

# 스케치 기반 메쉬 객체 결합 기법

권지용<sup>○</sup> 김나리 이인권 최윤철

연세대학교 컴퓨터과학과

{mage,wassupnari}@cs.yonsei.ac.kr, iklee@yonsei.ac.kr, ycchoy@rainbow.yonsei.ac.kr

## GluePaste : A Sketch-based mesh composition method

Ji-yong Kwon<sup>○</sup> Nari Kim In-Kwon Lee Yoon-Chul Choy

Department of Computer Science, Yonsei University

### 요 약

본 연구에서는 스케치를 기반으로 한 3D 모델의 어셈블링 기법을 제안한다. 본 연구에서 제안하는 시스템은 Teddy 모델링 시스템[4]을 기본으로 하여, 다양하고 복잡한 형태의 오브젝트의 모델링을 보다 단순한 오브젝트들의 합성을 통해 해결한다. 제안하는 시스템은 두 개 이상의 3D 모델을 생성하는 멀티 페이지 기능을 제공하며, 두 개 이상의 메쉬를 조립하여 하나의 메쉬를 만드는 조립 모드를 제공한다. 여기에서 두 개의 메쉬를 보다 직관적이고 쉽게 조립하기 위해 스케치 기반의 객체 조합 기법을 제안한다. 제안하는 시스템을 사용하면 복잡한 형태의 오브젝트를 보다 쉽고 빠르게 제작할 수 있다.

### 1. 서 론

간단한 스케치를 통해서 3D 모델을 생성할 수 있다는 것은 숙련되지 않은 사용자들에게 많은 편의를 제공하며 다양한 흥미를 유발시킬 수 있다. 전문가들이 사용하는 3D 모델링 프로그램들은 일정 기간 이상의 학습이 없으면 간단한 물체를 만드는 데에도 많은 시간과 노력이 필요하다. 반면 기존에 연구된 스케치 기반의 모델링 결과들[1], [2], [3]은 빠른 시간과 적은 노력으로 사용자가 얻고자 하는 3D 모델을 얻어 낼 수 있다는 점에서 상당한 장점들을 가지고 있다.

특히 Igarashi등이 제안한 'Teddy' 시스템[4]은 자유 곡선 형태의 입력을 기초로 하여 매우 쉽게 원하는 3D 모델을 만들 수 있다. 그러나 [5]에서 지적하였듯이, 이러한 스케치 기반의 모델링 시스템은 여러 개의 3차원 객체를 복합적으로 편집하는 데에는 편리하지 않다.

본 논문에서는 스케치를 입력으로 하여 임의의 두 개의 오브젝트를 손쉽게 결합시키는 인터페이스를 제안한다. 이러한 인터페이스를 추가한 스케치 기반의 모델링 도구는 여러 개의 오브젝트로 구성된 복잡한 형태의 오브젝트를 손쉽게 모델링할 수 있도록 한다. 우리는 Teddy 시스템의 기본적인 부분들을 구현하고 여기에 추가적으로 여러 개의 오브젝트를 편집할 수 있는 기능과 함께 본 논문에서 제안하는 GluePaste 인터페이스를 구현하여, 제안하는 인터페이스의 효용성을 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련된 여러 연구들을 살펴보고 3장에서는 시스템의 사용자

인터페이스를 설명 하며 4장에서는 본 연구에서 구현한 내용과 상세한 알고리즘을 설명한다. 5장에서는 구현 결과와 평가에 대해 논하고 마지막으로 6장에서는 결론을 기술한다.

### 2. 관련 연구

최근 10여년간 다양한 형태의 스케치 기반 모델링 연구들이 제시되어 왔다. 3차원 객체를 편집하기 위한 스케치 기반 모델링 시스템은 크게 제스처를 이용한 모델링 기법과 자유 곡선을 이용한 모델링 기법으로 나누어 볼 수 있다.

