

논문 2021-16-28

# 불규칙 3차원 데이터를 위한 기하학정보를 이용한 딥러닝 기반 기법 분석

## (Survey on Deep Learning Methods for Irregular 3D Data Using Geometric Information)

조성인\*, 박해주  
(Sung In Cho, Haeju Park)

Abstract : 3D data can be categorized into two parts : Euclidean data and non-Euclidean data. In general, 3D data exists in the form of non-Euclidean data. Due to irregularities in non-Euclidean data such as mesh and point cloud, early 3D deep learning studies transformed these data into regular forms of Euclidean data to utilize them. This approach, however, cannot use memory efficiently and causes loses of essential information on objects. Thus, various approaches that can directly apply deep learning architecture to non-Euclidean 3D data have emerged. In this survey, we introduce various deep learning methods for mesh and point cloud data. After analyzing the operating principles of these methods designed for irregular data, we compare the performance of existing methods for shape classification and segmentation tasks.

Keywords : 3D deep learning, Irregular data, Mesh, Point cloud, Classification, Segmentation

### I. 서론

딥러닝은 다양한 범주의 문제들을 해결하는 강력한 도구로 빠르게 발전해왔다. 컴퓨터 비전 분야에서도 딥러닝을 활용한 수많은 연구가 등장하였고, 이들은 분류 (classification) [1], 분할 (segmentation) [2], 탐지 (detection) [3] 등에서 주목할 만한 성과를 이루었다. 기존의 딥러닝 연구는 주로 2차원 영상과 같은 격자 (grid) 형태의 데이터를 기반으로 설계되었고 3차원 데이터에서는 복잡성과 불규칙성으로 인해 여전히 딥러닝 기반의 기법보다는 handcrafted descriptor의 활용 비중이 높았다 [4-6]. 최근에는 다양한 형태의 데이터 표현을 지원하는 딥러닝 기법이 활발히 연구되면서 3차원 데이터 처리에도 딥러닝 기술이 큰 주목을 받고 있다 [7-10].

3차원 데이터는 유클리드 (euclidean) 데이터와 비유클리드 (non-euclidean) 데이터로 분류된다. 유클리드 데이터는 다차원 선형 공간에서 모델링 되는 데이터로 volumetric, multi-view, RGB-D 데이터 등이 대표적이다. 비유클리드 데이터는 유클리드 공간에서 표현되지 못하는 모든 데이터를 의미하며 mesh와 point cloud 데이터 등이 대표적이다.

초기의 3차원 딥러닝 연구들은 불규칙한 비유클리드 데

이터를 voxel [7, 11, 12] 또는 multi-view images [8, 13]와 같은 유클리드 데이터의 규칙적인 형태로 변형해 인공신경망의 입력으로 취하는 방식을 사용했다. 이러한 방식은 기존의 인공신경망 구조를 그대로 적용할 수 있어 편리하지만 데이터를 변형하면서 메모리 낭비를 유발하고 입력 데이터의 원형의 정보가 소실된다는 한계가 존재한다. 이를 극복하고자 입력 데이터에 변형을 가하지 않고 비유클리드 3차원 데이터에 인공신경망을 직접적으로 적용할 수 있는 기법을 설계하여 3차원 객체 형상의 본질적인 정보를 학습하는 방식에 대한 여러 연구가 등장하였다 [14-22]. 본 논문에서는 이 중 mesh와 point cloud를 기반으로 하는 딥러닝 기법들의 동작 원리를 분석하고, 실험 결과를 통해 분류와 분할 작업에서의 성능을 비교한다.

### II. 기존 연구의 분석

#### 1. Mesh 기반 딥러닝 기법

Mesh는 3차원 공간에서 정점 (vertex)들의 집합으로 만들어진 다각형 (polygon)들의 집합이다. Mesh는  $(V, F)$ 로 정의되고 이때  $V = \{v_1, v_2, \dots\}$ 로  $R^3$ 에서 정점 좌표들의 집합이다. 정점은 간선 (edge)으로 연결되고 간선들의 닫힌 집합이 다각형 면  $F$ 를 이룬다. 일반적으로 객체의 형상을 근사할 때 삼각형이 사용되기 때문에 본 논문에서는  $F$ 를 삼각형 면 (face)으로 간주한다. Mesh 구조는 렌더링 과정이 단순하고 객체 형상의 복잡한 표면 (surface)과 위상 (topology)을

\*Corresponding Author (csi2267@dongguk.edu)

Received: Aug. 20, 2021, Revised: Sep. 15, 2021, Accepted: Sep. 17, 2021

S.I. Cho: Dongguk University (Prof.)

H.J. Park: Dongguk University (Integrated M.S. program Student)

\* This research was supported by Samsung Electronics, the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1C1C1009662), and Basic Science Research Program through the NRF funded by Ministry of Education (No. NRF-2020X1A3A1093880)

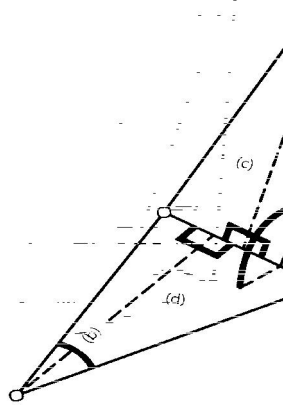


그림 1. 간선 특징 벡터의 다섯 가지 요소 [14]. (a)와 (b)는 내각, (c)와 (d)는 간선과 삼각형 수선의 길이 비, (e)는 2면각  
 Fig. 1. Five Elements of Edge Feature Vector [14]. (a) and (b) are two inner angles, (c) and (d) are two edge-length ratios, and (e) is the dihedral angle.

