

형상 차이 기반 홀 패치의 파라메트릭 블렌딩 기법

박정호^{1O}

박상훈²

윤승현^{3*}

^{1,3}동국대학교 멀티미디어공학과

²동국대학교 영상대학원 멀티미디어학과

wjdgh283@hanmail.net

mshpark@dongguk.edu

shyun@dongguk.edu

Parametric Blending of Hole Patches Based on Shape Difference

Jung-Ho Park^{1O}

Sanghun Park²

Seung-Hyun Yoon^{3*}

^{1,3}Dept. of Multimedia Engineering, Dongguk University

²Dept. of Multimedia, Dongguk University

요약

본 논문에서는 삼각 메쉬의 홀을 채우는 새로운 기법을 제시한다. 첫째, 임의 모양의 홀을 검출한다. 둘째, 삼각화(triangulation), 세분화(refinement), 공정화(fairing), 스무딩(smoothing) 과정을 통해 소스 및 타겟 홀 패치를 생성한다. 마지막으로, 두 패치 사이의 형상 차이를 분석하고 패치간 블렌딩을 통해 특징이 강조된 홀 패치를 얻는다. 다양한 모양의 홀을 갖는 삼각 메쉬 모델에 홀 채움 기법을 적용하여 모델을 복원함으로써 제안된 기법의 효과성을 입증한다.

Abstract

In this paper, we propose a new technique for filling holes in triangular mesh. First, arbitrary shaped holes are detected. Second, source and target hole patches are generated through triangulation, refinement, fairing, and smoothing. Finally, the shape difference between the two patches is analyzed and a patch with enhanced features is obtained through blending between patches. The effectiveness of the proposed technique is demonstrated by applying the hole filling technique to the triangular mesh model with various shaped holes.

키워드: 메쉬 홀 채움, 메쉬 페어링, 형상 블렌딩

Keywords: Mesh Hole Filling, Mesh Fairing, Shape Blending

1. 서론

삼각 메쉬는 형상에 대해 일반적이고 유연하며 곡면 표현을 위한 쉽고 효율적인 수단으로써 컴퓨터 그래픽스, 기하 모델링에서 널리 사용된다. 실제 물체로부터 삼각 메쉬를 만들어 내는 것은 컴퓨터 그래픽스에서 중요한 과제이다. 일반적으로 스캔 장비를 이용하여 물체의 표면을 샘플링하여 포인트 클라우드를 얻는 방식이 많이 사용된다. 실제로는, 물체의 표면이 빛을 반사하는 성질, 오클루전(occlusions), 접근성의 한계 등과 같은 문제들 때문에 물체의 표면은 완전히 샘플링되지 않는다 [1]. 이렇게 얻어진 메쉬는 아티팩트(artifact) 또는 원치않는 홀을 가지고 있을 수 있다. 이런 불완전한 메쉬는 때로 어플리케이션에 직접적으로 사용될 수 없다.

불완전한 메쉬를 수리하거나 원치않는 홀을 채우는 것은 컴퓨터

그래픽스에서 곡면 복원을 하기 위한 중요한 문제이다. 일반적으로 메쉬 홀 채움의 목표는 주변 메쉬와 부드럽게 연결되며 원래 모델과 잘 어울리는 홀 패치를 생성하는 것이다. 하지만 홀의 복잡성 때문에 만족스러운 홀 패치를 만들어내는 것은 쉽지 않은 문제이다. 본 논문에서는 빠르고 효율적으로 동작하는 새로운 홀 채움 알고리즘을 제시한다. 알고리즘은 여러 단계를 거쳐 진행된다(Figure 1 참조). 첫째, 메쉬의 홀을 검출하고, 둘째, 삼각화, 세분화, 공정화 과정을 거쳐 타겟 패치를 생성하고, 타겟 패치를 스무딩하여 소스 패치를 생성한다(Section 3). 마지막으로, 패치간 형상 차이 기반의 블렌딩을 통해 최종 홀 패치를 합성한다(Section 4). 타겟 패치는 경계에서의 높은 연속성을 만족하며 주변 메쉬의 굴곡을 충분히 반영하고 있는 홀 패치이다. 이를 스무딩하여 생성한 소스 패치와 형상 차이를 분석하여 패

*corresponding author: Seung-Hyun Yoon/Dongguk University(shyun@dongguk.edu)

시간 블렌딩을 수행하면 사용자가 원하는 수준의 홀 패치를 얻을 수 있다. 제안된 삼각 메쉬 홀 채움 기법은 다음과 같은 장점을 가진다.

1. 홀의 크기가 크더라도, 삼각화 과정이 단순하기 때문에 알고리즘이 빠른 시간내로 수행된다(Figure 9 참조).
2. 공정화 과정으로 인하여 임의 모양의 홀에 대해서도 강건하게 작동한다(Figure 10(d) 참조).
3. 소스와 타겟 패치 사이의 형상 차이를 고려함으로써 주변 메쉬와 어울리고 특징이 더욱 향상된 홀 패치를 얻을 수 있다.

2. 관련연구

지금까지 다양한 메쉬 홀 채움 알고리즘이 개발되어 왔다. Barequet 등 [2]은 메쉬에 존재하는 홀 또는 갭(gap)을 채우기 위해 최소 면적 삼각화(minimum area triangulation)방법을 찾기 위한 동적 프로그래밍 방식을 사용했다. 이 방식은 다른 많은 메쉬 리페어(repair) 알고리즘의 기반이 되었다 [3]. 그러나 홀 경계에서의 부드러움은 고려하지 않으며 거친(rough) 홀 패치를 생성한다는 점에서 제한적이다. Pfeifle 등 [4]은 조각 다항식 곡면(piecewise polynomial surfaces)에 존재하는 홀을 채우기 위해 들로네(Delaunay) 삼각분할을 이용한 세분화와 공정화 기법을 사용하여 부드러운 홀 패치를 생성했다. Liepa 등 [5] 또한 홀을 채우기 위해 삼각화, 세분화를 사용했고 더 나아가 홀 패치의 삼각형의 밀도를 주변 메쉬와 비슷하게 했다. 홀 패치를 공정화시키기 위해 umbrella 연산자를 제안했지만 삼각화 과정에서 $O(n^3)$ 성능 때문에, 크기가 큰 홀의 경우에는 시간이 오래 걸리는 한계점이 존재한다. Chen 등 [6, 7]은 날카로운 부분에 의존적인 홀 채움 접근법을 제안했다. 그들은 방사형 기저 함수(radial basis function)에 기반한 보간법을 이용하여 초기 음함수 패치를 생성하고 만약 홀 경계에 날카로운 특징이 존재한다면 홀 패치에 베이즈 분류법(Bayesian classification) [8]과 날카로움 의존 필터(sharpness dependent filter) [9] 기반의 특징 향상 과정을 적용한다. 그러나 날카로움에 의존적인 필터에 사용되는 매개변수에 따라 홀 패치의 형상이 크게 달라진다 [10].