Zelevnik[6]의 SKETCH에서는 정의된 제스처의 형태로 인식하여 미리 정의된 제스처에 해당 하는 3D 모델을 생성한다. 한편 Yang 등[1]은 오브젝트와 그에 해당하는 제스처 스케치를 미리 정하여 입력으로 들어오는 스케치와 유사한 형태의 오브젝트를 자동으로 생성하는 기법을 제안하였다. Shesh와 Chen이 제안한 SMARTPAPER[7]에서는 자유로운 형태의 스케치를 입력으로 하여 일정한 시점에서의 3차원 모델을 생성하는 데에 적합한 시스템을 제공한다. 이와 같은 제스처 기반의 모델링 기법은 정해진 형태의 오브젝트를 이용하므로, 자유로운 형태의 오브젝트를 만들어 내기는 힘들다는 단점이 있다.

한편 자유 곡선을 기반으로 한 스케치 모델링 시스템은 자유로운 형태의 모델들을 쉽게 생성할 수 있는 장점이 있다. Teddy[4] 시스템은 자유 곡선 스케치를 이용한 여러 가지 인터페이스와 오퍼레이션을 연결하여 자유 곡선 기반의 모델링에 대한 가능성을 보여주었다. Cherlin 등[5]은 간단한 스케치 입력으로

표 1. 그리기 모드

Key	Action
<i>Left mouse</i>	Drawing
<i>Ctrl + left mouse</i>	Rotation
<i>Shift + left mouse</i>	Panning

표 2. 조립 모드

Key	Action
<i>Left mouse</i>	Drawing
<i>Ctrl + left mouse</i>	Rotation
<i>Shift + left mouse</i>	Panning
<i>Ctrl + Shift + left mouse</i>	Scaling
<i>Left mouse click on small window</i>	Mesh positioning

다양한 형태의 3D모델을 생성하고 수정할 수 있는 기법에 대해 제안하였다. 하지만 논문의 마지막 부분에서 생성된 3D모델들을 어셈블링 하는 데에 많은 시간이 소요 되었다고 밝히고 있다.

한편 두 개의 오브젝트를 부드럽게 이어주는 연구 또한 많이 이루어졌다. Sharf 등[8]은 두 메쉬 간의 자연스러운 합성 방법에 대한 알고리즘을 제안한 바 있다. 본 연구에서는 직접적으로 이러한 알고리즘을 이용하지는 않으나, 제안하는 인터페이스를 통해 두 메쉬의 이어 붙이고자 하는 영역을 계산할 경우 위 알고리즘을 이용하여 위상학적으로 하나의 오브젝트를 만들 수 있을 것이다. 그러나 본 연구의 목적은 제안하는 방식의 인터페이스가 모델링에 효율적인지 여부에 대하여 논하는 것이므로 단순한 일차변환이동을 통해 두 오브젝트를 붙이는 것으로 대신하였다.

### 3. 유저 인터페이스

제안하는 시스템은 그리기 모드와 조립 모드라는 두 가지 모드를 제공한다. 그리기 모드에서의 사용자 인터페이스는 Igarashi[4]의 Teddy 시스템을 기본으로 한다. 이는 Teddy 시스템이 발표된 이후 통상적인 자유 곡선 입력 기반의 모델링 시스템이 Teddy와 유사하기 때문에 Teddy 시스템을 가장 일반적인 형태의 스케치 기반 모델링 시스템으로 보고 채택하였다. 다만 본 논문이 제안하는 시스템은 여러 개의 오브젝트를

