

딥러닝과 구체의 형태 변형 방법을 이용한 단일 이미지에서의 3D Mesh 재구축 기법

3D Mesh Reconstruction Technique from Single Image using Deep Learning and Sphere Shape Transformation Method

김 정 윤*, 이 승 호**

Jeong-Yoon Kim*, Seung-Ho Lee**

Abstract

In this paper, we propose a 3D mesh reconstruction method from a single image using deep learning and a sphere shape transformation method. The proposed method has the following originality that is different from the existing method. First, the position of the vertex of the sphere is modified to be very similar to the 3D point cloud of an object through a deep learning network, unlike the existing method of building edges or faces by connecting nearby points. Because 3D point cloud is used, less memory is required and faster operation is possible because only addition operation is performed between offset value at the vertices of the sphere. Second, the 3D mesh is reconstructed by covering the surface information of the sphere on the modified vertices. Even when the distance between the points of the 3D point cloud created by correcting the position of the vertices of the sphere is not constant, it already has the face information of the sphere called face information of the sphere, which indicates whether the points are connected or not, thereby preventing simplification or loss of expression. can do. In order to evaluate the objective reliability of the proposed method, the experiment was conducted in the same way as in the comparative papers using the ShapeNet dataset, which is an open standard dataset. As a result, the IoU value of the method proposed in this paper was 0.581, and the chamfer distance value was It was calculated as 0.212. The higher the IoU value and the lower the chamfer distance value, the better the results. Therefore, the efficiency of the 3D mesh reconstruction was demonstrated compared to the methods published in other papers.

요 약

본 논문에서는 딥러닝과 구체의 형태 변형 방법을 이용한 단일 이미지에서의 3D mesh 재구축 기법을 제안한다. 제안한 기법은 기존의 방식과 다른 다음과 같은 독창성이 있다. 첫 번째, 기존의 근처의 가까운 점들을 연결하여 모서리 또는 면을 구축하는 방식과 다르게 딥러닝 네트워크를 통하여 구체의 꼭짓점의 위치를 사물의 3D 포인트 클라우드와 매우 유사하게 수정한다. 3D 포인트 클라우드를 이용하므로 메모리가 적게 필요하며 구체의 꼭짓점에 오프셋 값 사이에 덧셈 연산만을 수행하기 때문에 더 빠른 연산이 가능하다. 두 번째, 수정한 꼭짓점에 구체의 면 정보를 씌워 3D mesh를 재구축한다. 구체의 꼭짓점의 위치를 수정하여 생성한 3D 포인트 클라우드의 점들의 간격이 일정하지 않을 때에도 이미 점들 사이의 연결 여부를 나타내는 구체의 면 정보라는 3D mesh의 면 정보를 가지고 있어 표현의 단순화나 결손을 방지할 수 있다. 제안하는 기법의 객관적인 신뢰성을 평가하기 위해 공개된 표준 데이터셋인 ShapeNet 데이터셋을 이용하여 비교 논문들과 같은 방법으로 실험한 결과, 본 논문에서 제안하는 기법의 IoU 값이 0.581로, chamfer distance 값은 0.212로 산출되었다. IoU 값은 수치가 높을수록, chamfer distance 값은 수치가 낮을수록 우수한 결과를 나타내므로 다른 논문에서 발표한 기법들보다 3D mesh 재구축의 결과에서 성능의 효율성이 입증되었다.

Key words : 3D Mesh, Reconstruction, Reparameterization Trick, Latent Vector, Deep Learning

* Dept. Electronic Engineering, Hanbat National University

★ Corresponding author

E-mail : shlee@cad.hanbat.ac.kr, Tel : +82-42-821-1137

Manuscript received Mar. 30, 2022, revised May. 29, 2022, accepted May. 31, 2022.

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

I. 서론

최근 가상현실(Virtual Reality), 증강현실[1](Augmented Reality) 등의 분야의 시장 규모가 급격히 성장하고 있으며, 두 분야에서 많이 사용되고 있는 3D map, 3D 모델 시장의 규모도 급격히 성장하고 있다. 또한, 4차 산업혁명으로 인하여 3D 프린트 시장이 성장함에 따라 3D 모델 데이터에 대한 중요도가 커지고 있다. 기존의 3D map, 3D 모델을 획득하는 방식은 대표적으로 적외선이 무언가에 반사되어 되돌아오는 시간을 측정하여 사물과의 거리를 재는 LiDAR[2]를 사용하고 있다. 그러나 이 방식은 태양광에 섞여 들어오는 적외선에 영향을 심하게 받아 주로 실내에서만 사용되며 사물의 색에 따른 반사율의 차이로 인하여 거리 오차가 발생하는 문제가 있다. 또 다른 방식으로는 여러 장의 2D 이미지를 촬영하고 촬영 시의 카메라 좌표를 정밀하게 입력하여 3D로 재구축하는 방식이 있다. 그러나 이 방식은 카메라의 위치를 정밀하게 위치시키기 위해 별도의 카메라 부스나 레일이 필요하며 부스와 레일의 사이즈를 벗어나는 물체에 대해서는 적용하지 못한다는 문제가 있다. 기존의 두 방식의 공통점은 여러 개의 점으로 이루어진 3D 모델인 3D 포인트 클라우드를 얻는 방법으로 면이 없는 불완전한 3D 모델이며 바로 응용하기에는 부적합하기 때문에 별도의 표면 생성 알고리즘을 통해 면을 생성하는 후처리가 필요하다라는 문제 또한 내포하고 있다.

