

캐널곡면의 복원 (Reconstruction of Canal Surfaces)

이 인 권 [†] 김 구 진 ^{**}
(In-Kwon Lee) (Ku-Jin Kim)

요약 본 논문에서는 입력으로 주어진 점집합 (point cloud)으로부터 캐널곡면을 복원 (reconstruction) 하는 알고리즘을 제시한다. 캐널곡면은 반경이 변화하며 중심점이 기준곡선(spine curve)을 따라 이동하는 구(moving sphere)의 스윕트곡면(swept surface)이다. 이 논문에서는 수축기법(shrinking method)과 moving least-square 방법을 이용하여 주어진 점집합을 세곡선(thin-curve)형태의 점집합으로 수축시킴으로써 캐널곡면의 기준곡선을 근사한다. 근사된 기준곡선과 입력으로 주어진 점집합에 포함된 점들 사이의 거리를 이용하여, 캐널곡면을 구성하는 이동 구의 반경을 계산한다.

키워드 : 캐널곡면, 곡면 복원, 기하 모델링

Abstract We present a method to reconstruct a canal surface from a point cloud (a set of unorganized points). A canal surface is defined as a swept surface of a moving sphere with varying radii. By using the shrinking and moving least-squares methods, we reduce a point cloud to a thin curve-like point set which can be approximated to the spine curve of a canal surface. The distance between a point in the thin point cloud and a corresponding point in the original point set represents the radius of the canal surface.

Key words : Canal surface, surface reconstruction, geometric modeling

1. 서론

캐널곡면(canal surface)은 토러스(torus), 듀핀 싸이클라이드(Dupin cyclide), 2차곡면(natural quadric), 파이프곡면(pipe surface) 등을 포함하는 곡면으로서 CAD/CAM 또는 컴퓨터그래픽스에서 물체의 모델링(modeling)에 자주 사용되며 곡면의 블렌딩(blending)에 있어서도 반드시 필요한 곡면이다[1,2]. 많은 경우 캐널곡면만으로도 물체를 모델링할 수 있고, 특히 둥그스름하고 긴 형태의 물체를 나타낼 경우 캐널곡면이 가장 적합하게 사용된다. 그림 1은 캐널곡면만으로 구성된 모델들을 예로 제시한다. 캐널곡면은 반경이 변화하며 중심점이 곡선을 따라 이동하는 구(sphere)의 스윕트 곡면(swept surface)으로 정의할 수 있으며, 중심점의 궤적 $C(t)$ 와 반경 함수 $r(t)$ 로 표현된다.

본 논문에서는, 점들 간의 순서나 이웃에 대한 정보가

없는 상태의 점집합(point cloud)으로부터 캐널곡면을 복원(reconstruction)하는 방법을 제시한다. 주어진 점집합은 캐널곡면으로 구성된 모델의 표면 상에 분포하는 점들의 집합이라고 가정한다. 이러한 문제에 대해서, 입력되는 점집합은 일반적으로 모델을 3차원 스캐너(scanner)로 스캔하여 얻어진다.

관련 연구로는 역설계(reverse engineering)분야에서 나선형곡면(helical surface)과 회전체(surface of revolution)의 복원[3], 프로파일곡면(profile surface)의 복원[4], developable 곡면의 복원[5], planar face의 복원[6]이 있으며, Pottmann 등[7,8]이 이러한 연구들을 자세히 소개하고 있다. Ueng 등[9]은 평행이동 스윕핑(sweeping)에 의해 생성된 스윕트곡면(swept surface)의 복원 방법을 연구했다. 기존의 복원 연구는 주어진 점집합을 다각형 메쉬(polygonal mesh)나 스플라인곡면(spline surface)으로 출력하는 데 중점을 둔 것에 비해 Ueng 등의 연구는 프로파일곡선(profile curve)을 복원한 뒤 kinematic motion을 이용하여 이동하는 프로파일곡선의 궤적을 구하는 방법을 사용한다. 이들의 방법을 사용하여 얻을 수 있는 결과 곡면은 절차적으로 정의된 기하학적인 모델(geometric model)이므로, 결과 곡면의 저장에 필요한 공간을 절약하고 물체의 디자인(design)

* 본 연구는 정보통신부 대학 IT연구센터 육성지원사업의 연구결과로 수행되었습니다.

