰 수 있으며, 단말 토러스 패치들로 형성된 토러스 패치 집합은 자유곡면 전체와 대응된다. 자유곡면조각들의 경계 주위에서는 여러 토러스 패치가 마치 물고기의 비늘처럼 겹칠 수 있지만, 자유곡면조각과 토러스 패치가 1:1로 대응되므로 자유곡면조각 자신에게 대응되는 토러스 패치가 어떤 것인지 쉽게 구분할 수 있다. 더불어 토러스 패치가 겹치는 영역으로 토러스 패치 간 위치정보를 파악하는데 용이하게 사용할 수 있다.

자유곡면을 자유곡면조각으로 분할할 때 큰 간격으로 분할한다면 BVH에서 사면체 BV들을 사용하는 단계가 적어지고, 최상위 단계의 토러스 패치 BV들의 근사오차가 커진다. 작은 간격으로 분할한다면 사면체 BV들을 사용하는 단계가 많아지고, 최상위 단계의 토러스 패치 BV들의 근사오차가 작아져 토러스 패치 BV들의 단계가 적어진다. 따라서 임의의 자유곡면이 주어졌을 때 자유곡면조각으로 분할하는 간격의 크기는 자유곡면의 차수, 허용오차의 크기, 그리고 하이브리드 BVH가 기하 연산을 해결하는데 소요되는 시간 등을 고려하여 실험적으로 결정하여야 한다.

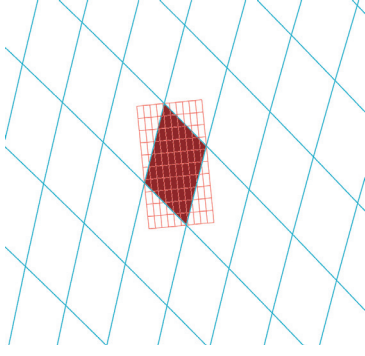
---

**Algorithm 2** Build Hybrid BVH

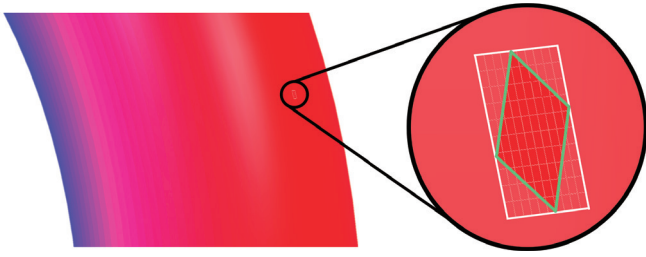
---

**Require:** Freeform surface  $\mathbf{S}$ 

- 1: Divide  $\mathbf{S}$  to  $\mathbf{S}_{00}, \mathbf{S}_{01}, \dots, \mathbf{S}_{mn}$
  - 2: Build hierarchical structure by tetrahedron BV ▷ Bottom-up
  - 3:  $\text{Stack.push}(\mathbf{S}_{00}, \mathbf{S}_{01}, \dots, \mathbf{S}_{mn})$
  - 4: **while** ! $\text{Stack.empty}$  **do**
  - 5:    $\mathbf{S}_{ij} = \text{Stack.top}$
  - 6:    $\text{Stack.pop}$
  - 7:    $\mathbf{T}_{ij} = \text{TorusPatchApproximation}(\mathbf{S}_{ij})$
  - 8:   **if**  $\text{GetError}(\mathbf{S}_{ij}, \mathbf{T}_{ij}) > \epsilon$  **then** ▷  $\epsilon$ : tolerance
  - 9:     Divide  $\mathbf{S}_{ij}$  to  $\mathbf{S}_{ij00}, \mathbf{S}_{ij01}, \mathbf{S}_{ij10}, \mathbf{S}_{ij11}$
  - 10:     Build hierarchical structure ▷ Top-down
  - 11:      $\text{Stack.push}(\mathbf{S}_{ij00}, \mathbf{S}_{ij01}, \mathbf{S}_{ij10}, \mathbf{S}_{ij11})$
  - 12:   **end if**
  - 13: **end while**
-



