자기폐색 물체의 2D 커브로부터의 3D 모델링
(3D Modeling of Self-Occluding Objects from 2D Drawings)

코디에 프레데릭[†] 서 혜 원[††] 조 영 상[†††]
(Frederic Cordier) (Hyewon Seo) (Young-Sang Cho)

요 약 본 논문은 2차원 평면상의 그림으로부터 3차원 물체를 복원하는 방법을 제시한다. 사용자가 입력하는 2차원 평면 그림은 3차원 물체의 윤곽선을 그린 것으로, 자신의 일부분이나 다른 물체에 의해 가려진 부분이 있는 윤곽선도 허용하는 것이 특징이다. 따라서, 복원된 3차원 물체 역시 자신의 일부, 혹은 다른 물체에 의해 가려진 부분이 존재할 수 있다. 본 논문에서 제안하는 방법은 2차원 윤곽선 분석, 3차원 골격 계산, 그리고 3차원 물체 복원의 세가지 단계로 구성된다. 본 논문의 주된 기여는 기여는 자신이나 다른 물체에 의해 가려진 2차원 윤곽선으로부터 3차원 골격을 계산하는 방법이며, 이를 위하여 일련의 최적화 문제를 정의하고 해결하였다. 최적화 문제는 골격의 생성, 물체의 충돌 제한, 그리고 C1 연속성 유지를 위하여 사용된다. 결과적으로, 제안된 방법은 기존의 실루엣 기반의 스케칭 인터페이스를 사용한 3차원 물체 모델링에 대하여, 상호 폐색 (가림/가려짐) 이 존재하는 형태에서도 허용되도록 확장하였다.

키워드 : 스케치 기반 모델링, 상호/자기 폐색, 최적화

Abstract *In this paper, we propose a method for reconstructing a 3D object (or a set of objects) from a 2D drawing provided by a designer. The input 2D drawing consists of a set of contours that may partially overlap each other or be self-overlapping. Accordingly, the resulting 3D object(s) may occlude each other or be self-occluding. The proposed method is composed of three major steps: 2D contour analysis, 3D skeleton computation, and 3D object construction. Our main contribution is to compute the 3D skeleton from the self-intersecting 2D counterpart. We formulate the 3D skeleton construction problem as a sequence of optimization problems, to shape the skeleton and place it in the 3D space while satisfying C1-continuity and intersection-free conditions. Our method is mainly for a silhouette-based sketching interface for the design of 3D objects including self-intersecting objects.*

Key words *: sketch-based modeling, (self-) occlusion, optimization*

1. 서 론

스케칭 인터페이스(sketching interface)는 디자이너가 2차원 스케치로부터 3차원 물체를 모델링 할 수 있도록 편리를 제공한다. Igarashi 외[1]는 사용이 간단하고 직관적인 실루엣 기반의 스케칭 시스템 Teddy를 제안함으로써 주목을 받은 바 있다. 그 이후에 제안된 스케칭 인터페이스 관련 연구들은 3차원 물체의 표현 방식에 중점을 두고 있다. 이러한 최근 연구로는 합성곱 곡면(convolution surface)[2], 볼륨 메트릭 형상(volumetric shape)[3], 변분 음함수 곡면(variatianal implicit surface)[4], 음함수 곡면(implicit surface) 관련 연구[5] 등이 있다.

본 논문은 기존 스케칭 인터페이스에서 불가능 했던 종류의 물체들, 즉 그림 1에서 보인 것과 같이 자신이나 다른 물체에 의해 가려진 부분이 있는 물체들도 모델링 할 수 있음을 보임으로써 스케치 기반 모델링의 영역을 확장한다. 앞서 소개한 Teddy와 같은 기존 모델러를 사용하여 이런 형태의 물체를 모델링을 하기란 매우 어렵고, 실제로 불가능하다고 할 수 있겠다. 예를들어, 그림 1에 보인 스케치의 경우 3개의 닫힌 곡선으로 이루어져 있으며, 이들을 Teddy에 입력할 경우 각각의 곡선에 대응하는 3차원 물체 조각들을 복원할 것이다. 그러나, 복원된 각각의 3차원 곡면은 스케치 평면과 수직인 방향으로는 평평하므로, 대상 물체의 곡면을 따라 변하는 깊이 정보, 그리고 가려진 부분에 대한 복원은 사용자의 매우 조심스러운 수작업에 의존한다.

본 논문은 이러한 문제에 대한 통합적인 해결 방법을 제안한다.

† 정 회 원 : 한국과학기술원 문화기술대학원
cordier@kaist.ac.kr
†† 정 회 원 : 충남대학교 전기정보통신공학부
hseo@cnu.ac.kr
††† 학생회원 : 한국과학기술원 전산학과
dingulx2@tclab.kaist.ac.kr
논문접수 : 2005년 12월 8일
심사완료 : 2006년 7월 28일

제안된 모델러의 접근 방법은 다음과 같다. 우선, 자신이나 다른 물체에 의해 가려진 부분이 있는 윤곽선의 집합이 주어지면 그것으로부터 2차원 골격(skeleton)을 추출한다. 다음으로, 2차원 골격으로부터 3차원 골격을 계산한다. 마지막으로, 3차원 골격을 기반으로 한 3차원 물체를 생성한다. 본 연구의 주된 기여는 2차원 골격의 교차점에서 C1 연속성과 충돌 억제 조건을 만족시키는 깊이 값을 계산하여 3차원 골격을 생성하는 과정에 있다. 이러한 3차원골격 생성 과정은 평면상의 2차원 골격을 반복적으로 수정하여 제약 조건을 만족하도록 일련의 제약적(constrained) 최적화 문제로 수식화하고 이를 해결하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구들에 대한 소개로 스케치 기반 모델링 시스템들에 대해 알아본다. 3장부터 5장에는 2차원 스케치로부터 3차원 물체를 복원하는 과정의 세 단계를 각각 기술한다. 6장에서는 구현에 대해 보다 자세히 설명하고, 완성된 모델러의 결과를 여러 예제 스케치에 대해 보인 후, 7장에서 결론과 향후 연구 방향에 대해 정리한다.

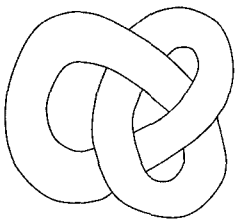

그림 1 매듭을 가진 원환면의 스케치

2. 관련연구

서로 가린 부분이 있는 물체의 윤곽선 스케치는 닫혀있지 않은 곡선을 다수 포함하므로 스트로크(stroke) 기반의 방법으로는 다룰 수 없다. 따라서, 사용자에 의해 새로이 그려지는 곡선 세그먼트(segment)들 각각을 입력 즉시 독립적으로 처리하는 것이 아니라, 전체를 이루는 모든 곡선 세그먼트에 대한 입력이 끝난 후에야 모델링을 시작할 수 있다. 그 외에는 스케치 기반 인터페이스와 관련된 기존의 연구들과 기술적으로 많은 공통점을 가지고 있다. 즉, 사용자가 입력한 2차원 곡선의 분석과, 2차원 곡선에 대응하는 3차원 물체의 복원 등에서는 예전 연구와 비슷한 기술을 사용한다.