[†] 정 회 원 : 연세대학교 컴퓨터과학과 교수
iklee@yonsei.ac.kr

^{**} 비 회 원 : 경북대학교 컴퓨터공학과 교수
kujinkim@knu.ac.kr

논문접수 : 2004년 6월 26일
심사완료 : 2005년 5월 4일

및 조작에 편의성을 제공하였다. Ramamoorthi와 Arvo [10]는 파이프곡면과 캐널곡면을 포함하여 다양한 모델을 복원하기 위한 시스템을 제안하였으며, 이 시스템은 주어진 점집합에 미리 사용자가 정의한 계층구조를 적용함으로써 복원을 수행한다.

본 논문에서는 이전에 제시되었던 파이프곡면의 복원 방법[11]을 확장하여, 점집합이 주어질 때 사용자의 관여를 최소화하면서 캐널곡면을 복원하는 방법으로서 “수축(shrinking)” 기법을 제시한다. 주어진 점집합은 여러 개의 캐널곡면으로 구성된 물체의 표면으로부터 얻어졌다고 가정한다. 주어진 점집합이 S 라고 할 때, S 는 수축 알고리즘을 적용하여 굵기가 가는 점집합(thin-point cloud) \tilde{S} 로 변형되며, 이후 \tilde{S} 는 moving least-square 방법을 적용하여 가는 곡선과 같은 형태(thin curve-like shape)의 점집합 \tilde{S} 로 변환된다. \tilde{S} 는 캐널곡면의 기준곡선(spine curve)으로 근사되고, S 에 포함된 각각의 점과 여기에 대응하는 \tilde{S} 의 점 사이의 거리를 이용하여 캐널곡면의 반경함수를 계산한다. 수축 알고리즘은 국소적 선형 최적화(local linear optimization)기법만을 사용하므로 iteration을 사용하는 반복 계산이 필요하지 않으며 닫힌 해(closed form solution)를 구할 수 있다. 따라서, 기존의 비선형 최적화(non-linear optimization)기법을 이용한 연구들[10]에 비해 더욱 효율적이고 안정적이다.

본 논문의 구성은 다음과 같다. 제2절에서는 주어진 점집합에서 캐널곡면을 복원하기 위한 알고리즘을 설명한다. 제3절에서는 캐널곡면의 복원에 대한 실험결과를 보이고, 제4절에서는 결론을 내리고 향후 연구방향을 제시한다.



그림 1 캐널곡면만으로 구성된 물체들

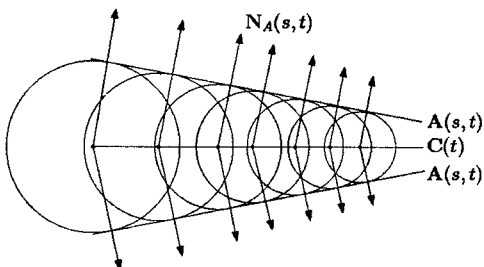


그림 2 캐널곡면의 대칭적인 법선벡터들

2. 캐널곡면의 복원

Paternell과 Pottmann[2]은 $C(t)$ 와 $r(t)$ 가 rational 곡선일 때 해당 캐널곡면 역시 rational 곡면으로 표현될 수 있음을 증명했다. 그림 2에서는 캐널곡면이 반경이 변하면서 중심점이 기준 곡선을 따라 이동하는 이동구(moving sphere)의 envelope 곡면임을 보인다. 여기에서 이동 구와 캐널곡면이 접하는 부분을 특징원(characteristic circle)이라 한다. 특징원에서 캐널곡면이 갖는 법선벡터와 이동 구의 법선 벡터가 일치한다는 것을 볼 수 있다. 캐널곡면에 포함된 하나의 특징원 상의 각 점에서 법선벡터의 방향을 따라 직선을 구성하면, 이러한 직선의 집합은 하나의 원뿔을 이룬다. 이 때 원뿔의 정점은 특징원을 포함하는 이동 구의 중심점에 위치하며, 동시에 캐널곡면의 기준곡선 위에 존재한다. 이러한 성질을 이용하여 기준곡선 $C(t)$ 와 반경함수 $r(t)$ 를 갖는 캐널곡면을 매개변수곡면 $A(s, t)$ 로 표현하면 $A(s, t) = C(t) + r(t)N_A(s, t)$ 이며, 여기에서 변수 s 와 t 가 각각 $s \in I_1$, $t \in I_2$ 이고 $N_A(s, t)$ 는 캐널곡면의 단위 법선벡터(unit normal vector)이다.

