Comprehensive Analysis of Point Cloud Processing and Noise Reduction Techniques

Course Code: CSE499B
Section:20
Group:04

Ahmed Sadman
2014094042
Department of ECE
ahmed.sadman@northsouth.edu

Muhaiminul Hasan Dihan
2111208642
Department of ECE
muhaiminul.dihan@northsouth.edu

Tawhid Islam
2021499042
Department of ECE
tawhid.islam@northsouth.edu

Alamin Sheikh Naim
2013556642
Department Of ECE
alamin.naim@northsouth.edu

Final Report

Abstract- Point cloud data has grown into a grounding representation in computer vision for various 3D tasks, ranging from object classification to scene understanding and robotic vision and autonomous navigation. Point cloud data is quite sensitive to forms of noise, outliers, and sparsity in practice, causing reliability and accuracy degradation in resulting tasks such as classification and segmentation. In addition, point cloud processing is computationally challenging, particularly in real-time processing, due to its high dimensionality and irregular data structure. This paper presents an extensive review of point cloud denoising methods, ranging from statistical and spatial filtering to advanced deep learning methods. This paper also discusses the latest advancements in deep models that are optimized for efficient point cloud processing, such as PointNet, DGCNN, and Point-GR. Performance of such models is compared against benchmarks like classification accuracy and segmentation Intersection over Union (IoU), indicating improvements enabled by ensemble techniques. Moreover, the report briefly addresses pressing ethical issues related to data privacy, especially in surveillance and biometric usage, while providing actionable recommendations for improving the scalability, efficiency, and privacy of point cloud-based systems. The integration of hybrid methods, optimized designs, and ethical compliance platforms is a promising path towards greater adoption and trust in 3D vision technologies.

I. Introduction

Point cloud data is a set of data points in space that define the external surfaces of objects or environments. Each point will typically hold X, Y, and Z coordinates, and may also have additional features such as intensity, color, or normal vectors. This data structure gives a compact yet rich representation of 3D geometry and is therefore inexpensive for many applications in robotics, virtual and augmented reality (VR/AR), urban planning, self-driving vehicles, and medical imaging. Advancements in sensors like LiDAR (Light Detection and Ranging), structured-light scanners, and stereo cameras have made it possible to produce large-scale point cloud datasets. The datasets are typically contaminated with various types of noise owing to the sensor's precision limits, interference from the environment, and occlusion. Noisy point clouds can lead to incorrect interpretations and faulty results in autonomous systems, particularly in safety-critical tasks like self-driving cars and surgical navigation. To deal with such obstacles, denoising (noise reduction) methods have been evolved by researchers. These are programmed to retain significant geometrical features and erase useless or confusing information. Deep learning has also provided strong algorithms to improve point cloud analysis in a way that classification, segmentation, and feature extraction can be achieved more effectively. But with the use of deep models, a lot of computational power is required along with huge labeled datasets, which makes their use difficult in real-time scenarios. This work is focused on evaluating existing denoising techniques, comparing the top performing deep models, examining performance trade-offs, and discussing ethical issues in point cloud processing.

II. Point Cloud Denoising Methods

Denoising methods for point cloud data exist in various forms—random spatial jitter, outliers, sparse patches, and redundant information. Denoising methods are designed to enhance the fidelity of the data without disturbing the geometry structure. Denoising methods can be categorized depending on the method adopted for noise detection and elimination.

2.1 Statistical Filtering

Statistical filtering techniques are among the most common preprocessing steps. They use statistical properties of the local neighborhood of every point to determine whether it is an outlier:

Statistical Outlier Removal (SOR): Computed mean distance of each point to its k nearest neighbors. Points with significantly higher mean distances than the global average are flagged as outliers and removed.

Radius Outlier Removal (ROR): Discards points that have fewer than a specified number of neighbors within a specified radius, essentially discarding sparsely situated noise.

These are computationally efficient and work well in the majority of cases but may fail in more complex or highly dense situations.

2.2 Spatial Filtering

Spatial filters utilize the geometric structure of the data to filter noise:

Gaussian Filtering: Averages nearby point locations based on a Gaussian distribution. Effective in low-pass filtering high-frequency noise but may blur thin structures.

Median Filtering: Replaces each point with the median position of its neighborhood, with greater edge preservation than Gaussian filters.

