Digital Signal Processing on LiDAR data

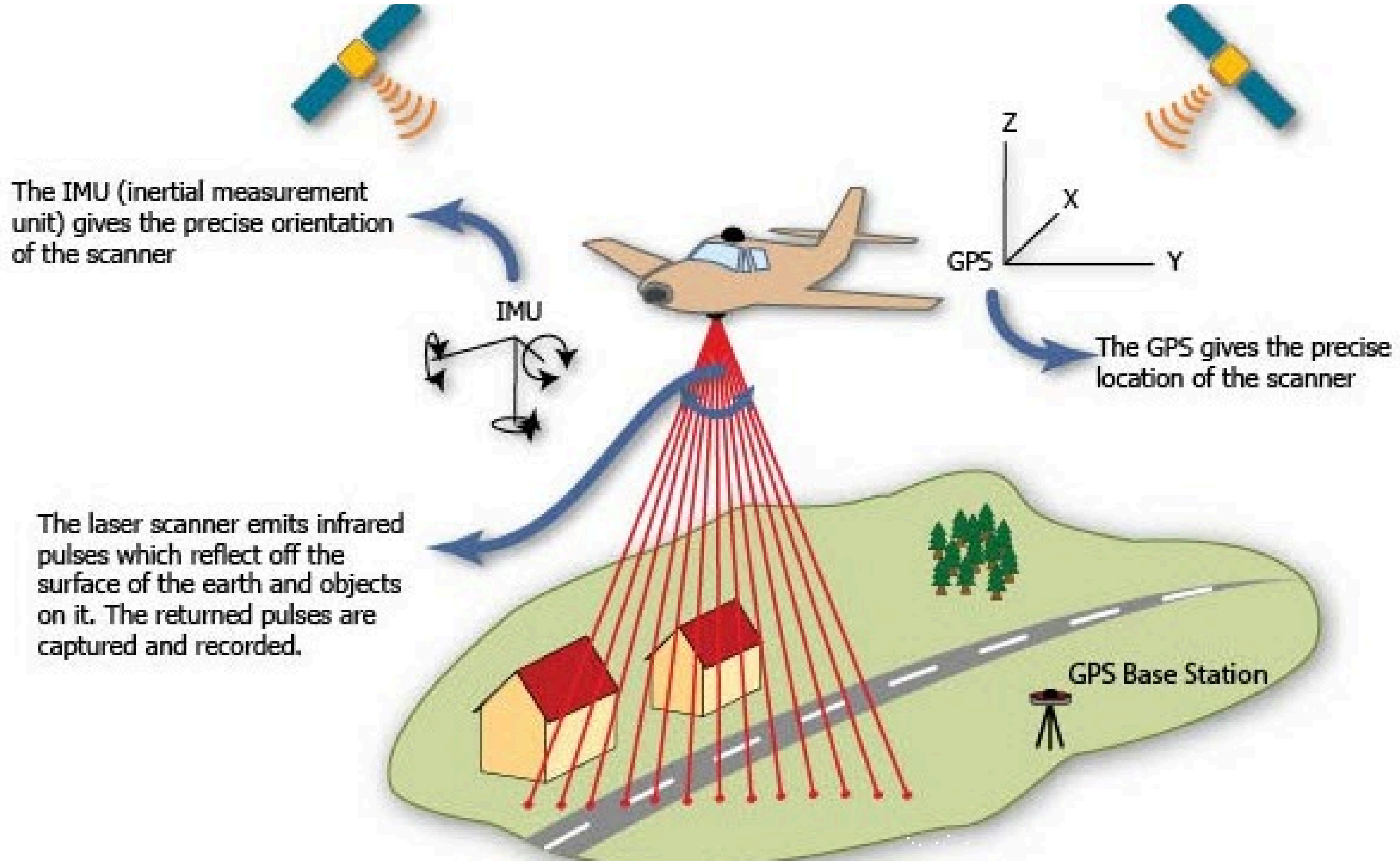
"Exploring the Depths with Precision"

PRESENTED BY:

