Sandia National Laboratories

FLIGHT TRAJECTORY STITCHING

Bryce Castle, Sean Lee,
Tanay Athreya, Christopher Ban,
Kabir Batra, Damian Cantu,
Joseph Luo, Yazan Mahmoud

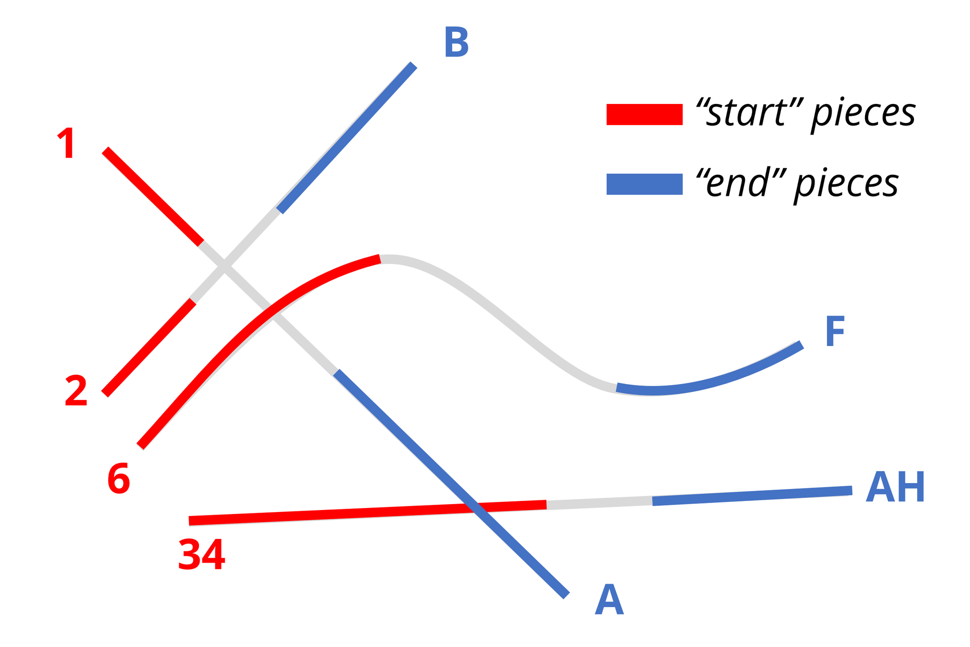
Introduction & Definitions

Given fragments of a flight, the goal is to reassemble flight fragments back together (to complete the motion path) using features of the flight fragments since fragments from the same flight should share similar feature values.

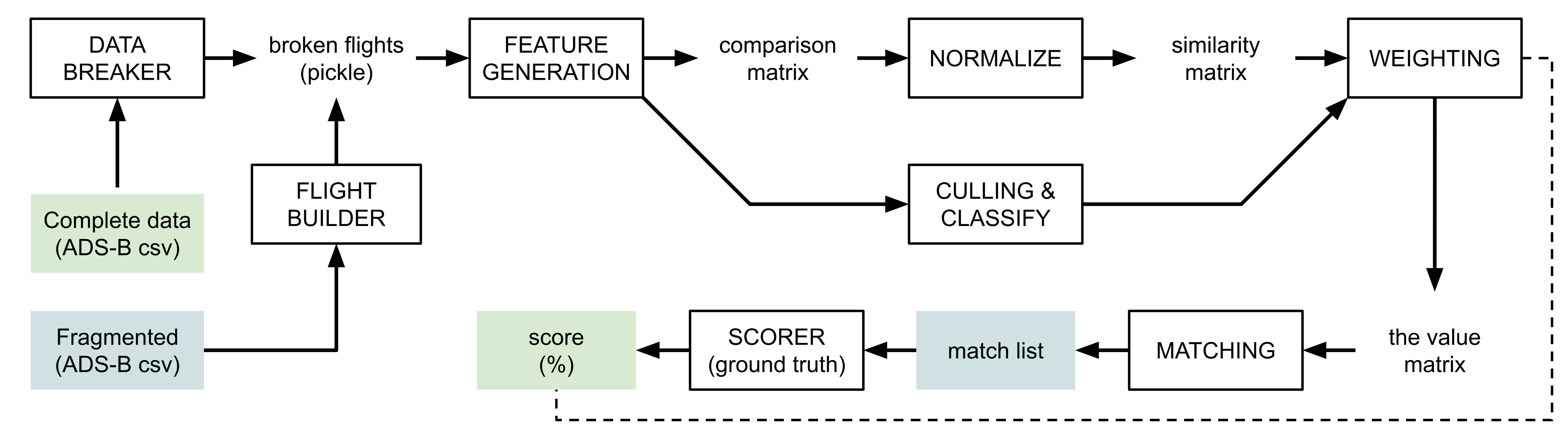
Flights might be broken by:

- Individuals tampering with transponder data stream
- Data collectors are sparse in some geographical locations
- Loss of electrical power and location reporting ability

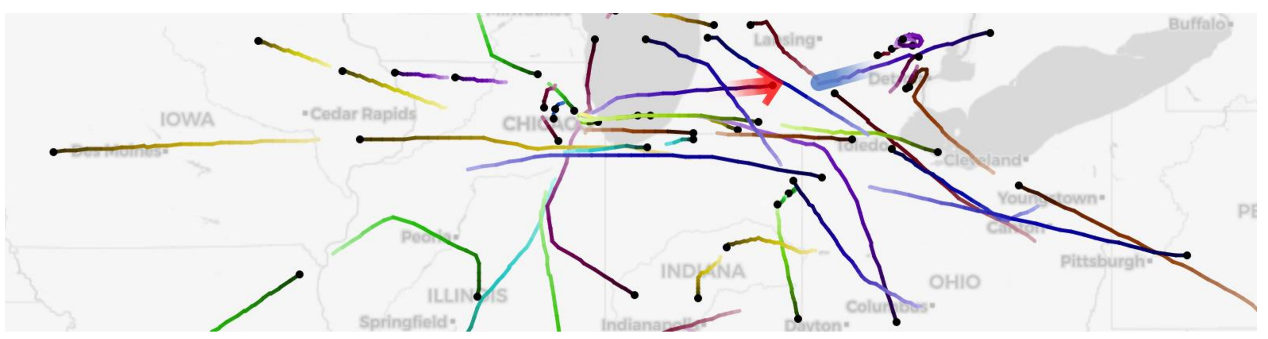
Trajectory: *ordered set of time & position points that define a flight*
Alt: *shorthand for altitude (reported in feet)*
Feature: *a value that summarizes a set of points (ex: average alt)*
Transponder: *aircraft device that sends position data*



[1]: Fragmentation of four flights, "start" pieces are indexed while "end" pieces are labeled spreadsheet-style.



[2]: Our process diagram



[3] All flight pieces before culling step

[4] Remaining potential matches after culling step

1. Culling

Why? Wanted to narrow down potential endpoints for flights before using features to eliminate obvious unrealistic matches & efficiency reasons

How? Implemented 3 culling approaches:

1. time based (going backwards in time)
2. speed based (flights that pass threshold of mach-1)
3. ascent/descent rate based (flights that climb or descend faster than 4000 ft/min)

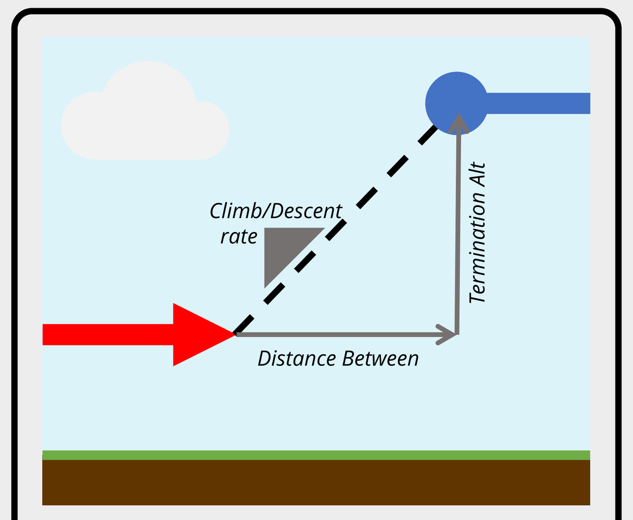
Results:
Time: 8.66% match-space reduction
Speed: 78.16% match-space reduction
Ascent/Descent: 9.62% match-space reduction

2. Features

Why? We use these features to compare the similarity of flight tracks. The assumption is that flight tracks with similar features are likely to be the same flight.

Top 5 Features:
•Distance Between Fragment Endpoints
•Climb/Descent rate
•Termination Speed
•Termination Altitude
•Maximum Speed

How? Features are fed into classification and matching.



[5] Features between pieces

3. Normalization

Why? We wanted to level the playing field of all the features, so our weighting algorithm doesn't bias towards certain features because their domain is much larger than another feature's domain.

How? We decided to use maximum normalization which would divide the feature matrix by the max value of the matrix to bound the matrix between 0 and 1.