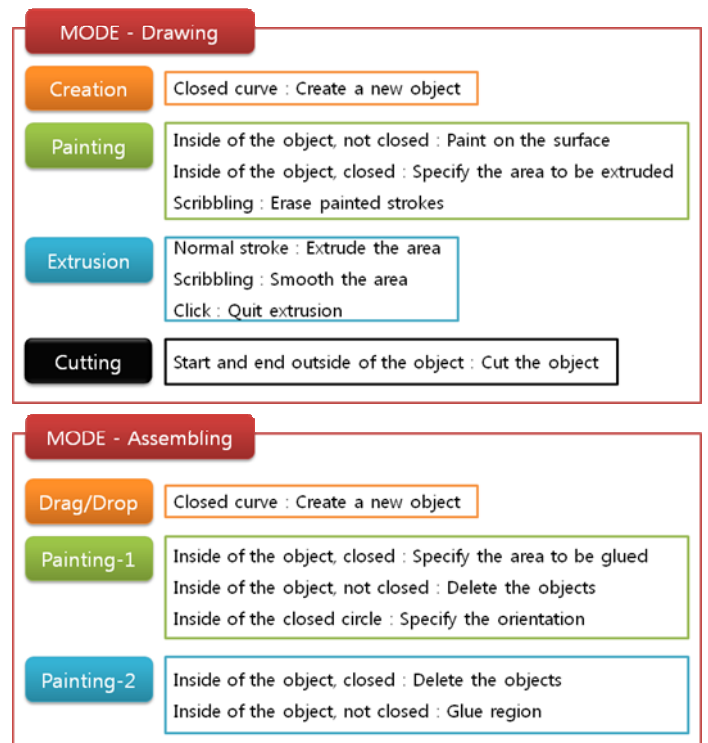


그림 1. 시스템의 제스처 인식 기준.

생성하고 이를 조합하는 기능이 필요하므로, 여러 개의 Teddy 프로그램을 동시에 실행한 것과 같은 멀티 페이지 기능을 추가적으로 구현하였다. 조립 모드에서는 앞의 그리기 모드에서 만들어진 오브젝트들을 조립하는 기능을 제공한다.

그리기 모드의 경우, 2D 평면 위에 사용자가 생성하고자 하는 모델의 실루엣을 달린 커브의 제스처로 그려주면 해당 실루엣을 바탕으로 풍선 모양으로 팽창된 3D 메쉬 오브젝트를 생성해 준다. 각 페이지에서 생성된 모델은 해당 캔버스에 저장되며 메인 화면 아래의 작은 화면을 클릭하면 다른 새로운 캔버스를 제공하여 여러 개의 오브젝트들을 생성할 수 있는 기능을 제공한다 (그림 2 참조).

조립 모드에서 사용자는 원하는 오브젝트들간의 결합을 할 수 있다. 캔버스 아래에 있는 작은 화면을 클릭하면 해당 페이지에 저장된 오브젝트를 화면으로 불러들일 수 있다. 불러들인 각각의 오브젝트는 마우스와 키 조작을 혼용하여 평행이동, 회전이동 및 크기조절 등 기본적인 일차변환이 적용 가능하다. 여기서 두 개의 모델에 붙여질 영역을 그려줌으로써 두 오브젝트를 쉽게 붙일 수 있다 (그림 3, 4 참조).

표1 및 표2는 우리가 실험적으로 구현한 시스템에서의 사용자 인터페이스 키 조작을 보여준다. 입력된 자유 곡선의 모양에 따른 시스템의 동작에 대한

정의는 그림 1에 자세하게 설명되어 있다. 스케치는 그리기 모드와 조립 모드에서 각각 다른 제스처로 인식된다. 즉, 그리기 모드에서 오브젝트 안쪽에 그리는 스케치가 열려있는지 혹은 닫혀있는지의 여부에 따라 각각 페인팅과 익스트루전으로 인식되는 반면, 조립 모드에서는 붙이기 수행과 오브젝트 삭제로 인식된다.

#### 4. 시스템 구성 및 알고리즘

##### 4.1 그리기 모드와 멀티 페이지 기능

기본적인 기능들은 그림 2과 같이 Igarashi[4]의 Teddy 시스템에서 제안한 기법들 중 모델링에 가장 필요한 기능들, 즉 초기 메쉬 생성, 자르기, 돌출, 페인팅 기능만을 구현하였다. 각각의 알고리즘은 [4]에서 소개한 방법을 그대로 사용하였다.

초기의 사용자 인풋으로 생성된 blob 메쉬를 돌출(Extruding), 자르기(Cutting)등의 기능을 사용하여 변형시킬 수 있다. 돌출을 할 경우에는 그림2의 (i)에서와 같이 메쉬 위에 닫힌 형태의 원을 그려주면 된다. 이때 원 안에 생기는 점은 해당 되는 영역의 법선 벡터 방향을 나타낸다. 그리고 법선 벡터의 방향이 현재 시점의 방향과 수직에 가깝게 물체를 회전 시킨 후 그림2의 (k)와 같이 추가하고자 하는 형태에 맞게 그려주면 돌출된 형태의 모델을 얻을 수 있다.

