Generalized Testing and Analysis of Storm Cell Tracking Algorithms

B. Root*, M. Yeary, and T.-Y. Yu

School of Meteorology, Norman, Oklahoma

Atmospheric Radar Research Center, Norman, Oklahoma

School of Electrical & Computer Engineering, Norman, Oklahoma

ABSTRACT

An organized and objective method for testing any centroid-based storm cell tracking algorithm is investigated. Several algorithms are currently used for these purposes, however, efficacy studies for these tracking algorithms have been of marginal use. Limitations such as the number of case studies, arbitrary metrics, and subjective analyzes restrict the scope and applicability of the methods used to determine efficacy. An objective, generalized method for testing and scoring any tracking algorithm is necessary for quantitative comparisons. It would also have an added benefit of allowing experiments to be constructed to study the behavioral characteristics of the algorithms in different environmental scenarios.

A system for testing and scoring any centroid-based storm cell tracking algorithm is described. The system simulates the track of the centroid of many storm cells in configurable scenarios. Tracking algorithms are then applied to the simulated data and the system uses this truth data to score the results. Several track scenarios, both idealized and realistic, are examined. The utility of this testing system for the purpose of determining an optimal tracking algorithm is discussed.

1. Introduction

a. Motivation

Storm cell tracking is an important component for storm evolution studies, real-time nowcasting and severe weather warning systems. The testing and evaluation of tracking algorithms is necessary for the optimal performance of these systems. Efficacy studies for these algorithms have involved both automated and human-made identification of radar features. While these studies are useful, it is impossible to fully isolate the impact the identification process has on the skill of the trackers. In addition, human analyzes are often limited in quantity, subject to operator errors, and can have differences in results due to different people performing analyzes.

This project seeks objectivity by analyzing simulated storm tracks. The storm tracks are presented to the tracking algorithms as identified radar features – wholly independent of any identification process. This approach has several benefits. First and foremost, the storm tracks are, by definition, a 'known' quantity in this analysis which allows for the use of established model skill analysis techniques. Second, all tracking algorithms receive the same input data, therefore, an 'apples-to-apples' comparison can be made between algorithms. Third, because of the ease of generating simulated tracks, the tracking algorithms can be extensively studied in a variety of storm scenarios. It is the goal of the project to produce a comprehensive and robust testing suite for centroid-based tracking algorithms.

b. Definitions

An 'object' is the term for a real-world body that is desired to be identified and tracked. A 'feature' is the computer vision of an 'object,' usually obtained from some sort of identification process. A key difference between an object and a feature is that a feature could be the result of noise in the data, while an object physically exists. The ideal identification process would produce a feature that perfectly corresponds with each object with no extra features and no un-identified objects. $Z(t)$ is the set of features reported for time t.

An 'association' is the joining of two features across time, while a feature that is not joined with any other feature is a 'non-association.' A 'track' is the set of associated features, at most one from the set $Z(t)$ for each t. Ideally, a track would contain only the features for a particular object. In other words, for each real-world object that exists in time, there would be a track that contains all of the features for that object with no features from other objects.

Because real data will inevitably contain false identifications of non-existent objects, the definition of a track should also be expanded to allow for tracks of zero length called 'false alarms.' These false alarms are also considered non-associations because they are not part of any object's track.

A 'hypothesis' is the set of tracks that covers the entire set of features for all t. For an arbitrary set of $Z(t)$, it is possible there exists multiple hypotheses that satisfy the data constraints.

* *Corresponding author address:* Benjamin Root, School of Meteorology and Atmospheric Radar Research Center at The University of Oklahoma, 120 David L. Boren Blvd., Suite 5900, Norman, OK, 73072
E-mail: ben.root@ou.edu

c. *Algorithms*

Any algorithm to be used for storm cell tracking must satisfy certain requirements. First, the algorithm must not require *a priori* knowledge of the number of tracks. Second, tracks must be able to enter and exit the domain. Third, tracks may be initiated and terminated at any time, at any location. Lastly, tracks may evolve independently of each other (i.e., no rigid-body assumptions). These requirements ought to be self-evident when considering the observed behaviors of unorganized, scattered storm cells and organized storms moving with a passing frontal boundary.

While a multitude of tracking algorithms could be studied with this approach of simulating storm tracks, only two particular algorithms were examined in this exploratory study: the Storm Cell Identification and Tracking (SCIT) and the Multiple Hypothesis Tracker (MHT) algorithms. Future work will examine other storm-cell tracking algorithms in a similar fashion.