주어진 점집합을 집합 $S = \{P_i | i = 1, \dots, n\}$ 로 나타내며, S 에 속한 각 점에 대한 단위 법선벡터가 미리 계산되어 집합 N 으로 다음과 같이 표기된다고 가정한다.

$$N = \{N_i | i = 1, \dots, n\}$$

주어진 점들의 법선벡터를 구하는 문제는 역설계 분야에서 활발히 연구가 되어왔으며 다양한 방법이 알려져 있다[12,13]. 본 연구에서는 각 점 P_i 에서 단위 법선벡터 N_i 를 계산하기 위하여 다음과 같은 방법을 사용하였으며, 계산된 단위 법선벡터들은 캐널곡면의 바깥 방향을 향한다고 가정한다.

- i) 각 점 P_i 마다 이웃 점(neighboring point)들의 집합 T_i 를 계산한다.
- ii) Least-squares 방법에 의해 T_i 의 회귀평면(regression plane)을 계산한다.
- iii) 회귀평면의 단위 법선벡터를 N_i 로 정한다.

입력으로 주어진 점집합은 자체적인 교차(self-intersection)가 없는 정규곡면(regular surface)인 캐널곡면으로부터 얻어졌다고 가정한다. 주어진 점집합으로부터 캐널곡면을 복원하는 과정은 수축, 기준곡선의 근사, 반경함수 계산의 세 단계로 구성된다.

1) 수축 알고리즘

수축 알고리즘은 각각의 점 P_i 마다 P_i 의 법선벡터 N_i 와 거의 유사한 법선벡터를 갖는 이웃 점들이 존재한다는 사실을 이용하여 구성되었다. 이러한 이웃 점들의 집

합 T_i 는 다음과 같이 나타낼 수 있다.

$$T_i = \{P_j | \|P_i - P_j\| < R, P_j \in S\}$$

여기에서 R 은 매우 작은 상수값이며, 사용자가 입력하는 값이다. T_i 는 다음과 같은 성질을 갖는다.

$$N_i \cdot N_j > 0, \text{ for all } P_j \in T_i \quad (1)$$

S 에 포함된 각 점 P_i 를 N_i 의 반대 방향으로 조금씩 평행이동시키며 기준곡선으로 수축시키는 방법을 고려해 보자. R 의 값은 고정되어 있고, P_i 의 이웃 점들의 집합 T_i 는 수축의 각 단계마다 새로 계산된다. 그러면, T_i 에 포함된 점들이 반대 방향의 법선벡터를 가질수록 수식 (1)에서 제시된 조건을 만족시키기는 어려워진다. 그림 3은 수축 알고리즘의 기본 개념을 설명한다.

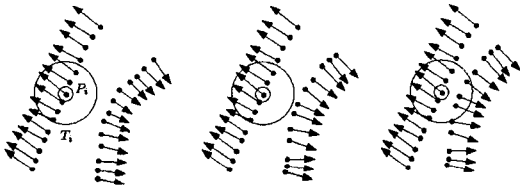


그림 3 수축 기법의 기본 개념

수축은 충분히 많은 개수의 점집합이 수식 (1)의 조건을 만족시키지 않을 때 중단된다. 캐널곡면 상의 하나의 특징원 상의 점들 중 서로 가장 먼 거리에 있는 점들에서 구한 곡면의 법선벡터를 대칭적인 법선 벡터라고 정의하자. 수축 알고리즘을 적용하기 위하여 다음의 사항들을 고려한다.

- 캐널곡면의 두 대칭적인 법선 벡터는 정반대방향을 갖지 않을 수도 있다(그림 2 참조). 일반적으로 두 개의 대칭 법선 벡터들은 구간 $(0, \pi]$ 사이의 각도를 갖는다.
- 사실적인 물체를 나타낼 때 $r'(t)$ 의 변화율이 매우 큰 경우는 거의 없다.
- 서로 다른 특징원 상의 점들은 서로 다른 반경으로 수축되어야 한다. 어떤 점들은 알고리즘 수행 도중에 수축을 중단해야 하는 반면 다른 점들은 주어진 최대 횟수만큼 수축을 수행해야 한다.

