SCHOOL OF DATA ANALYSIS

Hough Transform + ML

Mikhail Hushchyn, Andrey Ustyuzhanin

TrackML meeting, 21.02.2017

Goals of a baseline solution

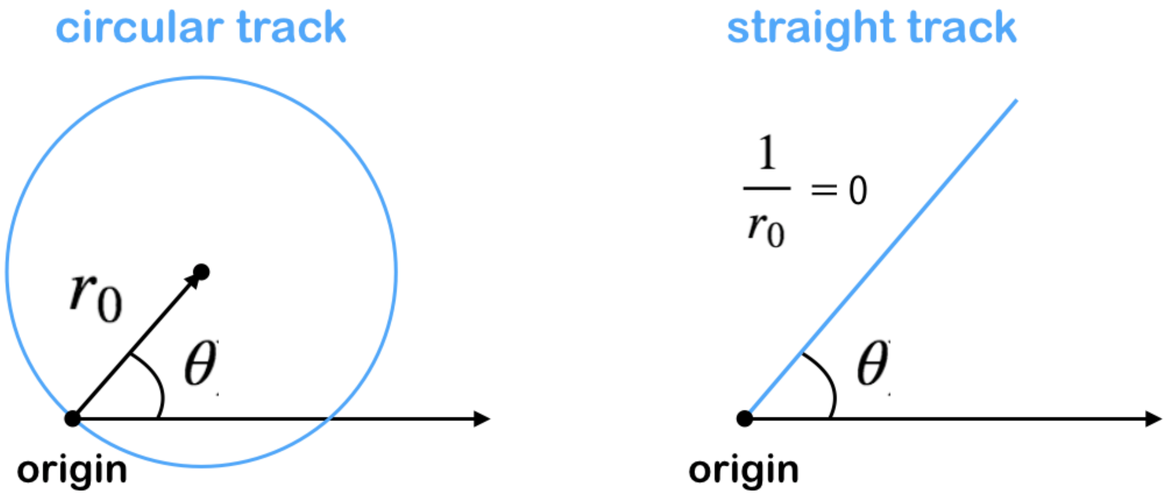
- Help participants to start
- Show how ML can be used in track pattern recognition

Hough Transform + ML

Hough Transform for a Hit

In polar coordinates (r, ϕ):

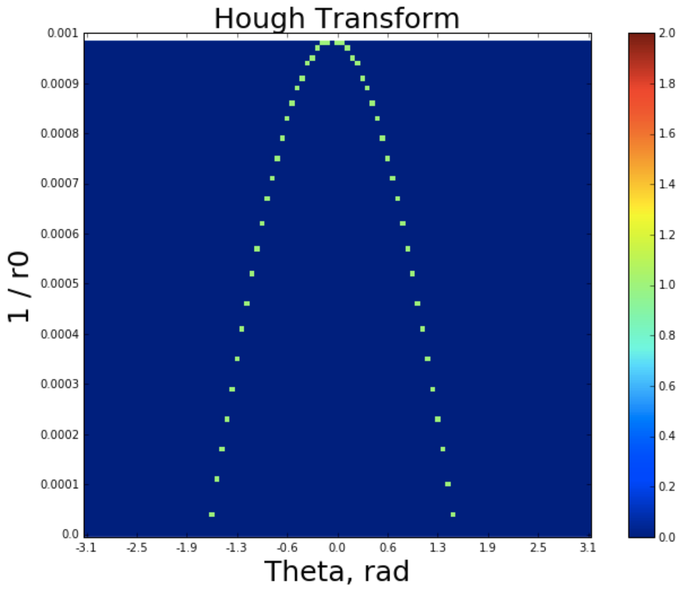
$$r = 2r_0 Cos(\phi - \theta)$$



One hit with coordinates (r, ϕ):