3차원 물체의 모델링을 위한 스케칭 인터페이스에 대한 관련 연구들을 살펴보겠다. 이들은 모델링하는 형상의 클래스에 따라 분류할 수 있는데, 가장 일반적인 방법은 사용자로 하여금 직선으로 이루어진 물체의 보여지는 윤곽과 감춰진 윤곽을 사용자가 그림으로써 모델링하는 방법이다[6-8]. 스케치에 나타난 직선과 3차원 물체의 직선 사이의 기하학적 상관관계 (이를테면 스케치 상에서 평행인 직선들은 3차원 물체에서도 평행)를 사용하는 이러한 복원 기술은 건축용 CAD 애플리케이션에서 사용되는 물체를 모델링하는 데 특히 적합하나, 직선으로 이루어진 물체만을 대상으로 한다.

몇몇 스케칭 도구에서는 제스처 기반 인터페이스를 사용한다. 예를 들어 "SKETCH"[9]는 미리 정해진 규칙에 따라 입력 스트로크를 인식하여 해석한다. 사용자에 의해 제공되는 제스처 기호가 기본 물체의 생성이나 이미 존재하는 물체의 변형에 매핑되도록 규칙집합 정의한다. Shesh[10]는 앞서 설명한 두 가지 방식을 조합한 형태, 즉 제스처 스케칭과 직선형태 스케치의 3차원 복원을 혼합한 스케칭 인터페이스를 개발하였다.

Igarashi 외, 그리고 Alex 외 등은 자유형태 (free-form) 모델링을 위한 스케칭 인터페이스[1,4,5]를 제안하였다. 이들 시스템에은 사용자가 모델링 될 3차원 물체의 실루엣을 2차원 평면에서 스케치하면, 그 실루엣을 만족시키는 3차원 물체를 생성한다. 주어진 실루엣에 의해 둘러싸인 영역을 3차원으로 팽창시킴으로써 3차원 모델을 복원한다. 실루엣의 폭이 좁은 곳이 복원되면 3차원 모델의 얇은 부분을 이루고, 폭이 넓은 곳은 두꺼운 부분에 대응되는 일반적인 속성을 사용한다. 또한 생성된 모델을 자르기, 덧붙이기, 구부리기, 또는 메쉬 위에 직접 그리기와 같이 모델을 인터액티브하게 변형시킬 수 있는 편집 도구들을 제공함으로써 생성된 물체에 변경을 더해 나갈 수 있도록 한다.

스케칭 도구 중에는 3차원 곡선의 모델링을 위해 개발된 것도 있다. Tolba[11]은 사용자가 2차원 스트로크로 장면을 그리면 그것을 다양한 관점에서 시각화시킬 수 있는 방법을 제안했다. 3차원 복원은 "perspective grid"에서 2차원 곡선을 정렬하여 얻을 수 있다. Cohen[12]은 3차원 곡선 모델링의 다른 방법을 제안하였다 – 사용자가 단일한 관점에서의 곡선과 바닥 면에서 그 곡선의 그림자를 그리면 3차원 곡선을 복원 할 수 있다.

3차원 모델의 변형을 위한 스케칭 인터페이스도 제안되었다. Kerautret[13]가 제안한 시스템은 사용자가 여러 방향의 조명에서의 쉐이딩(shading)을 그리면 그것으로부터 3차원 곡면을 변형하였다. 비슷하게, Nealen[14]과 Kho[15]는 3차원 모델에 사용자가 새로운 실루엣을 입력하면 그것에 맞도록 모델을 변형되는 기법을 소개하였다.

본 논문에서는 자신 혹은 다른 물체에 의해 가려진 부분이 있는 자유형태 물체를 모델링하는 것에 초점을 두었다. 자신으로부터 가려진 부분이 있는 3차원 물체를

효율적으로 복원하는 부분이 본 연구의 기여이다. 사용자 입력으로부터 2차원 곡선이 주어졌을 때, 3차원 물체의 복원은 다음과 같은 세 가지 단계로 진행된다. 우선 2차원 윤곽선을 분석하고, 다음으로 3차원 골격을 분석한 후, 마지막으로 3차원 물체를 생성한다. 다음 절들에서 각 단계의 세부 내용을 기술한다.

3. 2차원 윤곽선 분석

2차원 윤곽선 분석은 입력 스트로크를 부드럽게 하고 교차점을 찾는 스트로크 정리 부분과, 윤곽선 완성 부분으로 구성된다. 서론에서 소개한 바와 같이, 본 연구에서 입력 스케치는 부분적으로 가려진 부분이 있는 2차원 윤곽선의 집합이다. 이러한 윤곽선의 집합은 사용자가 모델링하고자 하는 물체의 실루엣을 2차원 평면상에 곡선으로 그림으로써 얻을 수 있다. 이 윤곽선들을 적절히 연결하여 최종적으로 하나 이상의 닫힌 곡선들로 표현해야 한다(윤곽선 완성). 또한 손으로 그려진 스트로크는 노이즈가 심하므로 이것을 없애기 위한 필터링 과정이 필요한데, 그러한 과정은 [16-18]에서 이미 다루었고, 본 연구에서도 그러한 방법을 참조하여 구현하였다.

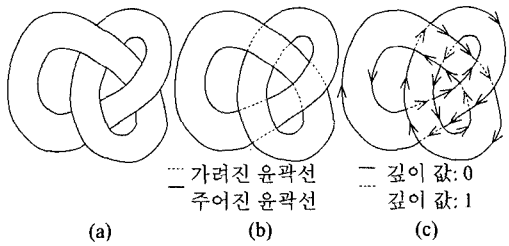

그림 2 스케치의 위상 정보 복원. (a) 입력 스케치 (b) 가려진 윤곽선 복원의 가능한 해 중 하나 (c) 모든 윤곽선에 대해 위상이 맞는지 검사

윤곽선 완성을 위해서는 가려진 윤곽선 세그먼트와 보이는 윤곽선 세그먼트를 연결하여 닫힌 곡선으로 표현해야 한다. 또한, 완성된 윤곽선이 위상적으로 맞는지도 함께 검사한다(그림 2 참조). 본 논문에서는 Williams[19]의 방법을 참조하였다. 즉, 닫힌 윤곽선을 만들기 위해 가능한 모든 윤곽선 세그먼트의 연결 후보 중 길이와 곡률을 고려하여 선택된 윤곽선 세그먼트의 끝점을 큐빅 베이지어 스플라인으로 보간한다.

윤곽선이 3차원 공간에서 물리적으로 가능한 곡선인지 아닌지를 판단하여 주어진 곡선 세그먼트의 집합이 위상적으로 맞는지 검사가 필요한데, 본 연구에서는 역시 Williams가 공식화한 정수 선형 프로그래밍을 사용하였다. 이 방법은 윤곽선의 교차 횟수가 안쪽, 그리고 바깥쪽 윤곽선의 개수와 같은 위상적인 성질을 이용하며, 정수 선형 프로그래밍의 해가 있으면 위상적으로 맞다고 판단한다. 위상적으로 옳은 닫힌 윤곽선 집합을 찾을 때까지 branch-and-bound 방법[6]으로 다음 후보를 찾아내는 것을 반복한다[19].

4. 3차원 골격 복원

본 절에서는 3차원 물체를 복원하기 위한 알고리즘을 기술한다. 3절에서 설명한 윤곽선 분석 단계에서 우리는 닫힌 곡선의 집합을 얻을 수 있었다. 각각의 곡선을 교차점에서 잘라(cut) 교차가 없는 여러 개의 곡선 세그먼트들로 나누었고, 각각의 곡선 세그먼트는 깊이 정보, 즉, 현시점에서 바라보았을 때 곡선을 가리는 곡면의 갯수로 라벨링 되었다(그림 2(c)). 본 연구에서 우리의 목표는 생성될 3차원 물체의 실루엣이 입력으로 주어진 윤곽선 그림과 일치하되, 상호 충돌이나 표면 교차가 일어나지 않도록 3차원 물체의 좌표점을 찾아내는 것이다. 앞으로 본 논문에서는 스케칭 평면은 $z=0$인 xy평면이며, 윤곽선 그림은 3차원 물체를 스케칭 평면에 직교 투영한 것을 그린 것이라고 가정하겠다.