홀 패치를 주변 메쉬와 부드럽게 연결되도록 하는 기법 또한 연구되어 왔다. Zhao 등 [11]은 AFM 기법 [12]을 적용함으로써 곡률이 큰 홀을 채우는 초기 패치를 생성하고, 디리클레(Dirichlet) 경계 조건을 만족하는 포아송(Poisson) 방정식 [13]을 사용하여 홀 패치의 정점을 재배치했다. Pernot 등 [14]은 들로네 삼각분할을 사용하여 초기 홀 패치를 생성하고, 주변 메쉬와의 블렌딩 조건을 만족시키면서 패치의 정점을 재배치했다. 하지만 그들의 방식은 곡률이 큰 홀을 채우는 데에는 적합하지 않다. Branch 등 [1]과 Xiao 등 [15]은 경계 주변의 이웃 정점을 샘플링하고 방사형 기저 함수(RBF)를 사용하여 홀 패치를 생성했다. RBF 기반의 형상 복원은 간단하며 유용하지만 새로 생성되는 홀 패치가 경계의

컨벡스 헐(convex hull)을 크게 벗어나지 않는다는 점에서 제한적이다.

NURBS 곡면 또한 메쉬 모델의 홀을 채우기 위해 활용된다. Charles 등 [16]은 곡면위의 다각형 홀을 채우기 위해 C^1 3차 스플라인 패치를 사용했다. Li 등 [17]은 특징을 보존하는 홀 채움 알고리즘을 제시했다. 그들의 방식은 메쉬 홀 경계에서 불안정한 특징 곡선들을 찾고, 특징 곡선들을 잇기 위한 다항식 블렌딩 곡선을 생성한다. 마지막으로, 특징 곡선으로 나뉘는 두개의 홀을 채우기 위해 베지에-라그랑주 혼합 패치(Bezier-Lagrange hybrid patch)를 구성한다. 이 방식은 홀 경계에서의 높은 연속성을 보장한다. Wang 등 [10] 또한 특징을 보존하기 위한 방식에 초점을 맞추었다. 그들은 소실된 코너와 날카로운 에지를 복원하기 위한 특징 보존 홀 채움 프레임워크를 제시했다. Fortes 등 [18]은 곡면 복원과 병렬 또는 직교 곡선의 네크워크를 기반으로 한 홀 채움 구조를 제시했다.

3. 두 개의 패치 생성

3.1 홀 검출

오직 한 개의 인접한 삼각형을 갖는 에지를 경계 에지라고 부른다. 만약 정점에 연결된 에지들 중 경계 에지가 존재한다면 그 정점은 경계 정점이라고 부른다. 홀은 경계 에지들로 이루어진 닫힌 경로이다. 한 개의 정점에 연결된 모든 에지들을 그 정점의 1-링 에지라고 부른다. 정점의 1-링 에지들이 갖는 정점들(자기 자신 제외)을 그 정점의 1-링 정점이라고 부른다. 2-링 정점은 1-링 정점들 중 각 정점의 1-링 정점을 조사함으로써 구할 수 있다. 이와 유사하게, 경계의 N -링 정점 또한 구할 수 있다(Figure 2 참조). 알고리즘을 위해 본 논문에서는 경계의 1, 2-링 정점을 살펴본다. 원래 메쉬와 홀을 채우는 메쉬는 각각 주변 메쉬와 홀 패치라고 부른다. 본 논문에서는 다양체(manifold) 메쉬만을 다룬다. 또한 삼각 메쉬를 표현하기 위해 Kettner [19]가 제안한 하프에지(half edge) 자료구조를 활용한다. 삼각 메쉬에 대한 하프에지 자료구조는 메쉬의 기하학적 연결 정보를 실시간으로 빠르게 얻을 수 있다. 앞서 설명한 개념들을 이용하면 메쉬의 홀을 빠르고 쉽게 검출할 수 있다. 특히, Pernot [14] 등은 만족스럽지 않은 홀 패치가 생성될 것을 방지하기 위해 홀을 채우기 전에 홀 경계를 다듬었다. 하지만 본 논문에서 제시된 기법은 복잡한 홀을 다룰 수 있기 때문에 홀 삼각화를 위한 전처리 단계를 생략할 수 있다.

3.2 삼각화

홀 검출이 완료되면, 홀에 완전히 맞는 초기 홀 패치를 생성하기 위한 삼각화 과정을 시작한다. 본 논문에서는 AFM 기법 [12]의 간략화된 버전을 사용하는데, 이는 다음과 같이 몇 가지 단계로 구성된다.

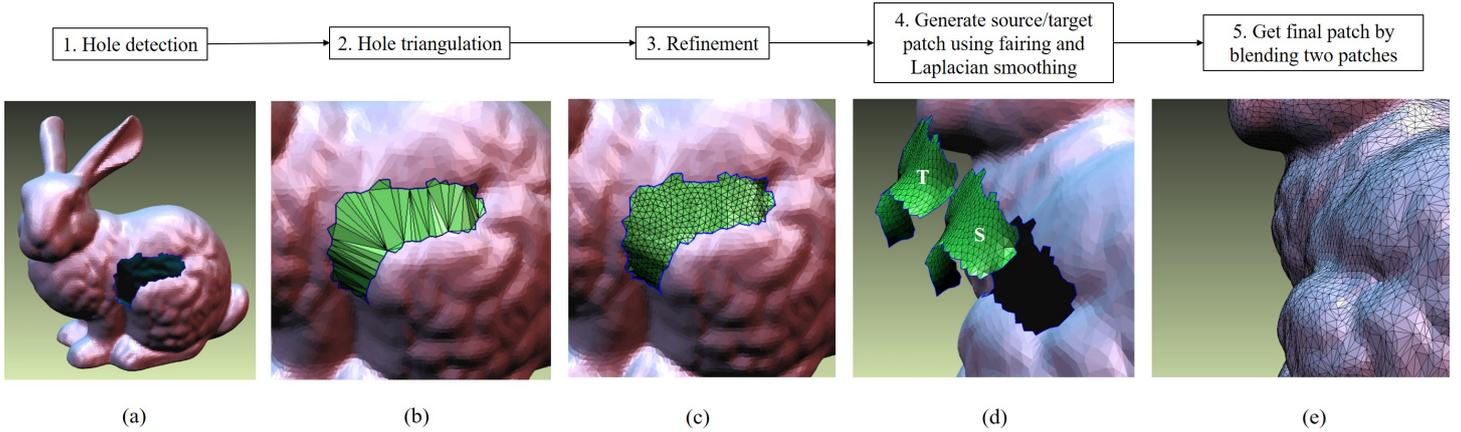


Figure 1: Pipeline of the algorithm and accompanying figures.