Hit (r,ϕ) $\xrightarrow{\text{Transfrom}}$

$$\frac{1}{r_0} = \frac{2Cos(\phi - \theta)}{r}$$

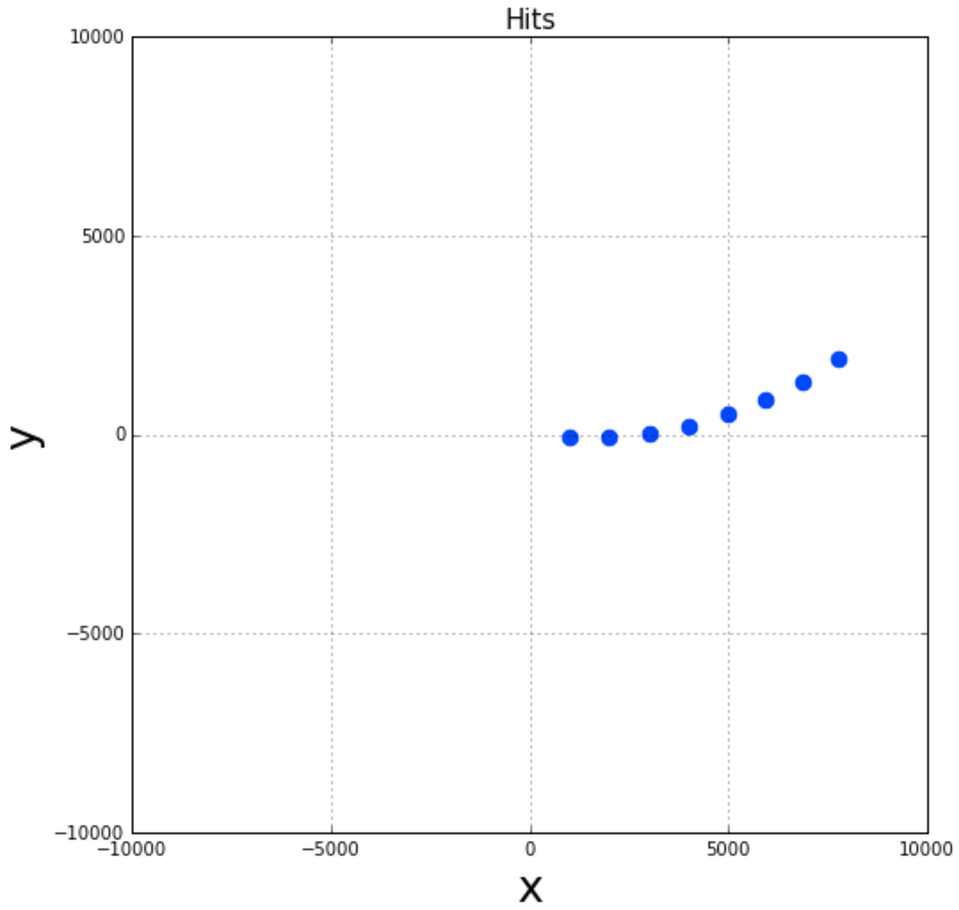


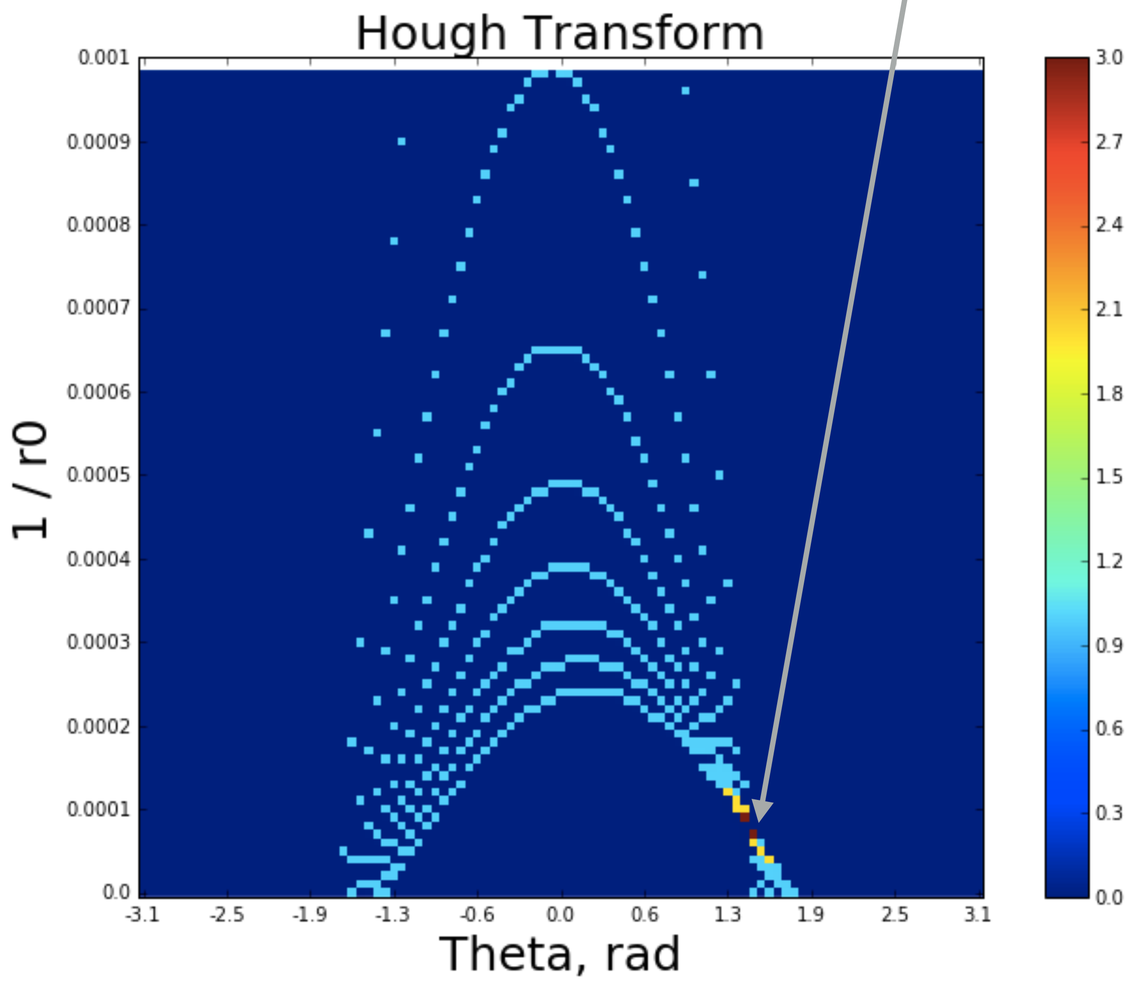
Hough Transform for a Track

One track