충실하게 나타내기 때문에 컴퓨터 그래픽스 분야에서 3차원 모델을 렌더링하고 저장하는데 가장 널리 사용된다. 그러나 구성 요소가 많고 불규칙하여 연결 관계가 복잡하고, 노이즈에 민감해 다루기 힘든 특성이 있다. 그럼에도 불구하고 현존하는 가장 강력한 3차원 객체 표현법이므로 mesh 데이터 고유의 특성을 활용하는 효과적인 인공신경망 구조를 설계하는 것이 매우 중요하고 어려운 과제이다. 본 장에서는 mesh 데이터가 가진 기하학적 특징과 연결 관계를 인공신경망의 입력으로 사용하는 두 가지 딥러닝 연구를 소개한다.

1.1 MeshCNN [14]

Mesh Convolution Neural Network (MeshCNN)은 mesh 구조에 특화된 객체 형상 분할과 분류를 수행하기 위해 제안된 딥러닝 기법이다. Mesh 데이터에 내재한 고유의 특성을 활용하기 위해 기존 Convolution Neural Network (CNN)의 convolution과 pooling 레이어의 개념을 차용하여 개별

간선에서 작동하는 mesh convolution과 mesh pooling 레이어를 설계하였다. 2차원 영상과 같은 격자 기반의 표현에서는 특징 (feature)과 공간상 인접 정보 (spatial neighborhood information)를 행렬의 형태로 쉽게 나타낼 수 있지만 불규칙한 형태의 mesh에서는 삼각형의 연결 정보로부터 형상의 특징을 규정해야 한다. 이를 위해 MeshCNN에서는 간선을 기본 구성단위 (unit)로 이용하는데, 이는 2차원 영상에서 픽셀의 역할과 같다.

그림 1은 mesh convolution의 입력으로 사용되는 간선의 다섯 가지 특징을 보여준다. 하나의 간선을 기준으로 인접한 두 삼각형에서 2면 각, 내각 2가지, 간선과 삼각형 수선의 길이 비 2가지로 구성된 5차원 기하 특징 벡터 (geometric feature vector)를 추출한다. 그리고 모든 간선이 두 삼각형과 인접한다는 점을 이용하여 하나의 간선에 대해 인접한 네 개의 간선을 convolution 이웃으로 정의한다. 그림 2의 (a)는 convolution 이웃을 나타낸다. 이웃 간선 입력 순서에 불변한 convolution 결과를 도출하기 위해 대칭 함수 (symmetric operation)를 이용하여 convolution 수용 영역 (receptive field)을 정의한다. 이후 convolution 커널을 이용해 수용 영역의 특징들을 결합한 새로운 특징 벡터를 추출한다.

Mesh pooling은 정해진 간선 수에 도달할 때까지 간선을 병합 (edge collapse) [23]하여 mesh 해상도를 감소시킨다. 이때 특징 벡터의 L2 norm이 최소인 간선부터 축소되는데, 그림 2의 (b)와 (c)에서 나타나듯이 객체 표면 위상을 유지 하면서 5개 간선을 2개로 축소 병합한다. Pooling 이후 각 간선의 특징 채널은 pooling 이전 간선들의 특징 채널의 평균으로 도출된다. 다시 말해 mesh pooling은 간선의 수를 줄이며 분류나 분할과 같은 훈련 목적에 따라 중요한 특징을 가진 간선은 보존하고 그렇지 않은 (즉, 정보가 적은) 간선은 병합하는 task-driven 과정이다. Mesh pooling이 끝나면 mesh 위에서 새로운 연결성 (connectivity)이 정의되어 후속 convolution을 위한 새로운 convolution 이웃이 규정된다. Mesh unpooling은 mesh pooling의 역과정이다. 분할 작업의

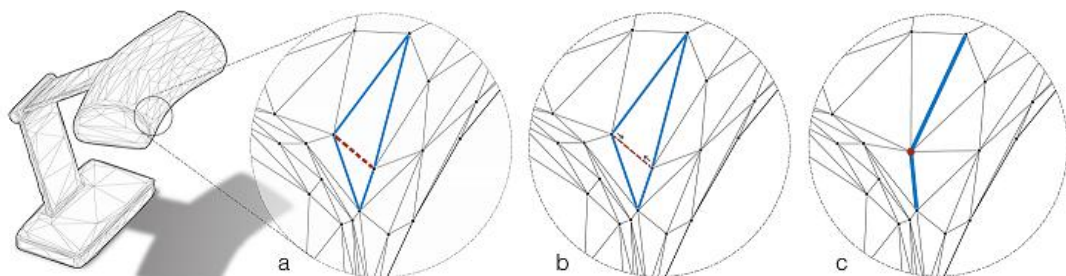


그림 2. Mesh pooling의 예시 [14]. (a) 하나의 간선 (점선)을 기준으로 인접한 두 삼각형이 만드는 네 간선이 convolution 이웃. Pooling 과정에서 점선으로 표시된 간선이 한 점으로 축소하면서 (b)의 기존 5개 이웃 간선이 (c)의 두 간선으로 병합.  
 Fig. 2. Example of Mesh Pooling [14]. (a) Four edges of the two incident triangles consist of neighborhood for convolution. In (b), the edge (denoted by dotted line) is collapsing to a point, and the five neighborhood edges merged into the two edges in (c).

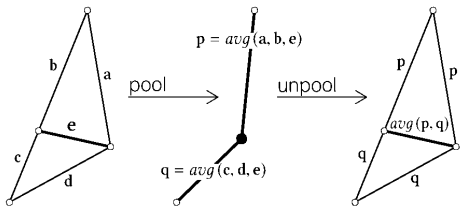


그림 3. Mesh pooling과 unpooling 과정에서 특징 결합 [14].  
Fig. 3. Feature Aggregation During Mesh Pooling and Unpooling [14].