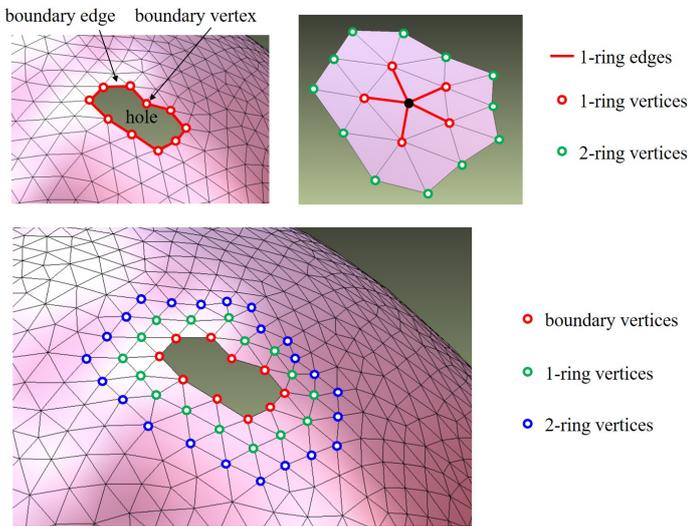


Figure 2: A hole and N -ring vertices.

1. 각 경계 정점에서 인접한 두 에지 사이의 가장 작은 각도를 찾는다.
2. 하나의 에지를 추가하여 두 개의 인접한 에지와 경계 정점을 포함하는 새로운 삼각형을 만든다.
3. 새로 생성되는 모든 삼각형으로 홀 영역을 채울 때까지 과정 1과 2를 반복한다.

이 방식을 사용함으로써 우리는 복잡하고 곡률이 큰 경계에 꼭 맞는 패치를 빠르고 쉽게 생성할 수 있다. 새롭게 생성되는 패치 안의 삼각형들은 거칠지만, 이어지는 세분화 단계에서 이러한 문제점은 자동적으로 해결된다. Figure 1(c)는 거친 삼각형들을 규칙적으로 만든 결과를 보여준다.

3.3 세분화(Refinement)

초기 패치 메쉬의 품질 향상을 위해 세분화 과정을 진행한다. 경계 에지를 제외한 내부 에지들은 매우 길며, 삼각형들은 비등방성

성질을 갖는다. 우리는 이러한 문제점을 해결하기 위해 점진적 리메싱 알고리즘을 채택한다 [3, 20, 21]. Algorithm 1은 목표 에지 길이를 입력으로 취하고, 에지 분할, 에지 제거, 에지 플립, 정점 이동을 순차적으로 수행한다.

목표 에지 길이 l 은 경계 에지의 평균 길이로 정하고, 세분화 후 패치와 주변 메쉬를 병합하기 위해 경계 에지를 분할 후보에서 제외한다. 에지 분할 단계에서는 임계 길이 $high$ 보다 긴 에지를 분할한다. 에지 제거 단계에서는 임계 길이 low 보다 짧은 에지를 제거하고, 만약 제거하여 임계 길이 $high$ 보다 긴 에지가 생성될 경우, 에지를 제거하지 않는다. 모든 에지 길이가 l 에 근접하고 삼각형 밀도가 주변 메쉬와 비슷하며 규칙적인 삼각형을 가진 패치를 얻을 때까지 알고리즘을 반복 수행한다. 품질이 향상된 홀 패치는 공정화 단계의 입력으로 사용된다. Figure 3은 알고리즘의 반복 수행 결과를 보여준다.

Algorithm 1 Incremental remeshing of the mesh.

Input: M : input mesh, l : target edge length
Output: M' : remeshed mesh.

- 1: $low = (\frac{4}{5}) * l$
- 2: $high = (\frac{4}{3}) * l$
- 3: SPLITEDGES($high$)
- 4: COLLAPSEEDGES(low , $high$)
- 5: FLIPEDGES()
- 6: RELAXATION()

3.4 공정화(Fairing)

세분화 후 홀 패치 정점의 위치를 재조정하고 부드럽게 만들기 위한 공정화(fairing) 과정을 시작한다. 공정화는 공정 에너지(fairness energy)를 최소화함으로써 곡면을 가능한 한 부드럽게 만드는 기술이다. 최소화하려는 공정 에너지 함수에 따라 최종적으로 원하는 곡면을 생성할 수 있다 [3]. 본 논문에서는 멤브레인(membrane) 에너지, 씰플레이트(thin-plate) 에너지, 곡률 변화율

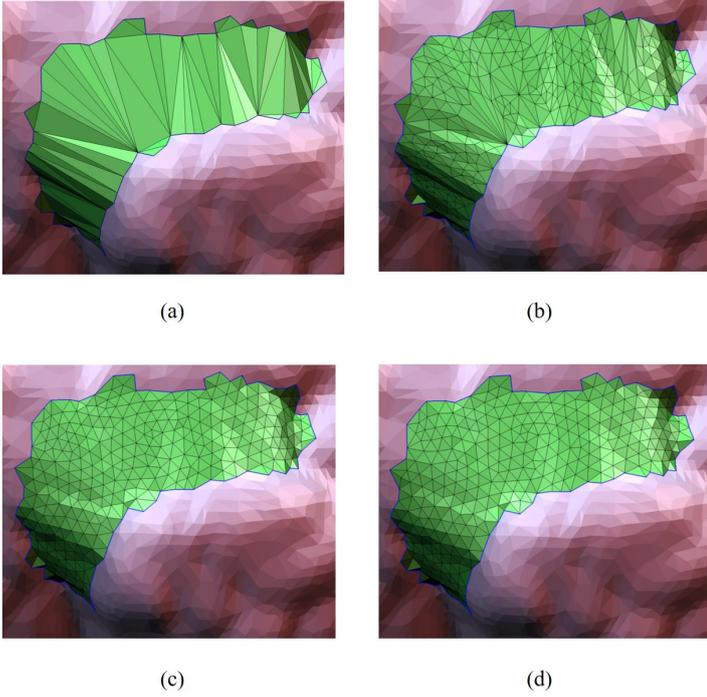


Figure 3: Process of incremental remeshing: (a) initial patch, (b) 1 iteration, (c) 3 iteration, (d) 5 iteration.

(curvature variation) 에너지 중에서 곡률 변화율 에너지에 초점을 맞춘다.