$$\frac{1}{r_0} = \frac{2Cos(\phi - \theta)}{r}$$

Recognized track

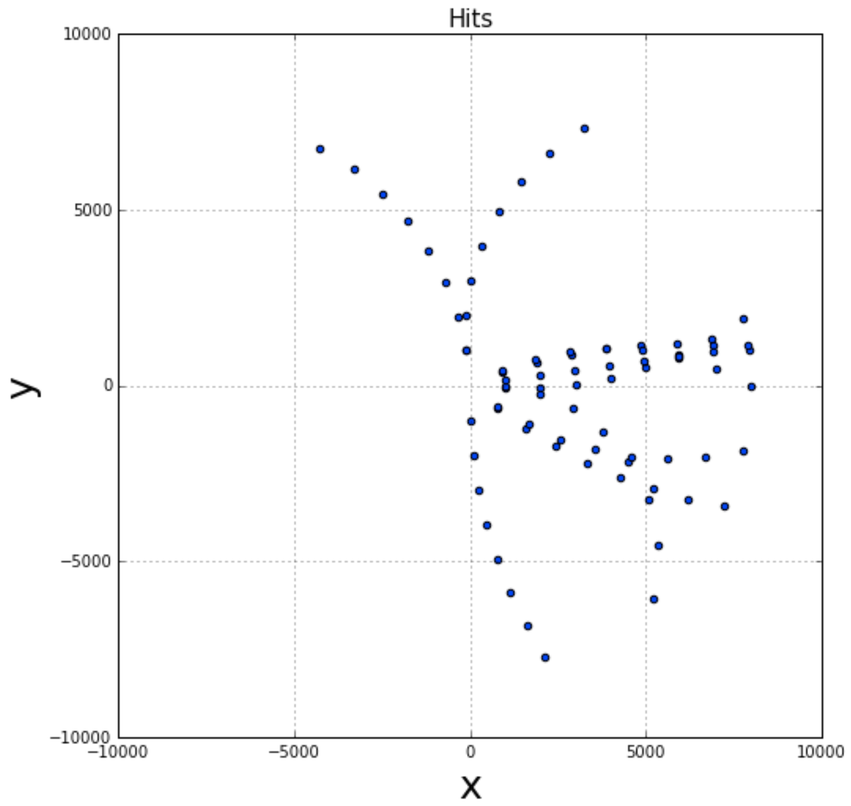


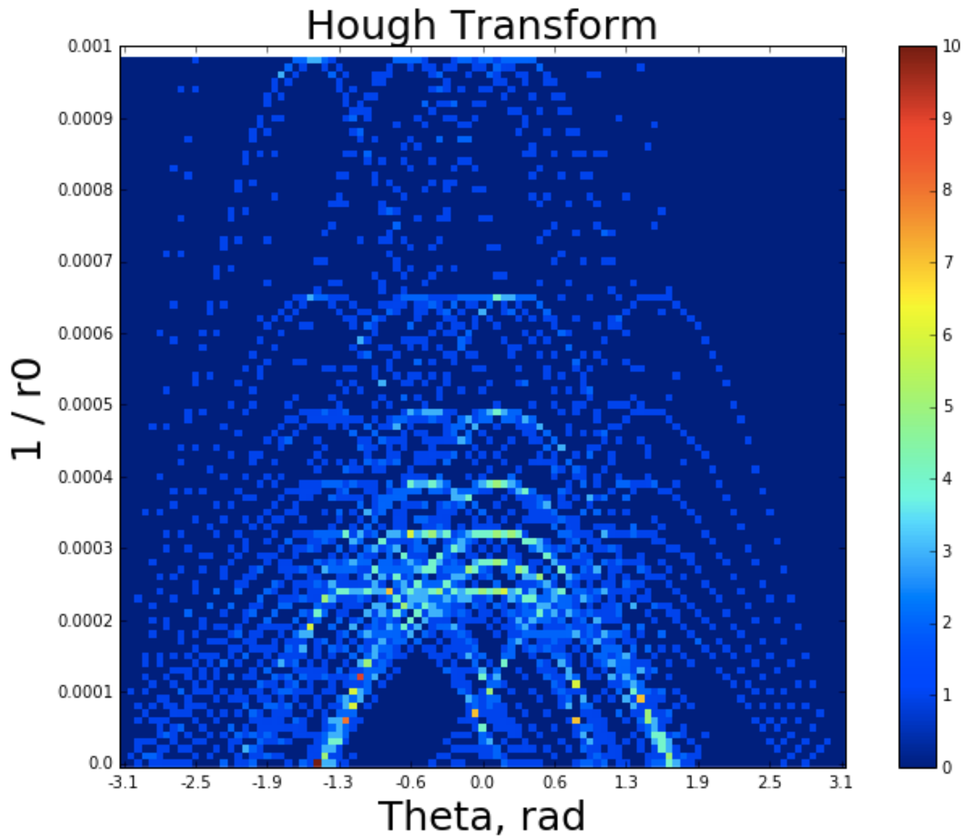


Hough Transform for an Event

Event ——— $\frac{1}{r_0} = \frac{2Cos(\phi - \theta)}{r}$ ⟶ Recognized tracks