문제를 단순화하기 위하여, 2차원 윤곽선의 z좌표를 직접 계산하기 보다는, 윤곽선의 골격에 대하여 z 좌표를 먼저 계산한 다음, 이것으로부터 나머지 부분에 대한 z좌표를 추론해 내도록 하였다. 마지막으로, 2차원 윤곽선이 실제 3차원 물체의 실루엣과 일치하도록 3차원 골격 주위에 3차원 볼륨을 생성한다.

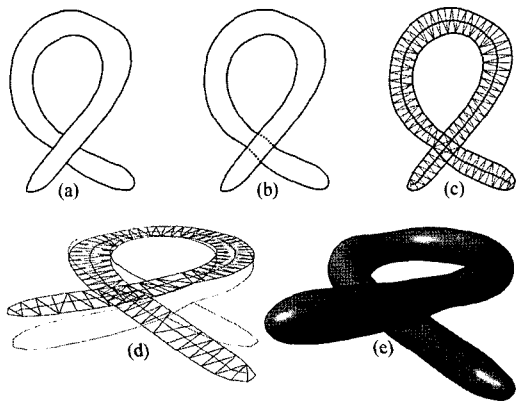

그림 3 복원 과정에 대한 개요: (a)스케칭 평면에 그린 입력 곡선 (b)윤곽선 분석에 의한 윤곽선 완성 (c)2차원 윤곽선 골격 (d)겹침이 없는 형태 복원을 위하여 스케칭 평면과 수직 방향으로 골격의 정점을 움직여 얻은 3차원 골격 (e)복원 결과.

4.1 골격 추출

골격을 계산하기 위해, "Teddy"[1]에서와 유사하게

실루엣 다각형[20]의 CAT(Chordal axis transform) 을 사용하였다. 현의 축(chordal axis)은 실루엣 다각형의 딜러니(Delaunay) 삼각화를 수행하여 생성되는 내부 에지들의 중점들을 연결한 곡선이다. Igarashi 외는 실루엣으로 둘러싸인 영역을 3차원 볼륨으로 팽창시킬 때 볼륨의 중심으로 현의 축을 사용하였다. 그러나, 본 논문의 경우는 실루엣 커브가 부분적 폐색(겹침)으로 인한 교차점을 포함할 수 있기 때문에, 제약 조건을 가진 딜로니 삼각화(constrained Delaunay triangulation)를 사용할 수 없다. 따라서, 본 연구에서는 plane-sweep 알고리즘의 변형된 버전인 Cordier와 Cheong이 제안한 방법[21]을 사용하였다.

4.2 복원될 물체의 질의 측정도

충돌 없이 실루엣이 입력 곡선과 일치되는 3차원 물체는 무수히 많다. 그 중 타당한 물체를 얻기 위해 3차원 형상의 질을 정량적으로 측정할 수 있도록 다음과 같이 세 가지 기준을 제안한다(그림 4 참조):

(1) 물체의 곡률(휨) 에너지; 틀어진 물체보다는 구부러지지 않은 물체를 더 자연스러운 형상으로 평가한다.
(2) 방향 (orientation)
(3) 스케치 평면으로부터 복원된 물체까지의 거리

위의 조건 (2),(3)은 복원된 물체를 스케치 평면에 가능한 가깝게 위치하기 위한 것이다.

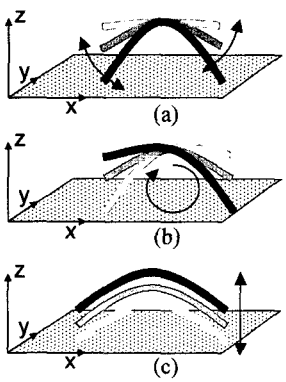

그림 4 골격의 z 좌표를 계산하기 위해 최소화 할 세 가지 기준: a)곡률 에너지, b)오리엔테이션, c)스케칭 평면과의 거리

4.3 골격의 표현

골격 둘레에 복원될 3차원 형상의 휨 에너지를 직접 측정하는 대신, 골격의 휨 에너지를 계산하여 2차원 골격에 적절한 z 좌표를 할당하는 것으로 대치하고자 한다. 안타깝게도, 곡선의 휨 에너지는 비선형이며[22], 이는 최적화 문제로 다루기 복잡하다. 따라서 본 연구에서는 각각의 골격 에지에 대응되는 조각 다항식 곡면(piecewise polynomial surface)의 집합으로 골격을 표현한다. 그림 5에도 보였듯이, 골격 에지는 자신의 두 점을 통과하는 곡면과 대응된다. 골격 z좌표가 할당됨에 따라, 대응되는 곡면도 따라서 변형될 것이다. 이렇듯 골격 자체 대신 곡면을 사용하면, 골격의 곡률 에너지 계산을 단순화할 수 있다. 예를 들어, 2차원 곡면을 사용하고 그 곡면의 휨이 적다고 가정 할 경우, 곡면 계수의 오목 2차 함수로 휨 에너지를 표현할 수 있다. 이러한 오목 2차 함수의 최소화 문제는 선형적으로 풀 수 있다. 휨 에너지에 대한 자세한 설명은 본 논문의 4.4.3 절에 있다.

골격이 구부러짐에 따른 곡면의 부드러운 변형을 유도하기 위해 C1 연속성이 필요하다. 골격의 정점(vertex) 스케치 평면으로부터 멀어지더라도 그 점에 인접한 두 곡면은 C1 연속성을 유지하며 부드럽게 휠것이다. 이렇듯 곡면의 휨 에너지로 스켈레톤의 그것을 대치하면 휨 에너지는 간단하게 계산할 수 있다.

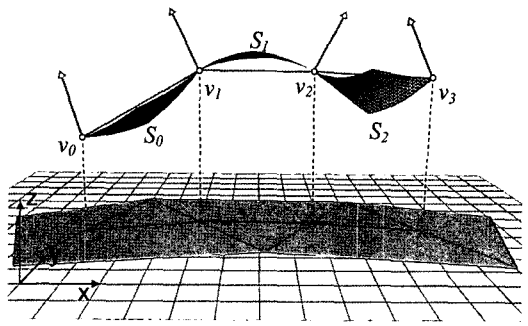

그림 5 다항식 곡면으로 표현된 골격

4.4 3차원 형상 복원을 최소제곱 문제로의 공식화

앞서 이미 설명한 바와 같이, 3차원 형상은 스켈리톤을 중심으로 하여 원형의 절단면을 가지는 볼륨을 생성함으로써 계산한다. 2차원 골격의 3차원 공간상에서의 위치를 계산하기 위해 골격의 각 정점 $v_i = (x_i, y_i)$에서의 z 좌표 z_i와 법선벡터의 방향$(x_{N,i}, y_{N,i})$을 계산하여 할당하는데, 이는 충돌을 피하고 연속성을 유지하면서 4.2절에서 언급한 골격의 질 - 3차원 물체의 질을 대신하는 - 을 최대화하는 최적화 문제로 볼 수 있다. 이를 다음과 같이 제약 조건과 목적 함수로 이루어진 최적화 문제로 공식화 하였다: 2차원에서 실루엣이 겹치는 물체가 3차원에서 충돌하는 것을 막기 위한 부등식 제약조건을 만들고, 인접한 곡면들 사이의 C1 연속성을 유지하기 위한 등식 조건을 추가한다. 목적 함수는 복원된 3차원 물체의 품질을 예측할 수 있는 골격의 품질로 정의한다. 정리하면, 다음 식과 같은 최적화 문제의 최소

제곱 해를 구하는 것이다:

$$\min_X |U_{Total} X| \text{ subject to } \begin{cases} M_{Cl} \cdot X = 0 \\ M_{Coll} X \geq d_z \end{cases} \quad (1)$$

위 식에서 $X=(z_0, x_{N,0}, y_{N,0}, \cdots, z_{m-1}, x_{N,m-1}, y_{N,m-1})$ 이고, m은 골격을 구성하는 정점의 개수이다.