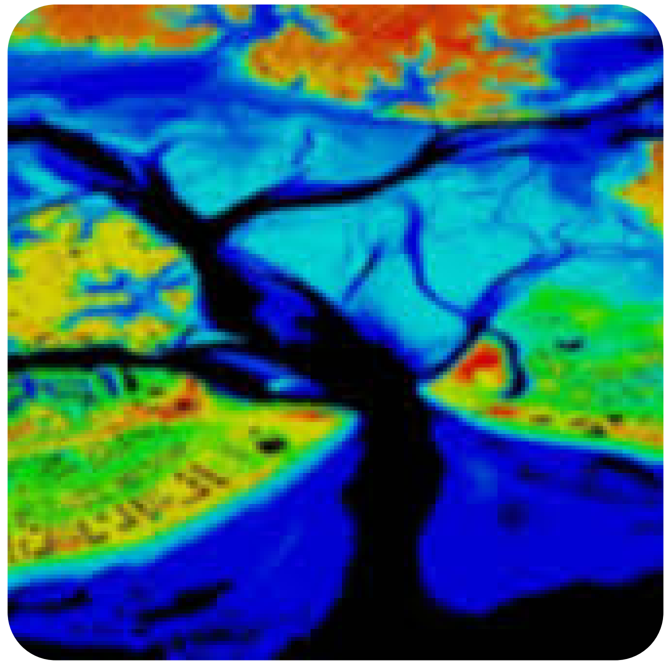
LYEBA ABID | SADIQAH MUSHTAQ

What is LiDAR?

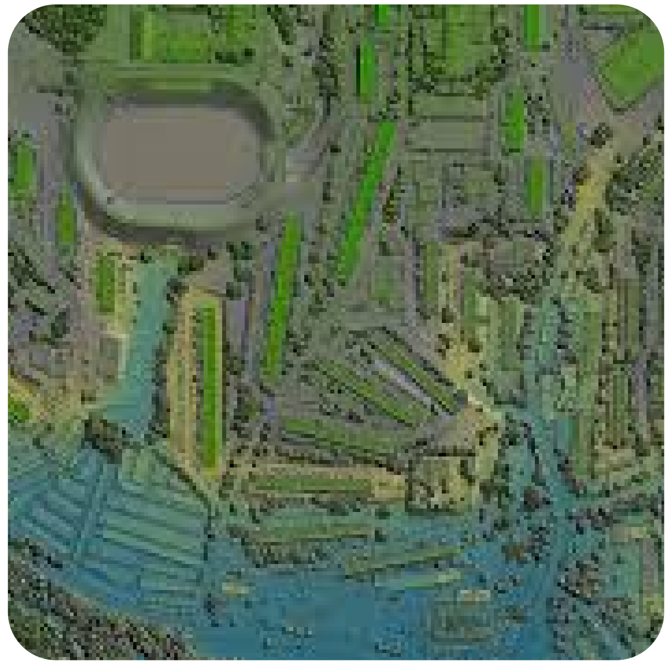
- LiDAR (Light Detection and Ranging) is a remote sensing method that uses light in the form of a pulsed laser to measure variable distances to the Earth.
- Principle: LiDAR systems emit laser pulses and measure the time it takes for the pulse to return after hitting an object. This information is used to create detailed 3D maps or models. Light travels with a finite and constant velocity in a given medium. The amount of time the for emitted light from a source to a reflective target and back to the source allows you to quantify the range. This is know as time of flight measurement.
- Lidars small footprint, independence of sunlight, higher vertical accuracy, less missing data by occlusion, low redundancy and because it does not rely on the existence of textured surfaces have lead to it's increase in popularity in recent years.



Applications of LiDAR



Topographic
Mapping



Urban Planning

Foestry
Management

Autonomous
Vehicles

Signal Acquisition and Pre-processing

- LiDAR utilizes **Time-of-Flight (ToF)** principle for distance measurement.
  - R is range,
  - c is light velocity,
  - ΔT is round trip time.

$$R = \frac{c\Delta T}{2}$$

- Several different LiDAR systems available for environment perception.
- **Scanning LiDAR system** consists of:
  - Transmitters (TX) with microelectromechanical mirror system (MEMS).
    - This type of transmitter reduces the system's price while providing high performance.
  - Receivers (RX) with microelectromechanical mirror system (MEMS)
    - Receivers include photo diode arrays with avalanche or single photon photo diodes with trans-impedance amplifiers and ADC.
- **Data Volume Challenge**
  - LiDAR systems generate millions of data points per second.
  - Resultant raw data equals several GBits/sec -> challenging hardware and software infrastructure.
  - Integration of multiple LiDAR sensors.

Data Compression Techniques

- Aim: Reduce data volume for efficient storage and transmission.

- **Low-Level Data Compression**
  - Binary arithmetic coding: Efficient for on-chip storage but slow.
  - LZMA: Dictionary-based compression with over 50% compression ratio.

- **Compressed Sensing** (CS)
  - Challenges Nyquist theorem, reconstructs signals from sparse samples.
  - Offers high compression ratios and lower sampling rates.