Feature	Value Range
Term Alt	[0, 14000]
Heading	[0, 180]
Max Speed	[0, 383]
Distance	[0, 5100]
Climb/Desc	[0, 13010]
Loiter Ratio	[0, 38]

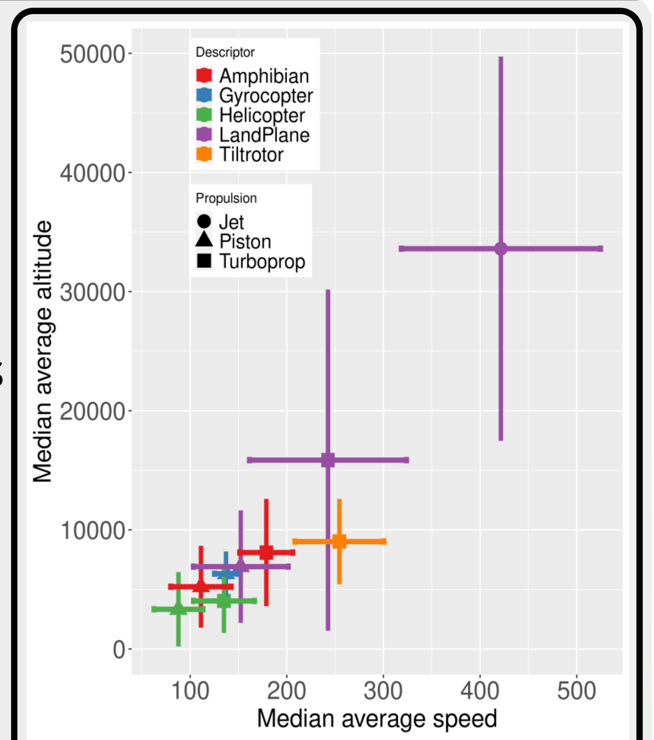
[6] Feature value ranges

4. Classify

Why? Classify flight tracks into categories by ICAO code, reducing matching space and allowing stitching algorithms to run more efficiently.

How? Characterize categories by generating bounding statistics for each, this allows for passing classified flights to stitching algorithm.

Results: Allowed for a 66% reduction in matching space.



[7] Scatterplot of class. stats

5. Weights

What? Scale similarity values by a weight before adding the features together.

Why? Give certain features more priority when matching to increase matching accuracy; can also be used to determine low value features.

How? Differential Evolution global optimization function; weights sum to 1; goal to maximize matching accuracy

Feature	Term Alt	Term Speed	Max Speed	Distance Between	Climb Descent
Weights	0.085	0.101	0.035	0.535	0.244

[8] set of computed feature weights

6. Matching/Scoring

Why? Evaluate if process is correctly pairing matches.

How? Multiply the weighted and normalized feature & culling matrices to get a resulting matrix of probabilities of track pairings. Search for highest probability in matrix, match track indices, produce pairing, remove indices from matrix and repeat. Test on known ground truth.

Values	1	A	2	B	3	C
1	0	0.99	0.33	0.70	0.25	0.66
A	0.45	0	0.62	0.11	0.67	0.38
2	0.22	0.45	0	0.51	0.15	0.92
B	0.72	0.54	0.33	0	0.44	0.22
3	0.08	0.27	0.48	0.90	0	0.10
C	0.25	0.33	0.74	0.19	0.80	0

Rows: Start Tracks
Cols: End Tracks

Pairings:
(1,A), (2,C), (3,B)

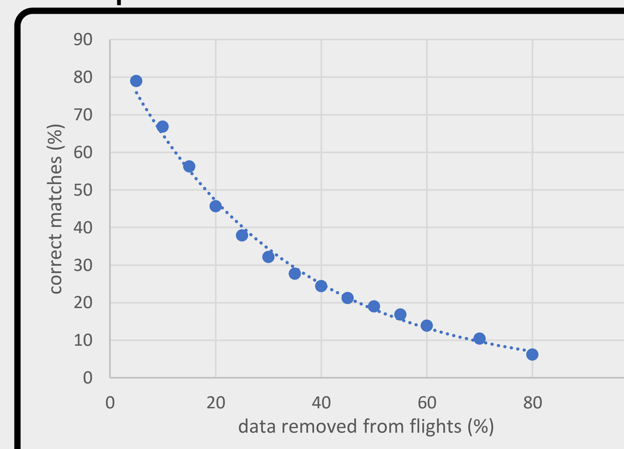
[9] Finding track pairings in final resulting matrix

Conclusion

Conclusion: We can correctly pair 79% of 1000 test flights that have been split in two after randomly removing 5% of their path. Additionally, as we remove more from the ground truth data, the score drops due to our reliance on distances between pieces.

Future Work: Creation of additional features and classification algorithms.

Acknowledgements:
Thanks to Data Mine staff and our mentors, Dr. Andy Wilson & Dr. Katrina Ward for their guidance and support throughout this year.



[10] % removed vs score