**Figure 4:** Torus patch approximates a small piece of freeform surface.



**Figure 5:** Torus patch uses a local region of torus.

## 2.4 양방향 하우스도르프 거리의 상한 계산

토러스 패치 집합과 자유곡면 사이의 양방향 하우스도르프 거리의 상한을 구하기 위해 이들의 최대근사오차를 먼저 계산할 필요가 있다. 이를 위해 토러스 패치가 근사하는 자유곡면상의 영역과 토러스 패치 사이의 오차를 모두 측정하고, 그 최대값을 구한다. 생성한 토러스  $\mathbf{T}(s, t)$  중 앞서 설정한 토러스 패치에 따라 토러스의 국소 범위만 사용하므로, 토러스 패치의 이용 범위를  $(p, q) \in [0, 1] \times [0, 1]$ 로 재매개화하여  $\mathbf{T}(s, t) = \mathbf{T}(s(p, q), t(p, q))$ 로 표현하고, 이 영역을 자유곡면에 투영시킨 영역을  $\mathbf{S}(u(p, q), v(p, q))$ 로 표현한다. Figure 7의 파란색  $\mathbf{T}(s(p, q), t(p, q))$ 는 토러스 패치를, 초록색 영역은 토러스 패치를 자유곡면  $\mathbf{S}(u, v)$ 에 투영시킨 영역  $\mathbf{S}(u(p, q), v(p, q))$ 을 나타낸다. 이때,  $(p, q)$ 로 재매개화된 토러스 패치는 토러스의 전체 영역에 비해 매우 국소적이므로, 계산의 편의성을 위해  $(p, q)$ 와  $(u, v)$ 사이의 관계는 아핀변환으로

가정한다.  $p, q$ 에서  $u, v$ 로 가는 아핀 변환이 다음과 같을 때

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} r_0 & r_1 & t_0 \\ r_2 & r_3 & t_1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p \\ q \\ 1 \end{pmatrix}$$

이를  $p, q$ 에 대해 편미분 한 결과는 아래와 같다.

$$\frac{\partial u}{\partial p} = r_0, \frac{\partial u}{\partial q} = r_1, \frac{\partial v}{\partial p} = r_2, \frac{\partial v}{\partial q} = r_3$$

이를 미분의 연쇄법칙을 이용하여 일차미분, 이차미분으로 확장시켜  $p, q$ 에 대한 미분값들을  $u, v$ 에 대한 미분값으로 표현할 수 있다.

$$\begin{aligned} \frac{\partial}{\partial p} \mathbf{S}(u(p, q), v(p, q)) &= \\ \frac{\partial \mathbf{S}}{\partial u} \cdot \frac{\partial u}{\partial p} + \frac{\partial \mathbf{S}}{\partial v} \cdot \frac{\partial v}{\partial p} &= r_0 \frac{\partial \mathbf{S}}{\partial u} + r_2 \frac{\partial \mathbf{S}}{\partial v} \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial q} \mathbf{S}(u(p, q), v(p, q)) &= \\ \frac{\partial \mathbf{S}}{\partial u} \cdot \frac{\partial u}{\partial q} + \frac{\partial \mathbf{S}}{\partial v} \cdot \frac{\partial v}{\partial q} &= r_1 \frac{\partial \mathbf{S}}{\partial u} + r_3 \frac{\partial \mathbf{S}}{\partial v} \end{aligned}$$

$$\begin{aligned} \frac{\partial^2}{\partial p^2} \mathbf{S}(u(p, q), v(p, q)) &= \\ r_0 \frac{\partial^2 \mathbf{S}}{\partial u^2} \cdot \frac{\partial u}{\partial p} + r_2 \frac{\partial^2 \mathbf{S}}{\partial v^2} \cdot \frac{\partial v}{\partial p} &= r_0^2 \frac{\partial^2 \mathbf{S}}{\partial u^2} + r_2^2 \frac{\partial^2 \mathbf{S}}{\partial v^2} \end{aligned}$$

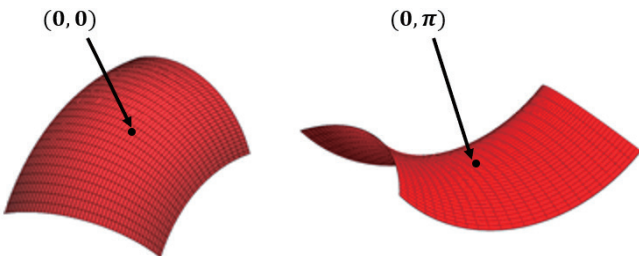
$$\begin{aligned} \frac{\partial^2}{\partial q^2} \mathbf{S}(u(p, q), v(p, q)) &= \\ r_1 \frac{\partial^2 \mathbf{S}}{\partial u^2} \cdot \frac{\partial u}{\partial q} + r_3 \frac{\partial^2 \mathbf{S}}{\partial v^2} \cdot \frac{\partial v}{\partial q} &= r_1^2 \frac{\partial^2 \mathbf{S}}{\partial u^2} + r_3^2 \frac{\partial^2 \mathbf{S}}{\partial v^2} \end{aligned}$$

$$\begin{aligned} \frac{\partial^2}{\partial p \partial q} \mathbf{S}(u(p, q), v(p, q)) &= \\ r_0 \frac{\partial^2 \mathbf{S}}{\partial u^2} \cdot \frac{\partial u}{\partial q} + r_2 \frac{\partial^2 \mathbf{S}}{\partial v^2} \cdot \frac{\partial v}{\partial q} &= r_0 r_1 \frac{\partial^2 \mathbf{S}}{\partial u^2} + r_2 r_3 \frac{\partial^2 \mathbf{S}}{\partial v^2} \end{aligned}$$

