Course Title: Environmental Data Analytics.
Degree Program: Masters in data science.
Instructor: Mohammad Mahdi Rajabi

Point Clouds

3D Data in Environmental Analytics

The environment is inherently three-dimensional. The land surface includes an elevation component (

topography), water bodies have depth, and the atmosphere varies vertically with elevation, influencing phenomena like temperature, pressure, and air quality. These spatial dimensions are critical for understanding and analyzing environmental processes, as they directly affect interactions between ecosystems, human activities, and natural phenomena. For instance, flood modeling requires elevation data to determine water flow paths, while studying vegetation structures or atmospheric dynamics demands accurate 3D representations. Thus, 3D data naturally aligns with environmental systems, making it indispensable for accurate and actionable insights.

Common Types of 3D Data in Environmental Analytics include Point Clouds, 3D Meshes. TINs, Digital Elevation Models (DEMs), 3D Volumetric Data (Voxels) and Cross-Sections and Profiles.

What are Point Clouds?

Point cloud data are collections of points in 3D space that represent the surface geometry of objects or environments. Each point in the dataset has spatial coordinates (x,y,z), and it may also include additional attributes such as color, intensity, reflectance, or classification labels.

Point clouds are among the most important types of 3D data in environmental data analytics because they serve as the basis for generating other 3D data types, such as DEMs, meshes, and 3D models, partly because widely used sensors like LiDARs naturally generate point cloud data as their primary output. Additionally, point clouds contain so much detail that other formats can be derived from them, whereas the reverse is not always true; for example, a DEM or mesh cannot be used to reconstruct a point cloud with the same richness of detail and attributes. Furthermore, their raw, unstructured nature makes them highly adaptable and directly compatible with many statistical and ML algorithms designed for spatial and geometric data.

Point clouds are not merely spatial data composed of points; they represent continuous 3D surfaces or volumes through a dense collection of points, often used to capture the geometry of objects, terrain, or environments in high detail. In contrast, spatial data containing discrete points, which represent specific locations or events in space—such as city centers, sensor locations, or occurrences of phenomena—are not considered point clouds.

Characteristics of Point Clouds

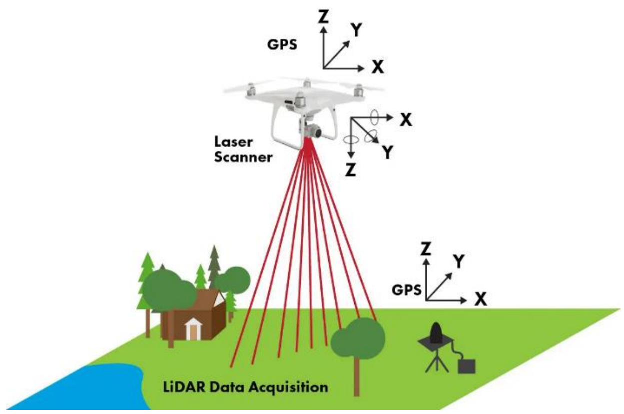
Point cloud have a set of rather unique characteristics:

- Unlike grids or meshes, point clouds do not have a predefined structure or connectivity between points. Each point exists independently in 3D space without a fixed relationship to its neighbors.
- Point clouds are inherently unordered, meaning there is no inherent sequence to the points, so the order of points can be changed, and the point cloud will not change.
- Point clouds don't involve any implicit assumptions about the underlying geometry, and they can represent fine-grained features, making them invaluable for applications requiring high-resolution spatial data.
- When complex, large domains are captured in a fine-grained way by point clouds, the resulting dataset could easily become huge. So working with point clouds may involve working with big data.
- Point clouds are not inherently high-dimensional since the only mandatory parts are the 3 dimensions x,y,z but they can become high-dimensional if they involve many features.
- Point clouds are often sparse and unevenly distributed, with some regions densely sampled and others under-sampled. Sparsity can lead to difficulty in learning meaningful patterns.
- Point clouds often contain noise and outliers due to sensor inaccuracies, environmental conditions, or reflections. Furthermore, occlusions or sensor limitations can result in missing points or incomplete regions in a point cloud. Another key issue is that in dynamic point clouds (e.g., moving objects or time-series scans), the number of points and their positions vary over time. All of these characteristics create complications in ML on point clouds.