다음 절 4.4.1~4.4.5에서 제약 조건들과 목적함수를 구체적으로 공식화하는 과정을 기술하겠다.

4.4.1 충돌 방지를 위한 선형 부등식 제한조건

2차원에 투영된 이미지 상에서 교차되는 곡면들로 표현되는 경우, 복원된 3차원 물체가 충돌하는 것을 막기 위하여 곡면들 사이에 최소한의 거리를 유지해야 한다. 이를 위하여 골격을 구성하는 정점들의 z좌표에 일련의 부등식 제약조건을 두었다. 우선, 충돌의 가능성이 있는 부분, 즉 부등식 제약조건이 정의되어야 하는 골격 부분의 정점의 쌍을 찾아야 한다. 본 연구에서는 Williams [19]의 패널링(paneling) 구축을 사용하였다. 패널링은 스케치 평면에 2차원 패널(혹은 영역)의 집합을 만든다. 각 패널은 3차원 형상 곡면의 일부가 스케치 평면에 투영된 그림이고, 윤곽선 완성 단계(3절 참조)에서 획득한 곡선 세그먼트들의 집합에 의해 경계가 정해진다. 그림 6은 패널링 구성의 한 예다. Williams의 논문[19]에 패널링 알고리즘의 상세한 설명이 언급되어 있다.

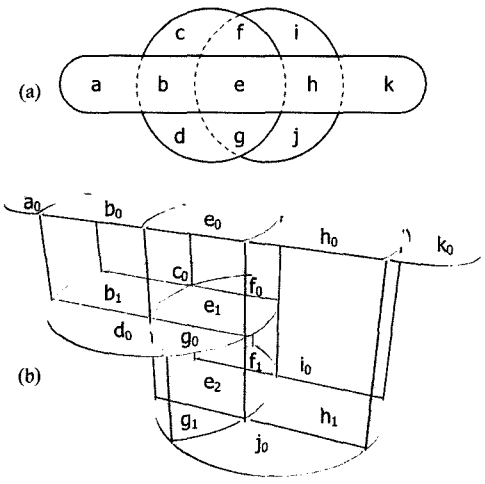

그림 6 스케치 (a)에 대응되는 penaling 구성 (b)

스케치 평면에 투영된 이미지가 서로 겹치는 형상의 경우, 패널링 구성은 깊이 인덱스 번호가 할당된 패널들의 스택을 만든다. 깊이 인덱스 번호는 시점으로부터 해당 패널 사이에 놓여있는 다른 패널의 갯수를 의미한다. 예를 들면, 그림 6에서 크기가 가장 큰 스택은 세 패널 (e_0, e_1, e_2)로 이루어진 것이고, 각각의 패널에 깊이 인덱스가 각각 0, 1, 그리고 2로 할당되어 있다. 이러한 패널들을 사용하여 충돌 제한 조건을 두어야 하는 골격 점들의 모든 쌍을 찾는 것은 다음과 같다: 우선 골격의 정점에 대응되는 딜로니 에지 중 패널에 연결된 것을 찾을 수 있고 그러한 에지를 가지고 있는 정점의 목록을 각 패널에 대해 만든다. 또한, 패널 스택로부터 각각의 패널에 대하여 자신의 바로 위에 있는 패널을 알아낸다. 이 두 패널이 겹쳤을 경우, 각각의 패널에서 속한 골격 정점을 하나씩 모아 만든 모든 정점쌍에 충돌 제약 조건을 둔다. 골격 에지가 아닌 골격 정점에만 이러한 충돌 제한조건을 고려함에 유의하라. 골격 에지의 길이가 골격의 크기에 비해 상대적으로 작다면 이러한 근사가 성립된다.

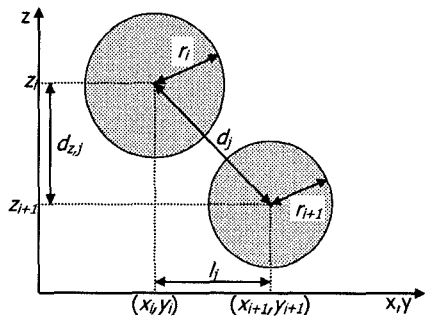

그림 7 정점 $v_i(x_i,y_i)$와 정점 $v_{i+1}(x_{i+1}, y_{i+1})$ 사이의 충돌 억제를 위한 제약조건

둘째, 각각의 제약 조건에 대해 최소 거리가 필요하다. 제약 조건은 행렬식으로 표현된다. 그림 7에서 보는 골격의 두 정점 $v_i=(x_i, y_i)$와 $v_{i+1}=(x_{i+1}, y_{i+1})$에 대하여, 이 두 점 사이의 최소 거리 d_i는:

$$d_j^2 = \left(r_i + r_{i+1} + d_{MinSurf}\right)^2 = l_j^2 + d_{z,j}^2 \quad (2)$$

이 된다. 이 식에서 r_i와 r_{i+1}는 각각 정점 v_i와 v_{i+1}을 중심으로 하는 물체의 반지름이고, $d_{MinSurf}$는 복원된 물체의 경계면 사이의 최소 거리, 그리고 l_j는 에지(v_iv_{i+1})의 길이이다. 식 (2)로부터 점 v와 v_{i+1} 사이의 z축 상의 최소 거리를 계산하면 다음과 같이 표현된다:

$$d_{z,j} = \sqrt{\left(r_i + r_{i+1} + d_{MinSurf}\right)^2 - l_j^2}$$

마지막으로, 두 정점에 대해 선형 부등식 제약 조건을 정의한다.

$$z_i - z_{i+1} \geq d_{z,j}$$

위 식에서 $d_{z,j}$의 부호는 겹쳐진 두 골격 사이의 가려진 순서에 따라 결정된다. 모든 q 충돌에 대한 부등식 제약 조건은 다음과 같이 행렬식으로 쓸 수 있다:

$$M_{Coll} X \geq d_z \tag{3}$$

$$M_{Coll} = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \dots & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & \dots & \dots & 0 & 1 & 0 & 0 & -1 \end{bmatrix}$$

$$d_z = \left(d_{z,0}, \dots, d_{z,q-1}\right)^T$$

경우에서 따라서는, 그러나, 선형 부등식으로 이루어진 시스템이 해를 가지지 않을 수도 있다. 이런 경우는 복원된 형상이 서로 투과되지 않고는 곡면이 생성될 수 없을 정도로 가까운 경우이다.