매개변수 곡면 $\mathbf{x}(u, v) : \Omega \rightarrow \mathcal{S}$ 의 멤브레인 에너지는 다음 식으로 표현될 수 있으며 디리클레(Dirichlet) 에너지라고도 불린다:

$$\tilde{E}_M(\mathbf{x}) = \iint_{\Omega} \|\mathbf{x}_u\|^2 + \|\mathbf{x}_v\|^2 dudv,$$

여기서 $\mathbf{x}_u = \partial\mathbf{x}/\partial u$ 이고 $\mathbf{x}_v = \partial\mathbf{x}/\partial v$ 이다. 곡면의 좌표 함수 $\mathbf{x}(u, v)$ 를 삼각 메쉬 정점의 좌표 $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T$ 로 대체하고, 이산 라플라스-벨트라미(Laplace-Beltrami) 연산자를 사용함으로써 위의 연속적인 영역의 공식을 삼각 메쉬로 이산화한다. 따라서 미지의 정점의 좌표 \mathbf{x} 에 대하여 다음 이산 라플라스 방정식을 풀 수 있다:

$$\Delta\mathbf{x} = \mathbf{0}, \mathbf{x} |_{\partial\Omega} = \mathbf{x}^* |_{\partial\Omega}$$

여기서 Δ 는 이산 라플라스-벨트라미 연산자이며 \mathbf{x}^* 는 경계 조건 $\partial\Omega$ 를 만족하는 알려진 정점의 좌표이다. Figure 4에서 삼각 메쉬 위의 한 정점 \mathbf{v}_i 를 공유한 n 개의 삼각형을 가정하자. 국소 영역 [22] A_i 를 갖는 \mathbf{v}_i 에 대한 이산 라플라스-벨트라미 연산자는 다음과 같이 정의된다:

$$\Delta f(\mathbf{v}_i) = \frac{1}{2A_i} \sum_{\mathbf{v}_j \in N_1(\mathbf{v}_i)} (\cot\alpha_{i,j} + \cot\beta_{i,j})(f_j - f_i)$$

f 는 정점의 좌표의 x, y, z 성분 중 하나가 될 수 있다.

만약, 곡면의 곡률을 최소화하는 것이 목적이라면 다음 싹플레

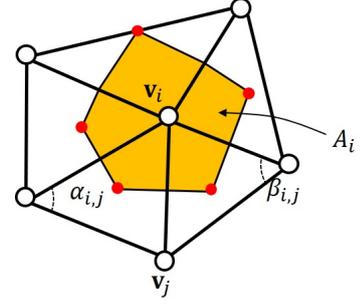


Figure 4: Local area and angles for the Laplace-Beltrami operator.

이트 에너지 함수를 사용한다:

$$\tilde{E}_{TP}(\mathbf{x}) = \iint_{\Omega} \|\mathbf{x}_{uu}\|^2 + 2\|\mathbf{x}_{uv}\|^2 + \|\mathbf{x}_{vv}\|^2 dudv.$$

$\tilde{E}_{TP}(\mathbf{x})$ 를 최소화 하기 위한 이산 라플라스 방정식은 $\Delta^2\mathbf{x} = \mathbf{0}$, $\mathbf{x} |_{\partial\Omega} = \mathbf{x}^* |_{\partial\Omega}$ 이다. 또한, 곡률의 변화율이 최소화된 곡면을 얻기 위해 Moreton 등 [23]이 제시한 곡률 변화율 에너지를 사용할 수 있다:

$$\tilde{E}_{CV}(\mathbf{x}) = \iint_{\Omega} \left(\frac{\partial\kappa_1}{\partial\mathbf{t}_1} \right)^2 + \left(\frac{\partial\kappa_2}{\partial\mathbf{t}_2} \right)^2 dudv,$$

여기서 κ_1, κ_2 는 주곡률이고 $\mathbf{t}_1, \mathbf{t}_2$ 는 그에 대응되는 주방향이다. $\tilde{E}_{CV}(\mathbf{x})$ 를 최소화 하기 위한 이산 라플라스 방정식은 $\Delta^3\mathbf{x} = \mathbf{0}$, $\mathbf{x} |_{\partial\Omega} = \mathbf{x}^* |_{\partial\Omega}$ 이다.

Figure 5은 홀 패치 메쉬의 공정화 결과를 보여준다. Figure 5(a)에서는 홀 패치의 경계 정점들이 고정되어 C^0 경계 조건이 만족되고, Figure 5(b)에서는 경계 정점과 경계의 1-링 정점이 고정되어 C^1 경계 조건이 만족되고, Figure 5(c)에서는 경계 정점과 경계의 1, 2-링 정점이 고정되어 C^2 경계 조건이 만족된다. 우리는 이 중에서 주변 메쉬의 형상을 가장 잘 반영하는 곡률의 변화율이 최소화된 곡면(Figure 5(c))을 블렌딩을 위한 타겟 패치로 사용한다.

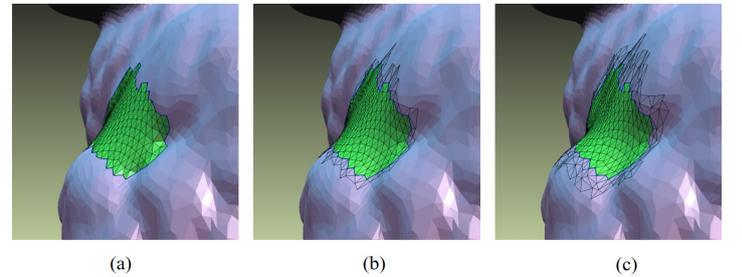


Figure 5: (a) A hole patch by minimizing membrane energy($\Delta\mathbf{x} = \mathbf{0}$), (b) A hole patch by minimizing thin-plate energy($\Delta^2\mathbf{x} = \mathbf{0}$), (c) A hole patch by minimizing curvature variation energy($\Delta^3\mathbf{x} = \mathbf{0}$).

3.5 라플라시안 스무딩

타겟 패치에 라플라시안 스무딩을 적용하여 블렌딩을 위한 소스 패치를 생성한다. 타겟 패치 각각의 정점 \mathbf{v}_i 에 대하여 다음 umbrella 연산자를 사용하여 정점의 위치 \mathbf{v}'_i 을 계산한다:

$$\mathbf{v}'_i = -\mathbf{v}_i + \frac{1}{W(\mathbf{v}_i)} \sum_{\mathbf{v}_j \in N_1(\mathbf{v}_i)} w(\mathbf{v}_i, \mathbf{v}_j) \mathbf{v}_j,$$

여기서, $W(\mathbf{v}_i) = \sum_{\mathbf{v}_j \in N_1(\mathbf{v}_i)} w(\mathbf{v}_i, \mathbf{v}_j)$ 이다. 본 논문에서는 타겟 패치의 삼각형의 모양을 유지시키기 위해 다음과 같은 cotangent 가중치 [24]를 사용하였다(Figure 4 참조):

$$w(\mathbf{v}_i, \mathbf{v}_j) = \cot \alpha_{i,j} + \cot \beta_{i,j}$$

타겟 패치의 삼각형 모양을 유지시켜주는 라플라시안 스무딩을 적용하여 소스 패치를 생성함으로써 패치간 자연스러운 블렌딩 방향을 설정할 수 있다.