자르기의 경우 그림2의 (g)와 같이 메쉬 모델이 아닌 바탕 화면에서부터 시작되는 스트로크를 제스처로 인식하여 인식된 스트로크를 단면으로 하여 객체를 잘라 내는 기능을 수행한다. 이때 스트로크의 왼쪽 부분을 남기고 오른쪽 부분을 삭제하게 된다.

기존의 Teddy 시스템과는 달리 본 논문이 제안하는

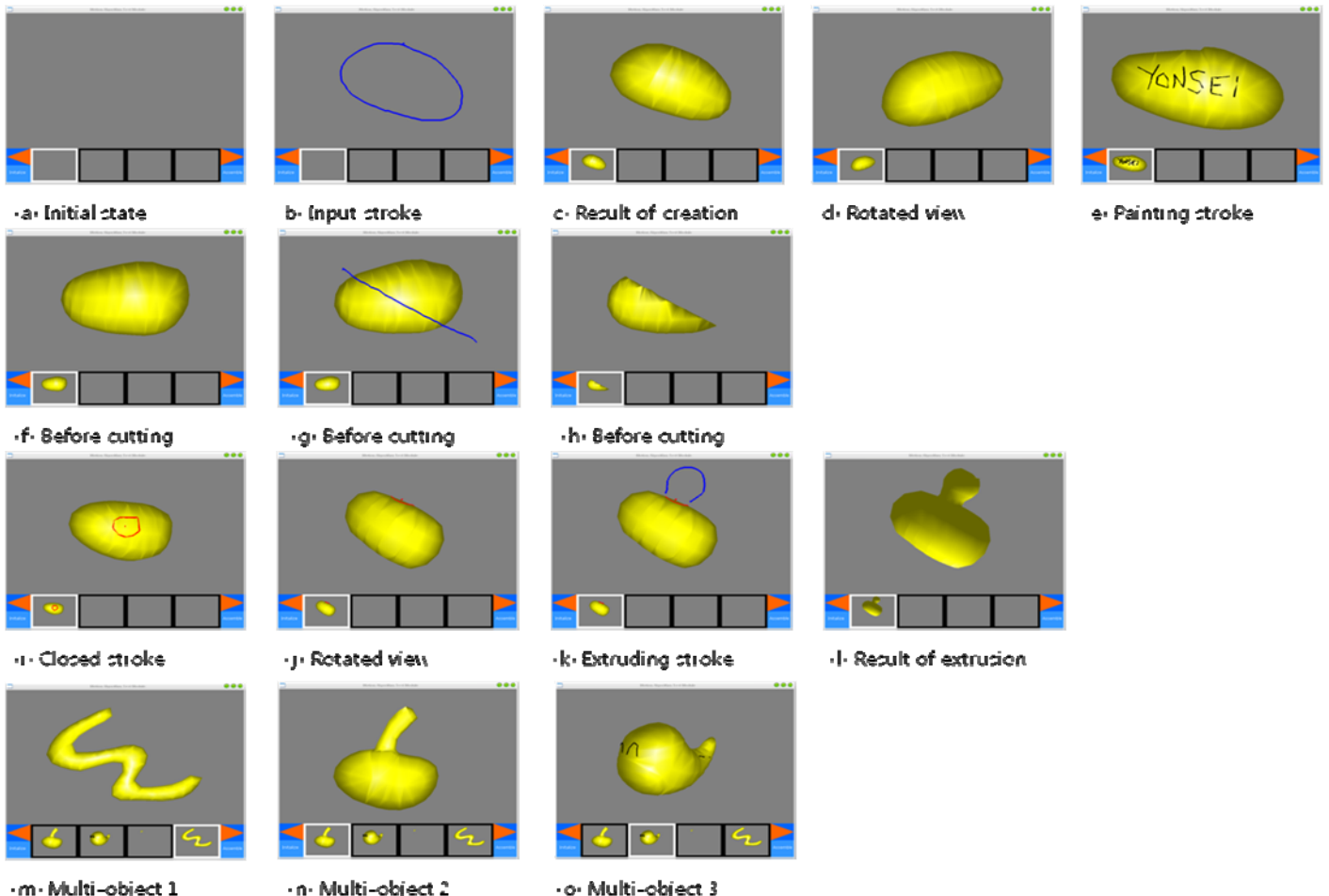


그림 2. 그리기 모드의 기능 예시. (a)첫 실행 화면. (b)(c) 2D 캔버스에 닫힌 형태의 스트로크를 그려주면 3D 메쉬 오브젝트 생성. (d)생성된 모델을 다른 각도로 회전한 모습. (e)오브젝트 위에 열린 스트로크를 그려 페인팅을 한 예. (f)(g)(h)오브젝트 전체를 가로지르는 열린 스트로크를 통해 오브젝트를 자른 예. (i)(j)(k)(l)오브젝트 위에 닫힌 스트로크를 그린 후 회전한 뒤 열린 스트로크를 그려 돌출된 모델을 생성한 모습. (m)(n)(o)멀티 페이지의 1,2,4번째 페이지에서 메쉬를 생성한 모습.

시스템은 여러 개의 오브젝트를 각각의 페이지에 저장하는 것이 가능하다 (그림2(m)(n)(o) 참조). 이와 같이 각각의 페이지에 저장된 오브젝트들은 조립 모드에서 페이지 단위로 복사해 붙여 넣는 것이 가능하다. 두 개의 오브젝트를 조립하는 GluePaste 인터페이스에 대한 설명은 다음 절에 상세히 소개하겠다.

#### 4.2 GluePaste

앞 절에서 간략히 언급한 바와 같이 조립 모드에서는 두 개 이상의 오브젝트를 동일한 캔버스 화면에 불러올 수 있다. 