따라서 본 논문에서는 별도의 측정 장비 없이 2D RGB 이미지 1장만을 이용하여 사물의 윤곽을 3D 포인트 클라우드로 나타내고 구체의 형태를 변형시켜 완전한 3D mesh 모델을 재구축하는 기법을 제안한다.

II. 본론

1. 전체 개요도

본 논문에서 제안하는 딥러닝과 구체의 형태 변형 방법을 이용한 단일 이미지에서의 3D mesh 재구축 기법의 전체 개요도는 그림 1과 같다.

본 논문에서 제안된 기법의 학습 과정은 ResNet-50을 사용한 feature map 추출, Reparameterization trick을 사용한 latent vector 변환, FCN을 사용한 오프셋 값 생성, 오프셋 값을 적용한 구체의 형태 변형 및 3D mesh 재구축, Loss 계산 및 딥러닝 네트워크 학습 등의 6가지 과정으로 이루어져있으며, 학습 과정의 흐름도는 그림 2와 같다.

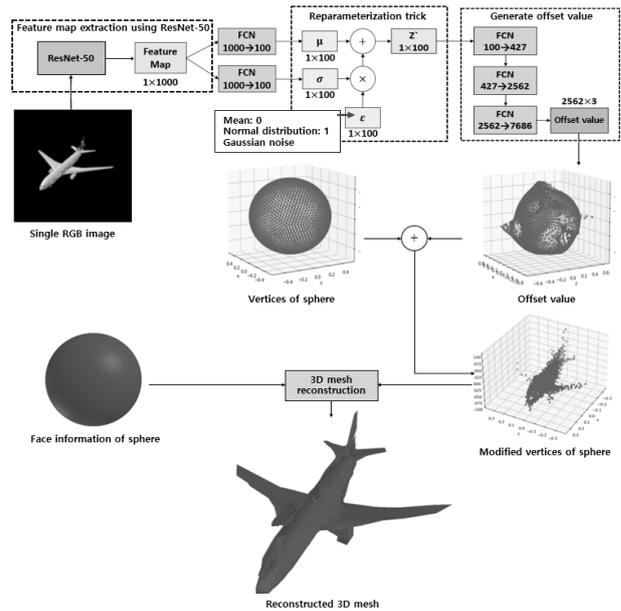


Fig. 1. A complete overview of the proposed method in this paper.

그림 1. 본 논문에서 제안한 기법의 전체 개요도

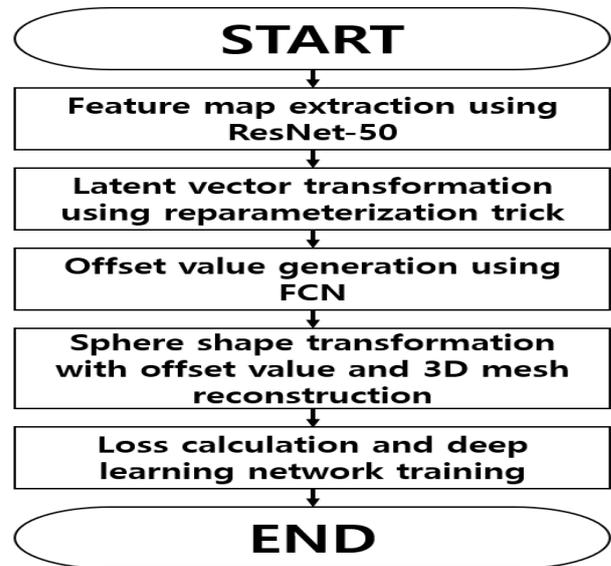


Fig. 2. The learning process of the proposed method in this paper.

그림 2. 본 논문에서 제안한 기법의 학습 과정

2. ResNet-50을 이용한 feature map 추출

ResNet(Residual Network)은 기존의 심층 딥러닝 네트워크의 gradient vanishing 문제를 해결하기 위해 고안된 딥러닝 네트워크로서 특징이 유실되지 않고 심층까지 효과적으로 전달되는 특징이 있다. 그림 3은 CNN과 ResNet에서 사용되는 residual learning의 차이를 나타낸다.

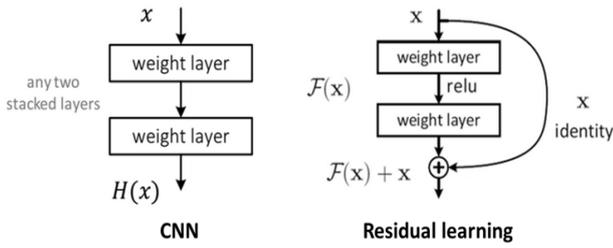


Fig. 3. Difference between CNN and residual learning.
그림 3. CNN과 residual learning의 차이

CNN은 x 를 입력으로 2개의 weight layer를 거친 후 $H(x)$ 라는 출력을 내며, 학습을 통해 최적의 $H(x)$ 를 찾기 위해 weight layer의 가중치를 수정한다. 그러나 residual learning은 $H(x) = F(x) + x$ 이며, $F(x) = H(x) - x$ 가 된다. 이러한 특징으로 인하여 x 의 미세한 변화를 검출하는 것에 높은 성능을 발휘하며 많은 논문에서 사용되고 있다. 본 논문에서는 50층으로 구성된 ResNet-50을 사용하여 128×128 크기의 입력 RGB 이미지로부터 1×1000 크기의 feature map을 추출한다. ResNet이 1000개의 카테고리를 분류하도록 설계되어 출력 크기가 1×1000 인 이유도 있으나, 후술할 reparameterization trick으로 크기를 축소시키기 위해서는 충분히 많은 특징들이 필요하기 때문에 출력 크기를 그대로 사용하였다.