Algorithm: ShrinkingOfCanalSurface

단계 1: for $i = 1$ to n do

$$\hat{P}_i \leftarrow P_i;$$

stop _{i} \leftarrow FALSE;

단계 2: BreakCount \leftarrow 0;

단계 3: for $i = 1$ to n do

for $j = 1$ to n do

if stop _{i} = FALSE and $\|\hat{P}_j - \hat{P}_i\| < R$ and $N_j \cdot N_i < \tau$ then

BreakCount \leftarrow BreakCount + 1;

stop _{i} \leftarrow TRUE;

단계 4: if BreakCount $\geq \mu n$ then

stop and return $\hat{S} = \{ \hat{P}_i | i = 1, \dots, n \}$;

단계 5: for $i = 1$ to n do

if stop _{i} = FALSE then

$$\hat{P}_i \leftarrow \hat{P}_i - \delta N_i;$$

단계 6: Goto 단계 3;

Algorithm: ShrinkingOfCanalSurface는 캐널곡면을 수축시키는 과정을 보인다. 수축한 결과는 $\hat{S} = \{ \hat{P}_i | i = 1, \dots, n \}$ 에 저장되며 이것은 캐널곡면의 기준곡선의 형태를 갖는다. 매개변수 τ 는 두 대칭적인 법선벡터 사이의 방향 차이의 허용오차(tolerance)를 제어하기 위해 사용된다. 제 3절의 실험에서는 $\tau = 0.1736 = \cos(80^\circ)$ 를 사용하였다. 변수 stop _{i} 는 P_i 가 수축을 중단했는지의 여부를 나타내기 위해 사용되며, stop _{i} 가 FALSE인 P_i 만이 기준곡선 방향으로 수축을 계속하게 된다. 그림 4는 캐널곡면의 복원을 위해 수축이 적용된 예를 보인다.

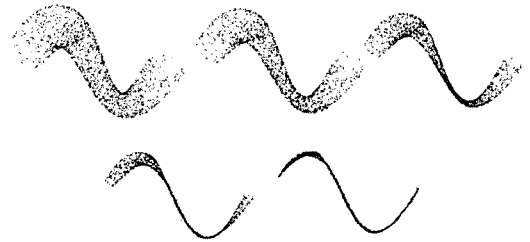


그림 4 캐널곡면의 복원을 위해 주어진 2000개의 점집합에 수축을 적용한 결과: 좌에서 우의 방향으로 각각 0, 0.8, 998, 1609개의 점들의 수축이 중단된 상태임.

2) Moving Least-Squares 기법

캐널곡면의 기준곡선을 구하기 위해 moving least-squares 기법을 사용하여 수축 알고리즘의 결과로 얻어진 \hat{S} 를 가느다란 곡선 형태의 점집합 \hat{S} 으로 변환시킨다. Moving least-squares 방법은 McLain 등[14,15]에 의해 개발되었으며 Lee[16], Levin[17] 등은 점집합을 가는 곡선과 같은 형태로 변환하기 위하여 이 방법을 사용하였다. 이 절에서는 Lee 등[16]이 moving least-squares 방법을 사용한 예를 간략히 소개한다.

각 점마다 이웃 점들에 fitting하는 곡선 또는 곡면은

가중치 회귀(weighted regression)방법을 적용하여 계산하며, 적용된 결과로 해당 점은 근사된 곡선 또는 곡면 상의 새로운 위치로 이동한다. Moving least-square 방법은 국소적인 오차가 지역적인 최적 방정식(local best polynomial) 근사의 오차로 한정된다는 면에서 거의 최적의 방법이라고 할 수 있다. 점집합이 가느다란 곡선과 같은 형태로 축소되면, 점 집합에 순서를 부여하여 근사곡선(approximation curve)을 계산할 수 있다. 다음과 같이 2차원 점의 집합 S 가 주어졌다고 가정하자.

$$S = \{P_i = (x_i, y_i) | i = 1, \dots, n\}$$

적절한 가중치 함수(weighting function)를 이용하여 각각의 점 P_i 에 대하여 0보다 크거나 같은 값의 가중치(weight) w_i 를 얻었다고 가정할 때, 집합 S 에 속하는 한 점 P_* 에 대해 국부적 회귀직선(local regression line) $L_*: y = ax + b$ 는 다음과 같은 2차식을 최소화(minimizing)함으로써 계산될 수 있다.

$$D_i = \sum_{j=1}^n (ax_j + b - y_j)^2 w_j$$

가중치 함수는 P_* 에서 거리가 먼 점에 대해 더 큰 penalty를 주는 함수가 선택되어야 한다. 본 연구에서는 $r = \|P_i - P_*\|$ 이고 H 가 가중치 반경(weighting radius)을 나타낸다고 할 때, 다음과 같은 함수를 선택하였다[18].