How Are Point Clouds Generated?

1) LiDAR (Light Detection and Ranging):

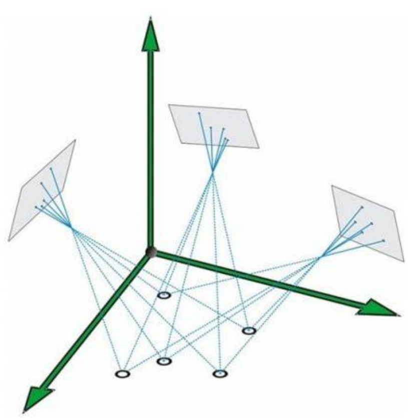
A laser scanner emits pulses of light and records the time taken to return after hitting a surface (known as the "time of flight"). Using the time of flight and the speed of light, the LiDAR system calculates the distance to the surface. The LiDAR sensor then determines the 3D position of each point based on the calculated distance and the the angle at which the laser pulse was emitted (using internal sensors to track the LiDAR system's orientation and position). This gives the position in a spherical coordinate system which is then used to compute the x,y,z coordinates of the point in a 3D Cartesian coordinate system.



LiDAR scanning.

2) Photogrammetry

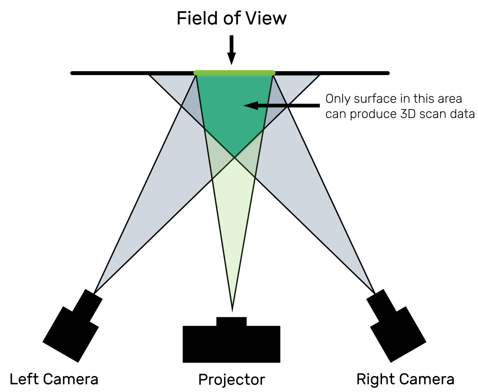
Unlike LiDAR, which directly measures points in 3D space using laser pulses, photogrammetry computes 3D points indirectly by analyzing and triangulating corresponding features from multiple overlapping images. Photogrammetry relies on identifying and matching points in 2D overlapping images of the object or scene from different angles and uses geometric principles (such as perspective projection and camera pose estimation) to reconstruct their 3D coordinates.



Schematic of how Photogrammetry works.

3) 3D structured light scanners

3D structured light scanners create point clouds by projecting a known pattern of light, such as stripes or grids, onto an object's surface and capturing the deformed pattern with one or more cameras. The deformation is analyzed to determine how the object's surface alters the light, and triangulation is used to calculate the 3D coordinates of points on the surface. This process generates a dense point cloud representing the object's geometry. Structured light scanners are highly precise, fast, and suitable for capturing fine details, making them ideal for applications like quality inspection, and 3D modeling.



3D structured light scanning

4) Synthetic Sources:

Point clouds can also be simulated using computational models or derived from existing datasets like meshes or voxels.

Platforms for Point Cloud-based Observations

1) Satellite-Based Point Clouds

Satellite missions equipped with LiDAR or photogrammetry capabilities can produce large-scale point cloud data. These are particularly useful for broad, regional-scale analyses.

Examples of Satellite Missions:

- **ICESat-2 (NASA)**: Uses the Advanced Topographic Laser Altimeter System (ATLAS) to measure land elevation, forest canopy height, and ice sheet thickness.

- **TanDEM-X (DLR)**: Generates a global digital elevation model (DEM) through radar interferometry, which can be converted into point clouds.
- **GEDI (Global Ecosystem Dynamics Investigation, NASA)**: Mounted on the International Space Station, it provides high-resolution LiDAR data for forest structure analysis.

Use cases include Regional Ecosystem Monitoring (Assessing forest biomass, canopy height, and carbon storage on a continental scale), Topographic Mapping (Generating global DEMs to model terrain, watersheds, and coastal processes), and Cryosphere Studies (Measuring ice sheet dynamics and snow thickness to monitor climate change).

2) Airborne Point Clouds (Planes)

Airborne LiDAR systems mounted on planes provide higher resolution and more flexible data collection than satellites. They are widely used for detailed mapping of smaller areas.