(hot bins)





Recognized tracks: good tracks, clones, ghosts.

Hough Transform + Tracks Clustering

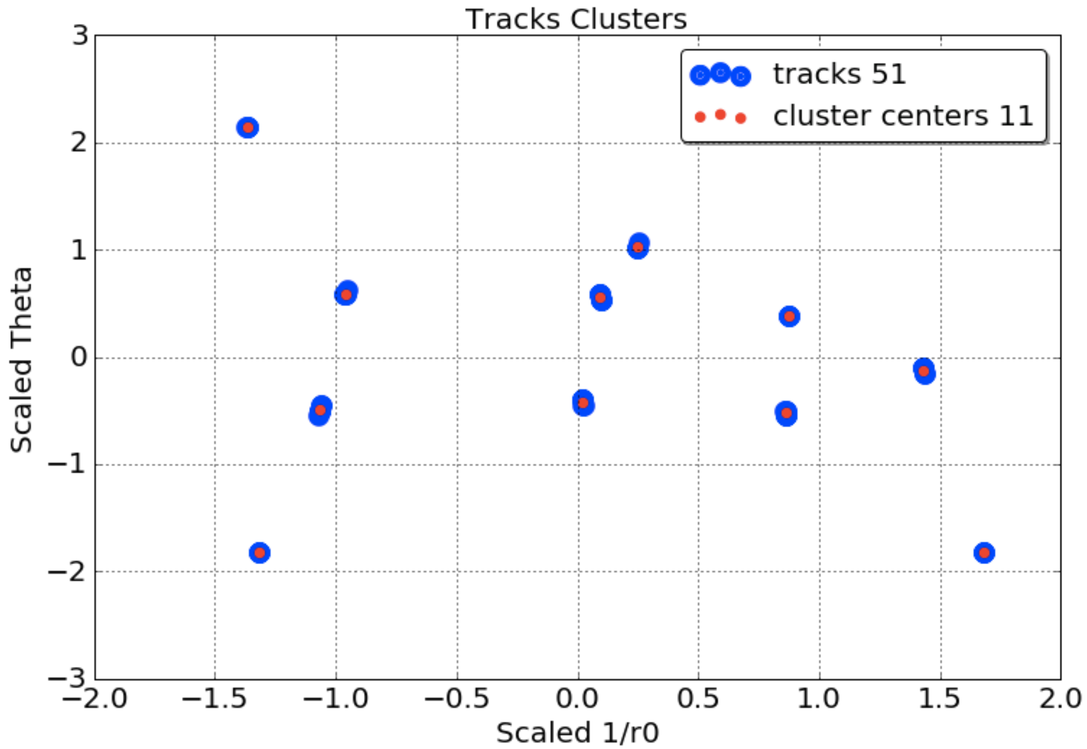
Tracks clustering to reduce a number of clones.