$$w_i = \begin{cases} 2\frac{r^3}{H^3} - 3\frac{r^2}{H^2} + 1, & \text{if } r < H \\ 0, & \text{if } r \geq H \end{cases} \quad (2)$$

수식 (2)의 가중치 함수는 반경 H 와 중심점 P_* 인 원의 바깥쪽의 각 점의 가중치가 0이 되게 한다. 그러므로, 우리는 P_* 로부터의 거리가 H 보다 작은 점의 집합만을 이용하여 회귀직선 또는 2차곡선을 계산할 수 있다.

가중치 회귀(Weighted regression)를 적용함으로써 우리는 P_* 에 대해 최적의 국부적 회귀직선 L_* 를 계산할 수 있다. x 축이 직선 L_* 와 평행하고 P_* 가 원점인 새로운 좌표계가 주어질 때 전체 점들의 집합을 새 좌표계로 변환하는 변환 M 을 적용하여 결과로 $\bar{S} = \{\bar{P}_i = (\bar{x}_i, \bar{y}_i) | i = 1, \dots, n\}$ 가 얻어졌다고 하자. 점의 집합 \bar{P}_* 를 위한 국부적 2차 회귀곡선(local quadratic regression curve)은 다음과 같다.

$$Q_*: \bar{y} = a\bar{x}^2 + b\bar{x} + c$$

위의 식은 다음의 식을 최소화하여 계산될 수 있다.

$$D_q = \sum_{i=1}^n (a\bar{x}_i^2 + b\bar{x}_i + c - \bar{y}_i)^2 w_i$$

이때 \bar{P}_* 를 Q_* 로 투사한 결과는 $(0, c)$ 이다. 마지막으

로 P_* 는 $(0, c)$ 에 역변환 M^{-1} 를 적용하여 얻어진 새로운 위치로 이동된다. 그림 7은 2차원 점의 집합을 세션화(thinning)한 예를 보인다.

Moving least-squares 방법은 3차원으로 직접 확장될 수 없다. 3차원 회귀곡선은 쉽게 계산할 수 있지만 3차원의 2차 회귀곡선을 계산하는 것은 어렵다. 그러므로, 3차원 점의 집합에 속한 각각의 점 P_* 에 대해 국부적 회귀 알고리즘을 적용하는 방법을 제안하면 다음과 같다.

1. P_* 의 이웃 점의 집합을 B 라 하면, $B = \{P_j | \|P_j - P_*\| < H\}$ 이다.

2. 다음과 같은 2차 함수 D_k 를 최소화함으로써 회귀평면 $K: z = Ax + By + C$ 를 계산한다.

$$D_k = \sum_{P_j \in B} (Ax_j + By_j + C - z_j)^2 w_j$$

3. B 에 속한 점들을 K 상에 투사한다.

4. K 상에서 2차원 moving least-squares 기법을 적용한다.

위의 알고리즘은 정규 공간 곡선(regular space curve) 상의 한 점은 지역적으로 접촉평면(osculating plane)을 갖는다는 사실에 근거하여 구성되었다. 기본적으로 d 차원의 점들을 $d-1$ 차원의 초평면(hyperplane)에 반복적으로 투사하여 2차원 문제로 축소할 수 있으므로 어떤 차원의 점들에 대해서도 위의 알고리즘을 적용할 수 있다.

Lee[16]는 안정성 있는 국부적 회귀(local regression)를 위해 EMST(Euclidean minimum spanning tree)를 적용하고 각 점들에 대해 적절한 H 값을 정하기 위해 correlation을 사용하였다. Moving least-square 기법을 이용하여 수축 알고리즘의 결과로 구해진 \hat{S} 를 가느다란 점집합 \tilde{S} 으로 변환시켰다. 가는 점집합 \tilde{S} 에 속한 점들에 순서를 부여해야 곡선으로 근사할 수 있다. 가장 간단한 방법은 두 개의 가장 가까운 점들을 재귀적으로 연결하는 방법이다. 점들에 순서를 부여한 후, 일반적인 곡선의 근사 또는 보간(interpolation)방법을 적용하여 \hat{S} 를 기준곡선으로 근사할 수 있다[12]. 그림 5는 \hat{S} , \tilde{S} , 그리고 \tilde{S} 로부터 근사된 기준곡선의 예를 보인다.



