11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, including the AIA
20 - 22 September 2011, Virginia Beach, VA

AIAA 2011-6825

Radar Data Tracking Using Minimum Spanning Tree-Based Clustering Algorithm

Chunki Park*, Hak-Tae Lee†, and Bassam Musaffar ‡
University of California Santa Cruz, Moffett Field, CA 94035, USA

This paper discusses a novel approach to associate and refine aircraft track data from multiple radar sites. The approach provides enhanced aircraft track accuracy and time synchronization that is compatible with modern air traffic management analysis and simulation tools. Unlike existing approaches where the number of aircraft in the radar data must be assumed, this approach requires no such prior knowledge. While commercial aircraft provide ID tags captured in the radar data in the form of Mode 3 transponder codes, general aviation often lacks such transponders, which precludes using the number of codes sensed to count the number of aircraft in the data. To meet this challenge, an approach to track an unknown number of unidentified aircraft using a clustering algorithm is proposed. The paper presents a method to relate aircraft between consecutive time frames and refine the trajectories of those vehicles. Experimental results from evaluating the algorithm and demonstrating its viability are provided.

Nomenclature

σ_L	Standard deviation for edge lengths [nmi]
Θ_h	Maximum allowable difference between headings of two aircraft [°]
Θ_p	Maximum allowable distance between the real position and the expected position of an aircraft [nmi]
d_i	Length of the ith edge in a spanning tree [nmi]
L_{max}	Maximum distance in a cluster [nmi]
m_L	Average of edge lengths [nmi]

I. Introduction

Unmanned Aerial Vehicle (UAV) traffic in the National Airspace System (NAS) is expected to grow and needs to be accommodated.[1] Currently, special permission is required to operate UAVs in the NAS, and studies are under way to plan and assess their safe operation in an environment of commercial and general aviation. Accurate air traffic data is required to support these studies. The need is to correlate and aggregate the data from multiple radar sites that cover the airspace of interest. These data may be consolidated in two ways – the data have to be properly associated with each aircraft in an environment where their number is arbitrary, and aircraft tracks need to be posted with regular time interval separation.

To address these challenges, a clustering and tracking algorithm is employed. The algorithm collects all radar hits that appear in a time window and generates a single consolidated data set for each time step, which is suitable for clustering algorithms. It identifies aircraft for each time step with a clustering method based on Minimum Spanning Tree,[2] and then constructs the trajectory for each aircraft by comparing identified vehicles between two consecutive time steps.

There are various clustering methods such as k-means clustering[3,4] and fuzzy c-means clustering.[5,6] These two clustering algorithms and their modified versions solve many clustering problems; however, they generally require a prior knowledge of the number of clusters and are therefore deficient since it is difficult

*Senior Systems/Simulation Engineer, UC Santa Cruz M/S 210-8, Moffett Field, CA 94035. chunki.park@nasa.gov
†Associate Research Scientist, UC Santa Cruz M/S 210-8, Moffett Field, CA 94035, AIAA Member. haktae.lee@nasa.gov
‡Director of Aerospace Systems, UC Santa Cruz M/S 210-8, Moffett Field, CA 94035. bassam.musaffar-1@nasa.gov

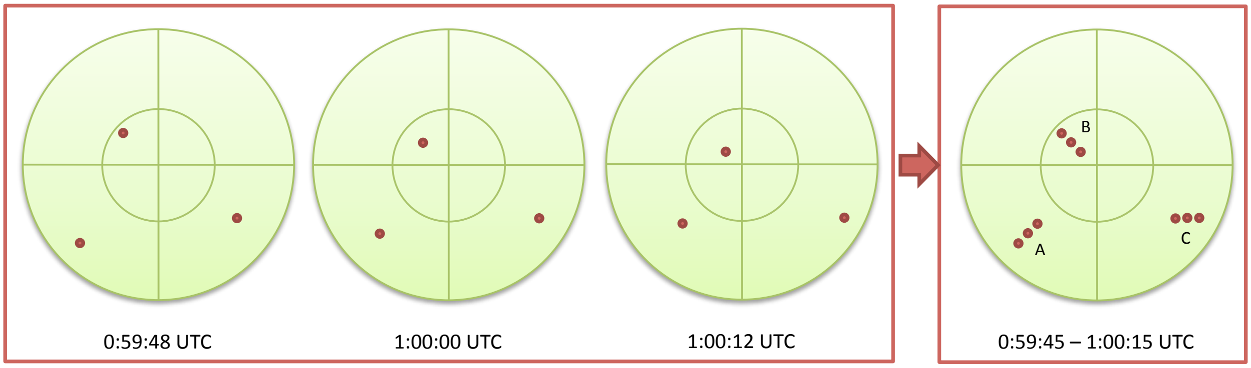

Figure 1. Three consecutive radar hits that have 12 second gaps between hits (left) and a collected radar hits for a time window (right)