Features: Track parameters

Methods: K-Means, Mean-shift, DBSCAN, Agglomerative clustering, ... (more)

Metrics: Fowlkes-Mallows scores, Homogeneity, Completeness and V-measure, Silhouette Coefficient, ... (more)

One event with 10 tracks:



Notebook: https://github.com/hushchyn-mikhail/TrackMLRamp/blob/develop/HoughML/HoughML.ipynb

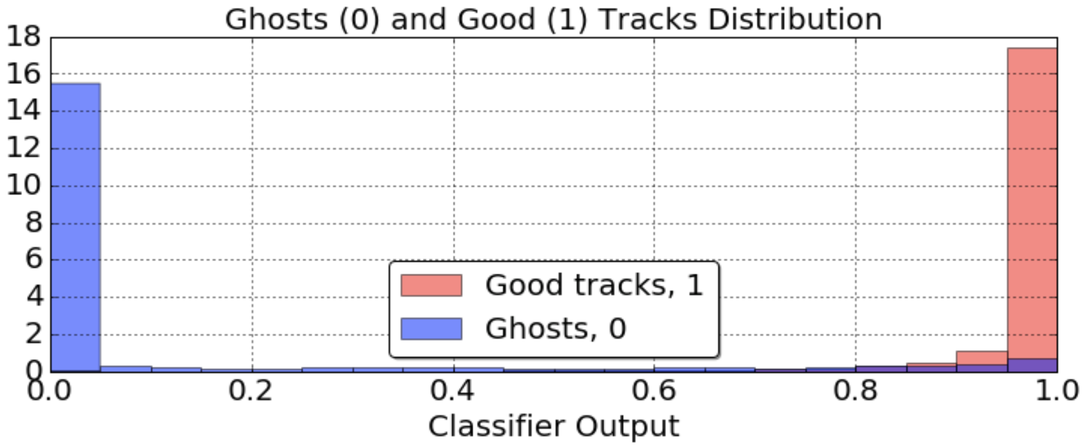
Hough Transform + Tracks Classification

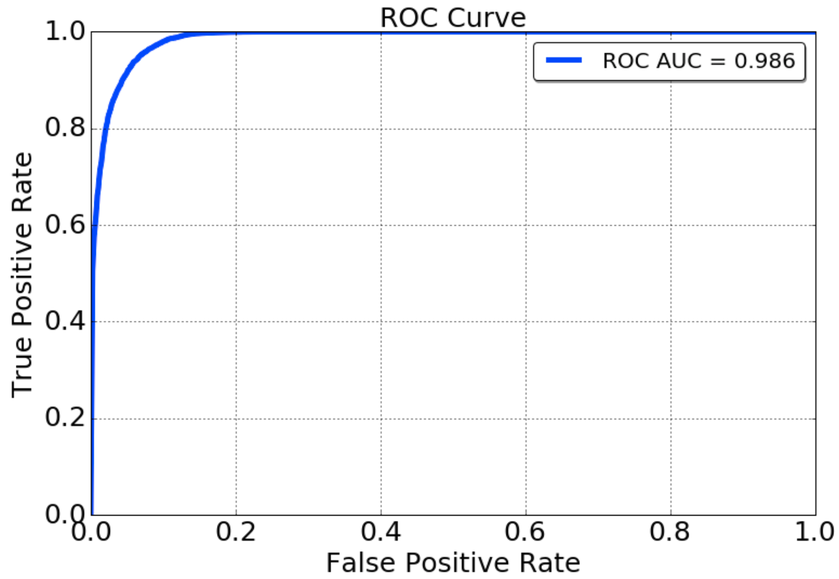
Tracks classification to reduce a number of ghosts.