4.4.2 C1 연속성

2차 곡면 $S_j(x,y)=a_j.x^2+b_j.x+c_j.y^2+d_j.y+e_j.x.y+f_j$ 과 두 정점 $v_i=(x_i, y_i, z_i)$, 그리고 $v_{i+1}=(x_{i+1}, y_{i+1}, z_{i+1})$에서 인접하는 2차 곡면들(그림 8에 도시화) 사이의 C1 연속성 제약 조건은 다음과 같은 선형 시스템으로 구성된다:

$$M_{C\to X,j} \cdot C_j = X_{i,i+1} \tag{4}$$

$$M_{C\to X,j} = \begin{bmatrix} x_i^2 & x_i & y_i^2 & y_i & x_i y_i & 1 \\ 2x_i & 1 & 0 & 0 & y_i & 0 \\ 0 & 0 & 2y_i & 1 & x_i & 0 \\ x_{i+1}^2 & x_{i+1} & y_{i+1}^2 & y_{i+1} & x_{i+1}y_{i+1} & 1 \\ 2x_{i+1} & 1 & 0 & 0 & y_{i+1} & 0 \\ 0 & 0 & 2y_{i+1} & 1 & x_{i+1} & 0 \end{bmatrix}$$

$$X_{i,i+1} = \begin{bmatrix} z_i & x_{N,i} & y_{N,i} & z_{i+1} & x_{N,i+1} & y_{N,i+1} \end{bmatrix}^T$$

$$C_j = \begin{bmatrix} a_j & b_j & c_j & d_j & e_j & f_j \end{bmatrix}^T$$

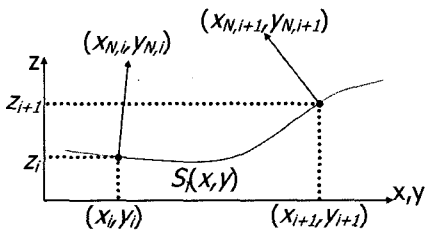

그림 8 곡면 Sj(x,y)의 정점(x_i,y_i)과 (x_{i+1},y_{i+1})에서 C1연속성을 위한 제약 조건

이 선형 시스템은 순위(rank)가 부족하며, 따라서, 이를테면 행렬 $M_{C\to X,j}$의 마지막 행 R_5는 나머지 5개의 행들의 선형 조합으로 표현 가능하다. 즉,

$$R_5 = m_{j,0}R_0 + m_{j,1}R_1 + m_{j,2}R_2 + m_{j,3}R_3 + m_{j,4}R_4,$$

$$m_{j,0} = \frac{2}{y_i - y_{i+1}}, \quad m_{j,1} = -\frac{(x_i - x_{i+1})}{y_i - y_{i+1}}, \quad m_{j,2} = -1,$$

$$m_{j,3} = -\frac{2}{y_i - y_{i+1}} \quad \text{and} \quad m_{j,4} = -\frac{(x_i - x_{i+1})}{y_i - y_{i+1}}.$$

따라서, $X_{i,i+1}$ 사이에도 비슷한 선형 의존성이 존재한다:

$$y_{N,i+1} = m_{j,0}z_i + m_{j,1}x_{N,i} + m_{j,2}y_{N,i} + m_{j,3}z_{i+1} + m_{j,4}x_{N,i+1}$$

모든 곡면에 대한 위의 선형 제약 조건을 결합하면, 다음과 같은 선형시스템을 얻는다:

$$M_{C1} \cdot X = 0 \tag{5}$$

$$M_{Coll} = \begin{bmatrix} \mathbf{m}_{0,0} & 0 & 0 & \mathbf{m}_{0,1} & 0 & 0 & \dots & 0 \\ 0 & \mathbf{m}_{1,0} & 0 & 0 & \mathbf{m}_{1,1} & 0 & \dots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \ddots & \dots & 0 \\ 0 & \ddots & \ddots & \mathbf{m}_{j,0} & 0 & 0 & \mathbf{m}_{j,1} & \vdots \\ \vdots & \dots & 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & \mathbf{m}_{n,0} & 0 & 0 & \mathbf{m}_{n,1} \end{bmatrix}$$

$$\mathbf{m}_{j,0} = \left(m_{j,0}, m_{j,1}, m_{j,2}\right), \quad \mathbf{m}_{j,1} = \left(m_{j,3}, m_{j,4}, -1\right).$$

4.4.3 곡률 에너지

휨이 적은 얇은 곡면 $s(x,y)$의 곡률 에너지의 근사값은 다음과 같다[23]:

$$E_{Curv} = \frac{D}{2}\iint_s \left(\left(\frac{\partial^2 s}{\partial x^2} + \frac{\partial^2 s}{\partial y^2}\right)^2 - 2(1-\nu)\left(\frac{\partial^2 s}{\partial x^2}\frac{\partial^2 s}{\partial y^2} - \left(\frac{\partial^2 s}{\partial x \partial y}\right)^2\right)\right) dxdy$$

상수 D는 Young(곡면의 탄력 계수)의 계수이고, v는 Poission의 비율이다. 실질적으로 Poission의 비율은 0에서 1/2 사이의 값을 사용하는 것이 일반적이며, 본 연구에서는 D=2, υ=½으로 설정하여 실험하였다. 에지 (v_i,v_{i+1}), $v_i=(x_i, y_i)$, $v_{i+1}=(x_{i+1}, y_{i+1})$를 표현하는 2차 다항식 곡면 $S_j(x,y)$ 근사화된 휨에너지는 다음 식과 같다:

$$E_{Curv,j} = l_j^2\left(4a_j^2 + 4c_j^2 + e_j^2 + 4a_j c_j\right)$$

$$l_j^2 = \left(x_{i+1} - x_i\right)^2 + \left(y_{i+1} - y_i\right)^2$$

이 식을 행렬식으로 다시 쓰면 아래와 같다.

$$E_{Curv,j} = l_j^2 \begin{pmatrix} a_j & c_j & e_j \end{pmatrix} H \begin{pmatrix} a_j & c_j & e_j \end{pmatrix}^T$$

$$H = \begin{bmatrix} 4 & 2 & 0 \\ 2 & 4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$E_{Curv,j}$가 볼록 2차 다항식 형태이므로, 이 목적함수를 최소화하는 계수 a_j, c_j, e_j는 다음의 최소제곱문제를 풀어 계산할 수 있다:

$$\min_{a_j, c_j, e_j} \left\| U \cdot \begin{pmatrix} a_j & c_j & e_j \end{pmatrix}^T \right\| \tag{6}$$

행렬 $U = \begin{bmatrix} 2 & 1 & 0 \\ 0 & \sqrt{3} & 0 \\ 0 & 0 & 1 \end{bmatrix}$는 H의 촐레스키(Cholesky) 분해로부터 계산된다. 근사 곡률 에너지는 선형식이기 때문에 비선형항을 포함하는 정확한 식을 다룰 때보다 계산

량이 줄어든다[22].