	Low-Level Compression	**Compressed Sensing (CS)**
Benefits	• Optimizes on-chip memory • Reduces link load	• Offers significant data reduction. • Potential for high compression ratios
Considerations	• Limited compression efficiency over samples	• Challenges for real-time implementation

Digital signal filtering

- Signal pre-processing techniques:
  - Signal difference filtering

$$y(n) = x(n) - x(n+m)$$

  - Smooth filtering.

$$y(n) = (1/N)\sum_{i=0}^{N-1} x(n+i)$$

- Used to address strong undershoot characteristics of target echo signals.
- Further improves the signal-to-noise ratio (SNR).

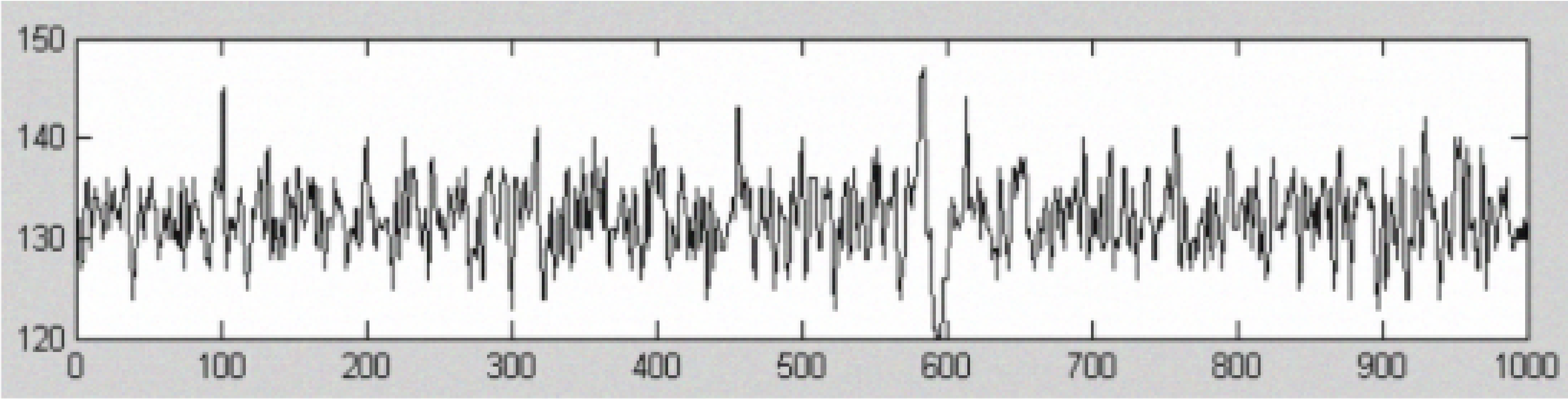


Fig. 3: The pre-filtering signal
(The signal amplitude 15, noise variance 3.51, SNR 4.27)