경우 각 간선마다 클래스 예측이 할당되어야 하므로 mesh pooling으로 소실된 mesh의 기존 해상도를 회복하는 unpooling 과정이 필요하다. Pooling 과정에서 병합 이전과 이후 간선들의 연관성을 그래프 형태로 기록하고 이를 이용하여 unpooling 과정에서 mesh 위상과 간선 특징 벡터를 upsampling 한다. 이때, unpooling 된 간선의 특징 벡터는 pooling 이후의 두 간선 특징 벡터의 가중치 합이 된다. 그림 3은 mesh pooling과 unpooling 과정을 설명한다.

MeshCNN은 이러한 mesh convolution과 mesh pooling 과정을 통해 중요한 정보를 가진 간선을 보존하면서 그 특징을 결합한다. 최종적으로 Global Average Pooling (GAP) 레이어를 연결해 전역 정보를 도출하여 객체의 클래스를 예측하거나 (분류), unpooling을 통해 각 간선 별 최종 특징 벡터를 도출하여 간선별로 클래스를 예측할 수 있다 (분할).

MeshCNN은 인공신경망을 mesh 데이터에 직접 적용하여 간선을 기반으로 특징 벡터를 결합하면서 mesh를 단순화하는 기법이다. 불규칙적이고 비균일 (non-uniformity)한 데이터 구조에 맞추어 설계된 convolution과 pooling 연산이 3차원 형상에서 고유한 특성의 분석과 활용을 용이하게 한다. 초기 mesh triangulation (tessellation)과 간선 입력 순서가 결과에 영향을 미치지 않으며 pooling 과정에서 자료구조

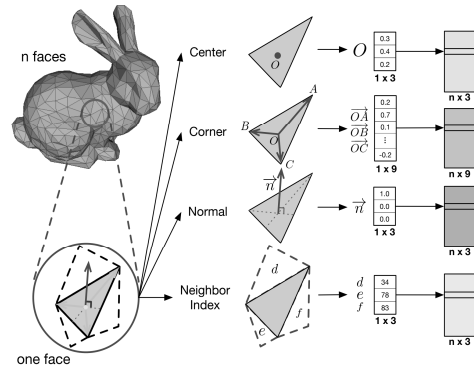


그림 4. 각 삼각형의 초기값 [15].  
Fig. 4. Initial Values of Each Face [15].

의 지속적인 갱신이 필요하다. 또한 task-driven 과정으로 목적에 맞추어 mesh 단순화를 진행하기 때문에 인공신경망의 출력이 목적에 부합하게 해석 가능하다는 특징이 있다.

1.2 MeshNet [15]

MeshNet은 삼각형의 특징을 공간적 (spatial), 구조적 (structural) 특징으로 나누어 분석하고 종합하여 객체 형상 분류, 검색 (retrieval)을 수행하기 위해 설계된 딥러닝 기법이다. MeshNet에서는 데이터 구성을 단순화하고, 충분한 정보를 얻기 위해 삼각형 면을 하나의 구성 단위로 간주한다. Mesh는 삼각형 면들의 list로 표현되고 하나의 삼각형은 최대 3개의 삼각형과 연결을 가질 수 있기 때문에 간선을 공유하는 삼각형 면 사이의 규칙적인 연결 관계가 형성되어 인공신경망에서의 활용을 쉽게 한다.

MeshNet은 그림 4에서 나타낸 바와 같이 각 삼각형 면에 대한 초기값으로 neighbor information (neighbor index)과 face information (center, corner, normal)을 정의한다. 삼각형 면은 위치 뿐만 아니라 형태에서도 정보를 담고 있기 때문에 이 정보를 공간적 특징 (center)과 구조적 특징

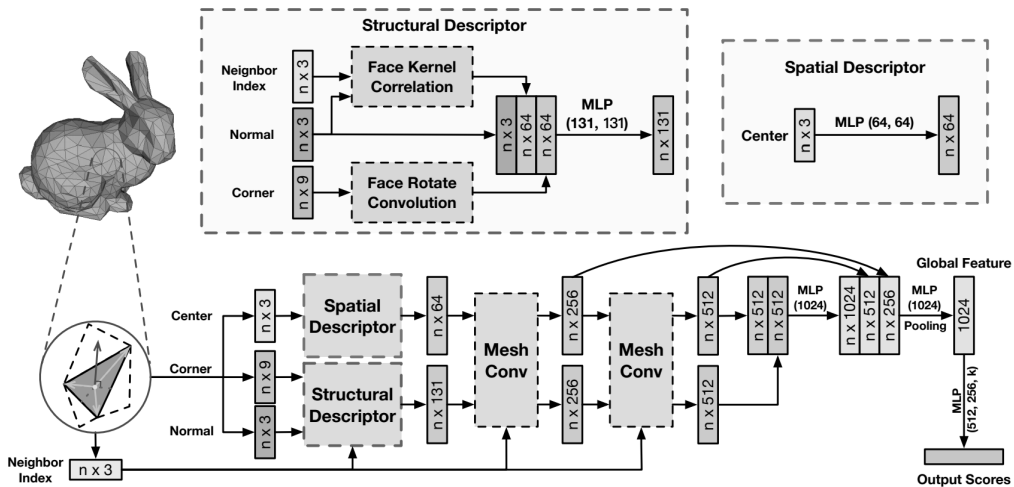


그림 5. MeshNet의 구조 [15].  
Fig. 5. The Architecture of MeshNet [15].

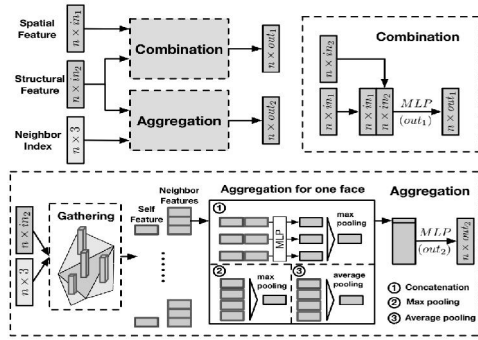


그림 6. Mesh convolution의 구조 [15].  
Fig. 6. The Architecture of Mesh Convolution [15].

