Novel Storm Cell Tracking with Multiple Hypothesis Tracking

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

The Multiple Hypothesis Tracking (MHT) algorithm is introduced as a centroid-based storm cell tracker for radar imagery. MHT improves upon current algorithms such as SCIT (Storm-Cell Identification and Tracking) and TITAN (Thunderstorm Identification, Tracking, Analysis, and Nowcasting) by addressing problems with detection drop-outs, false mergers, and spurious detections. This algorithm has been widely used in the field of video processing, and has now been adapted for storm cell tracking.

The design of the MHT algorithm is explained and the features that it provides are discussed in the context of addressing these inherent difficulties. MHT is demonstrated in a variety of scenarios using simulated and real data derived from radar imagery. Applications for both real-time and post-processing studies are 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 search continues for more reliable tracking algorithms that would supersede the widely used SCIT (Storm-Cell Identification and Tracking) and TITAN (Thunderstorm Identification, Tracking, Analysis, and Nowcasting). While the current crop of algorithms fare well in many situations, there remains issues with tracking efficacy with regards to detection drop-outs, false mergers and spurious detections.

Multiple Hypothesis Tracking (MHT) (Cox and Hingorani 1996) improves upon current technologies by addressing these problems in its design. Widely used in the field of video processing and military target tracking, it has now been adapted for use in storm cell tracking. A key benefit of the MHT algorithm is that it is highly configurable and robust to input data. An understanding of the parameter's impact upon the behavior of MHT is critical for its use. Therefore, the goal of this project is to explore these parameters.

* *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

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. The 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. $Z(t)$ is the set of features reported for time t. A 'track' is the set of features, at most one from the set $Z(t)$ for each t. Ideally, a track would contain only the features for a particular object, or would contain only a single reported feature that occurred from noisy data. Lastly, a 'hypothesis' is a set of tracks which 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.

c. Multiple Hypothesis Tracking

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.

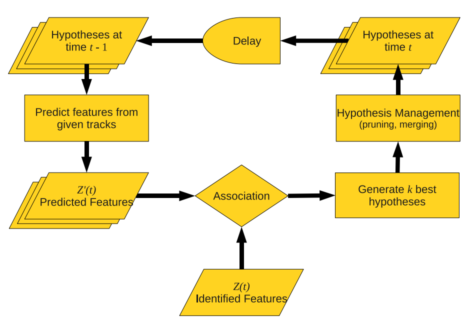

FIG. 1. Flowchart of the MHT tracking algorithm.

A multitude of tracking algorithms were reviewed, particularly the algorithms listed in the 2006 paper by Yilmaz, Javed, and Shah. The algorithm that satisfied the aforementioned requirements was "Multiple Hypothesis Tracking" (MHT) (Cox and Hingorani 1996). This tracking approach considers the time-associations globally, and is capable of 'correcting' its tracking decisions in subsequent iterations as more data is available. These are desirable properties for adaptive sensing and historical reanalyzes. While not used for this study, MHT can also utilize additional information about each feature to reduce tracking ambiguities (i.e., 'texture' data such as reflectivity).

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. The likelihoods are calculated as the conditional probability of a particular hypothesis, which explicitly models the probability of both spurious and non-spurious features. To limit the combinatorial explosion that occurs, only the k-best hypotheses are generated using Murty's method (Murty 1968). In addition, MHT employs techniques to 'prune away' remaining unlikely hypotheses. This algorithm is diagrammed in Figure 1.

2. Method

a. Source Code

The original source code for MHT was obtained from Ingemar Cox's website. The MHT source code, written in C++, required minor changes for storm cell tracking purposes. The MHT module and the tracker analysis system, ZigZag, are available on-line[1] and are maintained by the author.

b. Simulation

For the purpose of rigorously analyzing the MHT algorithm, the track simulator called 'ZigZag' (Root et al. 2011a) was used to generate many realistic tracks. The 'storm' tracks generated by ZigZag are essentially random walks. These tracks have random starting location, speed, and direction. The duration of the storm tracks is also randomly determined. ZigZag can also control many other aspects of the simulations, such as applying noise models and adding clutter objects.

For this parameter optimization study, the 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. The positions of these simulated tracks were then jittered randomly with a position variance of half a kilometer.

Next, the scenario simulates twenty spuriously created features that are uniformly distributed in time and uniformly distributed within the same squall line region. Last, the simulates 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.