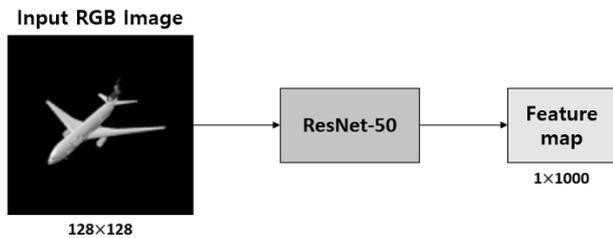


Fig. 4. Feature map extraction process using ResNet-50.
그림 4. ResNet-50을 사용한 feature map 추출과정

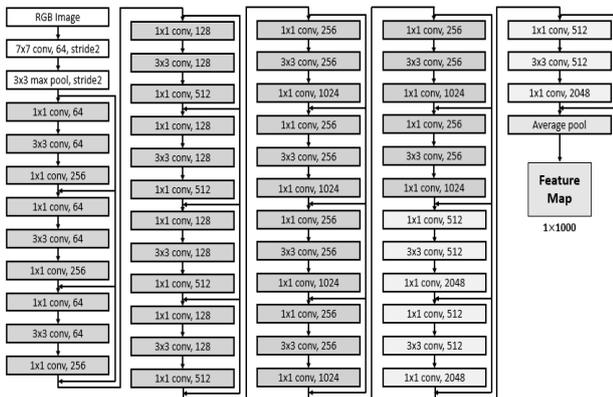


Fig. 5. Structure diagram of ResNet-50.
그림 5. ResNet-50의 구조도

그림 4는 ResNet-50을 사용한 feature map 추출과정을 나타내며 그림 5는 ResNet-50의 구조를 나타낸다.

3. Reparameterization trick을 사용한 latent vector 변환

Reparameterization trick은 주로 VAE(Variational AutoEncoder)[3]에서 사용되는 기법으로 딥러닝 네트워크를 통해 추출한 특징을 저차원의 확률 분포 형태의 latent vector로 변환하고 군집화 하는 기법이다. 본 논문에서는 feature map을 1×100 크기의 latent vector로 변환하여 데이터를 보다 분명하게 판단할 수 있게 하였다. Feature map을 1×100 크기의 latent vector로 변환하기 위해 FCN을 이용하여 1×100 크기의 특징 2개를 추출하고 각각을 latent vector Z 를 생성하기 위한 평균 μ 와 표준편차 σ 로 사용했다. σ 에 평균이 0, 표준편차가 1인 가우시안 노이즈 ϵ 를 곱해서 크기를 제한하여 불확실성을 낮췄다. 결과적으로 평균 μ 와 $\epsilon\sigma$ 를 더하여 확률 분포인 latent vector Z 를 생성한다. 그림 6은 reparameterization trick을 사용한 latent vector 변환 과정을 나타낸다.

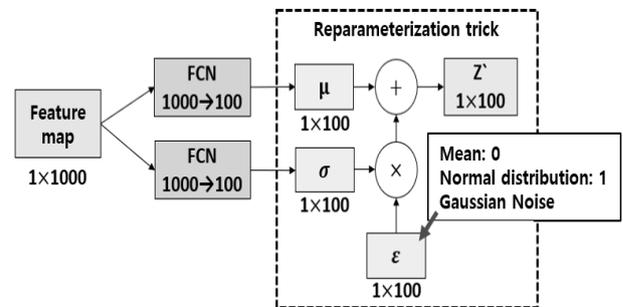


Fig. 6. Latent vector transformation process using reparameterization trick.

그림 6. Reparameterization trick을 사용한 latent vector 변환과정

4. FCN을 사용한 오프셋 값 생성

오프셋 값은 임의의 3D 포인트 클라우드의 점의 위치를 변경하기 위한 값으로써, 앞서 변환한 latent vector를 FCN의 입력 데이터로 사용하여 생성한다. 여러 층에 걸쳐 크기를 증가시켜야 LeakyReLU를 통해 비선형성을 높여 성능을 향상시킬 수 있다. 생성된 7686개의 값 중에서 앞의 2562개는 x 좌표이며, 중간의 2562개는 y 좌표, 마지막 2562개는 z 좌표이다. 그림 7은 FCN을 사용한 오프셋 값 생성과정을, 그림 8은 본 논문에서 사용된 FCN의 구조를 나타낸다.