(corner, normal, neighbor index)으로 분류한다.

MeshNet의 전체 구조는 그림 5에 나타나 있다. 공간적 특징과 구조적 특징으로 분류된 두 가지 정보는 각각 spatial descriptor와 structural descriptor의 입력으로 사용된다. 이때 structural descriptor는 삼각형의 내부 구조를 파악하는 face rotate convolution 블록과 삼각형의 외부 구조를 파악하는 face kernel correlation 블록으로 이루어진다. 두 가지 descriptor에서 각각 생성된 특징 벡터는 neighbor index와 함께 mesh convolution 블록의 입력으로 사용된다.

그림 6은 mesh convolution 블록을 나타낸다. Mesh convolution 블록은 인접한 삼각형의 정보를 취합하여 convolution의 수용 영역을 확장하도록 설계되었는데, 두 가지 블록으로 구성된다. 첫 번째는 공간적 정보와 구조적 정보를 연결 (concatenate)하고 Multi-Layer Perceptron (MLP)를 적용하는 combination 블록이다. 두 번째는 구조적 정보와 이웃 인덱스 정보를 입력으로 하여 이웃한 삼각형 면들의 특징을 결합하는 aggregation 블록인데, MeshNet은 제시한 aggregation의 3가지 방식 중 concatenation을 최종적으로 채택하였다 (그림 6의 ①). 이렇게 mesh convolution 블록에서 최종 출력으로 생성된 특징 벡터 두 가지를 연결하고 다시 MLP를 거쳐 분류나 검색을 위한 score가 도출된다.

이처럼 MeshNet은 삼각형 면에 내재한 풍부한 정보를 충분히 활용할 수 있도록 블록을 설계하였고, 삼각형 면을 한 단위로 처리하고 대칭 함수를 활용하면서 mesh의 복잡성과 불규칙성 문제를 해결하였다. 일반적으로, 계산 비용 (computational cost) 측면에서 mesh 기반의 방법보다 point cloud 기반의 방법이 더 효율적인 것으로 알려져 있었으나 MeshNet은 시간 복잡도와 공간 복잡도 측면에서 point cloud를 활용한 방법과 유사한 성능을 제공할 수 있음을 증명하였다.

2. Point Cloud 기반 딥러닝 기법

Point cloud 데이터는  $n$ 개의 3차원 point들의 집합  $\{P_i | i = 1, \dots, n\}$ 이며, 각 point  $P_i$ 는  $(x, y, z)$  좌표와 색상, 법선 등의 특징을 포함하는 벡터이다. LIDAR 센서 등을 통해 데이터를 손쉽게 얻을 수 있으며 다른 표현 방식으로의 변환도 쉽다. 그러나 순서가 없는 point들의 단순 집합이기 때문에 point 간 인접성이나 연결성이 정의되지 않는다. 이로 인해 국부 표면 (local surface)을 나타내는데 모호함이 있어 고해상도를 요구하는 날카로운 모양 등의 보존이 어렵고 객체의 원형을 정확히 추론하기 어려운 한계가 있다. 그러나 불규칙성과 복잡성을 가지는 mesh와 달리 균일하고 간편한 구조가 인공신경망에서의 용이한 학습을 가능케 하므로 최근 point cloud에 딥러닝 구조를 적용하는 많은 연구가 등장하여 높은 성능을 제공하였다 [9, 16-19]. 본 장에서는 point cloud 데이터를 인공 신경망의 입력으로 사용하여 분류 및 분할 작업을 수행할 수 있는 두 가지 딥러닝 연구를 소개한다.

2.1 PointNet [16]

PointNet은 point cloud 데이터에 직접적으로 인공신경망을 적용하여 객체 형상 분류, 객체 부분 분할 또는 장면 분할을 수행하기 위한 방법론의 선구자적 논문이다. Point cloud 데이터가 가진 세 가지 주요 특성에 초점을 맞추어 딥러닝 구조를 설계하였고, 이를 통해 point cloud 데이터 활용을 어렵게 해왔던 난제들의 해결 방법을 제시하였다. 뒤이어 발표된 많은 논문이 PointNet의 이러한 기법들을 이용하고 있다. 그

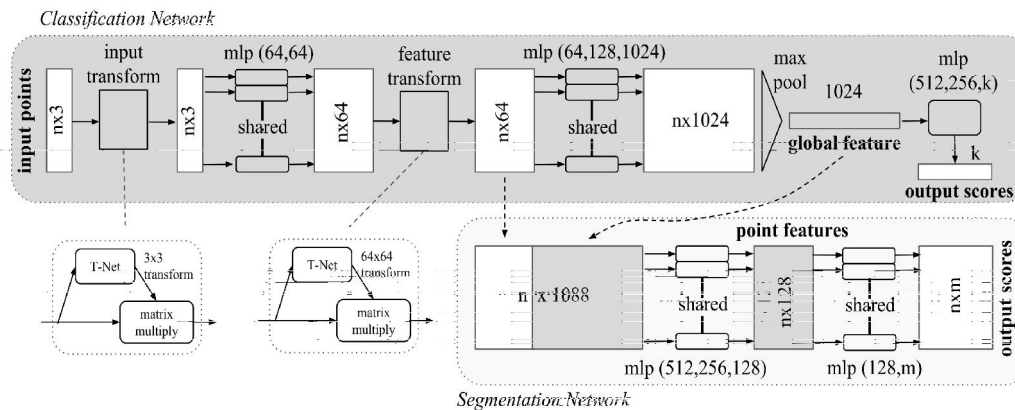


그림 7. PointNet의 구조 [16].  
Fig. 7. The Architecture of PointNet [16].

림 7은 PointNet의 전체 구조를 설명한다.