Use cases include Urban Planning (Creating high-resolution 3D city models for infrastructure development and zoning), Flood Risk Mapping (Generating detailed terrain data for hydrological modeling and floodplain delineation), and Forest Management (Assessing tree height, species distribution, and forest density at a regional scale).

3) Drone-Based Point Clouds

Drones equipped with LiDAR or photogrammetry cameras capture extremely high-resolution point clouds and are suitable for small to medium-scale areas with complex terrains.

Use cases include Agriculture (Monitoring crop health and generating terrain models for precision farming), Coastal Monitoring (Analyzing erosion and sediment transport in nearshore environments)., and Archaeological Studies (Creating 3D models of historical sites for preservation and study).

4) Ground-Based Point Clouds

Ground-based systems, including handheld and vehicle-mounted LiDAR scanners, are used for capturing highly detailed point clouds of localized environments.

Common Types of Devices:

- **Handheld Devices** like mobile scanners for indoor and outdoor use.
- **Vehicle-Mounted LiDAR**, commonly used in autonomous vehicles or mapping missions.

Use Cases include Infrastructure Assessment (Inspecting roads, bridges, and buildings for structural integrity), Urban Street Mapping (Generating high-detail 3D maps for smart cities and autonomous navigation), and Ecological Surveys (Measuring vegetation in urban or forested areas with extreme precision).

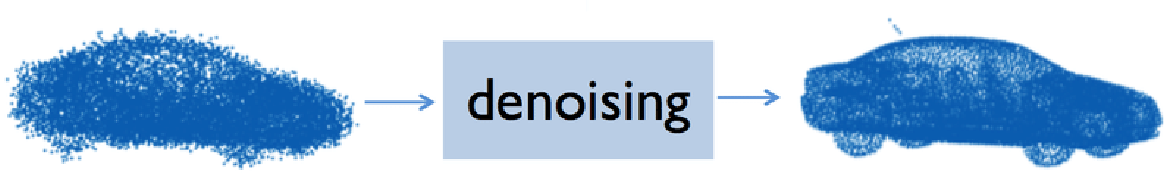
Preprocessing of Point Clouds

Analyzing point cloud data typically requires several preprocessing steps to ensure usability. This includes:

Denoising

Point cloud data often contains noise, which refers to small, random inaccuracies or distortions that typically affect many points within the dataset. While noise can slightly distort the surface or attributes of the data, it usually does not disrupt the overall structure. It commonly arises from errors or inaccuracies introduced during the data acquisition process, such as sensor imperfections, environmental conditions, or object movement. This noise can affect **coordinates** (spatial noise), causing points to deviate from their true locations, or **features** (attribute noise), such as intensity values in LiDAR scans varying due to inconsistent surface reflectivity. In some cases, both spatial and attribute noise are present at the same time.

Denoising is the process of correcting spatial and/or attribute noise in a point cloud without removing the data points. It aims to adjust noisy coordinates or features to align more closely with their true values, ensuring the dataset's integrity and structure are preserved for accurate analysis and processing.



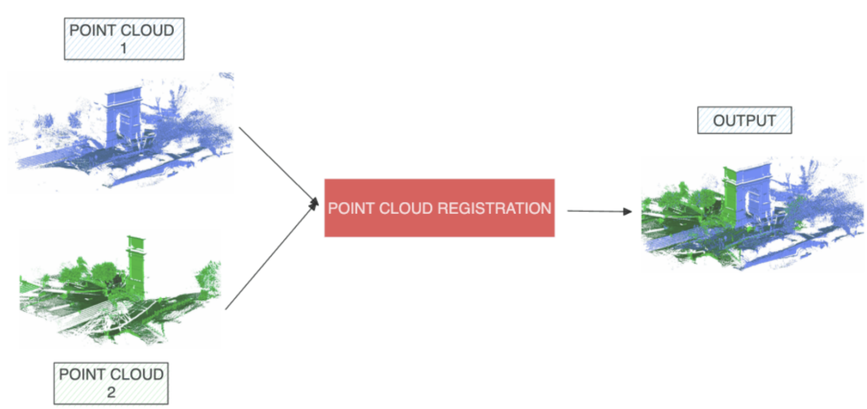
Denoising a point cloud.