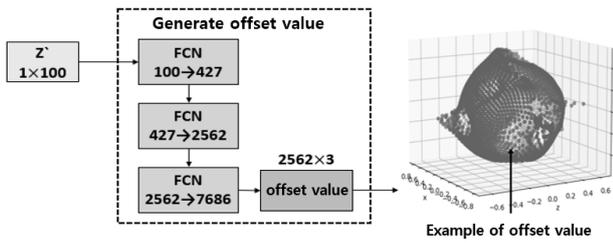


Fig. 7. Drawing of offset value generation process using FCN.

그림 7. FCN을 사용한 오프셋 값 생성과정

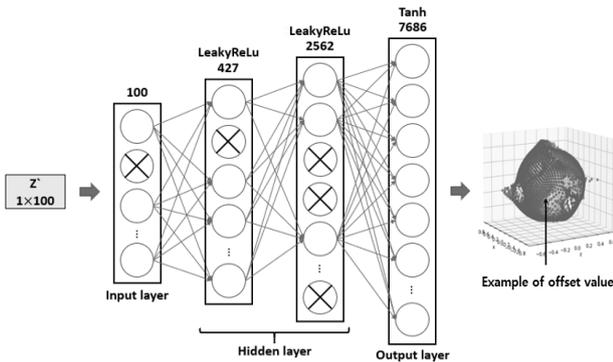


Fig. 8. Structure diagram of FCN used in this paper.

그림 8. 본 논문에서 사용된 FCN의 구조도

5. 오프셋 값을 적용한 구체의 형태 변형 및 3D mesh 재구축

오프셋 값을 구체의 꼭짓점에 적용하여 구체의 형태를 변형시켜 3D mesh를 재구축하는 기법은 본 논문에서 새롭게 제안하는 기법이다. 여러 가지 형태 중에 구체를 이용한 가장 큰 이유는 중심으로부터 모든 점이 똑같은 거리에 위치하고 있어 다양한 사물의 모양에 대응하기에 가장 적합하기 때문이다. 한편, 3D mesh 표면을 재구축하는 부분에 있어서 어느 꼭짓점끼리 연결하여 모서리, 혹은 면을 생성할지 결정하는 것은 굉장히 어려운 문제이다. 그러나 본 논문에서 제안하는 기법은 구체를 변형시키는 방법이므로 이미 구체의 면 정보를 가지고 있다. 따라서 해당 면 정보에 최적화된 방향으로 꼭짓점의 위치를 수정하는 것이 가능하다면 면 생성에 관련된 문제를 해결할 수 있다는 장점이 있다. 또한, 오프셋 값과 구체의 꼭짓점 사이에 덧셈 연산만을 수행하기 때문에 적은 연산으로 빠르고 효과적인 3D mesh 재구축이 가능하다.

5.1 반지름이 0.5인 구체 생성

형태를 변형시킬 재료로 중심의 좌표가 0, 0, 0이고 반지름이 0.5인 구체를 생성한다. 구체는 꼭짓점 개수가

2562개, 면의 개수가 5120개인 3D mesh 모델을 사용한다. 그림 9는 반지름이 0.5인 구체를 시각화한 것을 나타낸다.



Fig. 9. A sphere with a radius of 0.5 used in this paper. 그림 9. 본 논문에서 사용하는 반지름이 0.5인 구체

5.2 오프셋 값을 이용한 구체의 꼭짓점 위치 수정

앞서 생성한 구체의 꼭짓점 배열의 n번째 좌표 값과 RGB 이미지로부터 생성된 오프셋 값 배열의 n번째 좌표 값을 덧셈 연산하여 구체의 꼭짓점 위치를 변경한다. 그림 10은 오프셋 값을 적용한 구체의 꼭짓점 위치 변경 과정을 나타낸다.

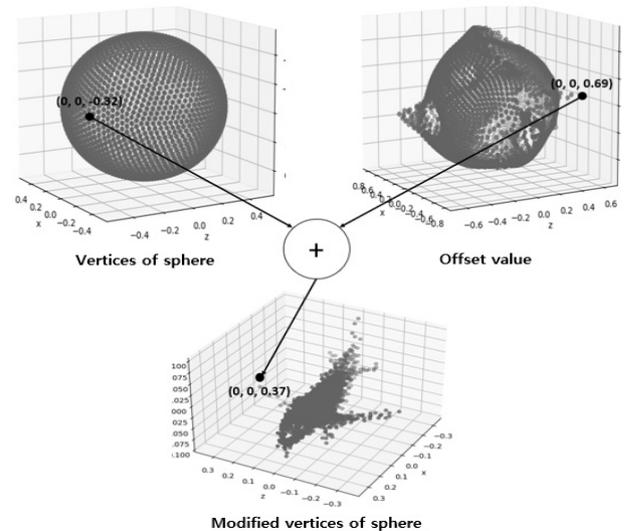


Fig. 10. The process of changing the position of the vertices of the sphere to which the offset value is applied.

그림 10. 오프셋 값을 적용한 구체의 꼭짓점 위치 변경과정

5.3 위치가 수정된 꼭짓점에 면 정보를 씌워 3D mesh를 재구축

3D mesh는 꼭짓점 정보와 면 정보로 이루어져 있다. 본 논문에서는 오프셋 값을 적용하여 위치를 변경한 구체의 꼭짓점의 배열을 3D mesh의 꼭짓점 정보로 사용하고 생성한 구체의 면 정보는 가공 없이 그대로 사용하여 3D mesh를 재구축한다. 그림 11은 꼭짓점 정보와 면 정보를 활용한 3D mesh 표면 생성의 예시를 나타낸다.