첫 번째 특성은 unordered problem이다. Point cloud는 자료 구조적 관점에서 정렬되지 않은 단순 점들의 집합이다. 따라서 크기가  $n$ 인 point 집합에서 데이터 입력 순서는  $n!$ 의 순열 (permutation)이 있고 네트워크의 출력은 이 순열에 영향을 받지 않아야 한다. PointNet은 이러한 문제를 단일 대칭 함수 (single symmetric function)를 통해 전역 정보를 결합하는 것으로 해결한다. 대칭 함수는 더하기 또는 곱하기 연산과 같이 변수의 교환에 대하여 불변인 연산이며 PointNet은 이 중 max pooling 연산자를 사용한다.

두 번째 특성은 인접한 point들이 유의미한 부분 집합을 형성한다는 것이다. 따라서 전역 특징 (global feature) 뿐만 아니라 인접한 point들 사이의 지역적 구조 (local structure)에서 결합적 상호작용 (combinatorial interactions)을 파악할 수 있어야 한다. PointNet은 그림 7의 segmentation network에서 전역 특징 벡터를 각 point의 특징 벡터와 연결하여 지역 정보와 전역 정보를 결합한 포인트 별 특징 벡터를 새롭게 추출한다. 이를 통해 국부 기하 (local geometry)와 global semantic을 함께 고려한 추론이 가능해진다.

마지막 특성은 point cloud에 회전이나 이동 등의 기하 변환을 가해도 객체의 분류나 분할 결과는 변함없어야 한다는 것이다. PointNet은 그림 7에서 T-net으로 표현된 두 가지 합동 정렬 네트워크 (joint alignment network)를 사용하여 이를 해결한다. 입력 집합을 표준 공간 (canonical space)에서 정렬하는 Spatial Transformer Network (STN) [24]에서 아이디어를 얻은 T-net은 입력 point 집합을 표준 공간으로 보내는 어파인 변환 행렬 (affine transformation matrix)을 추론하는 mini-network이다. 이 네트워크에서 계산된 input transform 행렬이 입력 point 집합의 좌표에 곱해지면서 입력 형태의 변환에도 네트워크 출력이 불변하도록 학습된다. STN은 embedding space에서도 마찬가지로 적용될 수 있기 때문에 feature space의 정렬에도 사용된다. 두 번째 T-net은 feature transform 행렬을 추론하여 특징 벡터가 표준 공간에서 정렬될 수 있도록 하는데, 이때 feature space에서의 변환 행렬은 차원이 매우 크기 때문에 직교 행렬 (orthogonal matrix)이 되도록 하는 정규화 항 (regularization term)을 사용하여 안정성 (stability)을 높인다.

PointNet은 상기 특성들에 맞추어 설계되었다. Classification network에서는 max pooling으로 전역 특징 벡터를 도출하고, MLP를 거쳐  $k$ 개의 각 클래스에 대한 prediction score를 출력한다. Segmentation network에서는 classification network에서 도출된 전역 특징 벡터와 각 point의 특징 벡터를 연결한다. 그리고 MLP를 거쳐  $n$ 개의 포인트 별로  $m$ 개의 클래스에 대한 prediction score를 출력한다.

이러한 특징 추출 과정을 통해 PointNet은 형상 학습에 유용한 정보를 제공하는 key point들을 요약 (summarize)할 수 있게 학습되고 그 결과가 객체 형상의 주요 골자 (skeleton)와 일치한다. 독립적인 point별 처리 후에 전역 특징을 종합하는 방식으로 연산 속도가 빠르며 point 손실 (missing)이나 손상 (corruption)과 같은 사소한 변형 (perturbation)에 강력하고 mesh 고유의 불규칙성을 해결할 수 있다. 그러나 형상의 지역적인 구조를 놓치는 한계가 있어 이를 보완한 후속 딥러닝 연구들이 대거 등장하였는데 대표적인 것이 hierarchical application을 제안한 PointNet++ [25]와 superpoint [26], 동적 그래프를 이용한 DGCNN [27] 등이 있다.

2.2 Relation-Shape CNN (RS-CNN) [17]

PointNet 이후로 point들의 지역적인 관계에 의거해서 국부 형상 정보를 추출하려는 여러 시도가 있었다. 그 중 하나인 RS-CNN은 Relation-Shape Convolution (RS-Conv) 연산자를 이용해 point 간의 기하학적 관계로부터 data-dependent weight를 학습한다. 개별적인 독립 point에서 특징 벡터를 추출한 이후 max pooling으로 종합하는 PointNet과는 달리 RS-CNN은 국부적인 인접 영역에서 기정 (predefine)된 기하 특징을 모아서 유의미한 형상 정보를 형성한다. 이는 입력 순서에 불변하고 기하학적 변형에도 견고한 PointNet의 특성을 유지하면서, 추가로 형상 인식도 (shape-awareness)를 높이는 결과를 이끌었다.

그림 8은 RS-Conv 연산을 나타낸다. Sample point  $x_i$ 를 중심으로 반지름  $r$  이내의 모든 이웃 point들을 구체 집합 (spherical neighborhood)  $N(x_i)$ 로 정의하면  $x_i$ 와  $N(x_i)$ 간에 3차원 유클리드 거리의 기하학적 관계가 규정된다.  $x_j \in N(x_i)$ 일 때,  $x_i$ 와  $x_j$  사이에서 10차원의 낮은 수준 관계 (low-level relation) 벡터인  $h_{ij}$ 가 도출된다.  $h_{ij}$ 는  $x_i$ 와  $x_j$

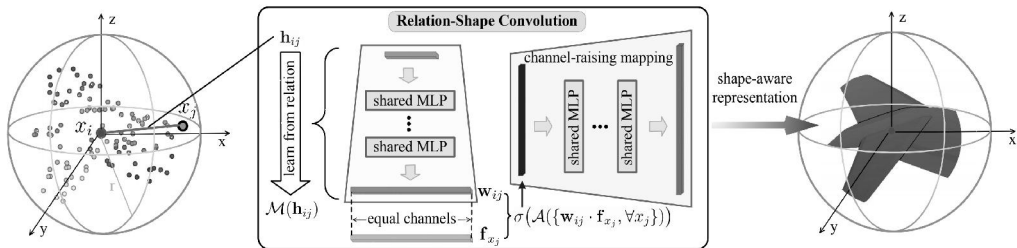


그림 8. RS-Conv의 구조 [17].  
Fig. 8. The Architecture of RS-Conv [17].