토러스 패치를  $n^2$ 개의 아주 작은 이중선형곡면으로 분할한 뒤, 앞서 구한 미분값들을 이용하여 Filip 등[33]의 방법에 따라 최대 근사오차를 측정할 수 있다.

$$\begin{aligned} \sup_{p, q \in [0, 1]} \|\mathbf{S}(u(p, q), v(p, q)) - \mathbf{T}(s(p, q), t(p, q))\| \\ \leq \frac{1}{8n^2} (M_1 + 2M_2 + M_3), \end{aligned}$$

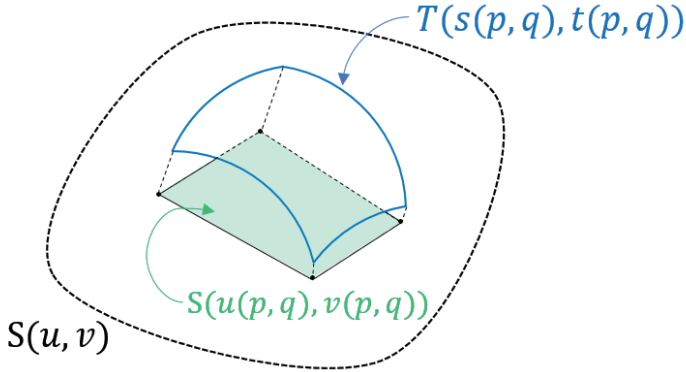
$$M_1 = \sup_{p, q \in [0, 1]} \left\| \frac{\partial^2 \mathbf{S}(u(p, q), v(p, q))}{\partial p^2} \right\|$$



**Figure 6:** Examples of torus patch.

$$M_2 = \sup_{p,q \in [0,1]} \left\| \frac{\partial^2 \mathbf{S}(u(p,q), v(p,q))}{\partial p \partial q} \right\|$$

$$M_3 = \sup_{p,q \in [0,1]} \left\| \frac{\partial^2 \mathbf{S}(u(p,q), v(p,q))}{\partial q^2} \right\|$$



**Figure 7:** Error measurement between a freeform surface and a torus patch.

토러스 패치 집합상의 임의의 점에서 자유곡면까지의 최단거리는 항상 최대근사오차보다 작거나 같다. 반대로, 자유곡면상의 임의의 점에서 토러스 패치 집합까지의 최단거리도 항상 최대근사오차보다 작거나 같다. 따라서 토러스 패치 집합을 구성하는 각각의 토러스 패치와 자유곡면 사이의 양방향 하우스도르프 거리 상한은 항상 그들의 최대근사오차보다 작거나 같다. 또한 토러스 패치의 최대근사오차가  $10^{-6}$  이상일 경우 자유곡면을 추가로 분할하여 계층적 구조를 가지게 하는 2.2절의 알고리즘에 따라, 양방향 하우스도르프거리의 상한은 항상  $10^{-6}$ 보다 작음이 보장된다.

## 2.5 오프셋 곡면 계산

토러스 패치 근사 기법을 사용하면 자유곡면의 오프셋 곡면을 쉽게 구할수 있다. 이는 토러스의 오프셋 곡면이 오목, 또는 볼록의 형태는 그대로 유지되면서 곡률의 크기만 달라지기 때문이다. 앞서 구한 토러스에서 작은 반지름에 오프셋 값을 더한 값을  $r_0$ 이라 할 때, 토러스의 오프셋 곡면  $\mathbf{T}_o(s, t)$ 는 다음과 같이 정의된다.

$$\widehat{\mathbf{T}}_o(s, t) = ((R + r_o \cos t) \cos s, (R + r_o \cos t) \sin s, r_o \sin t)$$

$$\mathbf{T}_o(s, t) = \mathbf{r} \widehat{\mathbf{T}}_o(s, t) + \mathbf{t}, (s, t) \in [0, 2\pi] \times [0, 2\pi]$$

오프셋 곡면의 회전행렬  $\mathbf{r}$ 과 이동행렬  $\mathbf{t}$ 는 입력 곡면의 회전행렬, 이동행렬과 동일한 행렬이다. 이런 토러스의 성질로, 오프셋한 토러스 패치 집합과 오프셋한 자유곡면간의 양방향 하우스도르프 거리의 상한은 입력 곡면과 토러스 패치 집합과의 양방향 하우스도르프 거리의 상한과 같아 오프셋한 토러스 패치 집합의 최대근사오차를 쉽게 계산할 수 있는 토러스 패치 근사 기법의 장점을 확인할 수 있다.

## 3. 실험결과

### 3.1 예제

토러스 패치 근사 기법의 근사정밀도를 측정하기 위해 세개의 자유곡면  $S(u, v)$ 에서  $u, v$  각 방향으로 128개씩 나누고 하이브리드 BVH 구조를 생성하였다. Figure 9의 예제 1, 2는 3차 베지어 곡면이며 예제 3은 2차 베지어 곡면이다. 문제의 크기를 동일하게 하기 위해 베지어 곡면의 제어점을 감싸는 바운딩 박스의 대각선 길이가 1이 되도록 조정하였다. 각 예제의 제어점은 다음과 같다.