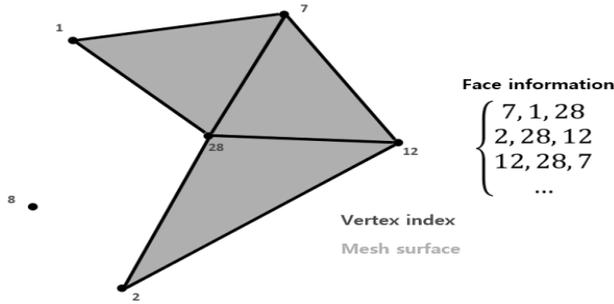


Fig. 11. Example of mesh surface creation using vertices and faces information.
 그림 11. 꼭짓점과 면 정보를 이용한 mesh 표면 생성의 예시

최종 재구성된 3D mesh는 그림 12와 같다.



Fig. 12. Example of mesh surface creation using vertices and faces information.
 그림 12. 꼭짓점과 면 정보를 이용한 mesh 표면 생성의 예시

6. Loss 계산 및 딥러닝 네트워크 학습

6.1 IoU loss

IoU loss는 2개의 3D 모델 간의 점유공간의 유사도를 평가하기 위한 방법이다. 3D 격자 안에서 각각의 3D mesh가 차지하는 점유공간을 비교하여 재구성된 3D mesh의 점유공간이 학습 데이터셋의 3D mesh의 점유공간과 유사해지도록 만든다. 그림 13은 IoU 계산을 도식화한 것을 나타낸다.

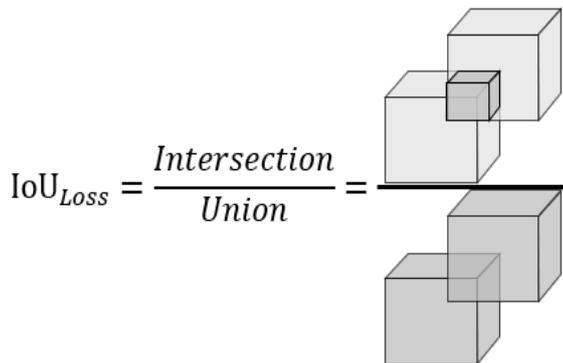


Fig. 13. Diagraming of 3D IoU calculation.
 그림 13. IoU 계산의 도식화

6.2 Laplacian smoothing loss

본 논문에서 사용한 laplacian smoothing loss의 정의는 다음과 같다. 만약 임의의 꼭짓점 i 와 j 가 같은 꼭짓점일 경우, -1 , i 와 j 가 모서리를 이룰 경우, $1/\text{deg}(i)$, 둘 다 해당되지 않을 경우에는 0 을 출력한다. 이때의 $\text{deg}(i)$ 는 임의의 꼭짓점 i 에 연결된 모서리의 개수를 의미한다. 모든 꼭짓점에 대해 laplacian smoothing loss를 계산하여 평균값을 학습에 이용한다. laplacian smoothing loss를 학습에 이용하는 이유는 꼭짓점 배열의 index를 고려하지 않고 면 정보를 적용하면 3D mesh가 엉망으로 재구성되기 때문에 꼭짓점 배열의 index를 정렬하기 위함이다. 그림 14는 두 꼭짓점 사이의 laplacian smoothing loss 계산의 예시를 나타낸다.

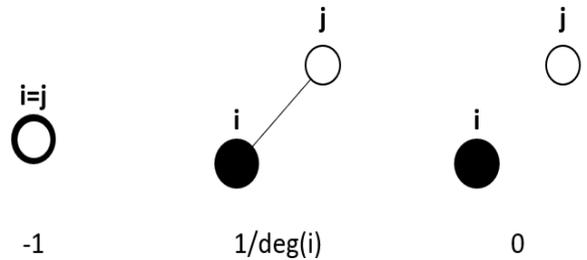


Fig. 14. Example of laplacian smoothing loss calculation using two vertices.
 그림 14. 두 꼭짓점을 이용한 laplacian smoothing loss 계산의 예시

그림 15의 (a)는 IoU loss, laplacian smoothing loss, chamfer distance loss를 학습에 사용했을 때의 결과를, (b)는 laplacian smoothing loss를 제외하고 학습했을 때의 결과를 나타낸다.