기존의 Teddy 방식에서는 단 하나의 오브젝트만 생성 및 편집이 가능하였으므로 단순한 카메라의 이동만으로 물체를 회전시킬 수 있지만 본 논문에서 제안하는 시스템에서는 각각의 물체를 개별적으로 조작하는 것이 가능해야 하므로, 카메라를 조절하는 것 보다는 각각의 객체에 로컬 좌표계를 조작하는 방법을 채택하였다. 또한 결합하고자 하는 두 개의 오브젝트가 서로 원하는 크기로 맞춰져 있지 않은 경우가 있을 수 있으므로, 크기 조절 기능을 추가적으로 지원하였다.

두 개 이상의 물체를 합성하는 스케치 인터페이스를 위해 본 논문에서는 두 가지의 조립 인터페이스를 제안하고 이를 구현하여 실험하였다. 첫번째 방법은 두 개의 물체에 각각 달린 영역을 그려주어 서로 붙을 부분임을 명시하는 방법이다. 두 개의 달린 스트로크를 정렬하는 문제는 많은 모호성 문제를 야기하므로, 여기서 우리는 달린 스트로크 이외에 추가적으로 방향을 결정하는 스트로크를 하나 더 그려넣는 방법을 택하였다 (그림 3 참조). 즉 하나의 오브젝트당 영역 스트로크와 방향 스트로크의 두 가지 스트로크를 입력받게 된다. 이를 이용하여 붙여질 부분의 지역좌표계를 생성하는 방법은 다음과 같다. 먼저 영역 스트로크의 모든 점에 대하여 무게 중심점을 계산하고, 또한 영역 스트로크를 가장 잘 포함하는 평면을 구하여 이의 법선 벡터  $z$ 를 계산한다. 다음 영역 스트로크의 무게 중심점과 방향 스트로크의 무게 중심점을 연결한 벡터를  $x$ 라 하고, 이 둘을 외적한 벡터  $y$ 를 구한 뒤 다시  $y$ 와  $z$ 를 외적함으로써  $x$ 를 계산한다. 이렇게 계산된 세 벡터  $x, y, z$ 는 지역좌표계를 생성한다. 두 개의 물체에 그려진 스트로크 위에 생성된 각각의 지역좌표계를 정렬하는 일차변환을 적용하면, 두 물체를 간단하게 붙일 수 있다.