#### • 예제 1

$$\frac{1}{83.666} \begin{bmatrix} (0, 0, 0) & (0, 10, 20) & (0, 20, 20) & (0, 30, 0) \\ (20, 0, 20) & (20, 10, 50) & (20, 20, 50) & (20, 30, 20) \\ (40, 0, 20) & (40, 10, 50) & (40, 20, 50) & (40, 30, 20) \\ (60, 0, 0) & (60, 10, 20) & (60, 20, 20) & (60, 30, 0) \end{bmatrix}$$

#### • 예제 2

$$\frac{1}{78.102} \begin{bmatrix} (0, 0, 0) & (0, 10, 20) & (0, 20, 20) & (0, 30, 0) \\ (20, 0, -20) & (20, 10, -20) & (20, 20, -20) & (20, 30, -20) \\ (40, 0, 20) & (40, 10, 20) & (40, 20, 20) & (40, 30, 20) \\ (60, 0, 0) & (60, 10, 20) & (60, 20, 20) & (60, 30, 0) \end{bmatrix}$$

#### • 예제 3

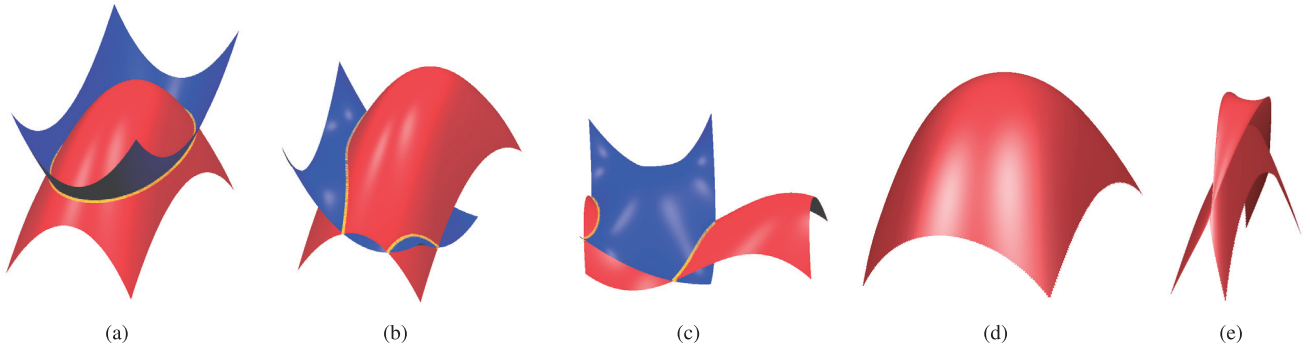
$$\frac{1}{48.990} \begin{bmatrix} (0, 0, 10) & (0, 10, 0) & (0, 20, 20) \\ (10, 0, 0) & (10, 10, 40) & (10, 20, 0) \\ (20, 0, 0) & (20, 10, 0) & (20, 20, 20) \end{bmatrix}$$

### 3.2 결과

세 개의 자유곡면 각각에 대해 토러스 패치 집합을 이용하여 모델링하였다. Figure 9는 3개의 예제 자유곡면과 토러스 패치 집합으로 감싼 모습을 나타낸다. 토러스 패치들의 구별을 위해, 각 토러스 패치들이 임의의 색으로 렌더링 되도록 하였다. Figure 9(a)는 주곡률이 양인 자유곡면이며, Figure 9(b)-(c)는 주곡률이 양과 음인 지점을 모두 포함하는 자유곡면이다. 토러스 패치가 자유곡면의 볼록, 오목 여부에 영향을 받지 않고 자유곡면을 잘 근사함을 확인할 수 있다.

Table 1에서는 각 자유곡면과 토러스 패치 집합에 대해 최대 근사오차를 측정하였다. 3개의 자유곡면 모두 최대근사오차가  $10^{-6}$ 보다 작으므로, 자유곡면과 토러스 패치 집합간의 양방향 하우스도르프 거리가  $10^{-6}$ 보다 작음을 알 수 있다.

Figure 8에서는 토러스 패치 근사 기법을 이용한 기하 연산을 수행한 예제를 제시하였다. Figure 8(a)는 예제 1 곡면(붉은색)과 이를 뒤집은 곡면(푸른색)을, Figure 8(b)는 예제 1 곡면과 예제



**Figure 8:** (a)-(c) Intersection curve of two freeform surfaces, (d) outward offset surface, (e) and inward offset surface.

3 곡면을, Figure 8(c)는 예제 2 곡면과 예제 3 곡면을 서로 교차하게 만들고 토러스 패치만을 이용하여 교차곡선(노란색)을 계산한 결과이다. 토러스 패치 내부에서 교차곡선 점들의 추적을 위해 marching method와 subdivision method[37, 38]를 활용하였다. 교차곡선상의 점들은 두 자유곡면과의 거리가  $10^{-6}$  이하이다. Figure 8(d)-(e)는 예제 1 자유곡면을 토러스 패치 근사 기법으로 토러스 패치 집합을 생성한 후 각각 0.15, -0.15씩 오프셋한 결과이다. 아직 자기교차(self-intersection)곡선은 제거하지 않은 상태의 예제이다.