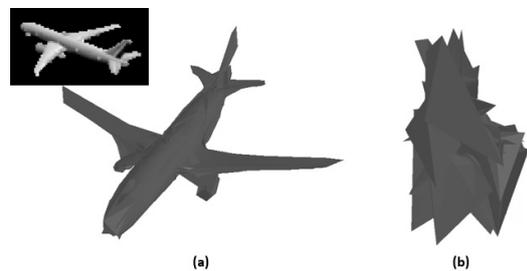


Fig. 15. Effect of laplacian smoothing loss on 3D mesh reconstruction.
 그림 15. Laplacian smoothing loss가 3D mesh 재구축에 미치는 영향

6.3 Chamfer distance loss

Chamfer distance loss를 학습에 이용하는 이유는 2개의 3D mesh에서 일정간격으로 샘플링한 3D 포인트

클라우드의 거리오차가 0에 가까워지도록 만들어 윤곽을 유사하게 만들기 위함이다. 식(1)은 chamfer distance loss의 계산식을, 그림 16은 2개의 3D 포인트 클라우드 사이의 chamfer distance 측정 예시를 나타낸다.

$$ChamferDistance(v_1, v_2) = \sum_{x \in v_1} \min_{y \in v_2} \|x - y\|_2 + \sum_{y \in v_2} \min_{x \in v_1} \|x - y\|_2 \quad (1)$$

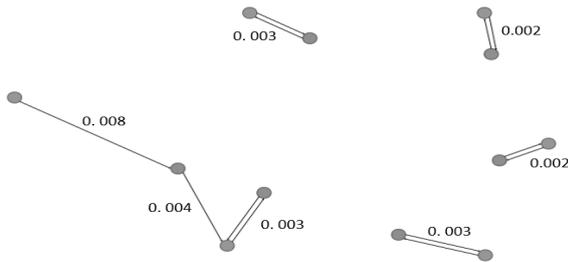


Fig. 16. Result image showing the extracted feature points.

그림 16. 추출된 특징점이 표시된 결과 이미지

6.4 딥러닝 네트워크 학습

본 논문에서는 IoU loss, laplacian smoothing loss, chamfer distance loss가 학습에 미치는 영향을 최적화하기 위해 각각의 loss에 α , β , γ 를 가중치로 적용하였다. 식(2)는 본 논문의 ResNet-50, FCN 등을 학습할 때 사용된 $Loss_{total}$ 을 나타낸다. 이때 α , β , γ 등은 학습 과정을 반복적으로 수행하여 최적의 결과를 얻기 위하여 계속 변경하며, 최고의 결과를 얻었던 가중치 값으로 결정된다.

$$Loss_{total} = \alpha * (1 - IoU_{Loss}) + \beta * chamfer\ distance_{Loss} + \gamma * laplacian\ smoothing_{Loss} \quad (2)$$

IoU와 chamfer distance가 목표한 값에 도달하지 못하면 가중치를 수정한 후에, 2000 epoch 학습하는 것을 반복하여 최적의 α , β , γ 를 도출한다. 그림 17은 학습 과정에서의 $Loss_{total}$ 값의 변화를 나타낸 그래프이다.

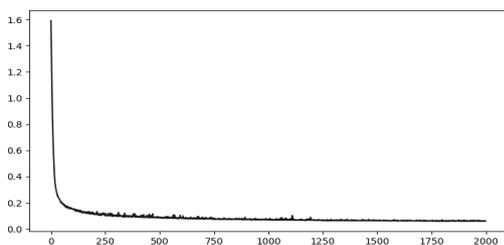


Fig 17. Graph of loss changes with epoch.

그림 17. Epoch에 따른 loss의 변화 그래프

8. 성능 실험

가. 실험 환경

실험에 사용된 운영체제 및 하드웨어는 Windows 10 64비트 운영체제를 기반으로 Intel(R) Core(TM) i7-10700k 3.8GHz CPU, RAM 16GB, NVIDIA GeForce RTX 3070 (VRAM 8GB) GPU로 구성되어 있다. 개발 도구는 Visual Studio Code와 Pytorch 1.9.1, CUDA 11.1, cuDNN 8.2.1 라이브러리를 사용하였다. 본 논문에서는 Princeton, Stanford 및 TTIC의 연구원들이 협업하여 구축한 데이터셋인 ShapeNet 데이터셋을 사용한다. 표 1은 실험에 사용된 ShapeNet 데이터셋의 13가지 카테고리를, 그림 18은 실험에 사용한 ShapeNet의 예시를 나타낸다.

Table 1. 13 categories of ShapeNet dataset used in the experiment.

표 1. 실험에 사용된 ShapeNet 데이터셋의 13가지 카테고리

Category	Contents
Object	bench, chair, lamp, speaker, gun, table, cabinet, monitor, long chair, cell phone
Vehicle	ship, airplane, car

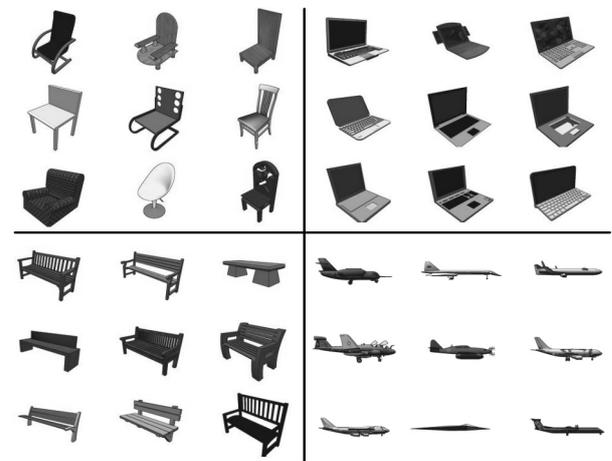


Fig. 18. Example of the ShapeNet dataset used in the experiment.