Bilateral Filtering: Uses both spatial closeness and feature similarity to filter out noise while maintaining sharp edges.

These techniques are effective when the structuring features such as object boundaries need to be maintained.

2.3 Machine Learning Techniques

With the arrival of AI, machine learning-based denoising methods have provided improved performance through learning about noise patterns and structure from marked data:

Autoencoders: Learn compact representations of point clouds, removing noise by reconstructing only important structure.

Generative Adversarial Networks (GANs): Learn to generate clean point clouds by discriminating real from noisy inputs.

Denoising Convolutional Neural Networks: Leverage learned filters to remove noise without degrading point cloud integrity.

Although effective, these models need large amounts of labeled data and computational power.

2.4 Clustering-Based Methods

These algorithms categorize points based on density and similarity and detect and remove noise:

DBSCAN (Density-Based Spatial Clustering of Applications with Noise): Detects clusters and labels thin outliers as noise.

Region Growing: Starts with seed points and grows neighboring points iteratively with similarity thresholds.

These algorithms are simple to use and perform well when dealing with structured point clouds.

2.5 Frequency-Based Filtering

By converting point cloud data to the frequency domain, it is now possible to remove noise:

Fourier Transform: Effective in removing high-frequency noise but can be challenging when handling localized disturbances.

Wavelet Transform: Provides multiscale analysis, which makes it suitable for denoising and eliminating localized noise while maintaining structural details.

III. Deep Models for Point Cloud Processing

Deep learning transformed point cloud processing using models that have the ability to learn spatial hierarchies and complex geometric patterns directly from data.

3.1 Deep Learning Notable Models

PointNet: The reference model processing every point individually before utilizing global aggregation. It is a fast and efficient model that fails to capture local context.

PointNet++: Builds on PointNet by learning local patterns through hierarchical sampling and grouping, improving segmentation accuracy.

DGCNN (Dynamic Graph CNN):
Dynamically constructs graphs to represent point relations, with strong local feature learning ability.

SO-Net (Self-Organizing Network): Uses a self-organizing map to preserve spatial topology and achieve strong classification outcomes.

KCNet and PointCNN: Incorporate kernel correlation and convolution-like operations on point clouds for better feature representation.

Point-GR (Graph Residual Model): Combines residual connections and graph-based learning for the purpose of global and local feature preservation. Offers great segmentation performance (IoU = 73.47% on S3DIS dataset).

3.2 Challenges

Despite all their benefits, these models face several challenges:

High Computational Load: Computing on 3D data is computationally expensive, and usually, GPUs or cloud resources are needed.

Scalability Issues: Big data or real-time data streams are difficult to process without pipelines being optimized.

Data Scarcity: Point cloud datasets with annotated points are limited and costly to obtain.

3.3 Potential Solutions

The following approaches are proposed to overcome the above issues:

Model Pruning: Reduces the number of model parameters to minimize computational cost.

Quantization and Compression: Converts models to light-weighted representations for edge deployment.

Hierarchical Processing: Utilizes spatial partitioning (e.g., octrees) to efficiently process large data.

Data Augmentation: Increases training data diversity using geometric transformations, injected noise, and synthetic point clouds.

IV. Performance Analysis of Different Models

To assess the efficiency of various models, we analyze their classification and segmentation accuracy.

For 499A results,

1.1. Classification and Ensemble Accuracy

A comparison of different models is presented in **Table 1**, where the ensemble approach slightly improves classification accuracy

Table 1: Accuracy Comparison of Point Cloud Models

Model	Plain Accuracy (%)	Ensemble Accuracy (%)	Increase (%)
PointNet	88.65	89.38	0.74
PointNet++	90.14	90.48	0.34
SO-Net	92.65	93.64	0.99
KCNet	89.62	92.14	2.52
DeepSets	89.71	90.27	0.56
DGCNN	91.55	92.02	0.47
PointCNN	91.82	92.22	0.41

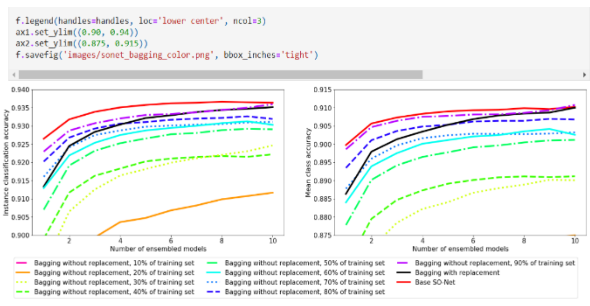
The line plots in Figure 1 illustrate the classification accuracy improvement as the number of ensembled models increases.