4. 홀 패치 블렌딩

메쉬 공정화를 통해 생성한 타겟 패치는 형상이 특징 표현이 두드러지지 못한 한계점이 존재한다. 때때로 주변 메쉬의 굴곡이 심한 경우에는 원하지 않는 형상의 홀 패치가 생성될 수 있다. 이러한 한계점을 극복하기 위해 본 절에서는 소스-타겟 패치간 형상 차이에 기반한 블렌딩을 제안한다.

타겟 패치에서 굴곡과 디테일이 충분히 제거된 소스 패치와 곡률의 변화율을 최소화한 타겟 패치 사이의 곡률 및 변위 즉, 형상 차이를 분석한다. 이러한 형상 차이에 기반한 블렌딩을 수행함으로써 굴곡이 더욱 강조되고 주변 메쉬와 어울리는 형태의 홀 패치를 사용자가 원하는 수준으로 얻을 수 있다.

4.1 형상 차이 측정

소스와 타겟 패치 메쉬를 각각 S 와 T 라고 하자. S 와 T 의 형상 차이를 계산하기 위하여 각각의 메쉬에 대해 이산 곡률 분석을 할 필요가 있다. 메쉬 위의 한 정점 \mathbf{v}_i 의 좌표가 \mathbf{x}_i 일 때 \mathbf{v}_i 에서의 평균 곡률의 절대값 H_i 은 다음과 같이 계산될 수 있다:

$$H_i = \frac{1}{2} \|\Delta \mathbf{x}_i\|,$$

여기서 Δ 는 이산 라플라스-벨트라미 연산자이다. 그리고 S 와 T 는 동일한 정점의 수와 연결 정보를 가지며 정점간의 일대일 대응 관계가 성립한다. 대응되는 정점 \mathbf{v}_i 와 \mathbf{v}'_i 의 형상 차이는 다음과 같이 계산될 수 있다:

$$\sigma_i = \sigma(\mathbf{v}_i, \mathbf{v}'_i) = (1 - \alpha) \frac{\Delta \kappa_i - \Delta \kappa^{min}}{\Delta \kappa^{max} - \Delta \kappa^{min}} + \alpha \frac{d_i - d^{min}}{d^{max} - d^{min}},$$

여기서 $\Delta \kappa_i$ 는 두 정점의 곡률 차이($\|H_i^S - H_i^T\|$)이고, d_i 는 정점간 변위($\|\mathbf{p}_i^S - \mathbf{p}_i^T\|$)이다. $0 \leq \alpha \leq 1$ 는 사용자 정의 파라미터이다. 만약 $\alpha = 0$ 이라면 형상 차이는 오직 곡률의 차이에만 비례한다. 반면 $\alpha = 1$ 이라면 형상 차이는 오직 변위에만 비례한다. 사용자는 α 를 조절함으로써 패치간의 형상 차이를 계산할 수 있다. Figure 6는 홀에서 생성된 소스-타겟 패치와 그 둘의 형상 차이를 컬러맵을 이용하여 가시화한 결과를 보여준다.

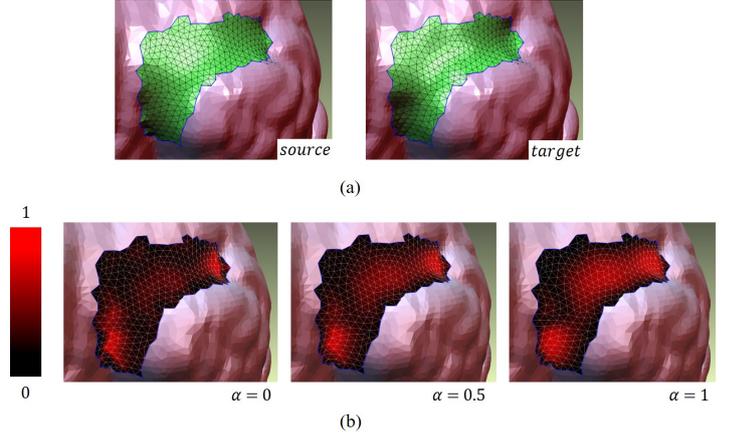


Figure 6: (a) Target patch T (right) by fairing, source patch S by applying Laplacian smoothing to target patch(left), (b) Color-coded shape difference between S and T

4.2 형상 차이 기반 블렌딩 함수

소스-타겟 패치는 형상은 다르지만 모든 정점에 대해 일대일 대응 관계가 성립하기 때문에 형상 블렌딩이 가능하다. 제어 파라미터 w 에 따라 소스 패치 S 와 타겟 패치 T 사이에 다음 선형 블렌딩을 적용하면 새로운 패치 P 를 합성해낼 수 있다:

$$P(w) = (1 - w)S + wT,$$

제어 파라미터 w 가 $0 \leq w \leq 1$ 인 경우, P 는 S 와 T 의 보간법(interpolation) 의해 합성된다. 반면 $w < 0$ 또는 $w > 1$ 인 경우, P 는 S 와 T 의 보외법(extrapolation)에 의해 합성된다. 본 논문에서는 제어 파라미터 w 를 정점별로 재매개화하여 변형 파라미터 \hat{w}_i 를 정의하였다:

$$\hat{w}_i = w \times \left(1 + \frac{1}{1 + e^{-10\sigma_i + 5}} \right).$$

적절한 \hat{w}_i 를 설정하기 위해 각각의 정점 \mathbf{v}_i 에 대하여 패치 사이의 형상 차이 σ_i 를 계산한다. 만약 형상 차이 σ_i 가 크다면, 이는 정점 \mathbf{v}_i 주변에서 패치 메쉬의 형상의 변화가 크다는 것을 유추할 수 있다. 형상 차이가 큰 정점 \mathbf{v}_i 는 상대적으로 홀 패치의 중요한 특징을 표현한다고 해석할 수 있다. 따라서 이러한 정점들은 변형 파라미터를 더 크게하여 블렌딩 과정에서 더욱 강조되도록 한다.

만약 한 정점에서 $\sigma_i = 0$, 즉 형상 차이가 거의 없다면 변형 파라미터 \hat{w}_i 는 w 와 같다. 반면 $\sigma_i = 1$, 즉 형상 차이가 큰 정점에서는 $\hat{w}_i = 2w$ 이므로 소스-타겟간 형상 차이가 강조된다(Figure 7참조).