그림 18. 실험에 사용한 ShapeNet 데이터셋의 예시

나. 실험 결과

제한하는 기법의 객관적인 성능을 평가하기 위해 비교 논문들과 같은 조건에서 실험 결과를 산출했다. 평가 지표로는 IoU, chamfer distance를 사용하였다. 표 2는 ShapeNet 데이터셋에서의 본 논문에서 제안하는 기법과 다른 논문의 기법들의 비교 결과를 그림 19는 ShapeNet 데이터셋의 RGB 이미지에서의 3D mesh 재구축 결과를 나타낸다.

Table 2. Comparison result of the proposed method in this paper and the methods of other papers in the ShapeNet dataset.

표 2. ShapeNet 데이터셋에서의 본 논문에서 제안하는 기법과 다른 논문의 기법들의 비교 결과

Method	IoU ↑	Chamfer Distance ↓
3D-R2N2[4]	0.493	0.278
PSGN[5]	-	0.215
Pix2Mesh[6]	0.480	0.216
AtlasNet[7]	-	0.175
ONet[8]	0.571	0.215
The proposed method	0.581	0.212

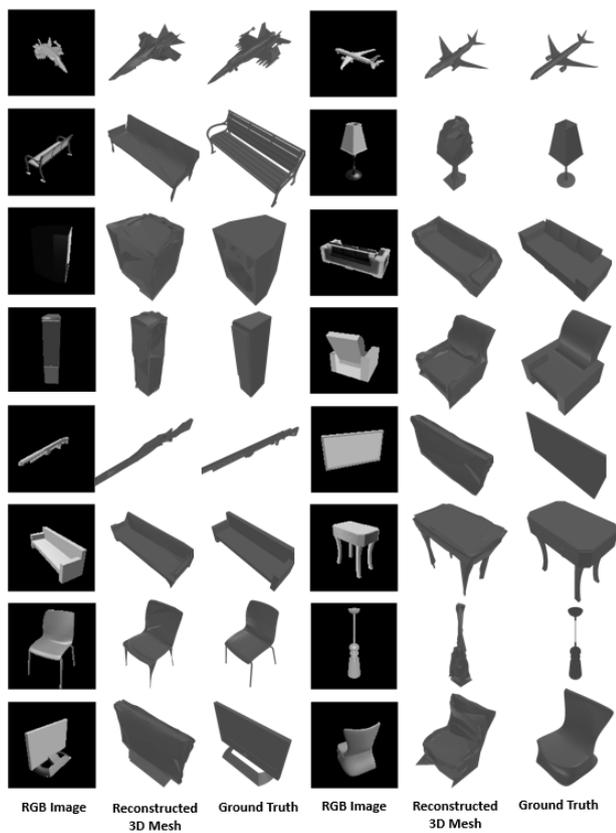


Fig. 19. 3D mesh reconstruction results from a RGB image of the ShapeNet dataset.

그림 19. ShapeNet 데이터셋의 RGB 이미지에서의 3D mesh 재구축 결과

본 논문에서는 IoU loss를 이용하여 학습했기 때문에 제안하는 기법의 IoU 값이 다른 논문에서 발표한 기법들보다 높은 수치를 나타내었다. IoU의 의존도가 지나치게 높으면 조각난 형태의 mesh가 3D 격자 안을 점유하고 있을 뿐인 비정상적인 모양의 3D mesh가 재구축 될 가능성이 있다. 그러나 본 논문에서는 laplacian smoothing

loss를 활용하여 이러한 문제를 방지하였다. 그림 20은 IoU를 비교 평가한 결과를 그래프로 나타내고 있다. IoU 값은 수치가 1에 가까울수록 우수함을 나타내므로, 다른 논문에서 발표한 기법들보다 우수한 것이 확인되었다.

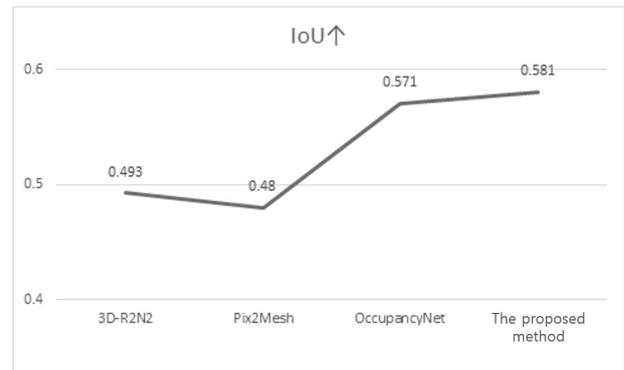


Fig. 20. Graph comparison results of IoU with the proposed method and proposed in other papers.

그림 20. 제안하는 기법과 다른 논문에서 제안한 기법들의 IoU 값의 그래프 비교 결과

그림 21은 chamfer distance를 비교 평가한 결과를 그래프로 나타내고 있다. chamfer distance 값은 수치가 0에 가까울수록 우수함을 나타내므로, 제안하는 기법이 다른 논문에서 발표한 기법들보다 우수한 것이 확인되었다. 단, AtlasNet은 다른 논문들과 다르게 RGB 이미지뿐만 아니라 3D 포인트 클라우드를 입력 데이터로 사용하여 학습하였기 때문에 같은 조건이라고 볼 수 없으며 따라서 RGB 이미지 기반의 기법들 중에서 본 논문에서 제안하는 기법의 chamfer distance 값이 가장 우수하다.