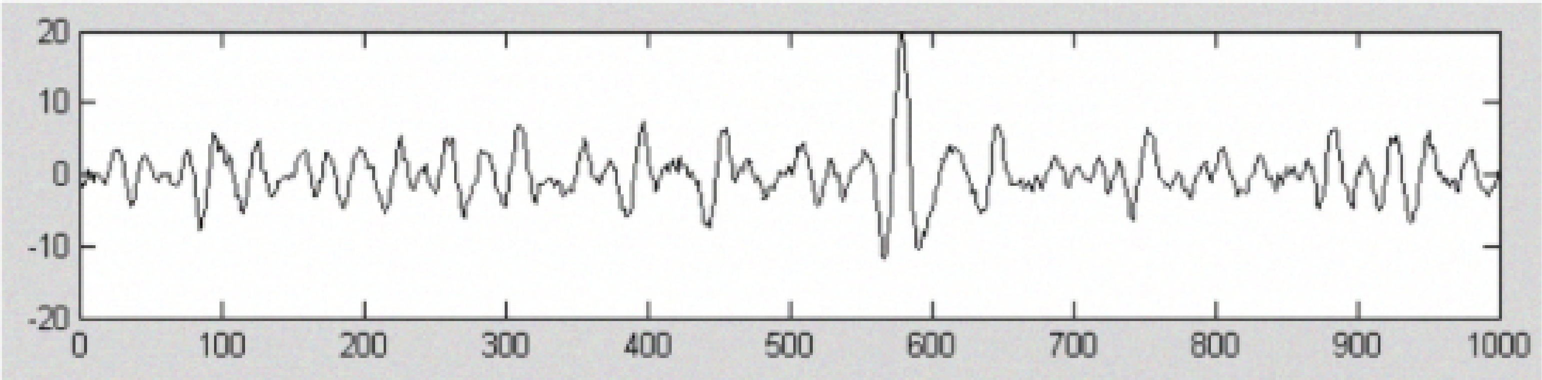
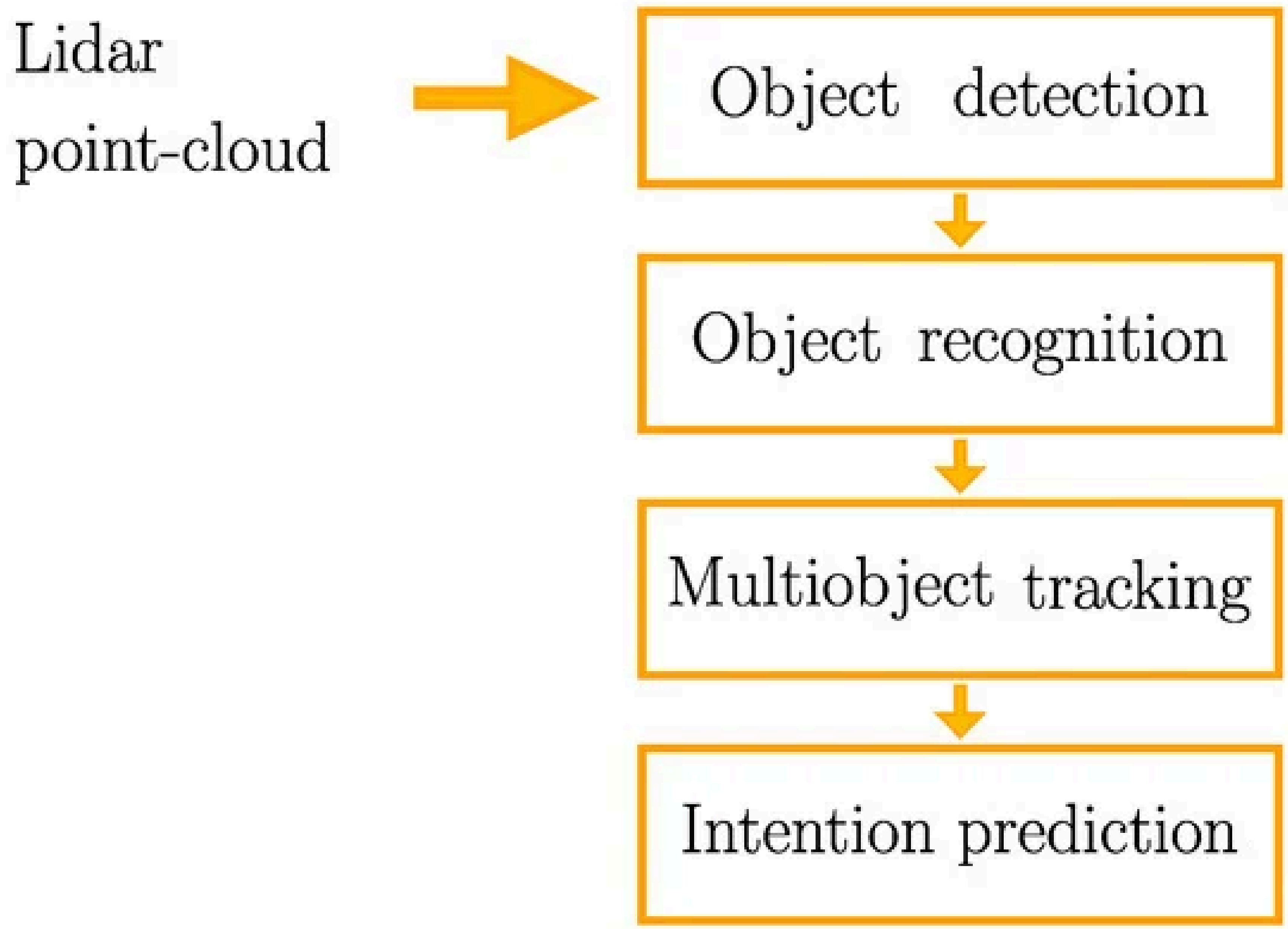


Fig. 4: The after filtering signal
(The signal amplitude 20, noise variance 2.55, SNR 7.9)