1) SCIT

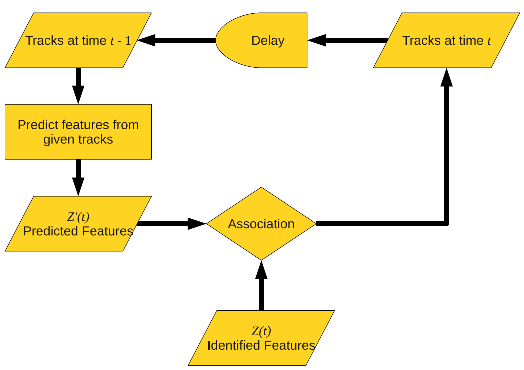

FIG. 1. Flowchart of the SCIT tracking algorithm.

The SCIT algorithm (Johnson et al. 1998) was chosen for study because it is currently used as the storm tracker for the WSR-88D system. SCIT satisfies the requirements mentioned above. This algorithm uses a greedy, nearest-neighbor approach to perform associations of the centroid of the identified features across time. At each iteration, a storm track has a predicted location for where the algorithm expects to find a feature. Each of the identified features at time t is associated with the track that has the closest predicted feature, without exceeding a distance threshold. For each identified feature that was left unassociated, new tracks are initiated. This process is diagrammed in Figure 1.

2) Multiple Hypothesis Tracking

The MHT algorithm (Cox and Hingorani 1996) considers the time-associations globally, in contrast to the sequential, order-dependent approach of SCIT. MHT is also capable of 'correcting' its tracking decisions in subsequent iterations as more data is available, which is a desirable property for adaptive sensing and historical reanalyzes. While not used for this study, MHT can also utilize additional information about the feature to reduce tracking ambiguities (i.e., 'texture' data).

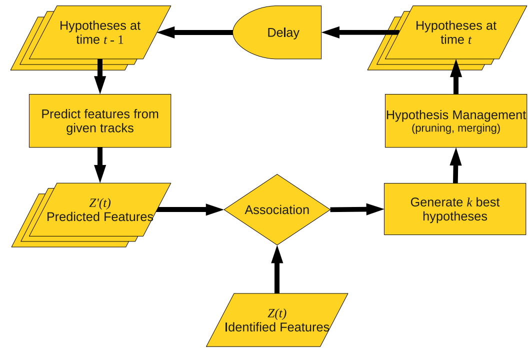

FIG. 2. Flowchart of the MHT tracking algorithm.

MHT treats the tracking problem as a global maximum finding problem. At each iteration, the MHT algorithm updates a list of hypothetical tracks, which are sorted by their calculated likelihoods. To limit the combinatorial explosion that can occur, only the k-best hypotheses are generated. Also, MHT employs techniques to 'prune away' the most unlikely hypotheses. This process is diagrammed in Figure 2.

2. Method

a. *Source Code*

This project seeks to analyze the tracking algorithms, independent of any identification methods. The original code for the SCIT and MHT algorithms come from the NWS's website for the WSR-88D Common Operations and Development Environment and Ingemar Cox's website, respectively. The tracking portion of SCIT was reimplemented as a Python module. The MHT source code, written in C++, required minor changes for storm cell tracking purposes. Both of these modules are available on-line[1] and are maintained by this author.

b. *Simulation*

For the purpose of rigorously analyzing the tracking algorithms, a track simulator called "ZigZag" was created to produce a variety of configurable scenarios. The 'storm' tracks generated by ZigZag are essentially random walks. These tracks are initialized with a random starting location, speed, and direction. The duration of the storm

[1] Available at: `https://github.com/WeatherGod`

Model	Reality	
	Correct Association H	False Association F
	Missed Association M	Correct Non-Association N

FIG. 3. Contingency table depicting the classifications that each line segment can be assigned. The number of segments in each category is used for calculating the skill score.

tracks are also randomly determined. ZigZag can also create tracks that split off of and/or merge into existing tracks.

Within a single scenario, a variety of track types can be added. For example, ZigZag can create 20 tracks that mimic the behavior of squall lines. In addition to those 20 tracks, ZigZag can simulate ground clutter features that do not move. The positioning of these components can be controlled to study their impact on the tracking results.

ZigZag can apply various noise effects to the tracks of a scenario. For example, 'occlusions' (or 'false-mergers') of nearby storm cells can be simulated by removing a feature from one of the occluding simulated tracks. Spurious features can be added at random locations and times. Also, the positions of each feature in a track can be jittered according to a given noise model.

c. *Measuring Results*

By breaking up the tracks into a list of associations and non-associations, a contingency table for the decisions of the tracker can be made. For each true association created by ZigZag, a 'hit' (H) is made when the tracker made the exact same association. A 'miss' (M) is when the tracker failed to make that association. For each truly unassociated feature created by ZigZag, a 'correct null' (N) is counted when the tracker also determined that the feature was not a part of any track. Conversely, it is considered to be a 'false alarm' (F) if the tracker associated that feature with a track[2]. A contingency table is shown in Figure 3.