Features: Track parameters, number of hits, RMSE of a track fit

Methods: ANN, Random Forest, Gradient Boosting, ...

Metrics: ROC-curve, ROC AUC





Notebook: https://github.com/hushchyn-mikhail/TrackMLRamp/blob/develop/HoughML/HoughML.ipynb

Two approaches:

1) Each hot bin is a recognized track. This means, that one hit can to belong to several recognized tracks:

reco. track 1: 1, 2, 3, 4, 5

reco. track 2: 4, 5, 6, 7, 8

…

2) One hit belongs to just one recognized track. This means, each hit has only one recognized track label:

Reco. hit labels: 1, 1, 1, 2, 2, 2, 3, …

True hit labels: 1, 1, 1, 1, 2, 2, 2, …

The 2nd approach goes from the 1st one. Not vice versa!

Metrics

- Track Finding Efficiency (purity of one recognized track)
- Reconstruction Efficiency (fraction of correctly recognized tracks)
- Ghost Rate (~ number of wrong recognized tracks)
- Clone Rate (~ number of clones)
- ‘Fraction of Correctly Recognized Hits’

Details in the backup slides

1000 events, 20 tracks/event, track_eff_threshold = 0.9, 450 mc/event.

Each hot bin is a recognized track (HitsMatchingEfficiencyTracks):

	Metrics	Hough	Hough + Clones Red.	Hough + Ghosts Red.	Hough + Clones and Ghosts Red.
0	Reconstruction Eff.	0.926	0.803	0.912	0.857
1	Clone Rate	3.505	0.636	3.371	0.653
2	Ghost Rate	2.024	0.686	0.409	0.261
3	Track Eff.	0.900	0.902	0.984	0.974

One hit for one recognized track (HitsMatchingEfficiencyLabels):

	Metrics	Hough	Hough + Clones Red.	Hough + Ghosts Red.	Hough + Clones and Ghosts Red.
0	Reconstruction Eff.	0.850	0.819	0.898	0.843
1	Clone Rate	0.000	0.000	0.000	0.000
2	Ghost Rate	0.115	0.140	0.041	0.088
3	Track Eff.	0.983	0.974	0.994	0.985

Fraction of Correctly Recognized Hits (RecoHitsEfficiency):

	Metrics	Hough	Hough + Clones Red.	Hough + Ghosts Red.	Hough + Clones and Ghosts Red.
0	Score	0.944	0.932	0.916	0.906

Notebook: https://github.com/hushchyn-mikhail/TrackMLRamp/blob/develop/HoughML/HoughML_0.9.ipynb

1000 events, 20 tracks/event, track_eff_threshold = 0.8, 450 mc/event.

Each hot bin is a recognized track (HitsMatchingEfficiencyTracks):

	Metrics	Hough	Hough + Clones Red.	Hough + Ghosts Red.	Hough + Clones and Ghosts Red.
0	Reconstruction Eff.	0.978	0.956	0.953	0.936
1	Clone Rate	4.441	0.883	4.017	0.836
2	Ghost Rate	1.037	0.287	0.085	0.056
3	Track Eff.	0.900	0.902	0.975	0.965

One hit for one recognized track (HitsMatchingEfficiencyLabels):

	Metrics	Hough	Hough + Clones Red.	Hough + Ghosts Red.	Hough + Clones and Ghosts Red.
0	Reconstruction Eff.	0.955	0.938	0.938	0.917
1	Clone Rate	0.000	0.000	0.000	0.000
2	Ghost Rate	0.010	0.022	0.004	0.016
3	Track Eff.	0.983	0.974	0.987	0.978

Fraction of Correctly Recognized Hits (RecoHitsEfficiency):

	Metrics	Hough	Hough + Clones Red.	Hough + Ghosts Red.	Hough + Clones and Ghosts Red.
0	Score	0.944	0.932	0.92	0.908

Notebook: https://github.com/hushchyn-mikhail/TrackMLRamp/blob/develop/HoughML/HoughML.ipynb

Several conclusions:

1) ML helps to reduce Ghost Rate :)

2) ML can increase Reconstruction Efficiency (@ @)

3) Tracks Clustering for Clone Rate reduction should be improved (by participants) :-]

Metric Proposal 1

Target metric: Reconstruction Efficiency

Restrictions (example) : Track Finding Efficiency > 0.8

Clone Rate < 0.1

Ghost Rate < 0.1

Pros:

〉 Works well with both approaches: a hit belongs to one or several recognized tracks

〉 Clear from physics point of view

〉 Hit weights can be applied

Cons:

〉 Non-user-friendly

Metric Proposal 2

Target metric: Fraction of Correctly Recognized Hits.