Removing Outliers and Invalid Points

Outliers are individual points or small clusters of points that deviate ***significantly*** from the general data distribution, often appearing as isolated or anomalous points far from the main dataset. These points typically result from sensor errors, occlusions, temporary obstructions, or artifacts during data acquisition, and they can disrupt analyses by skewing calculations or producing misleading results. ***Outlier detection*** involves identifying such points using techniques like statistical outlier removal (based on distance thresholds or neighborhood density), radius-based filtering (eliminating points with insufficient neighbors within a radius), or clustering algorithms (detecting points that do not belong to major clusters). Once identified, outliers can either be removed or flagged for further investigation, ensuring the integrity and reliability of the dataset for analysis or modeling.

Alignment and registration of point clouds

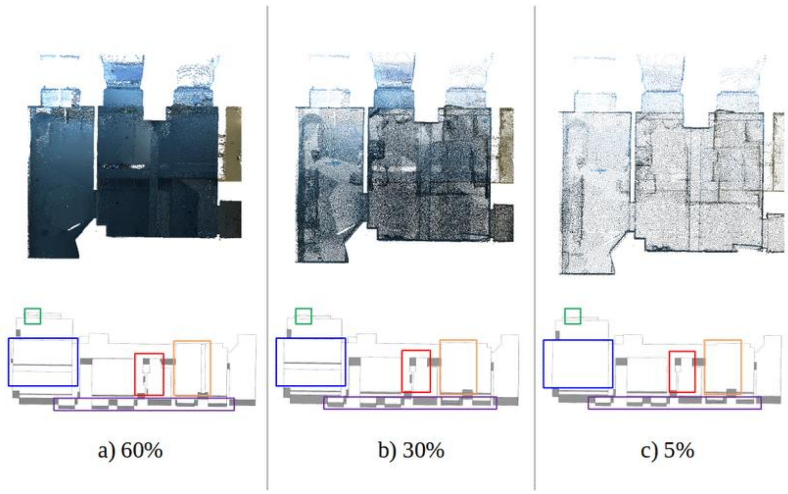
Alignment and registration of point clouds involve combining multiple point clouds into a single, unified coordinate system, ensuring proper alignment for analysis. This process is essential for various tasks, such as merging scans captured from different viewpoints, aligning point clouds from different times to detect changes, integrating data from different sensors (e.g., LiDAR and photogrammetry), combining multiple smaller scans to create a complete and continuous 3D model, and aligning point clouds with GIS data or maps for spatial consistency. Registration starts by roughly aligning the point clouds using methods that match recognizable features. Then, it fine-tunes the alignment using techniques like Iterative Closest Point (ICP), which adjusts the point clouds to minimize gaps or mismatches. Depending on the data, this may involve simple movements like shifting or rotating the clouds, or more complex adjustments.



Point cloud registration.

Downsampling

Point cloud downsampling is the process of reducing the number of points in a dataset while preserving its overall structure and essential details. It is necessary because raw point clouds often contain millions of points, which can be computationally expensive to process and store. Downsampling improves efficiency by reducing the data size, making it faster to analyze and easier to visualize without compromising significant features. Techniques like voxel grid downsampling, uniform sampling, and random sampling are commonly used to select representative points while maintaining the point cloud's accuracy and usability for tasks like modeling, machine learning, or visualization.



Various levels of downsampling applied to a point cloud.

Normalization and scaling

Point cloud normalization and scaling involve transforming the data to fit within a consistent coordinate system or range, ensuring uniformity and compatibility for analysis or machine learning. Normalization typically centers the point cloud by shifting its centroid to the origin (0, 0, 0), while scaling adjusts its size to fit within a unit sphere, cube, or other predefined range. This process is crucial because raw point clouds often vary in size, orientation, or units due to differences in acquisition methods. Normalization and scaling standardize the data, improving numerical stability in computations and enabling consistent input for algorithms, such as machine learning models or 3D processing pipelines.