to know in advance the number of aircraft in the airspace. The clustering algorithm proposed in this paper does not require a priori knowledge of the number of clusters. This paper has more in common with work on multiple-target tracking . The framework proposed in Ref. 7 can track and manage identities of multiple aircraft simultaneously using a modified version of joint probabilistic data association (JPDA) algorithm;[8,9] however, the main objective of research presented in Ref. 7 differs from the objective of this paper. While the JPDA based algorithm focuses on tracking multiple aircraft for air traffic controllers, the approach presented in this paper focuses on generating aircraft tracks in regular time interval separation, which are usable by modern air traffic analysis and simulation tools.

This paper is organized as follows: Section II presents the clustering and tracking algorithm; Section III describes experimental studies that evaluate the algorithm and show the tracking results with radar data; and Section IV provides conclusions.

II. Radar Data Tracking Algorithm

Suppose that there are three consecutive radar hits detected at 0:59:48 Coordinated Universal Time (UTC), 1:00:00 UTC, and 1:00:12 UTC respectively. These are depicted on the left hand side of Fig.1. Looking at these radar hits, humans may recognize three aircraft relatively easily. To enable automated tracking of each aircraft, the *time window* concept is employed. It is assumed that radar hits from same aircraft are closer in a time window compared to radar hits from different aircraft. Using a time window, groups of radar hits are generated. The radar hits of each group are likely to come from the same aircraft. For example, if the time window offset is ± 15 seconds at 1:00:00, all the radar hits that have time stamp between 0:59:45 and 1:00:15 falls into this time window as shown in the right hand side of Fig.1. Figure 1 shows three groups, one of which is located at the lower left (A), another around the center (B), and the other at lower right (C). Each group comprises three radar hits that are very likely to represent the same aircraft. The aircraft tracking method divides unidentified radar hits within a given time window to multiple groups, and then compares these groups with those in the next consecutive time window to determine whether each group in the new time window is a continuation of a group in the previous time window or not. This procedure is repeated to obtain the complete set of aircraft tracks.

A. Clustering Aircraft within a Time Window

Before presenting the clustering algorithm, the fundamentals of graph theory[10] are reviewed. A graph is defined as a collection of points and lines connecting subsets of these points. Points in a graph are commonly known as vertices or nodes. Lines connecting vertices of a graph are known as edges or links. An edge-weighted linear graph is composed of a set of nodes and a set of edges with a value assigned to each edge as weight. A circuit is a closed path composed of a sequence of edges from a node to itself. A connected graph has paths between any pair of nodes. A tree is a connected graph with no circuits and a spanning tree of a connected graph G is a tree in G which contains all nodes of G.

If the weight of a tree is defined as the sum of weights of its edges, a minimum spanning tree (MST) of

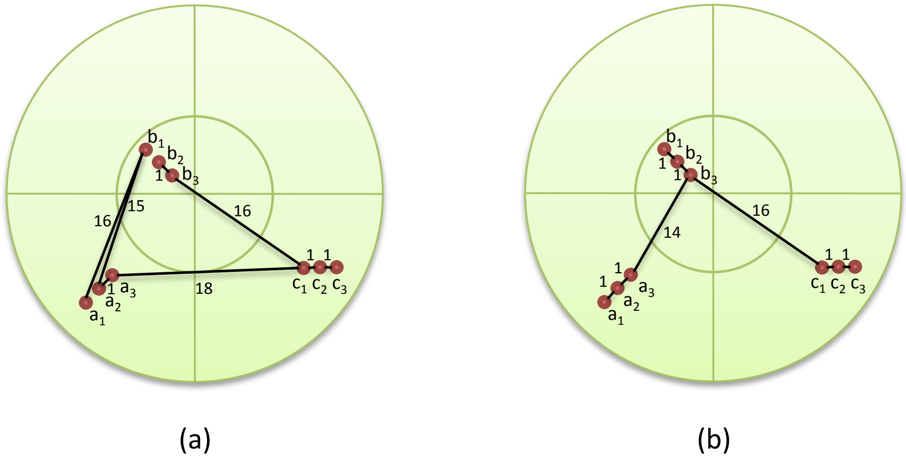

Figure 2. Two sample spanning trees: (a) is one among many possible spanning trees, and (b) is the minimum spanning tree