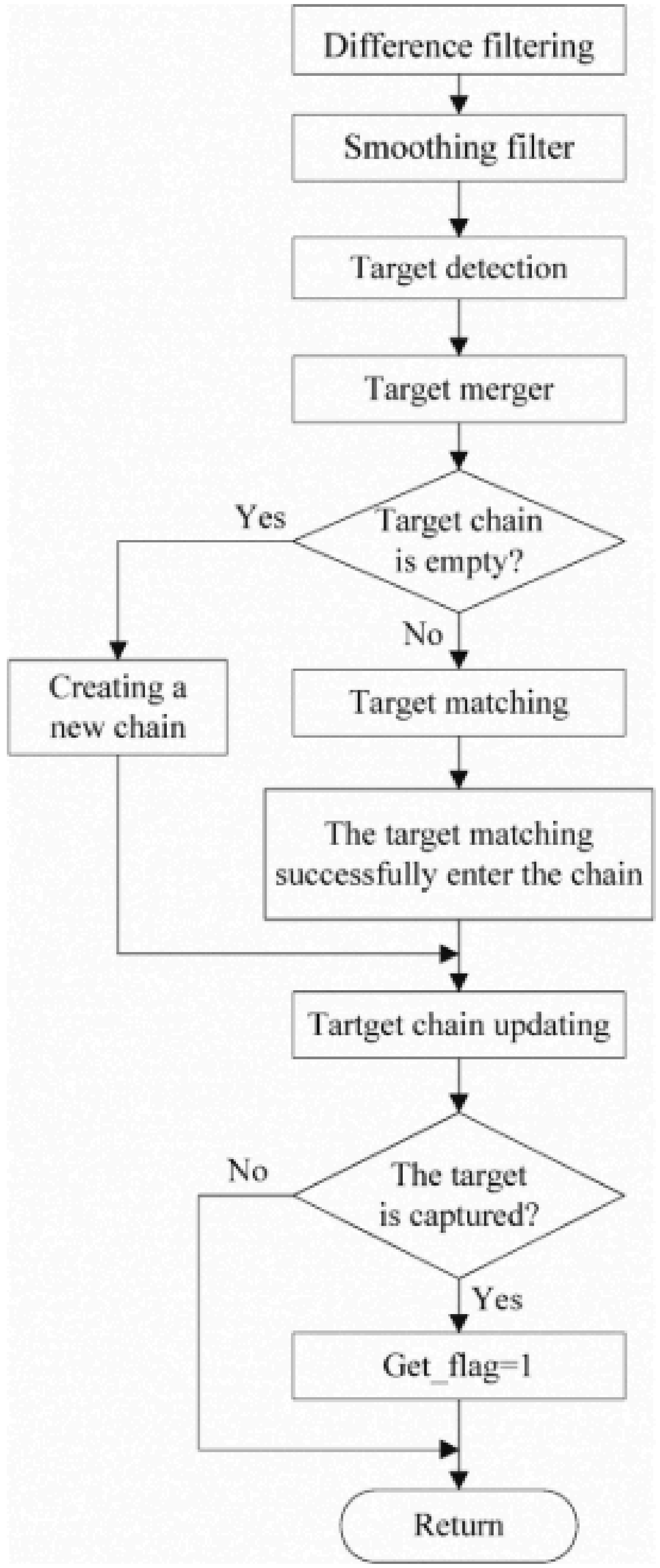
Signal Processing and Algorithms for Lidar



Signal/Object Detection and Ranging

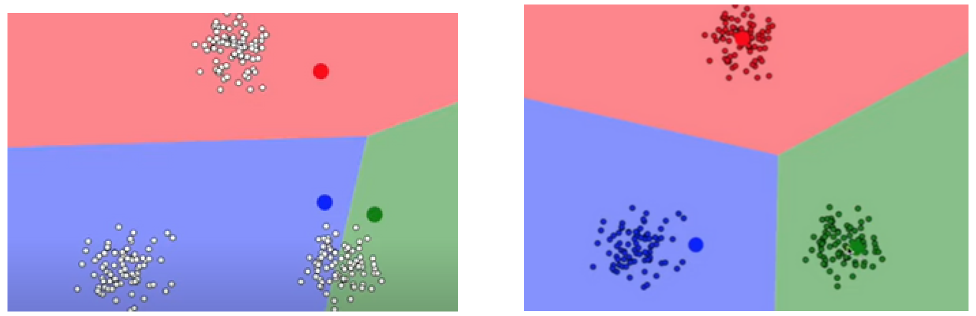
- **Automatic Threshold Adjustment**
  - Threshold level adapts to changes in noise level automatically.
  - Uses statistical calculations from continuous data frames to set threshold.

- **Correlation Detection**
  - Matches targets based on their characteristics like speed and strength.
  - Helps distinguish real targets from false ones caused by noise.