**Table 1:** Maximum, average, minimum value of maximum approximation error

최대근사오차	최대	평균	최소
예제 1	$5.792^{-9}$	$4.797^{-9}$	$2.667^{-9}$
예제 2	$1.501^{-8}$	$4.411^{-9}$	$4.168^{-10}$
예제 3	$5.536^{-9}$	$2.605^{-9}$	$1.734^{-10}$

## 4. 확장 가능한 주제들

### 4.1 다양한 기하 연산들의 가속화

토러스 패치 근사 기법을 이용하여 두 자유곡면의 교차곡선을 계산하였다. 향후 자유곡면에 속하는 토러스 패치간의 상관관계를 파악하고, 토러스간 교차곡선을 계산하는 이전 연구들[39, 40]을 확장할 수 있다. 이를 통해 더욱 정밀한 두 자유곡면의 교차곡선을 계산하고, 접하는 곡면들 사이의 교차곡선 계산 등 수치적으로 대단히 불안정한 경우의 문제들을 해결할 수 있을 것으로 기대된다.

오프셋 곡면에서는 자기교차를 제거하는 것이 중요한 문제로 남아있다. 입력 곡면에서 곡률이 높은 부분은 오프셋 크기가 커짐에 따라 국부적으로 꼬이는 오프셋 곡면을 생성하므로 자기교차가 일어난다. 이런 자가교차점이 발생하는  $(u, v)$ 의 집합이 아주 복잡한 위상을 가지기 때문에, 기존의 그리드 기반 접근 방법으로는 해상도의 문제로 인하여 정밀도가 높게 문제를 해결하는 데

한계가 있었다. 이 문제에 토러스 패치 근사 기법을 활용한다면, 고정밀도의 계산을 효율적이고 안정적으로 수행할 수 있을 것이다. 이를 응용하여 오프셋 크기를 변수로 고려하여, 오프셋 곡면들로 구성되는 3차원 볼륨 모델링으로 결과를 확장할 수 있다.

3차원 물체에 대한 보로노이 다이어그램(Voronoi Diagram)이나 중심축(Medial Axis) 계산을 하기 위해서는 서로 마주 보는 두 곡면 사이의 중간위치에 있는 이등분(Bisector) 곡면 계산을 정확하게 해야 한다. 이 곡면은 그 차수가 매우 높을 뿐만 아니라 근사 오차를 측정하기 어려워 안정적인 알고리즘의 개발이 어려웠다. 이를 근사정밀도가 높은 토러스 패치 근사 기법을 사용하여 접근한다면 거리계산 문제의 기하학적인 근사오차를 분석하는 데 큰 장점이 있을 것으로 예상된다.

최근 3차원 모델링에서 중요한 연구 주제가 된 사각 메시에서 사각형의 방향은 주방향을 따라가도록 생성되는 경우가 일반적이다. 하지만 곡면을 근사하는 사각 메시는 근사 과정에서 법벡터 정보가 부정확해진다. 또한 사각 메시지를 이루는 면인 쌍선형곡면(bilinear surface)은 항상 곡률이 음이므로, 자유곡면 중 곡률이 양인 부분을 표현하는데 한계가 있다. 이를 토러스 패치 근사 기법으로 접근한다면 향후 사각 메시에 대해서도 법벡터 정보를 유지하면서 매우 높은 근사정밀도를 가질 것으로 예상된다. 이를 통하여 자유곡면과 사각 메시 사이에 적용되는 다양한 알고리즘 개발에 중요한 역할을 할 것으로 기대된다.

## 5. 결론

이 논문에서는 3차원상의 자유곡면에 대한 효율적인 기하 연산을 지원하기 위한 새로운 자료구조인 토러스 패치 기반의 정밀 근사를 이용하는 자유곡면의 기하학적 처리 기법을 제안하였다. 이 기법으로 생성한 토러스 패치 집합과 자유곡면 사이의 양방향 하우스도르프 거리가 근사오차한계인  $10^{-6}$  보다 작음을 보였다. 두 자유곡면의 교차곡선과 자유곡면의 오프셋 곡면을 모델링하여 토러스 패치의 정밀 근사를 이용한 자유곡면의 기하학적 처리 기법이 다양한 알고리즘의 개발에 활용될 수 있음을 보였다.



