LiDAR and Thermal Image Situational Awareness

Eyassu Mongalo, Megan Hu
Advisor: Mizanur Rahman Jewel

May 2025

Overview of what we learned

Intro - 1

- Papers read
  - PointNet
  - PointNet++
  - Point Transformer
- Programming Project
  - Point Transformer "TITLE"

PointNet

PointNet - 1

Paper on Point Sets for 3D Classification and Segmentation (spend this time talking about the intro/conclusion)

- Point Cloud properties
- Abstract View of Process
- PointNet architecture
- Analysis and Experiments

Point Cloud

PointNet - 2

Point clouds have many interesting properties as a set of (x, y, z) coordinates

- **Unordered** - sets need to be invariant under N! permutations
- **Interactions among points** - neighboring points form a meaningful subset
- **Invariance under transformations** - segmentation and category should remain unchanged

PointNet architecture

PointNet - 3

There are 3 main factors to PointNet's architecture

- **Maxpooling** - Symmetric function for unordered input
- **Local/Global feature combination** - Information Aggregation
- **Joint Alignment Network** - Alignment of input points and features
- (Include image of the archtechture lol)

Why Maxpool?

PointNet - 4

3 Possible Methods for a way to work with a function S.T. it's invariant to permutations

- **Order** - Find a way to canonically order set
  - If possible requires an Bijection to 1D
- **Train an RNN** - Treat Input as a Sequence train
  - Impossible to totally omit order in a RNN
- **Simple Abelian Function** - Find a simplee symetric function
  - By commutative property such a function is invariant to permutations
  - Maxpooling has shown to have the best results out of the abelian functions

Maxpool's purpose

PointNet - 5

Maxpool attempts to approximate a general function

$$f(\{x_1, \dots, x_n\}) \approx g(h(x_1), \dots, h(x_n))$$

where $f : 2^{\mathbb{R}^N} \to \mathbb{R}$, $h : \mathbb{R}^N \to \mathbb{R}^K$, and

$$g : \underbrace{\mathbb{R}^K \times \cdots \times \mathbb{R}^K}_{n} \to \mathbb{R}$$

is a symmetric function.

- **g(x)** - Composition of single variable function and a max pooling function
- **h(x)** - Function that gets approximated by a multilayer perceptron

different h(x) can be used to approximate different f(x).

Local/Global feature Aggregation

PointNet - 6

The previous output for the previous slides is a vector

$$[f_1, ..., f_K]$$

- **Global Signature** - this is the signature of the input set
  - Easily able to train SVM or MLP classifier with this
- **Point segmentation** - A combination of both local and global features
  - For point segmentation we must feed the global features back
  - This is done by concatenating point features with its global features
  - Now it can extract new features based on the combined ones, this time aware of both local and global information

Joint Alignment Network

PointNet - 7

How to train model to recognize structure independant of orientation

- **Mini Network - JAN** is a Mini network designed to predict an affine transformation matrix
  - Mini Network resembles larger one
  - The transformed matrix gets constraineed to be close to the orthogonal one

$$L_{reg} = ||I - AA^T||_F^2$$

Analysis and Experiment

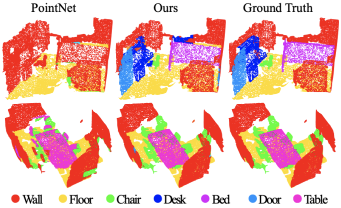
PointNet - 8

- **Universal Approximation** - Given enough neurons at the max pooling layer f can be arbitrarily approximated by our network
- **Bottleneck dimension** - small corruptions or extra noise points in the input set are not likely to change the output of our network
- **Experiments** - Shown to be better than state of the art

PointNet++ Motivation

PointNet++ - 1

- **PointNet** was revolutionary in handling unordered 3D point clouds directly.
- But it cannot capture *local geometric features* (e.g., edges, corners).
- **PointNet++** extends PointNet with a hierarchical framework to model *local structures*.
- Inspired by how CNNs process images through local patches.



Hierarchical Feature Learning

PointNet++ - 2

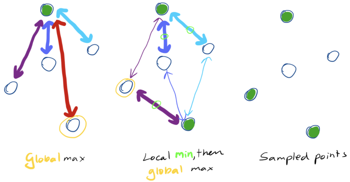


Illustration of local grouping & feature abstraction

How PointNet++ learns structure:

- **Local regions** are defined using Euclidean distance.
- Region centers chosen by **Farthest Point Sampling (FPS)**.
- PointNet applied to each region to extract features.
- Features grouped recursively into a hierarchy.
- Learns both *fine details* and *global context*.

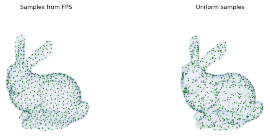
Hierarchical Set Abstraction

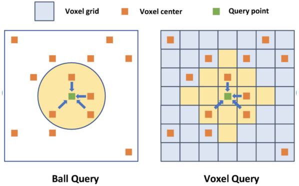
PointNet++ - 3

Each abstraction layer in PointNet++ contains:

- **Sampling layer**: Selects well-distributed centroids using *Farthest Point Sampling (FPS)*.
- **Grouping layer**: Forms local neighborhoods via *Ball Query* or *k-NN*.
- **PointNet layer**: Learns zone-level features using *relative coordinates*.