그림 5 좌에서 우의 방향으로 각각 \hat{S} , \tilde{S} , 그리고 \tilde{S} 로부터 근사된 기준곡선

3) 반경함수의 계산

캐널곡면의 반경함수 $r(t)$ 는 점집합의 각 점에 대해 구한 반경 $r_i = \|\tilde{P}_i - P_i\|$ 에 곡선의 근사기법을 적용하여 계산할 수 있다. 캐널곡면으로부터 2000개의 점들을 샘플링(sampling)한 뒤 perturbation을 주어 얻은 점집합을 캐널곡면으로 복원한 예를 그림 6에서 보인다. 수축과 moving least-square 단계는 각각 그림 4와 그림 5에서 제시되었다.



그림 6 캐널곡면 복원의 예: 좌로부터 우의 방향으로 원래(original) 캐널곡면, 2000개의 perturbation된 표본점들(sample points), 그리고 복원된 캐널곡면

3. 실험 결과

그림 6에 제시된 원래의 캐널곡면 $A(s, t)$ 는 표 1과 같은 제어점(control point)으로 구성된 기준곡선을 갖고 있다. 반경함수 $r(t)$ 는 knot vector (0, 0, 0, 1, 2, 3, 4, 4, 4)와 제어점 (2, 1.5, 1, 1.5, 2.5, 2)로 구성된 1차원의 2차 B-스플라인 곡선이다. 2000개의 표본점들은 수식 $A(s, t) + \epsilon(s, t)$, $-0.05 \leq \epsilon(s, t) \leq 0.05$,를 이용하여 생성하였다. 수축 알고리즘의 입력으로 주어진 변수값은 각각 $\delta = 0.1, \mu = 0.8, R = 0.1$, 그리고 $\tau = \cos(80^\circ)$ 이다. \hat{S} 은 수축 단계가 20번 수행된 후 계산되었으며 마지막 단계에서는 1609개의 점들이 수축을 중단하였다. 원래의 기준곡선과 근사로 얻어진 기준곡선 간의 최대거리는 1.772888이고 평균거리는 1.0936823이다. 원래의 반경함수와 근사로 얻어진 반경함수 간의 최대거리는 0.14423506이고 평균거리는 0.07538086이다.

그림 7에서는 캐널곡면 복원의 다른 예를 보인다. 원래의 캐널곡면 $A(s, t)$ 는 표 2와 같은 6개의 제어점과

표 1 그림 6에서 제시된 캐널곡면 기준곡선의 제어점과 수행시간

x	y	z	수행시간 (sec)	
-3.000	6.660	0.000	\hat{S} 계산	12.4
-0.337	5.660	-4.660	\tilde{S} 계산	15.1
0.410	0.430	-8.052	기준곡선근사	2.1
-2.037	-7.906	3.354	$r(t)$ 계산	5.2
1.813	-6.807	0.175		
6.760	-3.150	0.000		

(0, 0, 0, 0, 1, 2, 3, 3, 3, 3)의 knot vector를 갖는 1차원의 3차 B-스플라인 곡선을 기준곡선으로 가진다. 반경함수는 knot vector(0, 0, 0, 0, 1, 2, 3, 3, 3, 3)과 6개의 제어점(1.9999, 1.6666, 1.103463, 1.037336, 1.833333, 2.5)을 갖는 1차원의 3차 B-스플라인 곡선이다. 2000개의 표본점들은 수식 $A(s, t) + \epsilon(s, t)$, $-0.05 \leq \epsilon(s, t) \leq 0.05$ 으로부터 생성되었다. 수축 알고리즘의 입력으로 주어진 변수값은 $\delta = 0.1, \mu = 1.0, R = 0.1$, 그리고 $\tau = \cos(80^\circ)$ 이다. \hat{S} 은 수축 단계가 26번 수행된 후 계산되었으며 마지막 단계에서는 1995개의 점들이 수축을 멈췄다. 원래의 기준곡선과 근사로 얻어진 기준곡선 간의 최대거리는 1.4648859이고 평균거리는 1.023708이다. 원래의 반경함수와 근사로 얻어진 반경함수 간의 최대거리는 0.5901724이고 평균거리는 0.2694635이다.