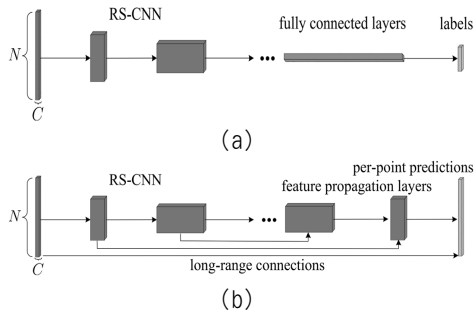


그림 9. (a) 분류 작업과 (b) 분할 작업에서 RS-CNN의 구조 [17].  $N$ 은 point의 개수,  $C$ 는 채널의 개수.  
Fig. 9. The Architectures of RS-CNN [17] for (a) Classification and (b) Segmentation.  $N$  is the number of points and  $C$  is the number of channel.

사이의 유클리드 거리,  $x_i - x_j$ ,  $x_i$ ,  $x_j$ 로 구성된다. 이후  $h_{ij}$ 는 MLP를 거쳐서 높은 수준 관계 (high-level relation)를 내포하는 data-dependent weight  $w_{ij}$ 로 변환된다.  $w_{ij}$ 를 convolution weight로 하여  $x_i$ 와  $N(x_i)$  사이의 관계를 종합하고 불규칙 point 집합의 공간적인 구조에 대한 inductive local representation을 얻을 수 있다. 이어서 channel raising mapping을 통해 채널 수가 증가되면서 더욱 강력한 형상 인식도를 가지게 된다.

이러한 RS-conv을 기본 연산으로 하여 RS-CNN은 맥락적 형태 인식 학습 (contextual shape-aware learning)을 이룰 수 있다. 그림 9는 RS-CNN의 전체 구조를 나타낸다. 분류와 분할 작업에서 공통적으로 RS-Conv연산이 수행되지만, 분류에서는 최종적으로 fully connected layer를 통해 전체 형상의 클래스에 대한 prediction을 도출하고, 분할에서는 feature propagation을 통해 upsampling을 수행하여 [25] point별 클래스 예측 (prediction)을 도출한다.

### III. 성능 비교 분석

본 장에서는 앞서 소개한 4가지 기법들 (benchmark methods) [14-17]의 3차원 객체 형상 분류 및 분할 성능을 비교한다. 비교에 사용되는 정확도 및 mean Intersection over Union (mIoU) 지표는 기존 논문에서 제공된 수치를 참고하였다.

#### 1. 3차원 객체 형상 분류

객체 형상 분류 작업은 주어진 3차원 형상이 속하는 클래스를 식별하는 것을 목표로 한다. 분류 실험에 사용된 데이터 세트는 3차원 형상 분류에서 가장 널리 사용되는 ModelNet40 [7]이다.

ModelNet40은 40개 카테고리의 12,311개 mesh 모델로 구성되어 있으며, 이는 훈련 샘플 9,843개와 테스트 샘플 2,468

표 1. ModelNet40 데이터 세트에서 top-1 분류 정확도.  
Table 1. Top-1 Classification Accuracy on ModelNet40 Datasets.

Method	ModelNet40
MeshCNN	-
MeshNet	91.9 %
PointNet	89.2 %
RS-CNN	93.6 %

개를 포함한다.

표 1은 ModelNet40에서 비교 방법들의 top-1분류 정확도를 보여준다. 해당 데이터 세트에서의 실험 결과가 제공되지 않은 MeshCNN을 제외한 세 가지 기법 중에서 RS-CNN이 가장 높은 분류 정확도를 보였다. 이는 반지름  $r$  이내의 모든 이웃 point와의 관계를 규정하여 PointNet의 지역적인 구조에서의 한계를 보완하고 이로부터 객체의 기하학적 형상에 대한 인식도를 높인 것이 성능 향상을 이끌었다고 볼 수 있다.

#### 2. 3차원 객체 형상 분할

객체 형상 분할 작업은 형상의 기본 구성 단위마다 부분 클래스 라벨을 추론하는 것이다. 분할 실험에 사용된 데이터 세트는 COSEG [28]와 ShapeNet-Part [29]이다. 그림 10과 그림 11은 각각COSEG 데이터 세트에서 MeshCNN의 분할 결과와 ShapeNet-Part 데이터 세트에서 PointNet의 분할 결과를 나타낸다. 그림 10과 그림 11에서 나타내고 있는 것과 같이 각 데이터 표현법의 기본 구성 단위 별로 클래스가 할당되는데, 예를 들어 mesh에서는 각 간선 또는 삼각형 면마다, point cloud에서는 각 point마다 컵 손잡이, 의자 다리 등과 같은 부분 클래스 예측이 할당된다.