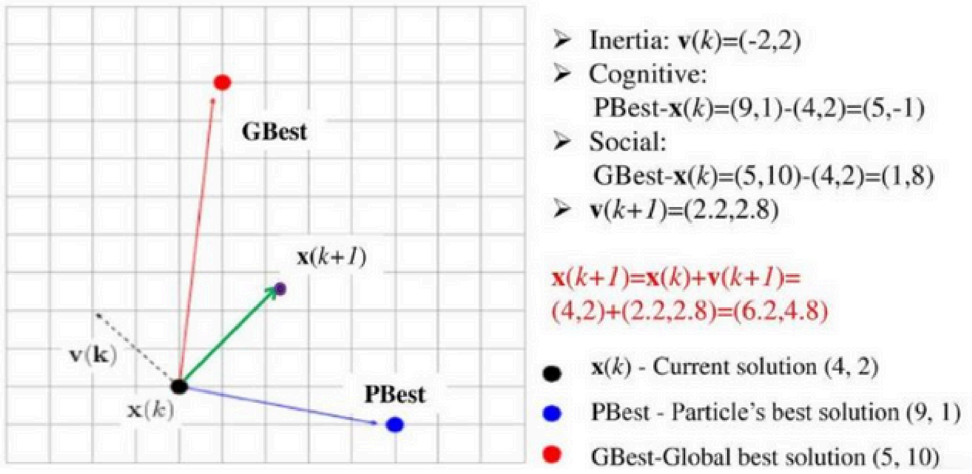


Clustering Algorithms to Process LiDar data

- K-means is a popular clustering algorithm for large datasets due to its linear computational complexity. However, it may converge to suboptimal solutions.
- Particle Swarm Optimization (PSO) is a technique that shows superior performance when applied to LiDAR data, especially in scenarios like urban planning with extensive datasets.
- Traditional methods might struggle with large datasets and identifying critical infrastructure like roads and bridges, often getting trapped in suboptimal solutions.



https://shabal.in/visuals/kmeans/2.html



Particle Representation:

$$\mathbf{x}_i = (\mathbf{m}_{i1}, \cdots, \mathbf{m}_{ij}, \cdots . \mathbf{m}_{iN_c})$$

Quantization Error:

$$J_e = \frac{\sum_{j=1}^{N_e} [\sum_{\forall \mathbf{z}_p \in C_{ij}}, d(\mathbf{z}_p . \mathbf{m}_j)/|C_{ij}|]}{N_c}$$

Algorithm:

1. Initialize each particle to contain N_c randomly selected cluster centroids.
2. For $t = 1$ to t_{max} do
   i. For each particle i do
   ii. For each data vector $\boldsymbol{z}_p$
      a. calculate the Euclidean distance $d(\boldsymbol{z}_p, \boldsymbol{m}_{ij})$ to all cluster centroids C_{ij}
      b. assign $\boldsymbol{z}_p$ to cluster C_{ij} such that distance $d(\boldsymbol{z}_p, \boldsymbol{m}_{ij}) = \min_{\forall c=1,\ldots N_c}\{d(\boldsymbol{z}_p, \boldsymbol{m}_{ic})\}$
      c. calculate the fitness using equation (3)
   iii. Update the global best and local best positions
   iv. Update the cluster centroids.

Position: $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,n}) \in \Re^n$

Velocity: $\mathbf{v}_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,n}) \in \Re^n$