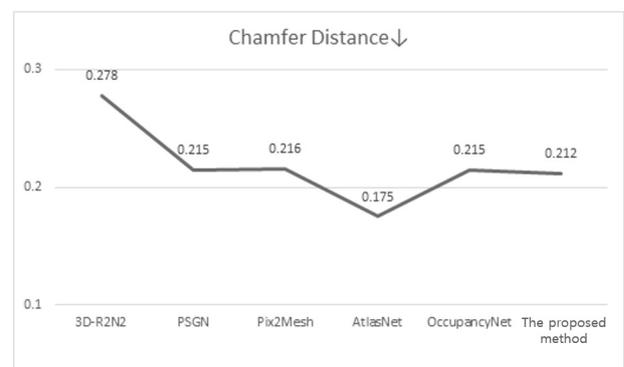


Fig. 21. Graph comparison results of chamfer distance with the proposed method and proposed in other papers.

그림 21. 제안하는 기법과 다른 논문에서 발표한 기법들의 chamfer distance 값의 그래프 비교 결과

III. 결론

본 논문에서는 딥러닝과 구체의 형태 변형 방법을 이용한 단일 이미지에서의 3D mesh 재구축 기법을 제안하였다. 제안한 기법은 기존의 방식과 다른 다음과 같은 독창성이 있다. 첫 번째, 기존의 근처의 가까운 점들을 연결하여 모서리 또는 면을 구축하는 방식과 다르게 딥러닝 네트워크를 통하여 구체의 꼭짓점의 위치를 사물의 3D 포인트 클라우드와 매우 유사하게 수정한다. 3D 포인트 클라우드를 이용하므로 메모리가 적게 필요하며 구체의 꼭짓점에 오프셋 값 사이에 덧셈 연산만을 수행하기 때문에 연산량이 적다. 두 번째, 수정한 꼭짓점에 구체의 면 정보를 씌워 3D mesh를 재구축한다. 구체의 꼭짓점의 위치를 수정하여 생성한 3D 포인트 클라우드의 점들의 간격이 일정하지 않을 때에도 이미 점들 사이의 연결 여부를 나타내는 구체의 면 정보라는 3D mesh의 면 정보를 가지고 있어 표현의 단순화나 결손을 방지할 수 있다. 제안하는 기법의 객관적인 신뢰성을 평가하기 위해 공개된 표준 데이터셋인 ShapeNet 데이터셋을 이용하여 비교 논문들과 같은 방법으로 실험한 결과, 본 논문에서 제안하는 기법의 IoU 값이 0.581로, chamfer distance 값은 0.212로 산출되었다. IoU 값은 수치가 높을수록, chamfer distance 값은 수치가 낮을수록 우수한 결과를 나타내므로 다른 논문에서 발표한 기법들보다 3D mesh 재구축의 결과의 효율성이 입증되었다.

후속 연구로는 구체의 꼭짓점 위치를 더 정확히 정렬시켜 해당 꼭짓점에 면을 씌워서 재구축한 3D mesh의 정확도를 향상시킬 수 있도록 새로운 loss 함수를 도출하는 연구가 필요하다.

References

[1] CIPRESSO, Pietro, et al. "The past, present, and future of virtual and augmented reality research: a network and cluster analysis of the literature," *Frontiers in psychology* 2086, 2018. DOI: 10.3389/fpsyg.2018.02086

[2] Gokturk, S. Burak, Hakan Yalcin, and Cyrus Bamji. "A time-of-flight depth sensor-system description, issues and solutions," *2004 Conference on Computer Vision and Pattern Recognition Workshop. IEEE*, 2004. DOI: 10.1109/CVPR.2004.291

[3] KINGMA, Diederik P.; WELLING, Max. "Auto-encoding variational bayes," *International Conference on Learning Representations (ICLR)*, 2014.

[4] Choy, Christopher B., et al. "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," *European conference on computer vision. Springer, Cham*, 2016.

[5] Fan, Haoqiang, Hao Su, and Leonidas J. Guibas. "A point set generation network for 3d object reconstruction from a single image," *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017. DOI: 10.48550/arXiv.1612.00603

[6] Wang, Nanyang, et al. "Pixel2mesh: Generating 3d mesh models from single rgb images." *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

[7] Groueix, Thibault, et al. "A papier-mâché approach to learning 3d surface generation," *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[8] Mescheder, Lars, et al. "Occupancy networks: Learning 3d reconstruction in function space." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

BIOGRAPHY

Jeong-Yoon Kim (Member)



2020 : BS degree in Electronic Engineering, Hanbat National University

2022 : MS degree in Electronic Engineering, Hanbat National University

2022~current : Ph. D degree course of Electronic Engineering Hanbat National University

Seung-Ho Lee (Member)

1986 : BS degree in Electronic
Engineering, Hanyang University
1989 : MS degree in Electronic
Engineering, Hanyang University
1994 : Ph. D degree in Electronic
Engineering, Hanyang University

1994~current : Professor, Department of Electronic
Engineering, Hanbat National University