## 감사의 글

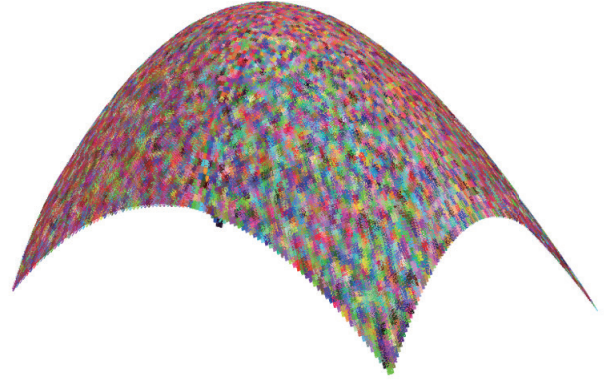
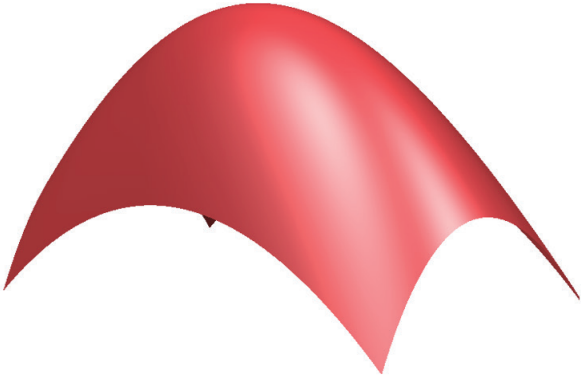
이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2019R1A2C1003490).

## References

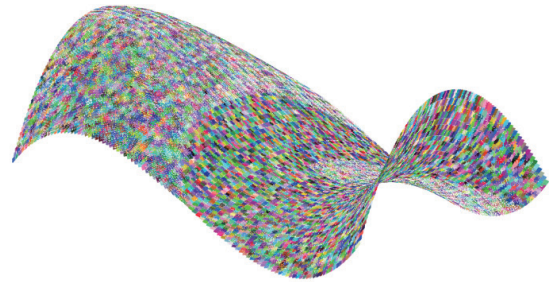
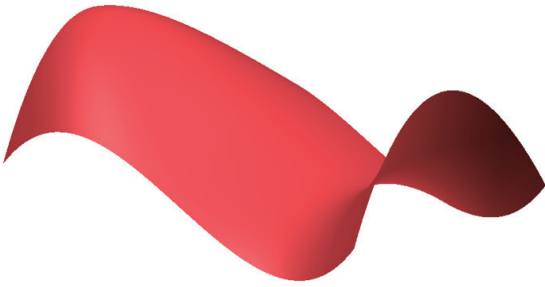
- [1] S. Gottschalk, M. C. Lin, and D. Manocha, "OBBTree A Hierarchical Structure for Rapid Interference Detection," *ACM Transactions on Graphics (SIGGRAPH 1996)*, vol. 15, no. 3, pp. 171–180, 1996.
- [2] E. Larsen, S. Gottschalk, M. Lin, and D. Manocha, "Fast Proximity Queries with Swept Sphere Volumes," Dept. of Computer Science, UNC, Chapel Hill, NC, Tech. Rep., 1999.
- [3] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha, "Fast distance queries with rectangular swept sphere volumes," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 4, 2000, pp. 3719–3726.
- [4] D. S. Meek and D. J. Walton, "Approximating smooth planar curves by arc splines," *Journal of Computational and Applied Mathematics*, vol. 59, no. 2, pp. 221–231, 1995.
- [5] M. Barton and G. Elber, "Spiral fat arcs—Bounding regions with cubic convergence," *Graphical Models*, vol. 73, no. 2, pp. 50–57, 2011.
- [6] Y.-J. Kim, Y.-T. Oh, J. Lee, S.-H. Yoon, and M.-S. Kim, "Data structures for accelerating geometric operations on freeform objects," in *Korea Computer Graphics Society 2011 Proceeding*, 2011, pp. 71–72.
- [7] Y.-J. Kim, J. Lee, M.-S. Kim, and G. Elber, "Efficient offset trimming for planar rational curves using biarc trees," *Computer Aided Geometric Design*, vol. 29, no. 7, pp. 555–564, 2012.
- [8] J. Lee, Y.-J. Kim, M.-S. Kim, and G. Elber, "Efficient offset trimming for deformable planar curves using a dynamic hierarchy of bounding circular arcs," *Computer-Aided Design*, vol. 58, pp. 248–255, 2015.
- [9] F. Aurenhammer, "Voronoi Diagrams — A Survey of a Fundamental Geometric Data Structure," *ACM Computing Surveys (CSUR)*, vol. 23, no. 3, pp. 345–405, 1991.
- [10] O. Aichholzer, W. Aigner, F. Aurenhammer, T. Hackl, B. Jüttler, and M. Rabl, "Medial axis computation for planar free-form shapes," *Computer-Aided Design*, vol. 41, no. 5, pp. 339–349, 2009.
- [11] O. Aichholzer, W. Aigner, F. Aurenhammer, T. Hackl, B. Jüttler, E. Pilgerstorfer, and M. Rabl, "Divide-and-conquer for Voronoi diagrams revisited," *Computational geometry*, vol. 43, no. 8, pp. 688–699, 2010.
- [12] J. Lee, Y.-J. Kim, M.-S. Kim, and G. Elber, "Efficient voronoi diagram construction for planar freeform spiral curves," *Computer Aided Geometric Design*, vol. 43, pp. 131–142, 2016.
- [13] S. Krishnan, M. Gopi, M. Lin, D. Manocha, and A. Pattekar, "Rapid and Accurate Contact Determination between Spline Models using ShellTrees," in *Computer Graphics Forum*, vol. 17, no. 3, 1998, pp. 315–326.
- [14] G. Bradshaw and C. O'Sullivan, "Sphere-Tree Construction using Dynamic Medial Axis Approximation," in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2002, pp. 33–40.
- [15] X.-M. Liu, L. Yang, J.-H. Yong, H.-J. Gu, and J.-G. Sun, "A torus patch approximation approach for point projection on surfaces," *Computer Aided Geometric Design*, vol. 26, no. 5, pp. 593–598, 2009.
- [16] S.-M. Hu and J. Wallner, "A second order algorithm for orthogonal projection onto curves and surfaces," *Computer Aided Geometric Design*, vol. 22, no. 3, pp. 251–260, 2005.
- [17] D.-S. Kim, P. Y. Papalambros, and T. C. Woo, "Tangent, normal, and visibility cones on Bézier surfaces," *Computer Aided Geometric Design*, vol. 12, no. 3, pp. 305–320, 1995.
- [18] M. Daniel, "Using a Convex Pyramid to Bound Surface Normal Vectors," in *Computer graphics forum*, vol. 15, no. 4, 1996, pp. 219–227.
- [19] K.-R. Park and G.-I. Kim, "Offsets of ruled surfaces," *Journal of the Korea Computer Graphics Society*, vol. 4, no. 2, pp. 69–75, 1998.
- [20] R. T. Farouki, "The approximation of non-degenerate offset surfaces," *Computer Aided Geometric Design*, vol. 3, no. 1, pp. 15–43, 1986.
- [21] G. Elber and E. Cohen, "Error bounded variable distance offset operator for free form curves and surfaces," *International*