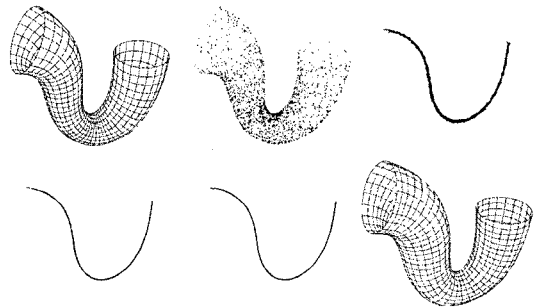


그림 7 캐널곡면 복원의 예: 좌로부터 우의 방향으로 원래의 곡면, S (2000개의 표본점들), \hat{S} , \tilde{S} , 그리고 \tilde{S} 로부터 근사된 기준곡선, 복원된 결과인 캐널곡면

표 2 그림 7에서 제시된 캐널곡면을 구성하는 기준곡선의 제어점과 수행시간

x	y	z	수행시간 (sec)	
-4.000	4.000	0.000	\hat{S} 계산	13.2
-0.478	4.547	0.000	\tilde{S} 계산	15.4
3.335	2.618	0.000	기준곡선근사	2.6
1.428	-1.214	0.000	$r(t)$ 계산	5.1
-3.543	-1.614	0.000		
-4.257	-4.700	0.000		

그림 8에서는 캐널곡면의 복원 알고리즘을 변형하여 파이프곡면을 복원한 예를 보인다. 원래의 파이프곡면 $A(s, t)$ 는 표 3과 같은 10개의 제어점과 (0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 7, 7, 7)의 knot 벡터를 갖는 1차원의 3차 B-스플라인 곡선을 기준곡선으로 한다. 반경함수는 상수 3이다. 2000개의 표본점들이 수식 $A(s, t) + \epsilon(s, t)$,

$-0.05 \leq \epsilon(s, t) \leq 0.05$ 으로부터 생성되었다. 수축 알고리즘의 입력으로 주어진 변수값은 $\delta = 0.05, \mu = 0.95, R = 0.1$, 그리고 $\tau = \cos(90^\circ)$ 이다. \hat{S} 은 수축 단계가 6번 수행된 후 계산되어졌으며 마지막 단계에서는 1934개의 점들이 수축을 멈췄다. 근사된 반경은 0.298778 이고, 원래의 기준곡선과 근사된 기준곡선 간의 최대거리는 1.60093152, 평균거리는 0.814391259 이다.

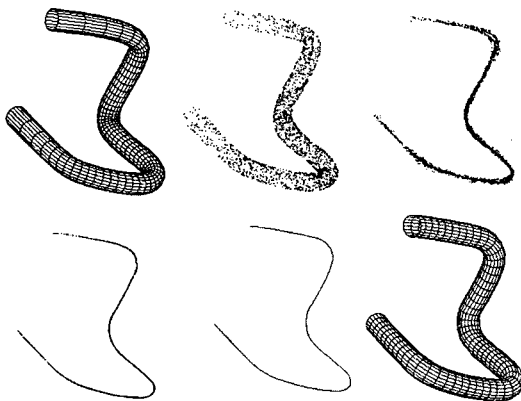


그림 8 파이프곡면 복원의 예: 좌로부터 우의 방향으로 원래의 곡면, S (2000개의 표본점들), \hat{S} , \tilde{S} , 그리고 \tilde{S} 로부터 근사된 기준곡선, 복원된 결과인 파이프곡면

표 3 그림 8에서 제시된 파이프곡면을 구성하는 기준곡선의 제어점과 수행시간

x	y	z	수행시간 (sec)	
0.000	0.000	0.000	\hat{S} 계산	9.3
2.773	-0.234	0.159	\tilde{S} 계산	13.4
3.091	0.058	-0.043	기준곡선근사	2.2
3.086	2.917	-0.365	r 계산	3.2
2.939	3.054	0.247		
0.066	3.159	-0.292		
-0.279	2.959	0.377		
0.411	3.108	2.569		
-0.045	2.942	2.928		
0.000	0.000	3.000		

4. 결론

본 논문에서는 점집합으로부터 채널곡면을 복원하는 방법을 제시하였다. 수축기법과 moving least-square 방법을 사용함으로써 주어진 점집합에서 기준곡선과 반경함수를 정확하고 효율적으로 추출하였다. 향후 연구의 방향은 이러한 채널곡면의 복원기법을 스웍트곡면과 같

이 좀더 복잡한 형태의 곡면을 복원하는 방법으로 확장하는 것이다.