graph G is defined as the spanning tree that has the minimum weight among all possible spanning trees of G. For example, in Fig.1, let the radar hits in the time window be nodes and let all possible connections between any two hits be edges. Figure 2 shows two possible spanning trees: (a) is one of many possible spanning trees, and (b) is the minimum spanning tree. After finding the minimum spanning tree,[2,10] the clustering algorithm needs to identify and delete edges connecting any two distinct groups such as edges $\overline{a_3b_3}$ and $\overline{b_3c_1}$ shown in Fig.2 (b). In this example, edges $\overline{a_3b_3}$ and $\overline{b_3c_1}$ have the weights of 14 and 16 respectively. These weights are much larger than the weights of all the other edges, which are about one. The clustering algorithm automatically deletes edges whose weights are larger than a given maximum permissible distance, L_{max}, and then further removes edges with weights larger than the average edge weight, m_L, by more than three times the standard deviation, σ_L. That is, the edge cut-off criteria is

$$
\begin{gathered}
d_i > L_{max} \\
\text{or} \\
d_i > m_L + 3 * \sigma_L,
\end{gathered}
$$

where d_i is the length of the ith edge. Once the MST is separated into sub-trees by trimming the long edges, this process is repeated recursively to the resulting sub-trees until there is no abnormal edge within each tree. After the algorithm finds several independent trees, it labels each tree with a distinct aircraft ID, and interpolates the position. Figure 3 describes the core clustering procedure.

B. Comparing Aircraft in Two Consecutive Time Windows

To relate a group (cluster) of newly identified aircraft in a time window to the one in the consecutive time window, the algorithm counts the number of overlapping radar hits between a cluster in the current time window and all clusters in the next time window. This procedure requires the proper size of the time window that is at least larger than the time spacing between two radar hits of a vehicle, since a hit should be included in two consecutive time windows. The cluster in the next time window with the highest number of overlapping radar hit gets the ID of the cluster in the current time window. If there is no overlapped aircraft, the cluster gets a new unique ID. After all clusters in a time window obtain their IDs, the position of each aircraft is calculated by interpolating the radar hits in the time window.

In addition to the overlapping radar hits, two more approaches are used to improve the comparison ability. First, the aircraft headings in consecutive time windows are compared. If the difference between headings of two aircraft is within a certain set threshold, Θ_h, the two aircraft are classified to be the same. Otherwise, the two aircraft are classified to be different group even though the number overlapping radar hits is high. Second, the expected position of an aircraft in the next time window is compared to its real position. If the two positions are within a given threshold, Θ_p, they are determined to be the same aircraft.

$tree\_queue \leftarrow \{\}$
$unidentified\_radar\_hits \leftarrow$ COLLECT_RADAR_HITS_IN_TIME_WINDOW($entire\_radar\_hits$)
$edges \leftarrow$ CALCULATE_ALL_EDGE_LENGTHS($unidentified\_radar\_hits$)
$minimum\_spanning\_tree \leftarrow$ KRUSKAL_METHOD($edges$)^2
add $minimum\_spanning\_tree$ **to** $tree\_queue$
repeat
  $pre\_num\_of\_trees \leftarrow$ SIZE($tree\_queue$)
  $new\_tree\_queue \leftarrow \{\}$
  for all $tree$ **in** $tree\_queue$ **do**
    $m_L \leftarrow$ CALCULATE_AVERAGE_OF_EDGES
    $\sigma_L \leftarrow$ CALCULATE_STANDARD_DEVIATION_OF_EDGES
    for all $edge$ **in** $tree$ **do**
      TRIM_EDGES_LARGER_THAN(MIN($L_{max}, m_L + 3\sigma_L$))
    end for
    $divided\_trees \leftarrow$ GET_DIVIDED_TREES($tree$)
    add $divided\_trees$ **to** new_tree_queue
  end for
  $tree\_queue \leftarrow new\_tree\_queue$
  $post\_num\_of\_trees \leftarrow$ SIZE($tree\_queue$)
until ($post\_num\_of\_trees - pre\_num\_of\_trees$)$\leq 0$

Figure 3. MST-based clustering algorithm

Otherwise, they are classified as different aircraft. The number of overlapping radar hits is first checked, then these two methods are used to verify the result. If at least one of two methods confirms that the two aircraft are identical, the aircraft is classified to be the same.