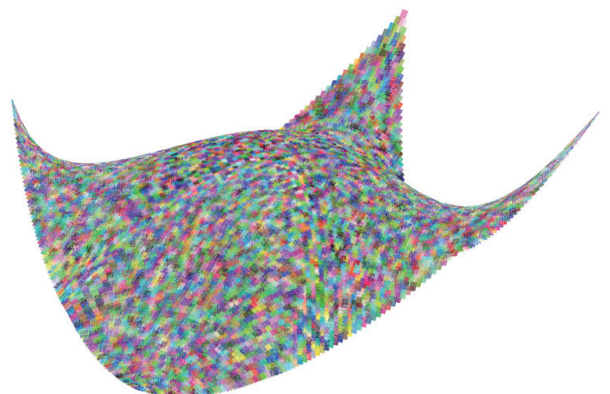
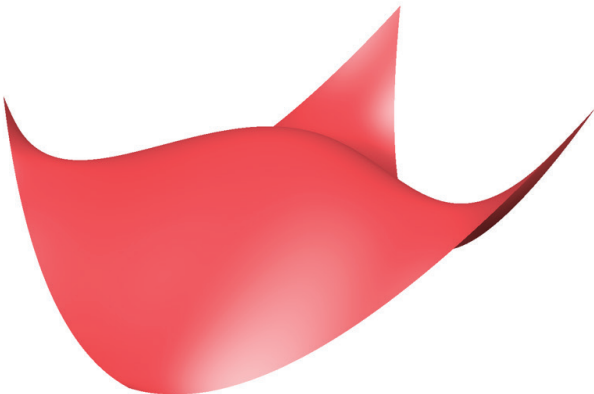
- Journal of Computational Geometry & Applications*, vol. 1, no. 01, pp. 67–78, 1991.
- [22] G. Elber, I.-K. Lee, and M.-S. Kim, “Comparing offset curve approximation methods,” *IEEE computer graphics and applications*, vol. 17, no. 3, pp. 62–71, 1997.
- [23] R. Martin, “Principal patches-a new class of surface patch based on differential geometry,” in *Eurographics*, vol. 83, 1983, pp. 47–55.
- [24] R. T. Farouki, “Exact offset procedures for simple solids,” *Computer Aided Geometric Design*, vol. 2, no. 4, pp. 257–279, 1985.
- [25] M. Tang, M. Lee, and Y. J. Kim, “Interactive hausdorff distance computation for general polygonal models,” *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, p. 74, 2009.
- [26] M. Bartoň, I. Hanniel, G. Elber, and M.-S. Kim, “Precise hausdorff distance computation between polygonal meshes,” *Computer Aided Geometric Design*, vol. 27, no. 8, pp. 580–591, 2010.
- [27] Y.-J. Kim, Y.-T. Oh, S.-H. Yoon, M.-S. Kim, and G. Elber, “Precise Hausdorff distance computation for planar freeform curves using biarcs and depth buffer,” *The Visual Computer*, vol. 26, no. 6-8, pp. 1007–1016, 2010.
- [28] M. Guthe, P. Borodin, and R. Klein, “Fast and accurate hausdorff distance calculation between meshes,” *Journal of WSCG*, vol. 13, no. 2, 2005.
- [29] Y. Kang, M.-H. Kyung, S.-H. Yoon, and M.-S. Kim, “Fast and robust Hausdorff distance computation from triangle mesh to quad mesh in near-zero cases,” *Computer Aided Geometric Design*, vol. 62, pp. 91–103, 2018.
- [30] I. Hanniel, A. Krishnamurthy, and S. McMains, “Computing the Hausdorff distance between NURBS surfaces using numerical iteration on the GPU,” *Graphical Models*, vol. 74, no. 4, pp. 255–264, 2012.
- [31] A. Krishnamurthy, S. McMains, and I. Hanniel, “GPU-accelerated Hausdorff distance computation between dynamic deformable NURBS surfaces,” *Computer-Aided Design*, vol. 43, no. 11, pp. 1370–1379, 2011.
- [32] Y.-J. Kim, Y.-T. Oh, S.-H. Yoon, M.-S. Kim, and G. Elber, “Efficient Hausdorff distance computation for freeform geometric models in close proximity,” *Computer-Aided Design*, vol. 45, no. 2, pp. 270–276, 2013.
- [33] D. Filip, R. Magedson, and R. Markot, “Surface algorithms using bounds on derivatives,” *Computer Aided Geometric Design*, vol. 3, no. 4, pp. 295–311, 1986.
- [34] M. P. Do Carmo, *Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition*, 2016.
- [35] Y. Kang, J. Jang, and M.-S. Kim, “Deformable quad mesh for accelerated geometric operations,” in *Korea Computer Graphics Society 2015 Proceeding*, 2011, pp. 73–74.
- [36] P. Du, Y. J. Kim, and S.-E. Yoon, “TSS BVHs: tetrahedron swept sphere BVHs for ray tracing subdivision surfaces,” *Computer Graphics Forum*, vol. 35, no. 7, pp. 279–288, 2016.
- [37] N. M. Patrikalakis, “Surface-to-surface intersections,” *IEEE Computer Graphics and Applications*, vol. 13, no. 1, pp. 89–95, 1993.
- [38] K.-J. Kim, J.-K. Seong, and M.-S. Kim, “Computing intersection between freeform surfaces,” *Journal of the Korea Computer Graphics Society*, vol. 10, no. 3, pp. 28–33, 2004.
- [39] K.-J. Kim, M.-S. Kim, and K. Oh, “Torus/Sphere Intersection Based on a Configuration Space Approach,” *Graphical Models and Image Processing*, pp. 77–92, 1998.
- [40] K.-J. Kim, “Circles in torus–torus intersections,” *Journal of Computational and Applied Mathematics*, vol. 236, no. 9, pp. 2387–2397, 2012.
- [41] G. Elber and T. Grandine, “Hausdorff and Minimal Distances between Parametric Freeforms in  $\mathbb{R}^2$  and  $\mathbb{R}^3$ ,” in *International Conference on Geometric Modeling and Processing*, 2008, pp. 191–204.