$$\mathbf{v}_i(k+1) = \text{Inertia} + \text{cognitive} + \text{social}$$

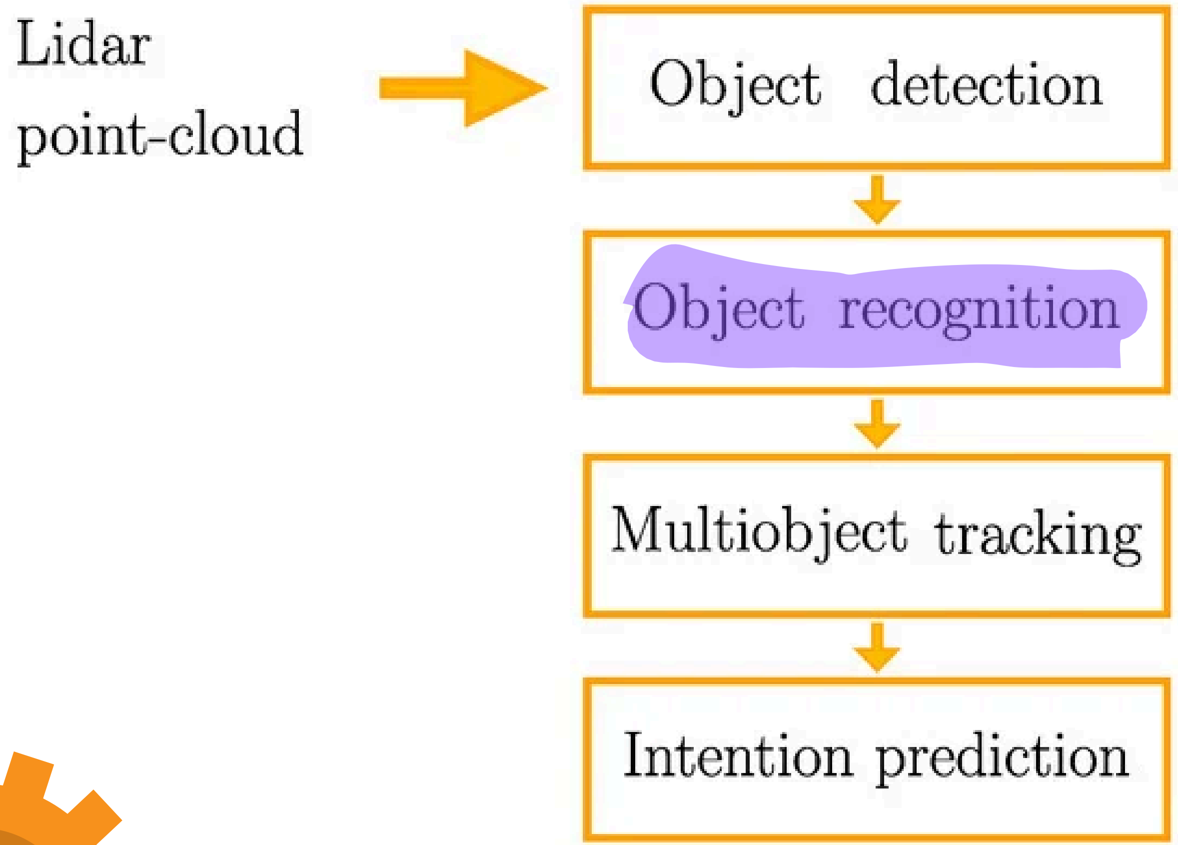
$$\mathbf{v}_i(k+1) = \omega \times \mathbf{v}_i(k) + c_1 \times random_1() \times (PBest_i - \mathbf{x}_i(k)) + c_2 \times random_2() \times (GBest - \mathbf{x}_i(k))$$

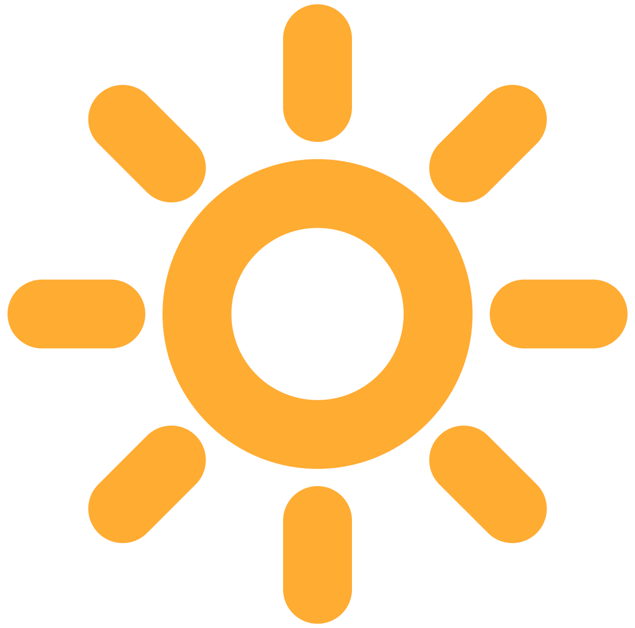
- w,c_1,c_2: Constant
- $random_1()$, $random_2()$: random variable