One more challenge lies in tracking aircraft correctly when the Mode 3 transponder codes of aircraft intermittently drop out. If the radar hit has the Mode 3 code, a clustering algorithm is not required to track the aircraft. However, if the code is intermittently unavailable, the aircraft is assigned a new ID by the clustering algorithm. To avoid splitting aircraft trajectories for the same aircraft, the algorithm compares the actual position of a vehicle without the Mode 3 code in the current time window to the expected next positions of all vehicles with the Mode 3 codes in the previous time window, and vise versa. Through this cross-comparison routine, continuous tracks of aircraft are obtained.

III. Experimental Evaluation

The method was verified using a radar data set around the Grand Forks Air Force Base, which was provided by the Air Force. The data set included all radar hits collected from six different radar sites. In this experiment, radar hits for a single day, a total of 116,790 radar hits, were used. Around 23 percent of these hits did not have a Mode 3 transponder codes, that is, 26,914 radar hits were unidentified and required clustering. Figure 4 depicts examples of radar hits for 24 consecutive times. This figure shows that the time spacing for the radar hits are irregular, and that it is difficult to differentiate multiple aircraft in those separate plots. Figure 5 shows the time window plot that includes all radar hits in Fig.4.

The current recursive MST-based method used a two-minute time window (± 60 seconds from current time) with a ten-second step time, and calculated the interpolated positions of clustered aircraft at each time. The size of the time window should be not only large enough to collect radar hits even when there were missing hits, but also small enough to avoid mixing the radar hits of different aircraft. The size of the time window, ± 60 seconds, was empirically determined after evaluating the algorithm with various time window sizes. Typical minimum separation standards for conflict are 5 nautical miles horizontally and 1000 feet vertically. Since horizontal motion is much greater than vertical motion, the vertical scale was adjusted before the Euclidean distances were calculated. The vertical distance was scaled up so that vertical 1000 feet was considered to be the same as horizontal 5 nautical miles. The minimum separation standards provide