Features:

- One hit for one recognized track
- This metric is similar to the metric used in TrackMLRamp repository
- Hit weights can be applied

Pros:

- User-friendly

Cons:

- Non-trivial accordance between the metric and Reconstruction Eff., Clone and Ghost Rates.

Final Conclusions

- Hough Transform provides a lot of possibilities for improvements
- ML reduces Ghost Rate and increases Reco. Eff.
- One hit for many tracks: highest Reco. Eff.
- One hit, one track: lowest Clone and Ghost Rates
- Metric Choice: two possible metrics

Backup

Track Finding Efficiency. Hit Matching.

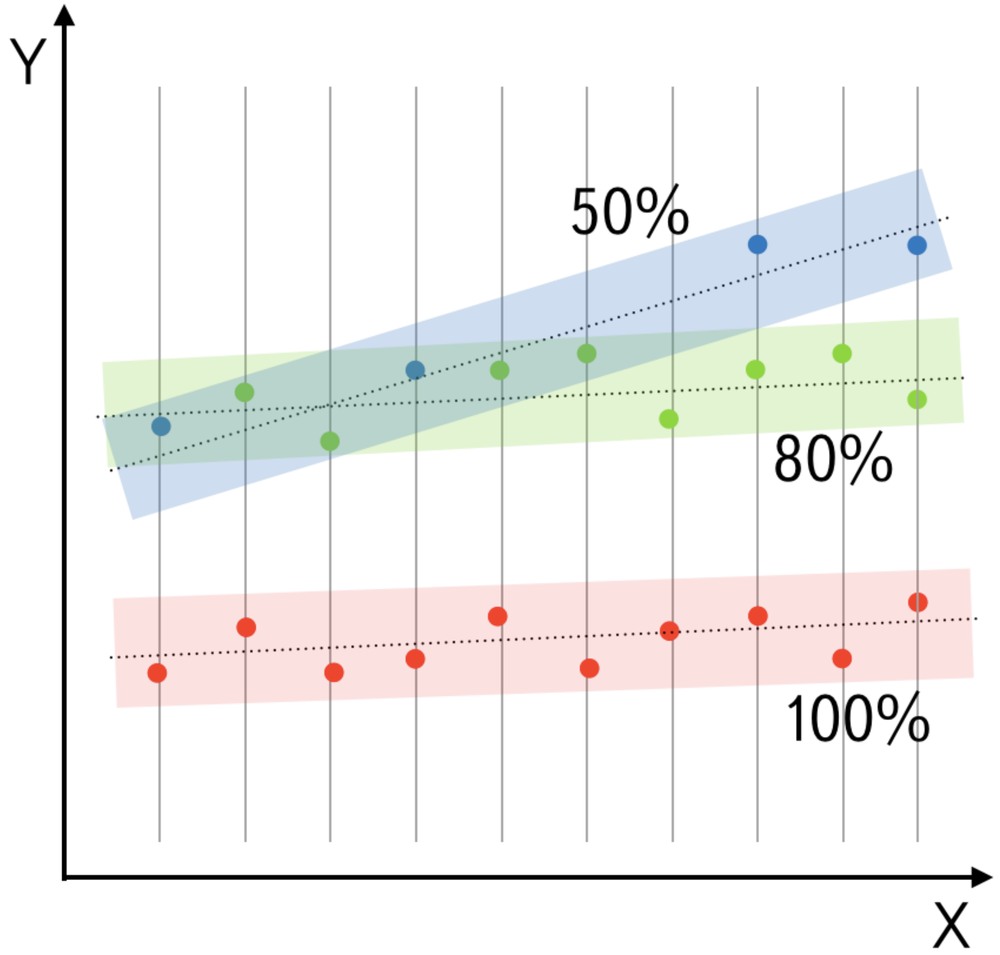
The track finding efficiency is defined as:

$$\epsilon_{track} = \frac{N_{reco\_true\_hits}}{N_{reco\_hits}} * 100\%$$

where N denotes the number of recognized true hits of a track and number of recognized hits respectively.

The track is considered to be reconstructed if its efficiency is higher than, for example, 70%.

This method is stable in the limit of very high track densities.