두 번째 방법은 두 오브젝트에 단순히 방향 스트로크를 길게 그려 이를 통해 붙여질 방향 및 영역을 추정하는 방법이다 (그림 4 참조). 방향 스트로크만으로 지역 좌표계를 생성하는 방법은 다음과

같다. 먼저 방향 스트로크를 이루는 점을 일정한 길이비율이 되도록 다시 샘플링하고 무게 중심점과 평균 방향 벡터를 계산하여 이를  $x$ 라 한다. 가장 문제가 되는 부분은 스트로크의 법선 벡터를 구하는 것이다. 법선 벡터를 구하는 방법은 스트로크의 각 선분이 지나고 있는 오브젝트의 삼각형들의 ID를 저장해둔 뒤, 법선 벡터를 계산할 때 스트로크가 지나고 있는 삼각형들의 법선벡터의 평균을 이용하였다. 이 때 각 점들의 법선 벡터에는 점과 스트로크의 각 선분간의 거리에 반비례하는 가중치를 부여하여, 무게중심에 가까운 삼각형의 법선일 경우 더 많은 가중치를 부여하도록 하였다. 이렇게 구해진 법선 벡터  $z$ , 그리고  $x$ 와  $z$ 를 외적한 벡터  $y$  세 개의 값을 정규화하면 지역 좌표계를 얻을 수 있다. 앞의 방법과 마찬가지로 두 오브젝트의 지역 좌표계를 정렬하면 두 오브젝트를 쉽게 붙일 수 있다.

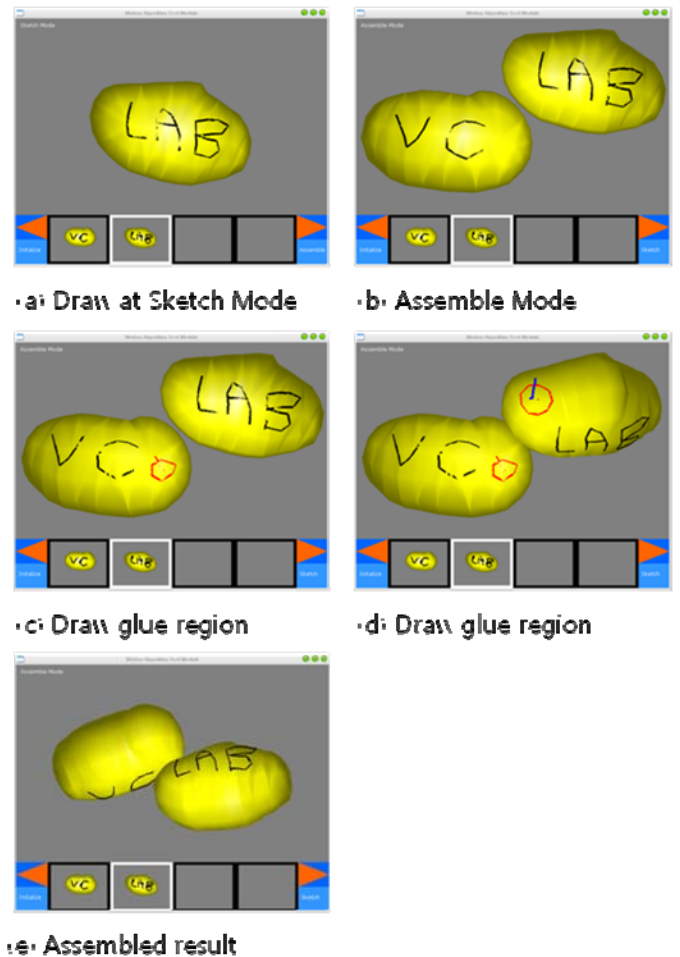


그림 3. 달린 스트로크를 이용한 조립 인터페이스. (a)그리기 모드에서 오브젝트를 생성. (b)조립 모드에서 두 오브젝트를 불러옴. (c)첫번째 오브젝트에 대하여 결합하고자 하는 부분과 오리엔테이션을 지정. (d) 두번째 오브젝트에 대하여 결합하고자 하는 부분과 오리엔테이션을 지정. (e) 결합 결과.

