Filtering of long-range pulsed lidars data using a spatial clustering algorithm

Leonardo Alcayaga

DTU Wind Energy Department

June 17, 2019
Outline

1. The problem

2. Filtering techniques

3. Performance, synthetic data

4. Performance, real data

5. Some final remarks
The problem

- Motivation: characterization of large-scale, turbulent, coherent atmospheric structures from long-range pulsed lidar measurements.
- Characterizing large scale structures requires...a large measuring area.
- Long-range lidars are good at this, but data is sometimes very noisy in the far region.
- How to identify reliable lidar observations without a reference?
Three approaches

- Threshold in the Carrier-to-Noise Ratio, CNR.
- Median-like filter.
- Filtering via DBSCAN clustering algorithm.
Reliable observations show CNR values in a range between -24 and 0. This CNR thresholds vary with site conditions, exp. setup and instrument characteristics.
Filtering using a CNR threshold

- Many reasonable observations are neglected with this approach, CNR decrease with distance. Spatial characteristics seem to be important.
Filtering using a moving median

- Median filter is usually recommended for image processing. Adapted to only identify and reject anomalous values.
- Four parameters: window sizes, $n_r$ and $n_\phi$, and a radial wind speed threshold, $\Delta V_{LOS,median}$.
- This approach is:
  - Fast.
  - Excellent in recovering data from reliable areas of the scan.
  - Arbitrary. Threshold and window size that may be adjusted to meas. height for instance.
  - Not very reliable in extended noisy regions.
Filtering using a density approach

- Main assumption is self-similarity of the data: reliable observations will be close to each other.
- High probability density regions were explored before (Beck and Kühn 2017), using CNR and $V_{los}$ as features. Thresholds for scaling and filtering need to be defined though.
- Problem is unbearable for higher dimensions/features using Kernel density estimates.
- Why more features? The more we consider, the greater the ”distance” between non-similar observations.
Filtering using a density approach, DBSCAN

- Density-based Spatial Clustering for Applications with Noise, DBSCAN (Ester et al. 1996). Why?
- Clustering algorithms are more efficient identifying high density regions.
- Previous knowledge of the number of clusters is not necessary and introduces the concept of noise.
- A robust filter needs few parameters. DBSCAN needs in practice just one.
Filtering using a density approach, DBSCAN

(a) NN = 5

(b)

(c)

(d) Cluster

Noise
Filtering using a density approach, DBSCAN

- NN can be fixed and $\epsilon$ can be estimated automatically according to the structure of the data.
- We end up choosing the number of features to consider and the amount of data/scans to filter per batch.
Performance comparison on synthetic data contaminated with noise

- Both filters were tested on synthetic data, sampled from 2-D Mann-turbulence box via a numerical lidar.
- Numerical lidar mimics the beam averaging and the accumulation of information on azimuthal direction.
- Real lidars have a weighting function that smooths $V_{LOS}$ even more on beam direction.
- No CNR.
Performance comparison on synthetic data contaminated with noise

- Synthetic scans contaminated with procedural (smooth) noise. Contaminated area increases with distance.
- Filters tested on approx. 2000 synthetic scans with different directions and turb. parameters.
Performance comparison on synthetic data contaminated with noise

- Features of the clustering filter:
  - $V_{LOS}$
  - Position, $r$ and $\phi$.
  - $\Delta V_{LOS} = \text{median}(V_{LOS,i} - V_{LOS,NN})$

- Fair comparison, the optimal set of $n_r$, $n_\phi$ and $\Delta V_{LOS,\text{median}}$ comes from maximum values of:
  - $\eta_{\text{noise}}$: fraction of noise detected.
  - $\eta_{\text{recov}}$: fraction of good measurements recovered.
  - $\eta_{\text{tot}} = f_{\text{noise}}\eta_{\text{noise}} + (1 - f_{\text{noise}})\eta_{\text{recov}}$. 
Performance comparison on synthetic data contaminated with noise

- The clustering filter performs better in noise detection, while keeping a good recovering rate.
Real data from The Østerild balcony experiments

- Wind speed data in space-time from two scanning lidars in Østerild Wind Turbine test centre (Karagali et al. 2018), at 50 and 200 meters.
- High resolution is space (range gates each 35 m.) and in time (45 seconds per scan). The lidars are not synchronous.
- Filters tested on data from \( \approx 1 \) week of measurements.
Performance comparison on real data

- Filters performance on less reliable areas of the scan (low CNR values).
- The recovery rate on reliable areas is also studied.
- The features used for the clustering algorithm this time are $V_{LOS}$, $r$, $\phi$, $\Delta V_{LOS}$ and CNR. More features are available, but it requires more scans per batch, due to euclidian distance.
Performance comparison on real data

Non-filtered, Line-of-Sight wind speed [m/s]
Performance comparison on real data

Filtered scan CNR threshold, Line-of-Sight wind speed [m/s]
Performance comparison on real data

Filtered scan median, Line-of-Sight wind speed [m/s]

North-South [m]

West-East [m]
Performance comparison on real data

Filtered scan DBSCAN, Line-of-Sight wind speed [m/s]
Performance comparison on real data

- Distribution $V_{LOS}$ for reliable CNR show few extreme values.
- On the less reliable side, distribution is heavy-tailed, but with many reasonable values.
- Distribution of the recovery fraction on the latter looks better for the clustering filter.
Performance comparison on real data

- Spatially, the clustering filter tends to reject more data in the far region of the scan.
- A fraction of reliable values are rejected in the near region. It can be combined with a CNR threshold.
Final remarks

- Clustering filter uses the best of two approaches: spatial continuity and CNR information.
- For this data set, recovery increased by 38%.
- Little user intervention, mostly in the definition of relevant features and the amount of data needed (more features, more observations are necessary).
- Need to be tested on different scanning patterns.
- A deeper analysis of the computational performance is necessary. DBSCAN have a $O(n \log(n))$ to $O(n^2)$ computational complexity. Very efficient median filters can achieve up to $O(n)$. 