Feature Engineering for Point Cloud Data

Feature engineering for point clouds is generally similar to other types of data in that it involves extracting, transforming, and creating meaningful features to improve model performance. Like tabular, image, or time-series data, point cloud feature engineering requires techniques such as normalization, dimensionality reduction, and statistical summarization. However, point clouds have unique characteristics that introduce specialized feature types and methods.

The unique features of point clouds can be categorized into three groups: **Local Features**, **Aggregated Features**, and **Attribute-Based Features**, based on their scope and the type of information they provide.

1) **Local Features** focus on the immediate geometry and spatial relationships around a point, such as normals, curvature, and point density, often derived from a point's local neighborhood. These features capture fine-grained, localized properties of the surface or environment.

   An example of local features is **point density**, which quantifies the number of neighboring points within a specified radius of each point. The density at a given point is calculated by counting the points within its defined neighborhood (e.g., within a fixed radius). This calculation is performed independently for each point in the point cloud, resulting in a **per-point density value**.

2) **Aggregated Features** summarize information across larger regions or the entire point cloud, providing broader context. Examples include statistical metrics like mean and variance, global shape descriptors, or multi-scale features that analyze patterns at varying levels of granularity.

   Examples of aggregated features include:

   - **Bounding Box**: A bounding box around the point cloud describes its approximate shape and dimensions, such as height, width, and depth, providing a concise representation of the overall spatial extent.
   - **Mean Coordinate**: The mean of all point coordinates in the point cloud represents the center or approximate location of the point cloud in 3D space.
   - **Variance**: The variance of the point coordinates quantifies how widely the points are spread, offering insight into the overall size or volume of the point cloud.

3) **Attribute-Based Features** are derived from additional data associated with points, such as color, intensity, timestamps, or classifications. These features are often sensor- or application-specific and add richness to the dataset by incorporating non-geometric information.

   For example, instead of using raw RGB values of the point cloud, features like **color histograms** or **dominant color clusters** can be extracted (as part of feature extraction) to summarize the distribution of colors in a region. For example, extracting a color histogram for points in a small neighborhood can help distinguish vegetation (mostly green) from buildings (gray or white).

Fusing point clouds with Other Data Modalities

Fusing point clouds with different data modalities is sometimes required to enhance the richness and usability of the dataset by combining the 3D spatial structure of the point cloud with complementary information from other sources. For instance, remote sensing imagery can add color or spectral features to a point cloud, GIS data can provide geospatial context such as land use or elevation, and sensor readings can contribute physical properties like temperature or pressure.

Different approaches can be used to combine point clouds with other data modalities, but a common approach is to use point clouds as the primary format. Point clouds inherently encode 3D spatial information, making them ideal for mapping additional data modalities, such as imagery, GIS data, sensor readings, or time-series data, as features onto individual points. This allows other data types to be attached without altering the 3D structure of the point cloud.

For example, fusing point clouds with remote sensing imagery begins with aligning both datasets to the same geographic coordinate system, typically achieved using GPS data, ground control points (GCPs), or sensor metadata for georeferencing. Once aligned, the spatial location of each point in the point cloud is used to identify its corresponding pixel in the remote sensing image. Attributes from the image, such as RGB values or infrared reflectance, are then extracted and assigned to the respective points in the cloud as extra features.

Machine Learning on Point Cloud

A point cloud is essentially a collection (or set) of points in 3D space. Sets inherently do not have an order, meaning that shuffling the points does not alter the underlying geometry or structure they represent. Therefore, machine learning (ML) models applied to point clouds should exhibit **permutation invariance**, meaning that the output of the model must not depend on the order in which the points are presented in the input data.

In other words, if you shuffle the points in a point cloud, the model should still produce the same result (e.g., the same classification label or segmentation output). Some ML models, however, are not inherently suitable for point cloud data due to this requirement. For instance, **Convolutional Neural Networks (CNNs)** in their standard form are not permutation-invariant. CNNs process data sequentially by convolving over fixed neighborhoods in the input, assuming a structured grid. If the order of input points changes, the neighborhoods and feature maps will also change, leading to inconsistent outputs. This makes CNNs unsuitable for direct application to point clouds.