지금까지 곡률에너지의 목적함수를 곡면 다항식의 계수로 구성된 변수 C_j에 대한 식으로 표현하였는데, 곡률에너지항이 식(1)과 같은 전역 최적화 문제에 통합될 수 있도록 이제 골격 변수 $X_{i,i+1}$의 함수 형태로 다시 쓸 필요가 있다. C_j와 $X_{i,i+1}$는 식 (4)에서 보였듯이 선형적 관계에 있다; 그러나, 행렬 $M_{C\to X,j}$는 순위(rank)가 부족하기 때문에 역행렬을 직접 구할 수 없다. 따라서 $M_{X\to C,j}$ $X_{i,i+1}$가 식 (6)의 곡률 최소화 문제의 해인 동시에 식 (4)의 선형 시스템의 해가 되도록 행렬 $M_{C\to X,j}$을 계산한다. 즉, $M_{X\to C,j}$ $X_{i,i+1}$ 는 선형 제약조건을 가지는 아래 최소 제곱법 문제의 해이다:

$$\min_{C_j}\left\|U'C_j\right\| \tag{7}$$

$$\text{subject to} \quad M'C_j = X_{i,i+1} \tag{8}$$

$$U'=\begin{bmatrix}2&0&1&0&0&0\\0&0&\sqrt{3}&0&0&0\\0&0&0&0&1&0\end{bmatrix}$$

$$M'=\begin{bmatrix}x_i^2 & x_i & y_i^2 & y_i & x_iy_i & 1\\ 2x_i & 1 & 0 & 0 & e_jy_i & 0\\ 0 & 0 & 2y_i & 1 & e_jx_i & 0\\ x_{i+1}^2 & x_{i+1} & y_{i+1}^2 & y_{i+1} & x_{i+1}y_{i+1} & 1\\ 2x_{i+1} & 1 & 0 & 0 & e_jy_{i+1} & 0\end{bmatrix}$$

위 식에서 M'은 $M_{C\to X,j}$의 6개행 중 5행으로 구성되어 있으며, 이것은 순위부족(rank deficient) 하지 않다. 등식 제약 조건을 가지는 최소 제곱 문제를 풀기 위한 방법으로 직접소거법[24]을 사용하였다. 우선 주어진 행렬들을 다음과 같이 분할(partitioning) 한다:

$$\begin{bmatrix}M'\\U'\end{bmatrix}=\begin{bmatrix}M_1' & M_2'\\ U_1' & U_2'\end{bmatrix}\begin{matrix}\}5\\\}3\end{matrix} \quad \text{and} \quad C_j=\begin{bmatrix}C_{j,1}\\C_{j,2}\end{bmatrix}\begin{matrix}\}5\\\}1\end{matrix}$$

(M_1', M_2' 아래 5, 1; U_1', U_2' 아래 5, 1)

$C_{j,1}$는 제약조건 (8)을 사용하여 다음과 같이 $C_{j,2}$의 함수로 표현된다.

$$C_{j,1}=M_1'^{-1}(X_{i,i+1}-M_2'C_{j,2}) \tag{9}$$

위 식을 식 (7)에 대입하여 $C_{j,1}$의 표현으로 치환하면 다음 식을 얻는다.

$$\left\|U'\cdot C_j\right\|=\left\|\tilde{U}_2'\cdot C_{j,2}+U_1'M_1'^{-1}X_{i,i+1}\right\|$$

위 식에서 $\tilde{U}_2'=U_2'-U_1'M_1'^{-1}U_2'$이고, 이 최소 제곱 문제의 해는 다음 식과 같다:

$$C_{j,2}=-\left(\tilde{U}_2'^T\tilde{U}_2'\right)^{-1}\tilde{U}_2'^TU_1'M_1'^{-1}X_{i,i+1}$$

이 식을 식 (9)에 대입하여 $C_{j,2}$를 치환하면,

$$C_{j,1}=M_1'^{-1}\left(I+M_2'\left(\tilde{U}_2'^T\tilde{U}_2'\right)^{-1}\tilde{U}_2'^TU_1'M_1'^{-1}\right)X_{i,i+1}$$

결과적으로, C_j를 $X_{i,i+1}$의 함수로 표현할 수 있게 하는 행렬 $M_{X\to C,j}$ 를 구한 것이 된다:

$$C_j=\begin{bmatrix}M_1'^{-1}\left(I+M_2'\left(\tilde{U}_2'^T\tilde{U}_2'\right)^{-1}\tilde{U}_2'^TU_1'M_1'^{-1}\right)\\ -\left(\tilde{U}_2'^T\tilde{U}'\right)^{-1}\tilde{U}_2'^TU_1'M_1'^{-1}\end{bmatrix}X_{i,i+1}$$
$$=M_{X\to C,j}\cdot X_{i,i+1}$$

따라서, $X_{i,i+1}$의 함수로 정의된 곡률 에너지의 목적함수는 다음과 같다.

$$\min_{C_j}\left\|U'C_j\right\|=\min_{X_{i,i+1}}\left\|U'M_{X\to C,j}X_{i,i+1}\right\|$$

위의 목적 함수를 모든 곡면에 대하여 결합하면 다음 식으로 정리된다.

$$\min_X\left\|U_{Curv}X\right\| \tag{10}$$

위 식에서 U_{Curv} 는 $l_jU'M_{X\to C,j}$ (j=0,…,n, n은 에지의 갯수) 들로 채워진 것이다.

4.4.4 골격의 오리엔테이션(방향)

복원된 형상의 질을 평가하기 위한 두 번째 기준은 골격이 스케치 평면 z=0과 이루는 방향각을 최소화하는 것이다.

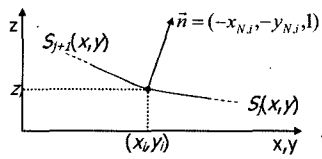

그림 9 정점 v_i 에서의 법선 벡터

함수 $S_j(x,y)$에 의하여 정의된 곡면의 법선 벡터는 다음과 같다:

$$\vec{n}=\left(-\frac{\partial S_j(x,y)}{\partial x},-\frac{\partial S_j(x,y)}{\partial y},1\right) \quad \text{(그림 9 참조)}$$

3.4.1절에서, $x_{N,i}=\dfrac{\partial S_j(x,y)}{\partial x}$ 와 $y_{N,i}=\dfrac{\partial S_j(x,y)}{\partial y}$ 으로 점 v_i에서의 변수($x_{N,i}$, $y_{N,i}$)를 정의하였고, 곡면 $S_j(x,y)$는 이 정점v_i에 연결된 2차 곡면이었다. 따라서, 골격의 방향각을 최소화하기 위한 목적 함수는 $\sum_{i=0}^{m-1}\left(x_{N,i}{}^2+y_{N,i}{}^2\right)$ 이다.

이 목적 함수를 행렬식으로 다시 쓰면 다음과 같다.

$$\min_X\left\|U_{Orient}X\right\| \tag{11}$$

$$U_{Orient}=\begin{bmatrix}0&1&1&0&\cdots&\cdots&\cdots&\cdots&0\\0&0&0&0&1&1&0&\cdots&0\\\vdots&&&&&&&&\vdots\\0&\cdots&\cdots&\cdots&\cdots&\cdots&0&1&1\end{bmatrix}.$$

4.4.5 스케칭 평면과의 거리

마지막으로 정의할 목적 함수는 복원된 물체가 스케칭 평면으로부터 최소화의 거리를 갖기 위한 것이다. 골격상의 각 정점 $v_j(x_j,y_j)$에 대한 목적함수는 z_i^2 이다; 모든 점에 대해 정의된 목적함수는 따라서,

$$\sum_{i=0}^{m-1}\left(z_i^2\right)$$

이다. 이 목적함수의 행렬식은 다음과 같다:

$$\min_X \|U_{Dist}\cdot X\| \tag{12}$$

$$U_{Dist}=\begin{bmatrix}1&0&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&0\\0&0&0&1&0&\cdots&\cdots&\cdots&\cdots&\cdots&0\\0&0&0&0&0&0&1&0&\cdots&\cdots&0\\\vdots&&&&&&&&&&\vdots\\0&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&0&1&0&0\end{bmatrix}$$

4.5 최적화 문제의 해법

최종 목적 함수는 앞서 정의한 세 개의 목적함수, 즉 식 (10)~(12)에 가중치를 곱해 더함으로써 얻는다:

$$\min_X\left\|\begin{bmatrix}w_{Curv}\cdot U_{Curv}\\w_{Orient}\cdot U_{Orient}\\w_{Dist}\cdot U_{Dist}\end{bmatrix}X\right\|=\min_X\|U_{Total}X\|$$

일반적으로 결과는 그림 10에서 보는 것처럼 가중치 (w_{Curv}, w_{Orient}, w_{Dist})의 선택에 의해 영향을 받는다. 본 연구에서는 사용자가 가중치를 모든 복원된 물체에 대해, 혹은 물체별로 조절할 수 있게 하였다.