(a)



(b)



(c)

**Figure 9:** Freeform surfaces and sets of torus patches: (a) positive curvature, (b)-(c) both positive and negative curvatures.

## 〈 저 자 소 개 〉



### 박 영 진

- 2016년 부산대학교 정보컴퓨터공학부 졸업(학사)
- 2016년~현재 서울대학교 컴퓨터공학부 석박사통합과정
- 관심분야: 컴퓨터그래픽스, 기하 모델링
- <https://orcid.org/0000-0002-3893-5421>



### 홍 규 연

- 2006년 서울대학교 컴퓨터공학부 졸업(학사)
- 2015년 Carnegie Mellon University Computer Science Dept. 졸업(석사)
- 2016년~현재 서울대학교 컴퓨터공학부 박사과정
- 관심분야: 컴퓨터그래픽스, 3차원 기하 모델링
- <https://orcid.org/0000-0002-3411-049X>



### 김 명 수

- 1980년 서울대학교 수학교육학과 졸업(학사)
- 1982년 서울대학교 수학과 졸업(석사)
- 1985년 Purdue University Dept. of Mathematics 졸업(석사)
- 1987년, 1988년 Purdue University Dept. of Computer Science 졸업(석사, 박사)
- 1989년~1999년 포항공과대학교 컴퓨터공학과 교수
- 1999년~현재 서울대학교 컴퓨터공학부 교수
- 2003년~2005년 서울대학교 컴퓨터 연구소장
- 2005년~2006년 서울대학교 컴퓨터공학부장
- 2007년~2008년 서울대학교 정보화본부장
- 관심분야: 기하 모델링, 기하 알고리즘, 변형 애니메이션
- <https://orcid.org/0000-0003-4755-5727>