To address this limitation, specialized deep learning architectures, such as **PointNet**, have been developed specifically for point clouds. PointNet aggregates features from all points using a symmetric operation, such as a max-pooling layer, ensuring that the order of the points does not affect the final global feature representation.

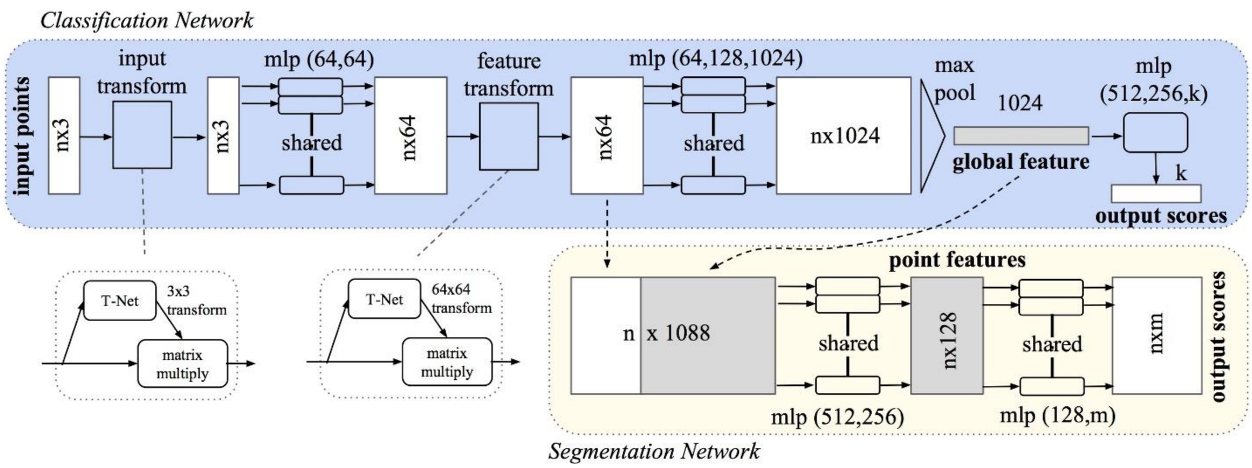
Another approach is to use **attention-based deep neural networks (DNNs)**, where attention mechanisms dynamically weigh the importance of different points. These mechanisms can be designed to focus on the **relative spatial relationships** between points rather than their order, ensuring permutation invariance while capturing meaningful geometric structures.

CNNs have also been adapted to accommodate point clouds. For example, **Dynamic Graph CNNs (DGCNN)** build a graph over the points based on their spatial relationships and apply **graph convolutions** instead of traditional convolutions. Graph convolutions operate on the graph structure and are inherently **invariant to the permutation of points**, ensuring that the order of points does not affect the output.

The PointNet Architecture

PointNet, introduced in 2017, is a pioneering deep learning architecture specifically designed to process 3D point cloud data. Unlike traditional methods that rely on transforming point clouds into structured formats (e.g., grids or meshes), **PointNet directly operates on raw point cloud data**. This groundbreaking approach opened new possibilities for efficiently handling 3D geometric data and set a foundation for subsequent architectures like PointNet++, DGCNN, and others, which build upon its concepts to address limitations such as capturing local geometric features and scalability for large datasets.

PointNet processes point cloud data by representing each point as a vector of coordinates (x,y,z), often with additional features like intensity or color (r,g,b). A point cloud with N points is represented as a set of N×d vectors, where d is the dimensionality of each point. To extract meaningful features, PointNet first applies shared Multi-Layer Perceptrons (MLPs) independently to each point. These MLPs transform the raw input points into high-dimensional feature vectors while ensuring permutation invariance through weight sharing. This produces a feature matrix of size N×K, where K is the dimensionality of the learned features for each point. To address the unordered nature of point clouds, PointNet aggregates the N×K feature matrix into a global feature vector using a symmetric function, such as **max pooling**. The result is a K-dimensional global feature vector summarizing the entire point cloud. Depending on the task, the architecture branches: for example for **classification**, the global feature vector is passed through fully connected layers to predict a class label for the entire point cloud, while for **segmentation**, both the global and individual point features are combined (via concatenation) to predict a class label for each point.



The PointNet architecture for Point Clouds.