또한, U_{total}이 악조건(ill condition)이 되지 않도록 가중치의 범위를 제안하였다. 그렇지 않으면 최소화 과정은 수치적 에러가 커져 실패할 수도 있다. 실험적으로 가장 큰 가중치와 가장 작은 가중치의 비율이 10^3보다 작을 때 좋은 결과가 나왔다. 그림 10(b)~(c)는 서로 다른 가중치를 사용하여 복원된 결과를 보인다.

LSIE 패키지[24]를 사용하여 식(1)에서 정의한 최소 제곱 문제의 해를 구하였다. 다만, 이 패키지의 선형대수 루틴을 Intel Math Kernal 라이브러리[25]의 루틴으로 대치하여 속도 향상을 꾀하였다. 형상 복원 알고리즘의 대부분의 계산시간은 최소 제곱 해의 계산에 소요된다. 그럼에도 전체적인 수행은 적절히 짧은 시간 안으로 유지되었다; 그림 10에 보인 모델들의 복원 시간은 완성된 스케치가 입력된 이후 5초 이내이다.

5. 골격으로부터 3차원 볼륨 생성

지금까지는 2차원 스케치로부터 추출해낸 골격을 3차원 공간에 위치시키는 최적화 문제에 촛점을 맞추었다. 앞서 정의한 최적화 문제의 해는 입력 실루엣과 일치하면서 충돌이 일어나지 않는 3차원 골격이다. 이 골격은

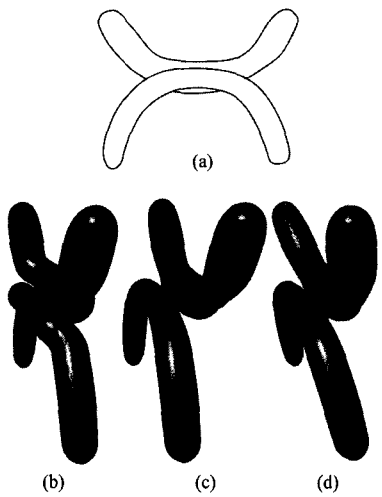

그림 10 예제 스케치(a)로부터의 복원 결과: (b) 곡률, (c)거리, (d) 오리엔테이션 목적 함수를 각각 작은 값으로 주었을 때 결과

또한 휨과 오리엔테이션, 그리고 스케칭 평면과의 거리를 최소화한다. 이제 이렇게 얻어진 골격 주위에 3차원 메쉬를 어떻게 생성하는지에 대해 기술하겠다.

간단히 말해서, 3차원 형상은 골격을 중심으로 하여 실루엣으로 의해 둘러싸인 영역을 팽창시킴으로써 얻어진다. 이와 같은 작업을 수행하기 위한 여러 가지 방법들이 제안되었는데, 대부분이 음함수 곡면(implicit surface)을 사용하여 곡면을 계산한다[4,2]. 본 논문에서는 구현이 비교적 간단한 Alexe 외가 제안한 곡면 모델링 방법[5]을 선택하였다. 그 방법에 의하면, 임의의 곡면은 골격을 따라 위치시킨 여러 개의 구형 음함수 곡면들을 블렌딩함으로써 생성된다. Alexe 외는 실루엣 곡선에 음함수 곡면을 일치시키기 위한 구형 음함수 곡면의 반지름 계산법을 제안했다. 본 논문에서는 구형 음함수 곡면의 식을 약간 변경하여, 곡면을 이루는 각각의 음함수를 가중치와 거리의 함수가 아닌 가중치만의 함수로 대치하고, 다음의 선형시스템의 최소 제곱해를 구하였다.

$$\sum_i f(w)=\sum_i \frac{w_i}{d_{ij}^2}=C, j=1,2,...,i$$

위 식에서 d_{ij} 는 음함수 i의 중심과 실루엣 상의 점 j 사이의 거리이고, C는 등위값 상수이다.

Alexe가 곡면 fitting을 위해 비선형 최소화를 사용했던 것을 선형식의 최소화 문제로 대치하였으므로, 자연스럽게 계산 시간이 단축된다. 보다 구체적으로 말해서, 동일한 개수의 구형 음함수 곡면을 사용하였을 때 계산 시간을 수 초에서 수백 밀리 초로 단축할 수 있었다.

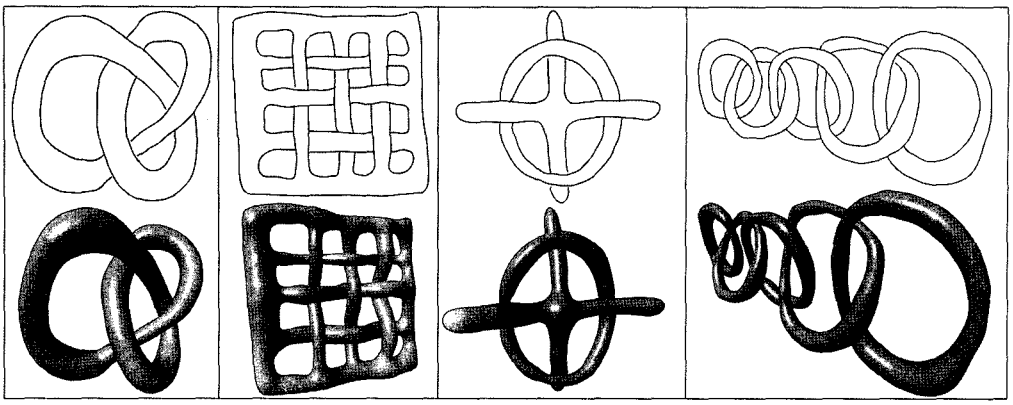

그림 11 다양한 입력 스케치를 사용한 3차원 모델의 복원

6. 구현 및 결과

2차원 드로우잉 툴과 3차원 형상 복원 소프트웨어 모두 상용 그래픽스 패키지인 Maya[26]의 플러그인을 사용하여 구현하였다. 사용자가 직교투영 화면에 스트로크를 그리면 결과는 원근투영 화면에 보여지게 된다. 최적화 알고리즘은 Intel Math Kernel 라이브러리[25]를 사용하여 구현하였다. 이 수치해석용 패키지는 본 논문에서 정의한 최소 제곱 문제에 포함된 선형 시스템을 풀기 위한 효율적인 방법들을 제공한다.

구현된 시스템을 다양한 스케치 예제에 대하여 적용하여 3차원 형상 복원을 테스트해 보았다. 실험에서 스케치를 위해 타블렛을 사용하였고, 그림 11은 예제 스케치와 복원 결과들을 보인다. 스케치 입력의 완료 후 5~10초 안에 3차원 메쉬가 생성된다. 결과 형상의 질은 [4]에서 얻은 결과물의 질과 비견할만 하다. 그러나, 새로운 모델러에서 사용자는 이제 서로 겹쳐진 물체들도 그릴 수 있다.