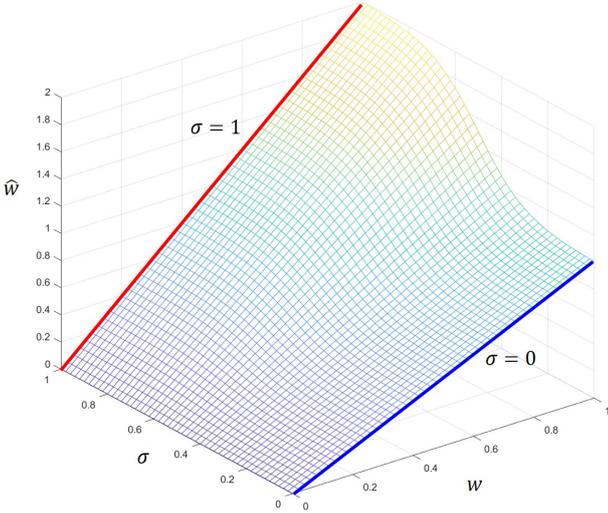


Figure 7: 3D graph of reparameterization by shape difference σ .

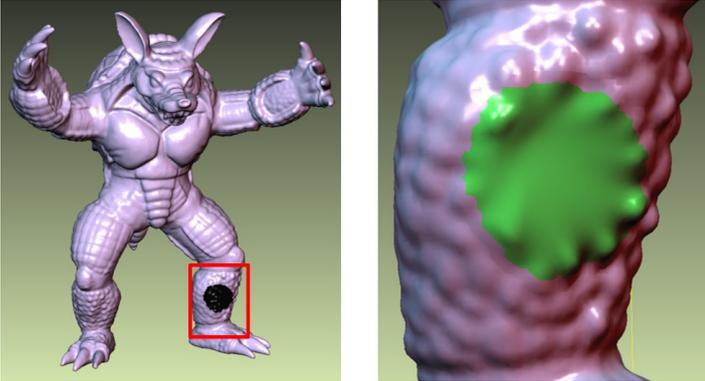


Figure 8: Difficulty to reconstruct a detail without enough information.

5. 실험결과

본 논문에서 제안한 형상 차이 기반 홀 채움 기법은 C++언어를 사용하여 Intel i7-9700F CPU, 16Gb의 메인메모리와 NVIDIA RTX 2060 그래픽 카드가 설치된 PC에서 구현되었다. 제안된 기법의 효과성을 실험하기 위해 다양한 홀을 가진 모델을 사용하였다. Figure 10은 각각의 모델에 대해서 최소 곡률 변화율 방식의 홀 채움 결과와 형상 차이 기반 블렌딩 방식의 홀 채움 결과를 비교하여 보여준다. 본 논문에서 제시한 기법은 특히 주변 메쉬에 굴곡 및 특징이 많이 존재하는 경우 이를 효과적으로 복원 가능하다. Figure 10(a)에서는 $\alpha = 0.5$ 로 설정하여 소스-타겟 패치간의 곡률과 변위를 절반씩 고려하여 굴곡을 과장시켰다. Figure 10(b)

에서는 토끼 모델의 귀 부근에 생성되는 소스-타겟 패치 사이의 변위를 크게 고려하였다. 따라서 변위가 큰 정점들은 블렌딩 과정 중 큰 폭으로 이동하여 부족한 귀의 형상이 복원되었다. Figure 10(c)처럼 움푹 패인 형상에 홀이 있는 경우 곡률과 변위를 충분히 고려하여 이를 복원하였다. Figure 10(d)에서는 생성된 소스-타겟 패치간의 곡률차가 심한 특성을 이용하여 홀 패치에 존재하는 돌기를 과장시키는 결과를 얻었다. Table 1은 홀의 크기를 늘려가면서 홀 패치를 생성하는 시간을 측정한 결과이고, Figure 9은 Table 1의 결과를 가시화한 결과이다. 실험 결과에서 보듯이 메쉬의 홀을 이루는 정점의 개수에 따라 홀 패치를 생성하는 시간은 선형적으로 증가하는 것을 볼 수 있다.

Table 1: Computational time for creating hole patch

Hole size (# vertices of hole)	Time (ms)
41	75
81	102
127	195
166	307
206	464
296	625

6. 결론

본 논문에서는 형상 차이 기반 블렌딩을 이용하여 삼각 메쉬의 홀을 채우는 기법을 제안하였다. 메쉬 홀을 채우는 소스 및 타겟 패치를 생성하여 그 둘의 형상 차이를 분석하였으며, 이를 기반한 형상 블렌딩을 통해 주변 메쉬와 어울리는 홀 패치를 사용자가 결정할 수 있도록 하였다. 크고 다양한 홀을 가진 메쉬 모델에 대하여 제안된 기법을 적용한 결과, 기존 방식으로는 복원될 수 없었던 주변의 굴곡이나 특징이 만족할 만한 수준으로 복원되는 결과를 얻었다.

제시된 방법은 공정화를 통해 생성한 패치를 블렌딩 타겟으로 선정했기 때문에 때로는 정점의 블렌딩 방향이 자연스럽지 못할 때가 있다. 또한 타겟 패치 생성 과정에서 주변 메쉬를 최대 2-링까지만 고려하였기 때문에 주변의 더 많은 특징을 고려하지 못하는 한계점이 존재한다(Figure 8참조). 향후 연구에서는 생성되는 홀 패치가 더 넓은 주변 영역의 디테일을 충분히 반영하도록 홀 채움 기법을 확장할 계획이다.

감사의 글

이 성과는 2020년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구 (No. NRF-2018R1D1A1B07048036)와 과학기술정보통신부 방송통신발전기금으로 한국전파진흥협회의 지원을 받아 수행된 연구임