COSEG 데이터 세트는 vases, chairs, 그리고 tele-aliens의 세 가지 세트로 이루어져 있으며 각각 200, 300, 400개의 모델로 구성되어 있다. 표 2는 COSEG 데이터 세트에서 분할 실험 결과의 정확도를 보여준다. Mesh를 구성하는 전체 간선들의 집합을  $E$ 이라고 할 때,  $e \in E$  인 간선  $e$ 에 대해 예측된 클래스 레이블이  $\ell_e$ 이고 ground-truth 레이블이  $\ell_e^*$ 이면 정확도는 다음과 같다.

$$acc = \frac{\sum_{e \in E} [\ell_e = \ell_e^*]}{|E|} \times 100, \quad (1)$$

여기서  $[\cdot]$ 는 Iverson Bracket을 나타낸다. COSEG 데이터 세트에서는 MeshCNN이 PointNet에 비해 높은 정확도를 보였다. 이는 형상의 국부적인 관계가 결과에 중요한 영향을 미치는 분할 작업의 특성으로 인한 결과라고 볼 수 있다. MeshCNN은 각 간선마다 인접 영역을 정의하고 이를 기반으로 mesh convolution과 mesh pooling 연산을 수행하

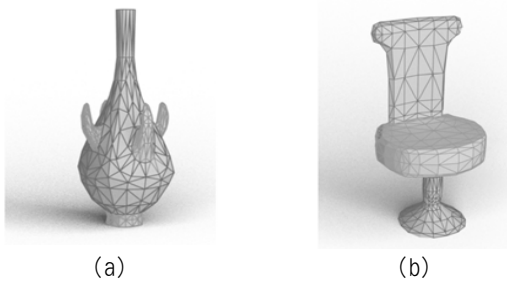


그림 10. COSEG 데이터 세트의 (a) Vase 와 (b) Chair 에서 MeshCNN의 분할 결과 [14].  
Fig. 10. The Segmentation Result of MeshCNN in (a) Vase and (b) Chair of the COSEG Datasets [14].



그림 11. ShapeNet-Part 데이터 세트의 (a) Mug, (b) Laptop, (c) Lamp 에서 PointNet의 분할 결과 [16].  
Fig. 11. The Segmentation Result of PointNet in (a) Mug, (b) Laptop, and (c) Lamp from the ShapeNet-Part Datasets [16].

기 때문에 PointNet에 비해 형상의 지역적인 정보를 효과적으로 이용할 수 있다. 또한 3차원 좌표만 입력으로 사용했던 PointNet에 비해 2면 각, 내각, 간선과 삼각형 수선의 길이 비와 같이 mesh 구조가 보유한 풍부한 기하학 정보를 입력으로 사용한 것이 성능을 향상시키는데 기여했다고 볼 수 있다.

ShapeNet-Part 데이터 세트는 16개 카테고리의 16,881개 모델로 구성되어 있다. 총 50개의 부분 클래스가 있으며 각 모델은 2~5개의 부분 클래스로 구성되어 있다. 표 3에서는 ShapeNet-Part 데이터 세트에서의 분할 정확도를 비교하는데, 이때 mIoU를 성능 평가 척도로 사용한다.

mIoU는 의미론적 분할 (semantic segmentation) 분야에서 가장 널리 사용되는 평가 지표로 IoU의 평균값이다. IoU의 정의는 다음과 같다.

$$IoU = \frac{|X \cap Y|}{|X \cup Y|}, \quad (2)$$

여기서  $X$ 는 예측된 분할이고  $Y$ 는 ground-truth 이다. 클래스당 IoU를 계산하고, 이를 평균 낸 mIoU 값을 통해 형상 분할 예측이 정답과 얼마나 일치하는지 판단할 수 있다.

표 2. COSEG 데이터 세트에서 분할 정확도.  
Table 2. Segmentation Accuracy on COSEG Datasets.

Method	COSEG		
	Vases	Chairs	Tele-Aliens
MeshCNN	97.27 %	99.63 %	97.56 %
MeshNet	-	-	-
PointNet	91.5 %	70.2 %	54.4 %
RS-CNN	-	-	-

표 3. ShapeNet-Part 데이터 세트에서 mIoU.  
Table 3. mIoU on ShapeNet-Part Datasets.

Method	ShapeNet-Part
MeshCNN	-
MeshNet	-
PointNet	83.7 %
RS-CNN	86.2 %

ShapeNet-Part 데이터 세트에서는 RS-CNN이 PointNet보다 우수한 분할 성능을 보였다.

#### IV. 결론

본 논문에서는 3차원 비유클리드 데이터인 mesh와 point cloud에서 분류와 분할을 위해 형상을 학습하는 딥러닝 기법들을 서술하고, 전통적 데이터 세트에 대한 정확도를 비교해 보았다.

Mesh 데이터에 내재한 형상 정보를 충분히 이용하여 학습하기 위해 MeshCNN에서는 간선을, MeshNet에서는 삼각형 면을 기본 단위로 하여 객체 형상의 기하 특징들을 추출했다. 이처럼 mesh에 잠재된 여러 기하학적 특징들을 효과적으로 추출하여 학습에 이용하고 형상을 추론하는 것이 mesh 기반 딥러닝 기법의 핵심이라고 할 수 있다.

Point cloud 데이터에서는 입력 point 집합의 변형 (순열, 회전, 이동)에 강건한 인공신경망을 제안했던 PointNet에 이어, RS-CNN에서 이웃 point간의 기하학적 관계를 정의하면서 이들의 상호작용에 주목했다. Point cloud 기반 딥러닝 기법에서는 국부적인 point들의 관계를 파악하여 형상의 지역적인 정보와 전역적인 정보를 균형 있게 활용하고, 입력 데이터의 다양한 변형에도 견고한 결과를 만들어내는 것이 핵심이다.

불규칙한 3차원 데이터를 직접 다루는 것은 여전히 까다롭고 복잡한 과정이다. 그러나 기존의 2차원 격자 (이미지) 기반 딥러닝 구조에서 벗어나 3차원 비유클리드 데이터의 불규칙성과 복잡성을 해결하고, 이들이 가진 풍부한 정보와 강력한 표현력을 최대한 활용할 수 있는 딥러닝 연구의 필요성이 더욱 증가할 것이다. 따라서 3차원 딥러닝 모델의 견고성을 향상시키고 다양한 3차원 데이터 표현을 목적에 맞게 사용하는 방향으로 발전시키는 것이 더욱 중요하다.