Reconstruction Efficiency

The reconstruction efficiency is defined as:

$$\epsilon_{reco} = \frac{N_{ref}^{reco}}{N_{ref}}$$

where N_{ref}^{reco} is the number of reference tracks that are reconstructed by at least one track. It lays in [0, 1] range.

Number of non-reference tracks ($N_{non-ref}^{reco}$) should also be controlled. Normally the relation:

$$\frac{N_{non-ref}^{reco}}{N_{total} - N_{ref}} \ll \epsilon_{reco}$$

should hold, otherwise the reference criteria might be too strict.

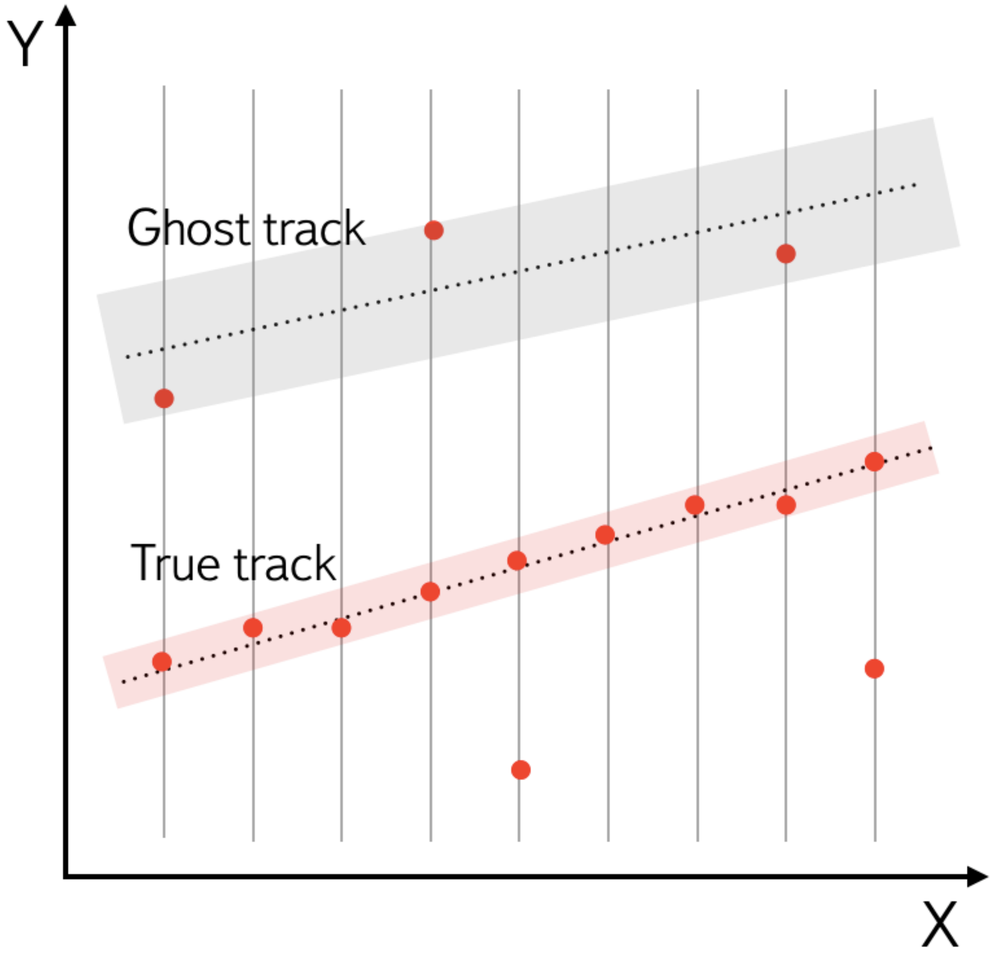
Ghosts

Ghosts are tracks produced by the pattern recognition algorithm that do not reconstruct any true track within or without the reference set.

A ghost rate is defined as:

$$\epsilon_{ghost} = \frac{N^{ghost}}{N_{ref}}$$

It can to take any non-negative values.



Clones

The definitions for efficiency and ghost rate are sensitive to multiple reconstructions of a track. Such redundant reconstructions are sometimes called clones.

For a given track m with N_m^{reco} tracks reconstructing it, the number of clones is

$$N_m^{clone} = \begin{cases} N_m^{reco} - 1, \text{if} N_m^{reco} > 0 \\ 0 \qquad\quad , \text{otherwise} \end{cases}$$

A clone rate then is

$$\epsilon_{clone} = \frac{\sum_m N_m^{clone}}{N_{ref}}$$

It can to take any non-negative values.