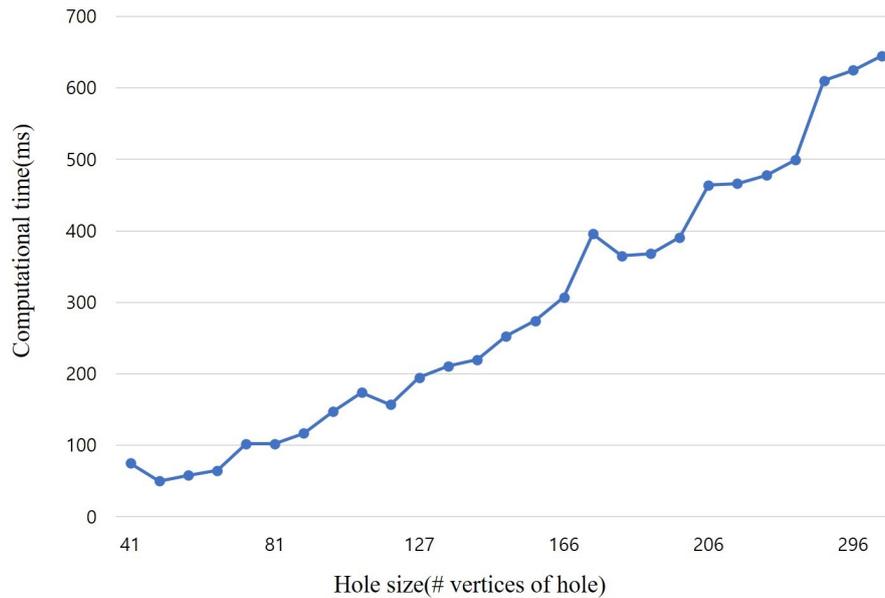
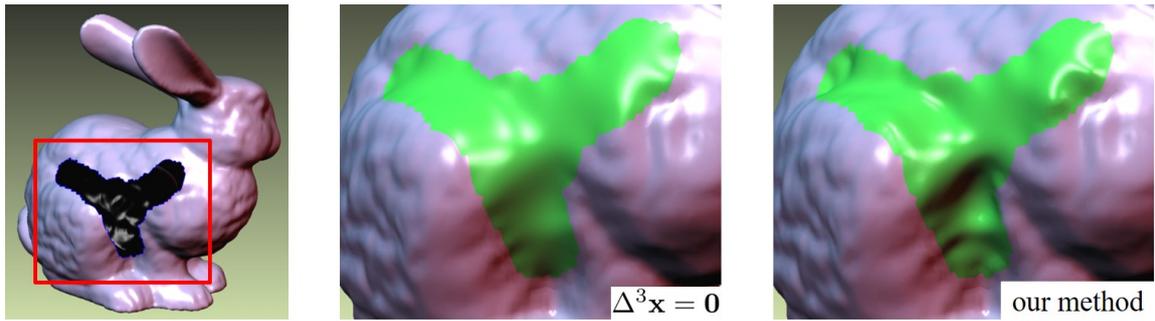


Figure 9: Computational time of creating hole patch for different size of hole.

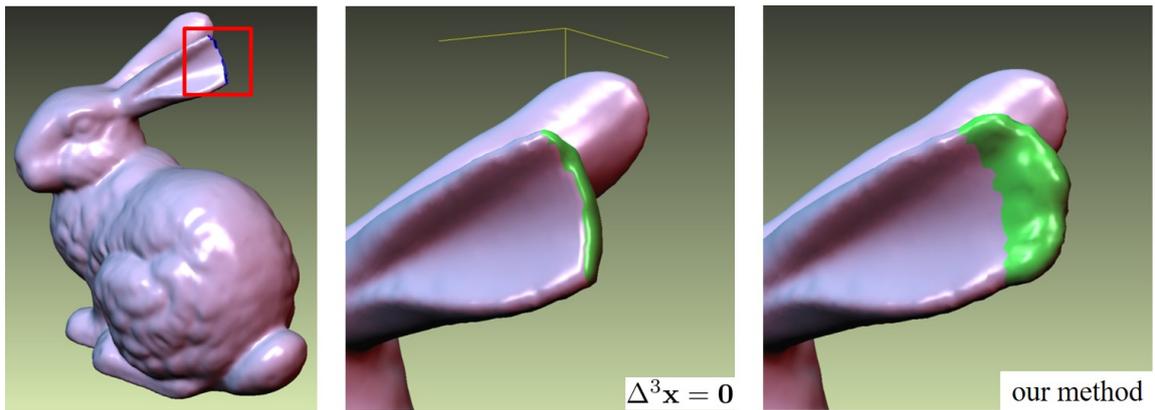
References

- [1] J. Branch, F. Prieto, and P. Boulanger, “A hole-filling algorithm for triangular meshes using local radial basis function,” in *Proceedings of the 15th International Meshing Roundtable*. Springer, 2006, pp. 411–431.
- [2] G. Barequet and M. Sharir, “Filling gaps in the boundary of a polyhedron,” *Computer Aided Geometric Design*, vol. 12, no. 2, pp. 207–229, 1995.
- [3] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy, *Polygon mesh processing*. CRC press, 2010.
- [4] R. Pfeifle and H.-P. Seidel, “Triangular B-splines for blending & filling of polygonal holes,” in *Graphics Interface*, 1996, pp. 186–193.
- [5] P. Liepa, “Filling holes in meshes,” in *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 2003, pp. 200–205.
- [6] C.-Y. Chen, K.-Y. Cheng, and H. Liao, “A sharpness dependent approach to 3d polygon mesh hole filling,” 2005.
- [7] C.-Y. Chen and K.-Y. Cheng, “A sharpness-dependent filter for recovering sharp features in repaired 3D mesh models,” *IEEE Transactions on visualization and computer graphics*, vol. 14, no. 1, pp. 200–212, 2007.
- [8] C.-Y. Chen, K.-Y. Cheng, and H.-Y. M. Liao, “Fairing of polygon meshes via bayesian discriminant analysis,” 2004.
- [9] C.-Y. Chen and K.-Y. Cheng, “A sharpness dependent filter for mesh smoothing,” *Computer Aided Geometric Design*, vol. 22, no. 5, pp. 376–391, 2005.
- [10] X. Wang, X. Liu, L. Lu, B. Li, J. Cao, B. Yin, and X. Shi, “Automatic hole-filling of CAD models with feature-preserving,” *Computers & Graphics*, vol. 36, no. 2, pp. 101–110, 2012.
- [11] W. Zhao, S. Gao, and H. Lin, “A robust hole-filling algorithm for triangular mesh,” *The Visual Computer*, vol. 23, no. 12, pp. 987–997, 2007.
- [12] P. L. George and É. Seveno, “The advancing-front mesh generation method revisited,” *International Journal for Numerical Methods in Engineering*, vol. 37, no. 21, pp. 3605–3619, 1994.
- [13] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H.-Y. Shum, “Mesh editing with poisson-based gradient field manipulation,” in *ACM SIGGRAPH 2004 Papers*, 2004, pp. 644–651.
- [14] J.-P. Pernot, G. Moraru, and P. Véron, “Filling holes in meshes using a mechanical model to simulate the curvature variation minimization,” *Computers & Graphics*, vol. 30, no. 6, pp. 892–902, 2006.
- [15] X. J. Wu, M. Y. Wang, and B. Han, “An automatic hole-filling algorithm for polygon meshes,” *Computer-Aided Design and Applications*, vol. 5, no. 6, pp. 889–899, 2008.