참고 문헌

- [1] T. Maekawa, N. M. Patrikalakis, T. Sakkalis, and G. Yu, "Analysis and applications of pipe surfaces," *Computer Aided Geometric Design*, 15(5):437-458, 1998.
- [2] M. Peternell and H. Pottmann, "Computing rational parametrizations of canal surfaces," *Journal of Symbolic Computation*, 23:255-266, 1997.
- [3] H. Pottmann and T. Randrup, "Rotational and helical surface approximation for reverse engineering," *Computing*, 60:307-322, 1998.
- [4] H. Pottmann, H.-Y. Chen, and I.-K. Lee, "Approximation by profile surfaces," A. Ball et al. Eds., *The Mathematics of Surfaces VIII*, Information Geometers, pp.17-36, 1998.
- [5] H.-Y. Chen, I.-K. Lee, S. Leopoldseder, H. Pottmann, T. Randrup, and J. Wallner, "On surface approximation using developable surfaces," *Graphical Models and Image Processing*, 61:110-124, 1999.
- [6] M. Peternell and H. Pottmann, "Approximation in the space of planes: applications to geometric modeling and reverse engineering," *Rev. R. Acad. Cienc. Serie A. Mat*, 96(2):243-256, 2002.
- [7] H. Pottmann and J. Wallner, *Computational line geometry*, Springer-Verlag, 2001.
- [8] H. Pottmann, S. Leopoldseder, J. Wallner, and M. Peternell, "Recognition and reconstruction of special surfaces from point clouds," *Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXIV, Part 3A, Commission III, pp. 271-276, 2002.
- [9] W.-D. Ueng, J.-Y. Lai, and J.-L. Doong, "Sweep-surface reconstruction from three-dimensional measured data," *Computer-Aided Design*, 30(10): 791-805, 1998.
- [10] R. Ramamoorthi and J. Arvo, "Creating generative models from range images," *SIGGRAPH '99 Proceedings*, 1999.
- [11] I.-K. Lee, "Pipe Surface Reconstruction Using Shrinking," 한국컴퓨터그래픽스학회 논문지, 제5권, 제2호, pp.1-7, 1999년 12월.
- [12] J. Hoschek and D. Lasser, *Fundamentals of Computer Aided Geometric Design*, A. K. Peters, 1993.
- [13] P. Krsek, G. Lukas, and R. R. Martin, "Algorithms for computing curvatures from range data," A. Ball et al. Eds., *The Mathematics of Surfaces VIII*, Information Geometers, pp.1-16, 1998.
- [14] D. McLain, "Drawing contours from arbitrary data points," *The Computer Journal*, 17:318-324, 1974.
- [15] D. McLain, "Two dimensional interpolation from

- random data," *The Computer Journal*, 19:178-181, 1976.
- [16] I.-K. Lee, "Curve reconstruction from unorganized points," *Computer Aided Geometric Design*, 17(2): 161-177, 2000.
- [17] D. Levin, "Mesh-independent surface interpolation," private communication.
- [18] G. Wyvill, C. McPheeters, and B. Wyvill, "Data structure for soft objects," *The Visual Computer*, 2:227-234, 1986.



이 인 권

1989년 연세대학교 전산학과 학사. 1992년 포항공대 컴퓨터공학과 석사. 1997년 포항공대 컴퓨터공학과 박사. 1997년~1999년 비엔나 공대 연구원. 1999년~2001년 포항공대 정보통신연구소 선임연구원. 2001년~2003년 아주대학교 미디어학부 조교수. 2003년~현재 연세대학교 컴퓨터학과 조교수. 관심분야는 컴퓨터 그래픽스, 컴퓨터 게임, 컴퓨터 음악 등



김 구 진

1990년 이화여자대학교 전자계산학과 학사. 1992년 한국과학기술원 전자계산학과 석사. 1998년 포항공대 컴퓨터공학과 박사. 1998년~2000년 Dept. of Computer Sciences, Purdue University, PostDoc 2000년~2002년 아주대학교 정보및컴퓨터공학부 BK교수. 2002년~2003년 Dept. of Mathematics and Computer Science, University of Missouri-St. Louis, Visiting Assistant Professor. 2004년~현재 경북대학교 컴퓨터공학과 조교수. 관심분야는 컴퓨터 그래픽스, 컴퓨터 애니메이션, 곡면 및 기하모델링 등