7. 결 론

본 논문에서는 자신 혹은 다른 물체에 의해 가려진 부분이 있는 물체를 대상으로 하여 스케치 입력으로부터 3차원 물체를 복원하는 방법을 제안하였다. 그림 11에 보인 예제들은 제안된 방법의 효용성을 잘 나타낸다. 우리가 아는 한 본 연구는 실루엣 기반 모델링 도구 중 겹쳐진 부분을 허용하는 최초의 시도이다.

본 연구의 또 다른 기여는 복원 물체의 깊이값 계산 과정을 최적화 문제로 표현한 것이다. 우리의 수학적 모델은 단순하고 또 구현을 위해서는 목적함수와 제약조건 행렬을 계산하기만 된다. 이러한 모델의 단순성에도 불구하고 본 알고리즘은 서로 가려진 여러 개의 물체로 이루어진 그룹에 대한 스케치, 혹은 자기 자신에 의한 가려짐이 있는 물체의 스케치에 모두 잘 동작한다. 본 모델러에서 모든 입력 스케치는 동일한 방법으로 처리되며, 이론적으로 물체의 개수에 제한을 두지 않는다.

향후에는 본 모델링 기법을 확장하여 대화형 편집 및 변경이 가능하도록 할 것이다. 예를 들어 Teddy에서와 같이 사용자가 이미 생성된 모델에 대해 자르기, 돌출(extrude), 왜곡 (distort)과 같은 연산을 수행할 수 있을 것이다.

참 고 문 헌

[1] T. Igarashi, S. Matsuoka, H. Tanaka. "Teddy: A Sketching Interface for 3D Freeform Design," SIGGRAPH 99 Conference Proceedings, pp. 409-416, 1999.

[2] C.-L. Tai, H. Zhang, J. C. Fong, "Prototype Modeling from Sketched Silhouettes based on Convolution Surfaces," Computer Graphics Forum 23(1): 71-84, 2004.

[3] S. Owada, F. Nielsen, K. Nakazawa, T. Igarashi: "A sketching interface for modeling the internal structures of 3D shapes," Proc. 3rd International Symposium on Smart Graphics, Springer-Verlag, Berlin, pp. 49-57, 2003.

[4] O. Karpenko, J. F. Hughes, R. Raskar, "Free-form sketching with variational implicit surfaces," Computer Graphics Forum 21(3), 2002.

[5] I. A. Alexe, V. Gaildrat, L. Barthe, "Interactive modelling from sketches using spherical implicit functions," Afrigraph 2004: 25-34.

[6] H. Lipson, M. Shpitalni, "Optimization-based reconstruction of a 3D object from a single free-hand line drawing," Computer-aided Design 28(8): 651-663, 1996.

[7] H.Lipson, M.Shpitalini, "Correlation-based reconstruction of a 3d object from a single freehand sketch," AAAI Spring Symposium on Sketch

Understanding, pp. 99-104, 2002.
[8] A. Piquer, R. Martin, P. Company, "Using skewed mirror symmetry for optimisation-based 3d line-drawing recognition. In Proc. 5th IAPR International Workshop on Graphics Recognition," pp. 182-193, 2003.
[9] R. C. Zeleznik, K. P. Herndon, J. F. Hughes: "SKETCH: An interface for sketching 3D scenes," In Proceedings of ACM SIGGRAPH 96, Addison-Wesley, Boston, Massachusetts, pp. 163-170.
[10] A. Shesh, B. Chen, "SMARTPAPER: An Interactive and User Friendly Sketching System," Computer Graphics Forum 23(3): 301-310, 2004.
[11] O. Tolba, J. Dorsey, L. McMillan, "A projective drawing system," Proc. I3D Symposium on Interactive 3D Graphics, 2001.
[12] J. Cohen, L. Markosian, R. Zeleznik, J. Hughes, and R. Barzel, "An interface for sketching 3D curves," ACM Symposium on Interactive 3D Graphics, pp. 17-21, 1999.
[13] B. Kerautret, X. Granier and A. Braquelaire, "Intuitive Shape Modeling by Shading Design," Proc. the 5th International Symposium on Smart Graphics, pp. 163-174, LNCS 3638, Springer-Verlag 2005.
[14] A. Nealen, O. Sorkine, M. Alexa, D. Cohen-Or, "A sketch-based interface for detail-preserving mesh editing," ACM Trans. Graph. 24(3): 1142-1147, 2005.
[15] Y. Kho and M. Garland, "Sketching mesh deformations," Proceedings of the ACM Symposium on Interactive 3D Graphics, April 2005.
[16] T. Pavlidis, C. J. Van Wyk, "An automatic beautifier for drawings and illustrations," ACM SIGGRAPH Computer Graphics, Volume 19, Issue 3, pp. 225-234, July 1985.
[17] M. Shpitalni, H. Lipson, "Classification of sketch strokes and corner detection using conic sections and adaptive clustering," ASME Journal of Mechanical Design 119(2): 131-135, 1997.
[18] T. M. Sezgin and R. Davis. "Handling Overtraced Strokes in Hand-Drawn Sketches," In Making Pen-Based Interaction Intelligent and Natural. 2004.
[19] L.R. Williams, "Perceptual Completion of Occluded Surfaces," PhD dissertation, Dept. of Computer Science, University of Massachusetts, Amherst, Mass., 1994.
[20] L. Prasad, "Morphological analysis of shapes," CNLS Newsletter, 139: 1-18, July 1997.
[21] F. Cordier, O. Cheong, "Delaunay triangulation of self-intersecting polygons," technical report, CS Depts., 2005, KAIST.
[22] W. Wesselink. "Variational Modeling of Curves and Surfaces," PhD thesis, University of Technology, Eindhoven, 1996.
[23] L. Landau, E. Lifshitz, "Theory of Elasticity," Butterworth-Heinemann, Boston, Mass., 1995.
[24] C. Lawson and R. Hanson, "Solving Least Squares Problems," Englewood Cliffs, NJ: Prentice-Hall, 1974.
[25] Intel Math Kernel Library 8.0, http://www.intel.com, 2005.
[26] Maya, http://www.alias.com, 2005.

코디에 프레데릭
1996년 프랑스 University of Claude-Bernard Lyon I 전산학 학사. 1998년 University of Claude-Bernard Lyon I 전산학 석사. 2004년 스위스 University of Geneva 전산학 박사. 2004년~2005년 KAIST 컴퓨터그래픽스 연구실 Post-doc 연구원. 2005년~현재 KAIST 문화기술대학원 초빙교수. 관심분야는 실시간 옷감 시뮬레이션, 스케치 입력 기반 모델링 및 텍스처매핑 등

서 혜 원
1996년 KAIST 전산학과 학사. 1998년 KAIST 전산학과 석사. 1999년 MIRA-Lab, University of Geneva 교환연구원 2004년 스위스 University of Geneva 전산학 박사. 2004년~현재 충남대학교 전기정보통신공학부 전임강사. 연구분야는 물리기반 시각 시뮬레이션, 인간과 컴퓨터 상호작용, 가상 현실 등

조 영 상
2003년 고려대학교 이과대학 컴퓨터학전공 학사. 2005년 KAIST 전산학과 전산학전공 석사. 2005년~현재 KAIST 전산학과 전산학전공 박사과정. 관심분야는 컴퓨터 그래픽스, 물리 기반 캐릭터 동작 생성