These layers are stacked to build a robust feature hierarchy.



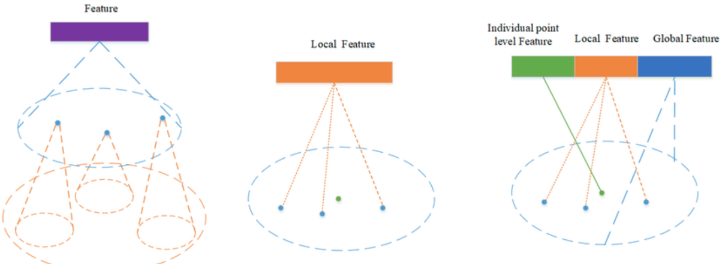


Density Adaptation

PointNet++ - 4

To handle **non-uniform sampling density**, PointNet++ introduces two strategies:

- **Multi-Scale Grouping (MSG)**: Extracts features at multiple radii and uses input dropout during training to improve robustness.
- **Multi-Resolution Grouping (MRG)**: Combines raw and abstracted features efficiently, adjusting based on local point density.

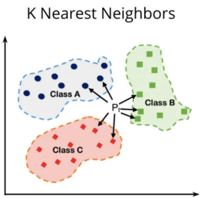


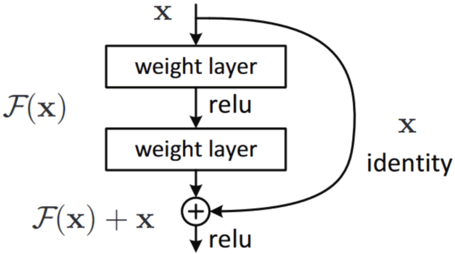
Feature Propagation for Segmentation

PointNet++ - 5

To restore point-level features for segmentation, PointNet++ combines three techniques:

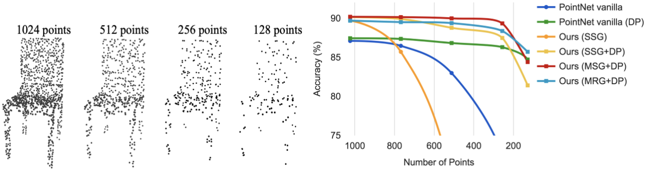
- **Interpolation**: Uses inverse distance weighted k-NN to estimate features at non-sampled points.
- **Skip connections**: Brings in earlier-layer features to preserve fine-grained context.
- **Unit PointNet**: Applies shared MLPs to refine each point's feature individually.





Experiments

PointNet++ - 6



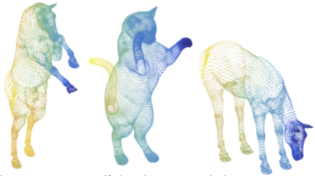
Evaluation overview:

- Tested on **MNIST** and **ModelNet40** for classification.
- Tested on **SHREC15** and **ScanNet** for segmentation and shape analysis.
- Shows robustness to point dropout — accuracy remains high even when reducing test points from **1024** to **256**.
- Thanks to **multi-scale (MSG)** and **multi-resolution (MRG)** strategies, PointNet++ handles sparse data effectively.
- Outperforms voxel-based baselines by avoiding quantization and directly learning from raw point clouds.

Non-Euclidean Classification

PointNet++ - 7

- On **SHREC15**, PointNet++ uses **geodesic distance** instead of Euclidean distance.
- Uses intrinsic features: **WKS**, **HKS**, and **multi-scale Gaussian curvature**.
- **Geodesic neighborhoods** preserve surface structure even under non-rigid deformations.
- Greatly outperforms XYZ + Euclidean baselines.



(a) Horse (b) Cat (c) Horse

Figure 7: An example of non-rigid shape classification.

Plans for Next Week

Focus Areas

- *More papers* – Read 2+ papers on Thermal Transformers
- *Mini Project #3* – Complete a short PyTorch Project on VisionTransformer
- *Thermal* – Deepen understanding of feature extraction from sensor data.

Weekly Flow

Read $\rightarrow$ *Test* $\rightarrow$ *Build* $\rightarrow$ *Reflect*

Works Cited I

Closing - 2

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). *PointNet: Deep learning on point sets for 3D classification and segmentation*. https://arxiv.org/pdf/1612.00593. Accessed 29 May 2025.

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). *PointNet++: Deep hierarchical feature learning on point sets in a metric space*. https://arxiv.org/pdf/1706.02413. Accessed 29 May 2025.

Zhao, H., Jiang, L., Jia, J., Torr, P. H. S., & Koltun, V. *Point Transformer*. https://arxiv.org/pdf/2012.09164. Accessed 29 May 2025.

Works Cited II

Closing - 2

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is All You Need*. https://arxiv.org/pdf/1706.03762. Accessed 29 May 2025.

Google Colab. (n.d.). *Neural Networks*. 3Blue1Brown, YouTube. https://colab.research.google.com/drive/18-r47vgJSdtQkfIzKkadfpQtEpEf0Y9Q. Accessed 29 May 2025.