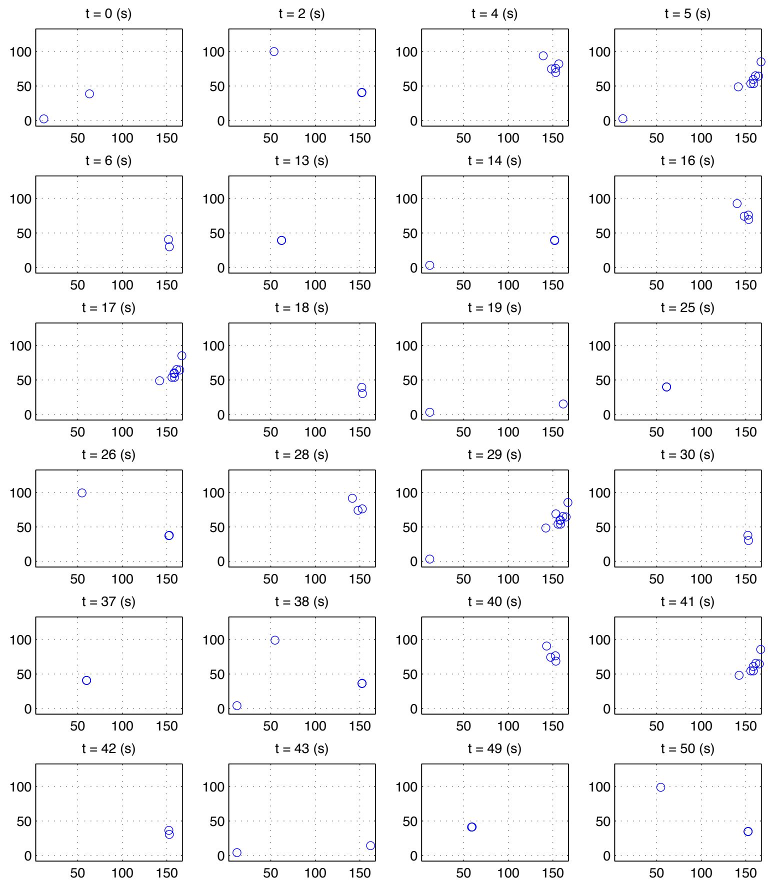

Figure 4. Examples of radar hits in the x-y coordinate for 24 consecutive times

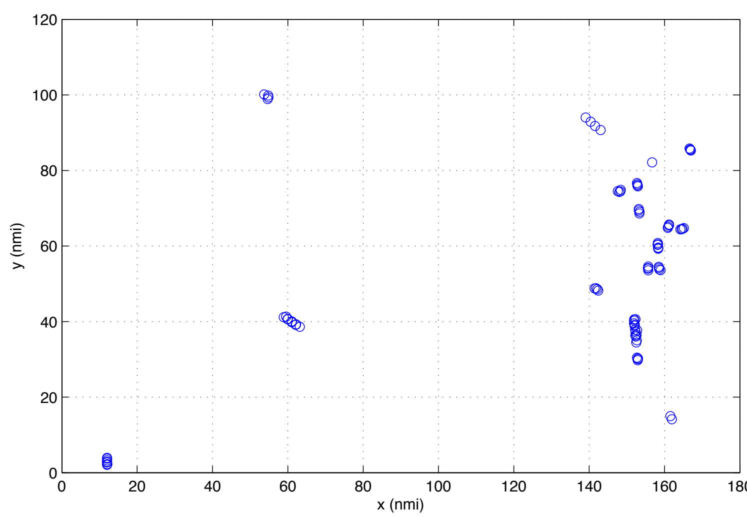

Figure 5. Time window plot that includes all radar hits in Fig.4

a reasonable scaling factor since it should be based on maximum unexpected movements. L_{max} was set to 5 nautical miles that is the typical horizontal separation distance. After clustering unidentified aircraft of all time windows, the method compared clustered aircraft in two consecutive time windows, and generated trajectories of aircraft. Θ_h was set to 90 °, which ensures that two groups in the consecutive time windows head to at least the same direction. Θ_p was set to 5 nautical miles.

A. Evaluating the Algorithm

To evaluate the MST-based clustering and tracking algorithm, trajectories of two aircraft with known Mode 3 transponder codes were considered. The Mode 3 information was not used by the algorithm. The estimated trajectories generated by the algorithm were compared with the known trajectories to validate the algorithm. One aircraft, the Mode 3 code of which is 4260, was randomly selected from the commercial flights that are flying around FL300, and the other, the Mode 3 code of which is 307, was selected from the low altitude flights.

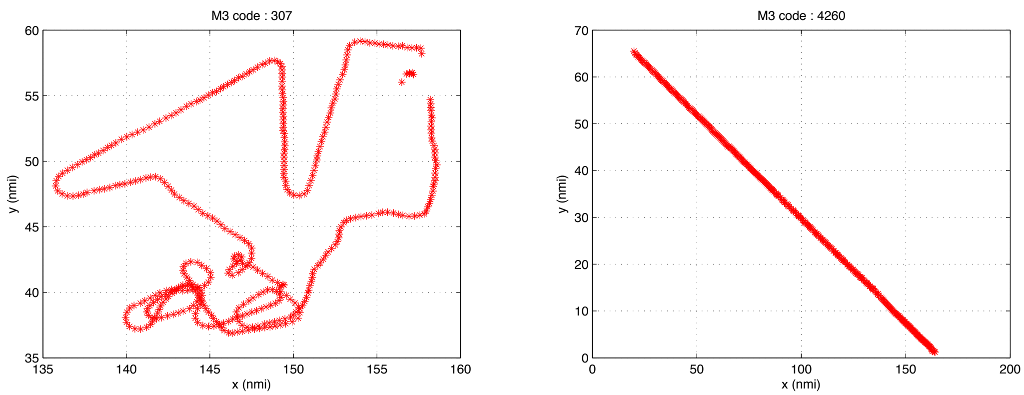

Figure 6. The trajectories of two aircraft, one of which was randomly selected from the low altitude flights (on the left side), and the other was selected from the regular commercial flights flying around FL300 (on the right side).