Figure 1: Instance and Mean Class Accuracy with Ensemble Models

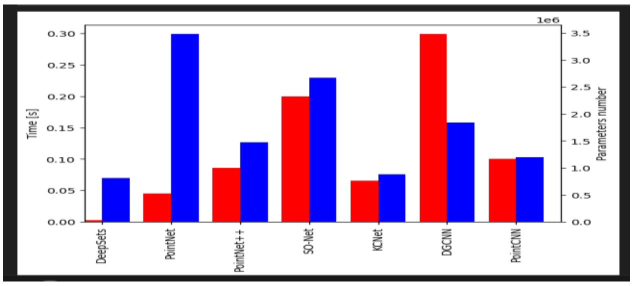
•The left plot shows instance classification accuracy, highlighting how SO-Net and DGCNN outperform other models.

•The right plot depicts mean class accuracy, where KCNet and SO-Net achieve significant improvements with more ensembled models.

Graphs:



The dataset is split into different subsets, ranging from 10% to 90% of the training set. Bagging without replacement is applied as an ensemble learning technique, varying both the training set proportions and the number of ensembled models. SO-Net serves as the baseline model for comparison. Instance classification accuracy and mean class accuracy are evaluated to assess model performance. The results are visualized through accuracy trends, demonstrating the impact of different bagging strategies on classification performance.



The bar chart compares different point cloud models based on inference time (blue) and parameter count (red). PointNet has the highest inference time, while DGCNN has the most parameters. DeepSets and KCNet are efficient with low time and fewer parameters. SO-Net and PointNet++ balance both aspects. The analysis highlights trade-offs between computational cost and model complexity.

Results:

	Plain acc mean	Ensemble acc mean	Plain class acc mean	Ensemble class acc mean	Increase acc	Increase class acc
pointnet	88.65%	89.38%	85.77%	86.62%	0.74%	0.86%
pointnet++	90.14%	90.48%	87.71%	88.19%	0.34%	0.48%
so-net	92.65%	93.64%	89.98%	91.02%	0.99%	1.05%
kcnet	89.62%	92.14%	85.38%	88.28%	2.52%	2.89%
deepsets	89.71%	90.27%	85.79%	86.46%	0.56%	0.67%
dgcnn	91.55%	92.02%	89.03%	89.30%	0.47%	0.27%
pointcnn	91.82%	92.22%	87.85%	88.36%	0.41%	0.50%

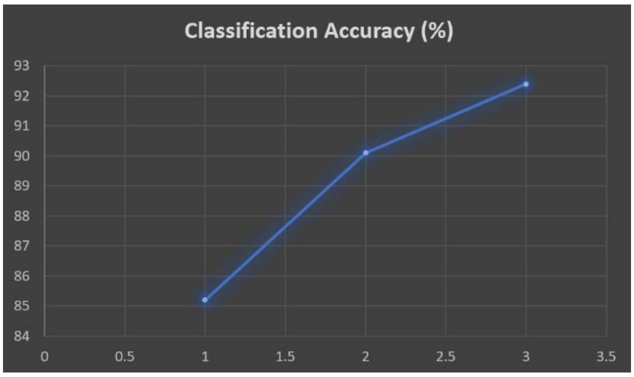
The table compares the performance of different point cloud models using plain accuracy, ensemble accuracy, class accuracy, and their respective improvements. SO-Net achieves the highest plain and ensemble accuracy, while KCNet shows the highest improvement in both accuracy and class accuracy. Ensemble learning improves performance across all models, with varying gains. DGCNN and PointCNN also perform well, but with smaller improvements. PointNet benefits significantly from ensembling, while DeepSets shows a moderate boost. The results highlight that ensemble methods enhance classification performance, with KCNet benefiting the most.