This project used 'Percent Correct' (PC) for measuring the performance of the trackers. For each tracking run in a simulation, the contingency table can be tabulated. By knowing the number of hits, misses, false alarms and correct nulls, one can use (1) to determine what percentage of tracking decisions made were correct.

$$PC = \frac{(H + N)}{(H + M + F + N)}. \qquad (1)$$

Values range from 0 (worst) to 1 (best).

d. *Analyzing Trackers*

With full control over the characteristics of the simulated scenarios and a method of measuring the performance of a tracker, one can analyze and compare many different aspects of tracking algorithms. For example, it is possible to estimate the optimal tracking parameters of a tracker for a given scenario. Also, it is possible to study how well a tracker performs in different scenarios. In particular, one can determine how well the performance of a tracker is maintained over changing conditions.

To prepare this study, both MHT and SCIT were applied to the simulations many times with varying parameters. As an example of how to optimally tune a tracker, we will demonstrate this approach for SCIT. SCIT has only one parameter – speed threshold – which is used to calculate the maximum distance an object can be from its predicted location. If no features are found within this maximum range, then that particular track is terminated.

MHT has many parameters, but only five were examined for this project. This optimization study was covered in a separate, concurrent study.

Using the optimal parameters determined for each tracker, one can then study the trackers in a variety of different scenarios. For this project, four scenarios were created with progressively more difficult properties. All scenarios represent thirty minutes of motion at one frame per minute.

The 'cleanest' scenario consisted of a squall line with twenty-five tracks moving at speeds between 1.0 and 2.5 km per minute (approximately 35 to 90 miles per hour). These tracks were initialized from an ellipse-shaped region with a minor axis length of 25 km and a major axis length of 125 km that moves slowly at about 0.5 kilometers per minute (approximately 20 miles per hour). These main squall line tracks have a 10% probability of termination at any time. In addition to the twenty-five main tracks, two additional tracks were created to merge into the existing tracks, as well as two tracks to split off of main tracks. A couple of simulated features were dropped due to false apparent mergers, as well as 1% of randomly chosen features in order to simulate identification dropouts. Lastly, the positions of the simulated tracks were jittered randomly with a position variance of half a kilometer.

The second scenario uses the same settings as the first simulation, but also includes twenty spuriously created features that are uniformly distributed in time and uniformly distributed within the squall line region. The third scenario uses the same settings as the first simulation, but also includes sixteen persistent clutter objects located at the heart of the squall line. These clutter objects were normally distributed over a radius of about ten kilometers. The 'dirtiest' (and Presumably hardest) scenario consisted of all parts of the first three scenarios.

For each of these scenarios, ten simulations were generated. The simulation count was kept low for the purpose of expediting the process of parameter optimization. More simulations are highly recommended for adequate statisti-

[2]A false alarm in this context is different from the false alarm mentioned previously in the Definitions section.

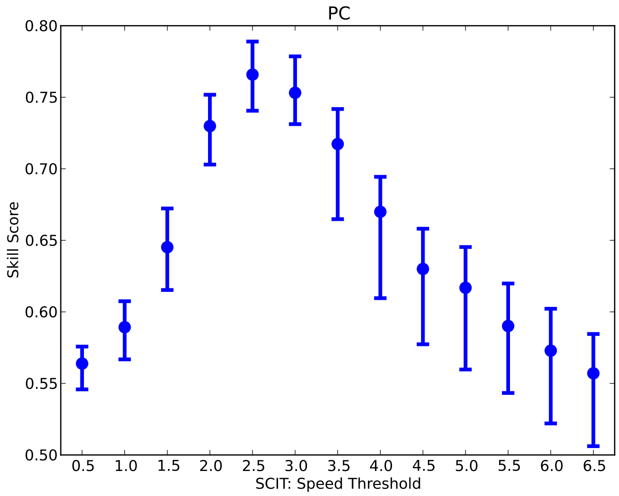

FIG. 4. The Percent Correct measure for SCIT when applied to the same scenario, but with increasing values for its speed threshold parameter. The 95% confidence interval for the average skill score is depicted by the error bars.

cal analysis. Using the optimized parameters determined previously, the tracking algorithms were then applied to each of these simulations and the results tabulated.