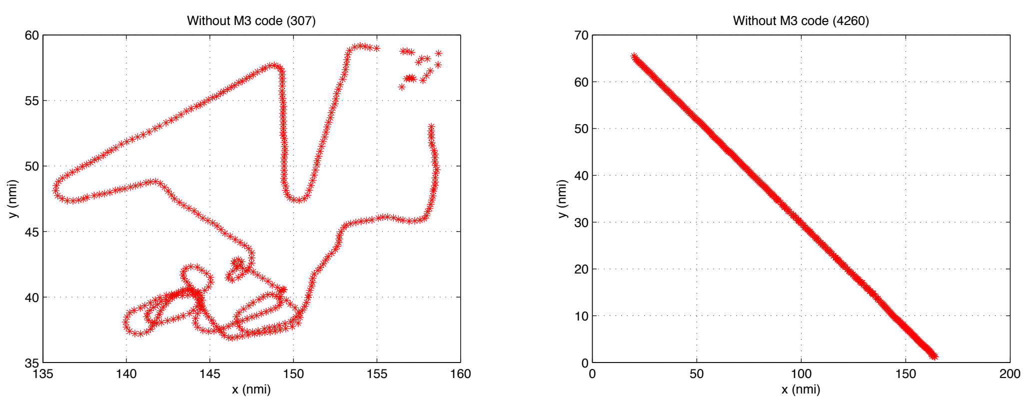

Figure 7. The trajectories of two aircraft that the MST-based clustering algorithm regenerated from the radar data including the radar hits of two aircraft in Fig. 6 from which the Mode 3 codes were removed.

Figure 6 shows the original trajectory of the two aircraft, and Fig. 7 shows the trajectory that the algorithm generated from the radar data including the radar hits from which the Mode 3 codes of two aircraft were removed. For the commercial flight (Mode 3 code 4260), the algorithm perfectly regenerated the flight trajectory from the unidentified radar hits as shown on the right sides of Figs. 6 and 7. For the low altitude flight (Mode 3 code 307), the algorithm also regenerated the flight trajectory almost perfectly. As shown on the left sides of Figs. 6 and 7, only the top-right parts of the plots are slightly different. Since there were two or three more unidentified aircraft that were flying close to each other in this airspace, the clustering algorithm could not track aircraft perfectly; however, the 97.8 % of data points on the original trajectory are completely fit to the clustered trajectory. This experiment shows that the proposed clustering and tracking algorithm can generate the aircraft tracks from unidentified radar hits.

B. Executing the Algorithm for the Radar Data

The algorithm was executed for the data set that contains 116,790 radar hits. It computed 90,921 interpolated aircraft positions from radar data and generated 1,029 aircraft trajectories. Table 1 summarizes the tracking results. The estimated trajectories are classified into three groups: trajectories that have Mode 3 information, trajectories that do not have Mode 3 information, and trajectories that intermittently have Mode 3 information. Some examples are shown in Fig. 8. These results demonstrate that the MST-based approach was able to track unidentified aircraft.

Table 1. Results of clustering and tracking aircraft from radar data

Case	Number of Clusters	Number of Interpolated Radar Hits
Mode 3 codes **on**	332	31,892
Mode 3 codes **off**	332	8,921
Mode 3 codes **on/off**	365	27,979 (Mode 3 on) 22,129 (Mode 3 off)
Total	1,029	90,921