## References

- [1] A. Krizhevsky, I. Sutskever, G.E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," In *Advances in neural information processing systems*, pp. 1097 - 1105, 2012.
- [2] J. Long, E. Shelhamer, T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431 - 3440, 2015.
- [3] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, "Overfeat: Integrated Recognition, Localization and Detection Using Convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.
- [4] J.W.H. Tangelder, R.C. Veltkamp, "A Survey of Content Based 3D Shape Retrieval Methods," *Multimedia tools and applications*, Vol. 39, No. 3, pp. 441-471, 2008.
- [5] K. Michael, T. Funkhouser, S. Rusinkiewicz, "Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors," *Symposium on geometry processing*, Vol. 6, pp. 156-164, 2003.
- [6] Z.B. Liu, S.H. Bu, K. Zhou, S.M. Gao, J.W. Han, J. Wu, "A Survey on Partial Retrieval of 3D Shapes," *Journal of Computer Science and Technology*, Vol. 28, No. 5, pp. 836 - 851, 2013.
- [7] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, "3D ShapeNets: A Deep Representation for Volumetric Shapes," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1912 - 1920, 2015.
- [8] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, "Multi-View Convolutional Neural Networks for 3D Shape Recognition," In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 945 - 953, 2015.
- [9] R. Klokov, V. Lempitsky, "Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models," In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 863 - 872, 2017.
- [10] E. Ahmed, A. Saint, A. Shabayek, K. Cherenkova, R. Das, G. Gusev, D. Aouada, "A Survey on Deep Learning Advances on Different 3D Data Representations," *arXiv preprint arXiv:1808.01462*, 2018.
- [11] D. Maturana, S. Scherer, "Voxnet: A 3D Convolutional Neural Network for Real-time Object Recognition," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 922-928, 2015.
- [12] M. Gadelha, R. Wang, S. Maji, "Multiresolution Tree Networks for 3D Point Cloud Processing," In *Proceedings of the European Conference on Computer Vision*, pp. 103 - 118, 2018.
- [13] Y. Feng, Z. Zhang, X. Zhao, R. Ji, Y. Gao, "GVCNN: Group-view Convolutional Neural Networks for 3D Shape Recognition," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 264 - 272, 2018.
- [14] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, D. Cohenor, "MeshCNN: A Network with an Edge," *ACM Transactions on Graphics*, Vol. 38, pp. 1-12, 2019.
- [15] Y. Feng, Yi. Feng, H. You, X. Zhao, Y. Gao, "Meshnet: Mesh Neural Network for 3D Shape Representation," *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, No. 1, 2019.
- [16] C.R. Qi, H. Su, K. Mo, L.J. Guibas, "Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.
- [17] Y. Liu, B. Fan, S. Xiang, C. Pan, "Relation-shape Convolutional Neural Network for Point Cloud Analysis," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.
- [18] M. Jiang, Y. Wu, T. Zhao, Z. Zhao, "PointSIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation," *arXiv preprint arXiv:1807.00652*, 2018.
- [19] J. Li, B.M. Chen, G.H. Lee, "SO-Net: Self-Organizing Network for Point Cloud Analysis," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9397 - 9406, 2018.
- [20] X. Liu, Z. Han, Y.S. Liu, M. Zwicker, "Point2sequence: Learning the Shape Representation of 3D Point Clouds with an Attention-based Sequence to Sequence Network," *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, No. 1, 2019.
- [21] I. Kostrikov, Z. Jiang, D. Panozzo, D. Zorin, J. Bruna, "Surface Networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2540 - 2548, 2018.
- [22] Y. Xu, T. Fan, M. Xu, L. Zeng, Y. Qiao, "Spidernn: Deep Learning on Point Sets with Parameterized Convolutional Filters," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 87-102, 2018.
- [23] H. Hoppe, "View-dependent Refinement of Progressive Meshes," In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 189 - 198, 1997.
- [24] M. Jaderberg, K. Simonyan, A. Zisserman, "Spatial Transformer Networks," *Advances in neural information processing systems*, Vol. 28, pp. 2017-2025, 2015.
- [25] C.R. Qi, L. Yi, H. su, L.J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," In *NeurIPS*, pp. 5099 - 5108, 2017.
- [26] L. Landrieu, M. Simonovsky, "Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs," In



Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4558 - 4567, 2018.

- [27] Y. Qiang, Y. SUN, "Dynamic Graph cnn for Learning on Point Clouds," ACM Transactions On Graphics, Vol. 38, No. 5, 2019.
- [28] Y. Wang, S. Asafi, O.V. Kaick, H. Zhang, D. CohenOr, B Chen, "Active Co-analysis of a set of Shapes," ACM Transactions on Graphics, Vol. 31, No. 6, 2012.
- [29] L. Yi, V.G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, L. Guibas, "A Scalable Active Framework for Region Annotation in 3D Shape Collections," ACM Transactions on Graphics, Vol. 35, No. 6, 2016.

### Sung In Cho (조성인)



2010 Electronics Engineering from Sogang University, Seoul, Republic of Korea (B.S.)

2015 Electrical and Computer Engineering from Pohang University, Pohang, Republic of Korea (Ph. D)

2015~2017 Senior Researcher with LG Display

2017~2019 Electronic Engineering from Daegu University, Gyeongsan, Republic of Korea (Assistant Prof.)

2019~Multimedia Engineering, in Dongguk University (Assist. Prof.)

Field of Interests: Image Analysis and Enhancement, Video Processing, Computer Vision, and Deep-learning.

Email: csi2267@dongguk.edu

### Haeju Park (박해주)



2018~Multimedia Engineering, in Dongguk University (Integrated MS program.)

Field of Interests: image processing, geometric deep learning, 3d data processing

Email: haeju620@gmail.com