3. Results & Discussions

Parameters for both MHT and SCIT were studied for determining optimal performance. The optimization study for MHT can be found in a concurrent paper (Root et al. 2011). The parameter values used for MHT are listed in Table 1.

TABLE 1. Parameters used for the MHT algorithm

Probability of Detection	0.9
Mean Rate of False Alarms	0.0002
Mean Rate of New Tracks	0.001
λ_x	30.0
Initial Estimate of Velocity Variance	1.0

The best score for SCIT in Figure 4 was for a speed threshold of 2.5 km per minute. This makes sense as because the simulated tracks were initialized with speeds ranging from 1.0 to 2.5 km per minute. If the speed threshold was set too high, then spurious points may get associated into a nearby track and terminated tracks might be incorrectly continued onto a nearby track. Conversely, if the speed threshold was set too low, then tracks would be prematurely terminated if the object moves too quickly or erratically. This is depicted in Figure 5. The track plots on the left (lower speed thresholds) has many gray lines depicting missed associations. While the track plots on the right (higher speed thresholds) has many red lines depicting incorrect associations.

A sample of four tracking results – one from each scenario tested – is shown in Figures 6, 7, 8, and 9. In each figure, the output from each algorithm – SCIT on the left, MHT on the right – operating on the same input is shown. In these tracking figures, green markings depicts a correct tracking decision by the respective algorithm. A false association is depicted by a red line, while a dashed, gray line depicts a missed association that should have been made by the tracker. The average, bootstrapped skill scores and the 95% bias-corrected, accelerated (BCa) confidence intervals (Efron and Tibshirani 1993) are depicted in Figure 10.

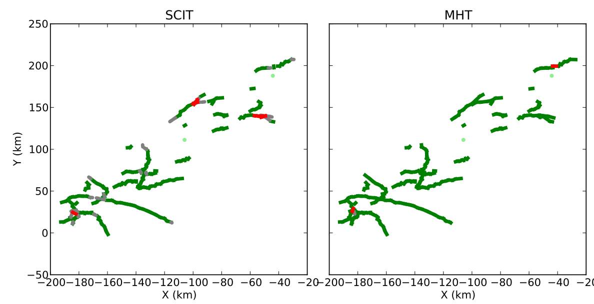

FIG. 6. A tracking result from a simulation in the squall-line scenario that does not have spurious points and does not have clutter objects. This should be the 'easiest' scenario for trackers to process of the four scenarios tested.

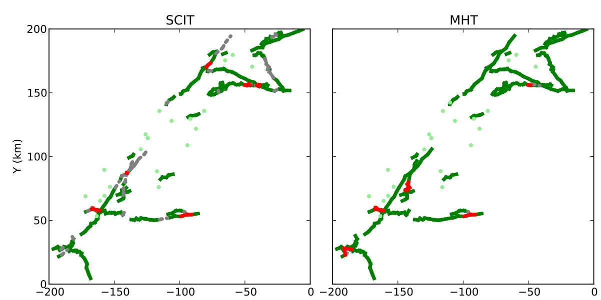

FIG. 7. A tracking result from a simulation in the squall-line scenario with spurious features, but no clutter objects.

According to these measurements of PC, MHT represented a significant improvement over SCIT. MHT consistently performed with 95% accuracy or better, while SCIT was returning results that were correct between 75% and 90% of the time. SCIT performed best when the objects being tracked moved with uniform speed (i.e., no clutter objects), consistently hitting between 85% to 90% accuracy. SCIT did not appear to be significantly impacted by

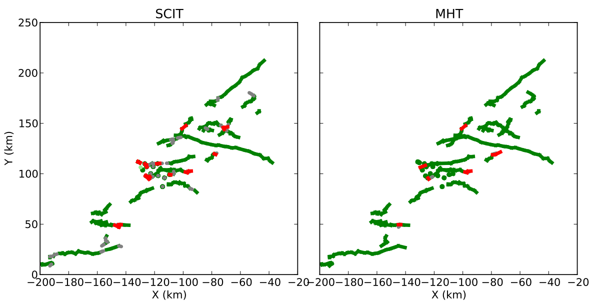

FIG. 8. A tracking result from a simulation in the squall-line scenario with clutter objects, but without spurious features.