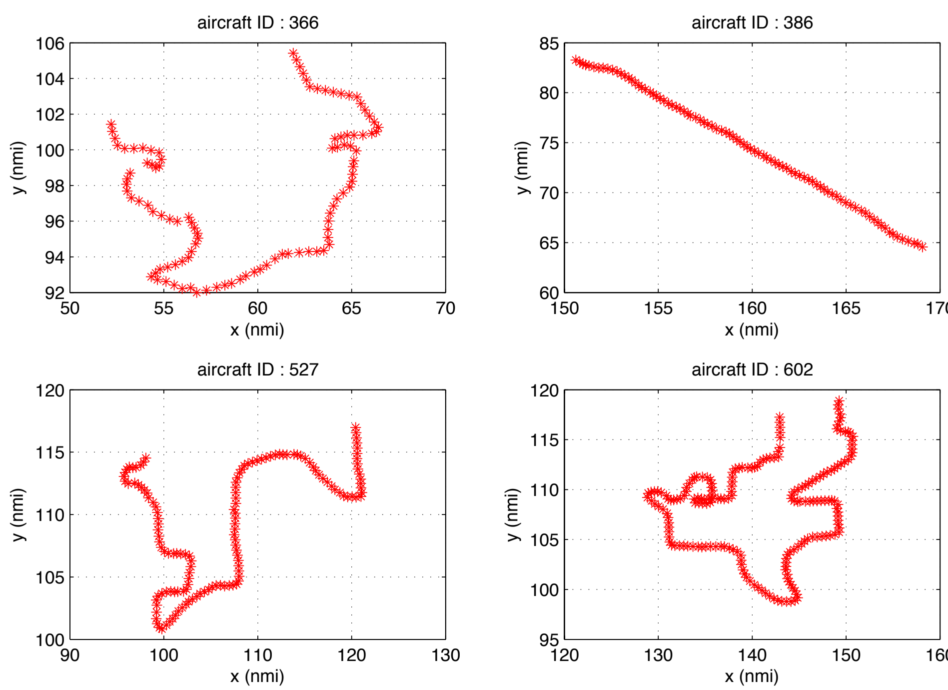

Figure 8. Examples of aircraft trajectories which the MST-based clustering algorithm generated from radar data without Mode 3 codes

IV. Concluding Remarks

This paper describes an approach for tracking multiple unidentified aircraft in radar data. The proposed recursive MST-based clustering algorithm enables identification of independent tracks as belonging to the trajectory of an aircraft whose identity is not known. This paper also presents a three-step method to compare clustered aircraft in two consecutive time windows to generate trajectories of aircraft, and reports evidence that the method regenerates the aircraft trajectory from the radar hits from which the Mode 3 transponder codes were removed to make them unidentified aircraft. The experimental result with the radar data of a single day shows that the method can track unidentified aircraft.

Although this method generates plausible aircraft trajectories from the radar data that contain unknown aircraft, there is room for improvement. The method currently uses altitude information of aircraft as well as the horizontal position information. Finding a way to consider the radar hits that do not provide altitude information could lead to the more robust tracking algorithm. Future research also includes investigating the sensitivity of the algorithm with respect to the distance between two neighboring aircraft.

Acknowledgments

The authors would like to thank Dr. Dallas Denery and Mr. Duc Trong Tran at NASA Ames Research Center and Dr. Jeffrey Bauer at NASA Dryden Research center for providing overall guidance and coordination with the US Air Force. Radar data provided by the US Air Force were invaluable for this study and the help from Lt. Col. Dallas Brooks, Lt. Col. Anthony Militello, Mr. James Evans, and 2nd Lt. Arnold Cline are greatly appreciated.

References

[1]Lacher, A., Zeitlin, A., Maroney, D., Markin, K., Ludwig, D., and Boyd, J., "Airspace Integration Alternatives for Unmanned Aircraft," Tech. rep., The MITRE Corporation, Feb. 2010.

[2]Kruskal, J. B., "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem," *Proceedings of the American Mathematical Society*, Vol. 7, No. 1, February 1956, pp. 48–50.

[3]MacQueen, J. B., "Some Methods for Classification and Analysis of MultiVariate Observations," *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, edited by L. M. L. Cam and J. Neyman, Vol. 1, University of California Press, 1967, pp. 281–297.

[4]Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y., "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7, July 2002, pp. 881–892.

[5]Bezdek, J. C., *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[6]Hussien, B., McLaren, R., and Bleha, S., "An application of fuzzy algorithms in a computer access security system," *Pattern Recognition Letters*, Vol. 9, No. 1, 1989, pp. 39 – 43.

[7]Hwang, I., Balakrishnan, H., Roy, K., and Tomlin, C., "Multiple-Target Tracking and Identity Management with Application to Aircraft Tracking," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 3, May 2007, pp. 641–652.

[8]Fitzgerald, R. J., "Development of Practical PDA Logic for Multitarget Tracking by Microprocessor," *Proceedings of the American Control Conference*, june 1986, pp. 889 –898.

[9]Zhou, B. and Bose, N., "Multitarget tracking in clutter: fast algorithms for data association," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 29, No. 2, apr 1993, pp. 352–363.

[10]Zahn, C. T., "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," *IEEE Transactions on Computers*, Vol. 20, No. 1, 1971, pp. 68–86.