$$\mathbf{x}_i(k+1) = \mathbf{x}_i(k) + \mathbf{v}_i(k+1)$$

Object Recognition

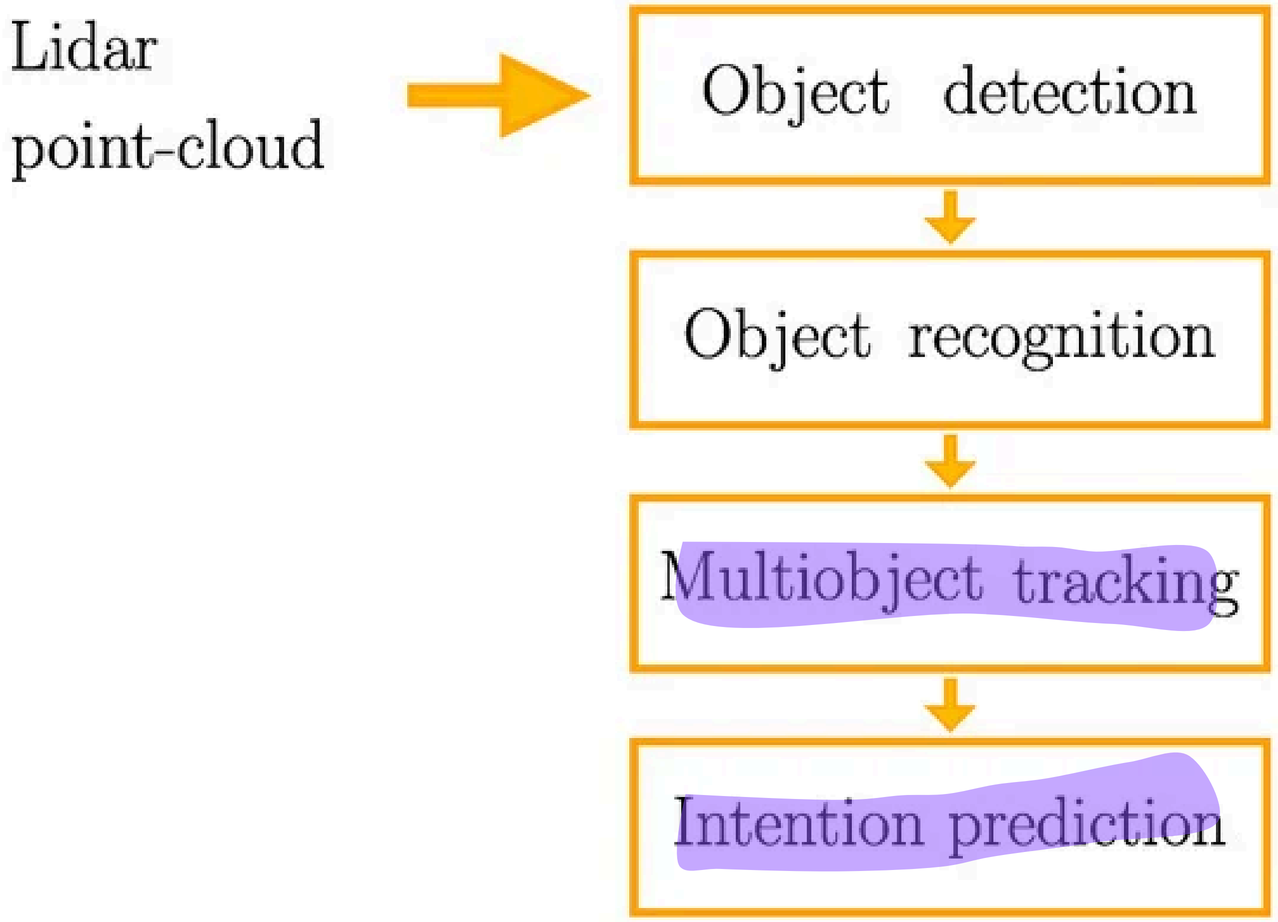
Once we have these objects, we can describe them using features like their size or the average brightness of the points. Then, we can use machine learning to classify these objects, even if we haven't seen them before, reducing mistakes.





Multiobject Tracking and Intention Prediction

1. **Object Tracking**: We can keep track of these objects over time. One method is using a Particle Filter, which is good at handling different shapes of objects. Another option is using Kalman Filtering.
2. **Intention Prediction**: In scenarios like autonomous driving, it's important to guess what other objects might do next. We can use machine learning to predict future actions based on their current state.
3. **Deep Learning Trends**: Nowadays, there's a move towards using deep learning methods to make sense of LiDAR data. These methods, like VoxelNet, HVNet, and SECOND, use neural networks to detect and classify objects in 3D point-cloud data.



Refrences

1. https://ieeexplore.ieee.org/document/1299577
2. https://slideplayer.com/slide/13389026/
3. https://shabal.in/visuals/kmeans/2.html
4. https://www.ukessays.com/assignments/lidar-digital-signal-processing-6771.php
5. https://sci-hub.se/10.1049/cp.2009.0187
6. https://www.mdpi.com/2504-3900/2/13/852
7. https://www.isprs.org/proceedings/XXXVIII/3-W8/papers/p102.pdf
8. https://www.mdpi.com/2674-0729/2/4/15

Thank You!