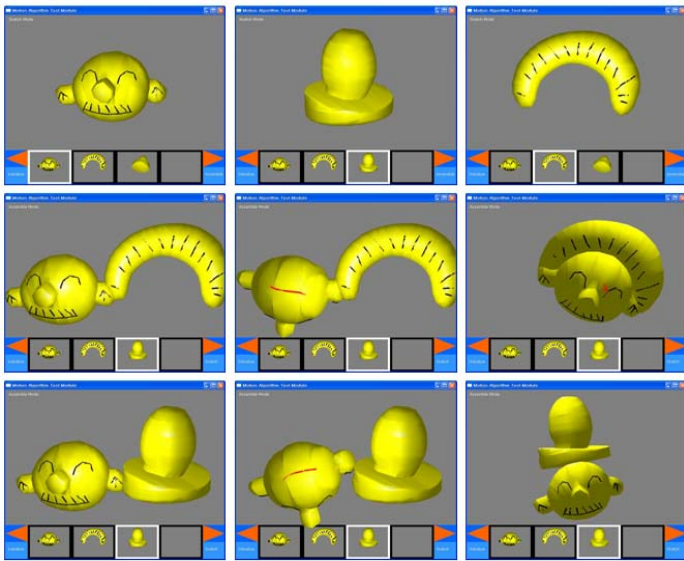


그림 4. 두 번째 조립 방법을 이용하여 두 물체를 결합하는 과정.

## 5. 결과

우리는 Intel Core2 2.13GHz, 2.0 GB 메모리, GeForce7600 그래픽 카드를 가진 컴퓨터를 이용하여 구현 및 실험을 하였다. 그림 3은 제안한 GluePaste의 첫 번째 인터페이스를 사용하여 두 오브젝트를 붙이는 과정을 보여주고 있다. 이 경우 평평한 두 개의 오브젝트를 붙일 때는 편리한 기능을 제공했으나, 울퉁불퉁한 영역 간의 붙이기를 시행할 경우 좋지 않은 결과를 얻을 가능성이 커지는 문제가 있었다. 또한 방향을 결정하는 스트로크를 잘못 그리게 되면 잘못된 결과를 얻기 쉽다는 단점이 있었다.

두 번째 GluePaste 방법의 경우 우선 첫 번째 방법에 비해 스트로크를 각 오브젝트당 하나씩만 그려주면 되므로 편리한 점이 있었다. 또한 스트로크를 어느 정도 이상으로 길게 그려주면 첫 번째 방법에 비해 방향 스트로크를 잘못 그리게 되는 일이 적어지기 되서 실수하는 정도가 많이 줄어드는 것을 확인할 수 있었다. 제안한 방법으로 계산된 지역 좌표계 정렬은 오브젝트를 다른 오브젝트의 원하는 위치에 정확하게 붙일 수 있었다. 그림 4는 두 번째 조립 방법을 통해 두 오브젝트를 결합한 예이다.

## 6. 결론

사용자로부터 단순한 몇 개의 스트로크를 입력 받아 3D 객체를 생성한다는 것은 고도의 정밀함을 요구하지 않는 범위 내에서 많은 편의를 제공한다. 본 연구에서는 Teddy 시스템[4]을 기본으로 한 스케치 기반의 메쉬 결합 인터페이스를 제안하였다. 제안하는 방법을 통해

사용자가 원하는 형태의 3D 객체를 생성하고 또 이들을 조립하는 데에 좋은 효용성을 보여주었다. 또한 일반적인 메쉬 조합 방법에 비해 직관적인 인터페이스를 제공하는 것이 본 연구에서 제안하는 인터페이스가 가지는 장점이라 할 수 있겠다.

시스템의 효용성을 입증하기 위해 한 가지 더 고려해야 할 것은 더 많은 일반 사용자들에게 우리의 시스템을 실험하여 설문 점수를 내는 것이다. 우리는 아직 제안하는 시스템을 일반 사용자들에게 공개적으로 평가하도록 요청하지 않았으나, 제안한 방법이 복잡한 오브젝트를 보다 간단한 오브젝트들의 조합으로 해결할 수 있다는 점에서 더욱더 효율적일 것이라고 믿어 의심치 않는다. 또한 시스템을 보다 강건하게 하기 위해서 메쉬 최적화 모듈을 붙여야 하는데, 그 이유는 표면이 고르지 못한 메쉬 오브젝트는 특히 표현의 스트로크를 이용하여 계산하여야 하는 결합을 위한 지역 좌표계 계산에 영향을 미칠 수 있기 때문이다.