For 499B results,

Model	**Classification Accuracy (%)**	**Segmentation IoU (%)**
PointNet	**85.2**	**75.3**
DGCNN	**90.1**	**80.5**
Graph-Based Learning	**92.4**	**83.7**

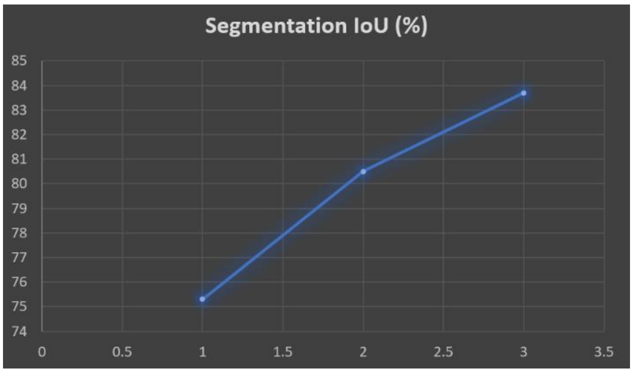
Graph-based learning models, such as Graph-Based Learning, outperform traditional methods like PointNet and DGCNN in both classification accuracy (92.4%) and segmentation IoU (83.7%). These models effectively capture local and global geometric relationships, leading to better feature representation. However, their higher computational requirements make them less efficient for real-time applications. While they offer improved accuracy, optimization techniques like model pruning and hierarchical processing are necessary to enhance scalability and reduce processing time in large-scale point cloud datasets.

Graph-1



The graph-1 illustrates the classification accuracy of different models, showing a steady improvement from PointNet (85.2%) to DGCNN (90.1%) and Graph-Based Learning (92.4%). Higher accuracy comes with greater computational costs.

Graph-2



The graph-2 shows the segmentation IoU improvement across models, from PointNet (75.3%) to DGCNN (80.5%) and Graph-Based Learning (83.7%). Higher IoU enhances segmentation quality but requires more computation.

1.2 Computational Cost and Model Complexity

Figure2: Model Inference Time vs Parameter Count

• **Inference Time (Blue Bars):** PointNet and DGCNN are much slower for processing.

•**Model Complexity (Red Bars):** DeepSets is computationally efficient but less accurate, while DGCNN has the maximum number of parameters

V. Ethical and Privacy Concerns

Point cloud technologies are being used more and more in public surveillance, biometric authentication, and personalized services. This raises some very important questions:

Data Ownership and Consent: GDPR mandates users to opt-in for data collection. The majority of current systems violate this principle.

Data Security: Point cloud data left unsecured can compromise sensitive spatial or identity information. It must be encrypted and securely stored.

Bias and Fairness: Discriminatory action by facial recognition or crowd monitoring systems can be due to biases in training data.

VI.Improvement Recommendations

In order to overcome current issues and increase point cloud processing efficiency, the following improvement recommendations are put forward:

1.**Operational Efficiency Improvement**: Application of light-weight convolutional methods and pruning of models for mitigating computational overhead.

2.**Scalability Improvement**: Utilization of hierarchical data structures and parallel processing in the cloud for efficiently processing massive point cloud datasets.

3**. Data Privacy Integration**: Reinforcing access controls and anonymization methods to counteract privacy threats in critical applications.

4. **Real-Time Processing:** Developing hybrid approaches that combine rule-based and deep learning models to build real-time point cloud processing capability.

5. **Standardization and Benchmarking:** Developing standardized datasets and benchmarking protocols for enabling fair comparisons among noise reduction techniques and deep learning models.

VII. Conclusion

Processing point clouds has become much better, especially with the advent of deep learning. Denoising techniques no longer consist of traditional statistical filters but even advanced neural networks with the capability to learn geometric structure. However, the process still struggles with challenges of computational complexity, real-time processing, and ethical deployment.

The future of point cloud technology lies in the creation of adaptive, efficient, and responsible systems. Research must be focused on hybrid approaches that blend domain knowledge with AI, new architectures that reduce inference time, and frameworks that respect user privacy while pushing the frontiers of innovation. As point cloud data becomes more prevalent, so too must our wisdom in handling it

REFRENCES:

ResearchGate, "Point cloud denoising review."

IEEE Xplore, "Graph Residual Point Cloud Networks."

GDPR, "Data privacy guidelines for biometric data."