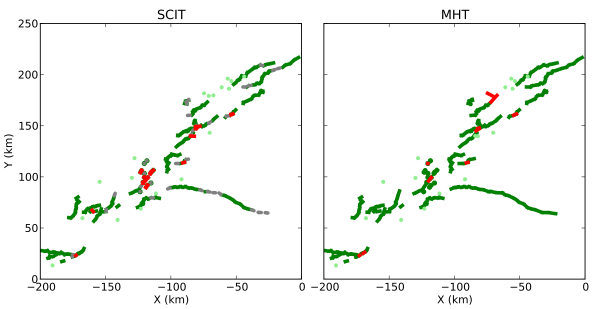

FIG. 9. A tracking result from a simulation in the squall-line scenario with both spurious points and clutter objects. Of the four scenarios tested, this should be the 'hardest' scenario to track.

spurious points. However, having a wide range of speeds and direction (i.e., clutter objects) significantly hampered SCIT. When clutter objects were present, the accuracy of SCIT dropped by about ten percentage points. Meanwhile MHT actually improved slightly by about one percentage point, but is not statistically significant. MHT also did not appear to have difficulties with spurious points.

4. Conclusions

Both SCIT and MHT performed well over a variety of scenarios, consistently producing results that were more than 75% correct. However, SCIT suffered significantly in the presence of clutter. In comparison to MHT, SCIT was unable to track the objects as accurately. MHT is not constrained by hard limits such as the speed thresholds that parametrizes SCIT. This allows for a wider variety of motions to be tracked by the identified objects without significantly increasing the risk of incorrectly joining tracks together.

This project demonstrated a method of objectively assessing the efficacy of a tracking algorithm. This method can be used to guide parameter optimizations. In addition, by producing multiple simulations of storm tracks under various constraints, fair comparisons can be made between different tracking algorithms and different scenarios. This allows for thorough testing and analysis of tracking algorithms so that the best tracker can be used.

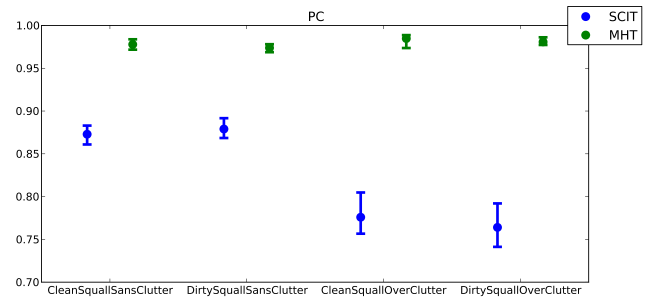

FIG. 10. Plot of the average Percent Correct of each tracking algorithm in each of the four simulation scenarios. The 95% confidence interval for the average skill score is depicted by the error bars.

Acknowledgments

This work was primarily supported by NOAA/NSSL under cooperative agreement of NA17RJ1227. We greatly appreciate the discussions and inputs from Michael Richman, and Valliappa Lakshmanan.

REFERENCES

Cox, I., 2009: Ingemar Cox Home Page. http://www.cs.ucl.ac.uk/staff/ingemar/.

Cox, I. and S. Hingorani, 1996: An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, **18 (2)**, 138–150.

Efron, B. and R. Tibshirani, 1993: *An Introduction to the Bootstrap*. 1st ed., Chapman & Hall/CRC.

Johnson, J. T., P. L. MacKeen, A. Witt, E. D. Mitchell, G. J. Stumpf, M. D. Eilts, and K. W. Thomas, 1998: The storm cell identification and tracking algorithm: An enhanced WSR-88D algorithm. *Wea. Forecasting*, **13 (2)**, 263–276.

National Weather Service, 2009: WSR-88D Common Operations and Development Environment. http://www.weather.gov/code88d/.

Root, B., M. Yeary, and T. Yu, 2011: Novel storm cell tracking with multiple hypothesis tracking. *27th Conference on International Interactive Information and Processing Systems*, Amer. Meteor. Soc., Seattle, WA.

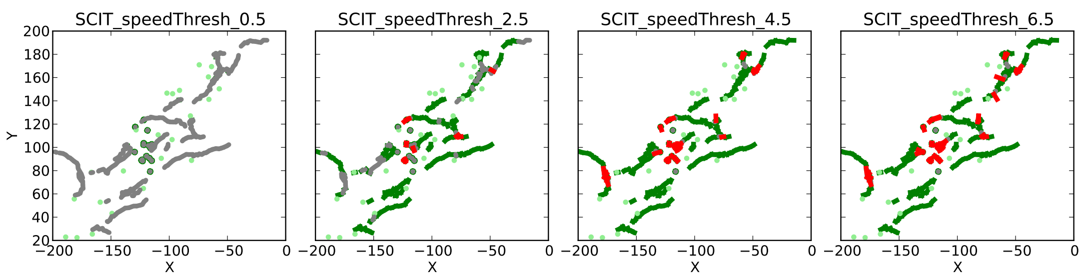

FIG. 5. Tracking results from SCIT on the same simulation, but with increasing values for its speed threshold.