우리가 제안한 시스템의 한계점은, 기본적으로 Teddy 시스템에서 사용된 스케치 인터페이스를 채용하였기 때문에 하나의 화면에서 한 개의 메쉬 모델만을 편집할 수 있다는 점이다. 이는 Teddy 시스템이 기본적으로 블롭(Blob) 메쉬를 생성할 때, 화면상에 생성된 메쉬가 없어야 하기 때문에 생겨나는 한계점이다. 물론 우리는 이러한 사항을 극복하기 위해서 멀티 페이지 기능을 도입하였으나, 멀티 페이지 보다는 하나의 페이지에서 여러 오브젝트를 동시에 편집할 수 있으며 조립까지 가능하도록 하는 편이 더 편할 것이다. 또 우리가 제안한 방법은 개념적으로 [8]에서 제안한 드래그 & 드롭 방식의 메쉬 결합 기법에 비해 더 많은 수의 입력을 요구한다. 우리가 제안한 방법은 두 개의 메쉬에 각각 한번씩의 스트로크를 요구하나, 드래그 & 드롭 방식은 한번의 드래그를 통해 메쉬를 붙일 수 있기 때문이다. 그러나 제안한 시스템은 단순히 두 메쉬의 결합 위치만을 결정하는 것이 아니라 결합 영역 및 방향 또한 결정이 가능하다는 장점을 가지고 있으므로, 결합 알고리즘 및 스케치 정보의 활용 부분을 개선하여 보다 나은 결과를 얻을 수 있을 것으로 기대된다.

향후 과제로는 현재 시스템이 가지는 몇 가지 결함들을 해결하고 조립 과정에서 기존에 연구된 바 있는 메쉬 연결 알고리즘을 적용하여 위상학적으로 연결된 메쉬 오브젝트로 편집하는 것이다. 또한 앞에서 언급한 바와 같이 리메싱(Remeshing) 알고리즘 등 메쉬 최적화 기법을 적용하여 보다 퀄리티 높은 결과물을 생성하는 것을 들 수 있다. 마지막으로 본 논문에서 제안한 방식보다 더 직관적이고 편리한 결합 인터페이스에 대한 연구가 이루어져야겠다.

### 감사의 글

본 연구는 문화체육관광부 및 정보통신연구진흥원의 IT원천기술 개발사업(2008-F-031-01)의 연구 결과로 수행되었음.

### 7. 참고문헌

[1] C. Yang, D. Sharon, M. Panne. Sketch-based modeling of parameterized objects. Eurographics 05, 2005.

[2] H. Wang, L. Markosian. Free-form Sketch. EUROGRAPHICS 07, 2007.

[3] J. Y. Oh, W. Stuerzlinger, J. Danahy. Comparing SESAME and sketching on paper for conceptual 3D design. EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling 05, 2005.

[4] T. Igarashi, S. Matsuoka, H. Hanaka. Teddy : A Sketching Interface for 3D Freeform Design. ACM

SIGGRAPH 99, 1999.

[5] J. J. Cherlin, F. Samavati, M. C. Sousa, J. A. Jorge. Sketch-based Modeling with Few Strokes. ACM 2005

[6] R. C. Zeleznik, K. P. Herndon, J. F. Hughes. SKETCH : An interface for sketching 3D scenes. SIGGRAPH 96 Conference Proceedings, pages 163-170, 1996.

[7] A. Shesh, B. Chen. SMARTPAPER : An interactive and user friendly sketching system. Eurographics 04, 2004.

[8] A. Sharf, M. Blumenkrants, A. Shamir, D. Cohen-Or. SnapPaste : an interactive technique for easy mesh composition. The visual computer, September 2006, pages 835-844, 2006