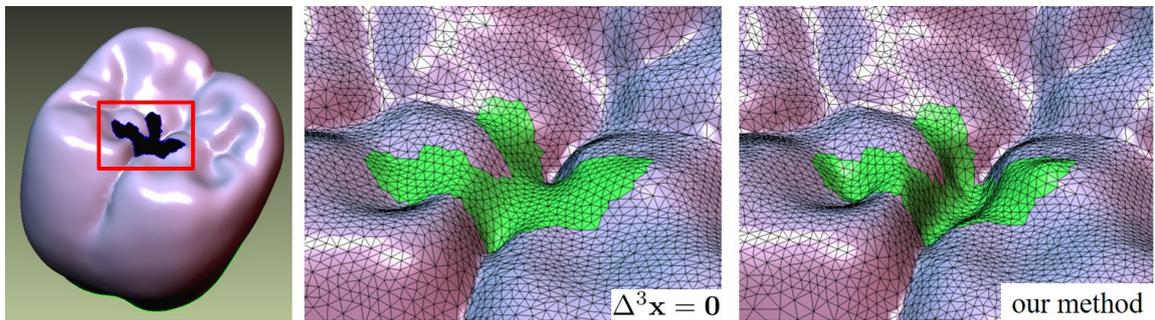
- [16] C. K. Chui and M.-J. Lai, "Filling polygonal holes using C^1 cubic triangular spline patches," *Computer Aided Geometric Design*, vol. 17, no. 4, pp. 297–307, 2000.
- [17] Z. Li, D. S. Meek, and D. J. Walton, "Polynomial blending in a mesh hole-filling application," *Computer-Aided Design*, vol. 42, no. 4, pp. 340–349, 2010.
- [18] M. Fortes, P. González, A. Palomares, and M. Rodríguez, "Filling holes using a mesh of filled curves," *Mathematics and Computers in Simulation*, vol. 164, pp. 78–93, 2019.
- [19] L. Kettner, "Using generic programming for designing a data structure for polyhedral surfaces," *Computational Geometry*, vol. 13, no. 1, pp. 65–90, 1999.
- [20] J. Vorsatz, C. Rossel, and H.-P. Seidel, "Dynamic remeshing and applications," *J. Comput. Inf. Sci. Eng.*, vol. 3, no. 4, pp. 338–344, 2003.
- [21] M. Botsch and L. Kobbelt, "A remeshing approach to multiresolution modeling," in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 2004, pp. 185–192.
- [22] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," in *Visualization and mathematics III*. Springer, 2003, pp. 35–57.
- [23] H. P. Moreton and C. H. Séquin, "Functional optimization for fair surface design," *ACM SIGGRAPH Computer Graphics*, vol. 26, no. 2, pp. 167–176, 1992.
- [24] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999, pp. 317–324.
- [25] L. P. Kobbelt, T. Bareuther, and H.-P. Seidel, "Multiresolution shape deformations for meshes with dynamic vertex connectivity," in *Computer Graphics Forum*, vol. 19, no. 3. Wiley Online Library, 2000, pp. 249–260.
- [26] M. Botsch and L. Kobbelt, "An intuitive framework for real-time freeform modeling," *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 630–634, 2004.
- [27] M. Wei, J. Wu, and M. Pang, "An integrated approach to filling holes in meshes," in *2010 International Conference on Artificial Intelligence and Computational Intelligence*, vol. 3. IEEE, 2010, pp. 306–310.
- [28] M. Attene, "A lightweight approach to repairing digitized polygon meshes," *The visual computer*, vol. 26, no. 11, pp. 1393–1406, 2010.



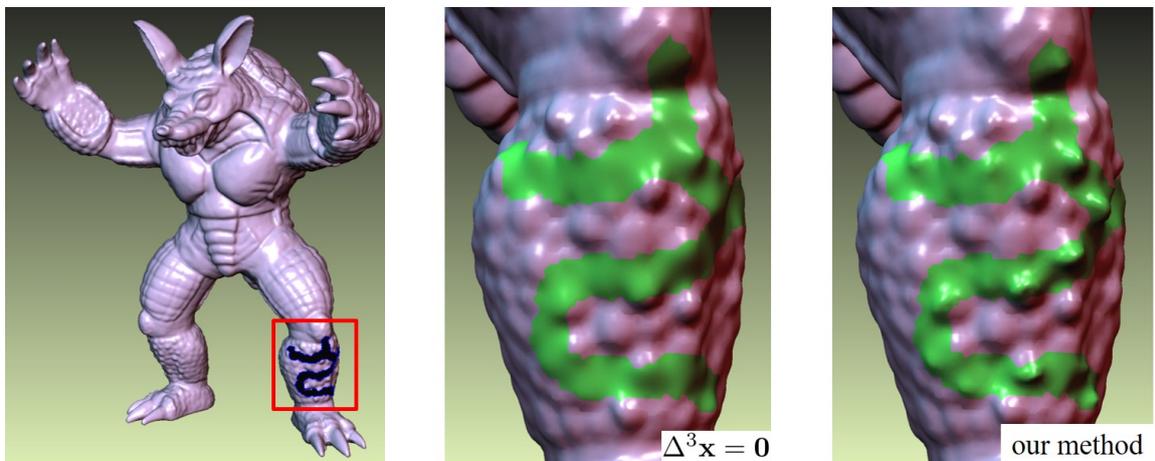
(a)



(b)



(c)



(d)

Figure 10: Four examples with complex holes. (a): left, a bunny model; middle: minimizing curvature variation($\Delta^3\mathbf{x} = \mathbf{0}$), right: shape difference-based blending(our method), $\alpha = 0.5, w = 1.2$. (b): left, a bunny model; middle: $\Delta^3\mathbf{x} = \mathbf{0}$, right: our method, $\alpha = 0.9, w = 3.3$. (c): left, a molar model; middle: $\Delta^3\mathbf{x} = \mathbf{0}$, right: our method, $\alpha = 0.5, w = 1.55$. (d): left, a armadillo model; middle: $\Delta^3\mathbf{x} = \mathbf{0}$, right: our method, $\alpha = 0.2, w = 1.4$.

〈 저자 소개 〉



박정호

- 2019 동국대학교 멀티미디어공학과 학사
- 2019 ~ 현재 동국대학교 멀티미디어공학전공 석사과정
- 관심분야 : 기하 모델링, 가상현실, 컴퓨터 그래픽스
- <https://orcid.org/0000-0002-6708-6692>



박상훈

- 1993 서강대학교 수학과 학사
- 1995 서강대학교 컴퓨터학과 석사
- 2000 서강대학교 컴퓨터학과 박사
- 2002 ~ 2005 대구가톨릭대학교 컴퓨터정보통신공학부조교수
- 2001 University of California, Davis 방문 연구원
- 2005 ~ 현재 동국대학교 멀티미디어학과 교수
- 관심분야: 실시간 렌더링, 사실적 렌더링, 과학적 가시화, 고성능 컴퓨팅 등
- <https://orcid.org/0000-0001-5383-7005>



윤승현

- 2001 한양대학교 수학과 학사
- 2007 서울대학교 컴퓨터공학과 박사
- 2007 ~ 현재 동국대학교 멀티미디어공학과 조교수/부교수/교수
- 관심분야 : 컴퓨터그래픽스, 기하모델링, 가상/증강/혼합현실
- <https://orcid.org/0000-0002-0015-8305>