For this scenario, 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 statistical analysis.

c. Parameters

The MHT algorithm has no fewer than fifteen parameters that control the various components within the algorithm. This many parameters are far too many to study each in detail for this project. However, several of these parameters can be excluded or simply kept at their default values as set by the original authors of the algorithm. For example, some parameters are related to the handling of the texture data, which have been excluded for this study. Other parameters that have been kept at their defaults are related to the control of the Kalman filter and the hypothesis pruning used for efficiency purposes. Four parameters remained for study:

1 **Probability of Detection**
Estimated probability for correct object identification

2 **Mean Rate of New Tracks**
Estimated number of newly established tracks per frame

3 **Mean Rate of False Alarms**
Estimated rate of spurious feature identification

4 $\lambda_{\mathbf{x}}$
Probability of track termination

In addition to these four parameters, it was found that the default value for the initial velocity variance estimate parameter was two orders of magnitude too large for this problem. In general, the initial value of this value should not significantly impact the tracking results because MHT should converge on the proper velocity variance estimate. However, it was found that by setting this parameter to

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

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

FIG. 2. 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.

one (as oppose to 200), the tracking results contained significantly fewer spurious track assignments.

d. Measuring Results

By breaking up the tracks from a tracking algorithm and the known tracks into lists 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 an identical 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 associated with any other feature. 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 2.

This project used 'Percent Correct' (PC) for measuring the performance of the trackers.

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

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

3. Results & Discussions

The average, bootstrapped skill scores and the 95% bias-corrected, accelerated (BCa) confidence intervals (Efron and Tibshirani 1993) are depicted in Figures 3, 4, 5, and 6.

First, the probability of detection (POD) parameter for MHT was examined. Values of ranging from 0.15 through 0.99 were tested. Throughout this range, MHT consistently scored an average of 95% correct with its tracking decisions. Setting the POD value too low resulted in more misses, while setting it too large increased the number of false associations. For this scenario, a POD value between 0.8 and 0.9 seems optimal, but fine-tuning this parameter may not be needed.

Next, the mean rate of new tracks (NTPS) parameter was examined. Note that this name might be misleading and further examination of the algorithm's source code is needed to determine the exact nature of this parameter.

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

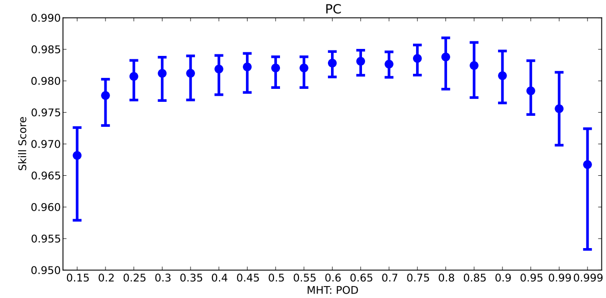

FIG. 3. Tracking scores for MHT with changing values of the probability of detection. The 95% confidence interval for the average PC value is depicted by the error bars.

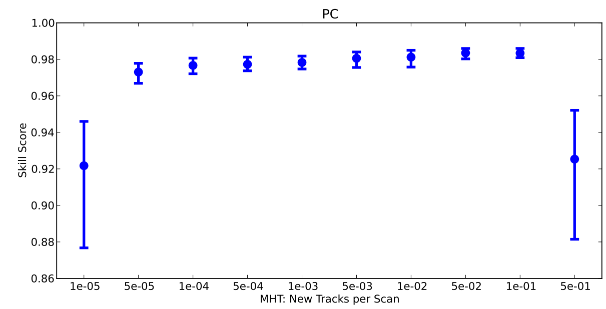

FIG. 4. Tracking scores for MHT with respect to the mean rate of new tracks parameter. The 95% confidence interval for the average PC value is depicted by the error bars.

Experimenting with values on a $\log_{10}$ scale from -5 to -1 revealed a similar shape to the PC values for POD, and is depicted in Figure 4. For most values tested, MHT achieved a 97% accuracy or better. Only at the extremes of this range did the accuracy drop slightly to about 92%. If NTPS was set too large, MHT would produce many short tracks and longer tracks would have many dropouts as it tries to satisfy the rate of new track creation. However, if NTPS was set to small, MHT would miss many possible tracks. The clutter objects seemed to be the most impacted by the extremes of this parameter. The optimal value appeared to be about 0.001 to stay away from either extremes.

The mean false alarm per scan (MFAPS) was the next parameter to be examined. Like the NTPS, this name might also be slightly misleading and a determination of the exact nature of this parameter is needed. Testing values on a $\log_{10}$ scale from -8 to -1 revealed an intriguing curve, part of which is depicted in Figure 5. For smaller values, MHT performed very well, achieving approximately 0.98% accuracy, whereas the largest values of MFAPS (not shown) resulted in near-zero accuracy. Further examination of the results revealed that this sudden drop in skill scores occurs whenever MFAPS is somewhat larger than

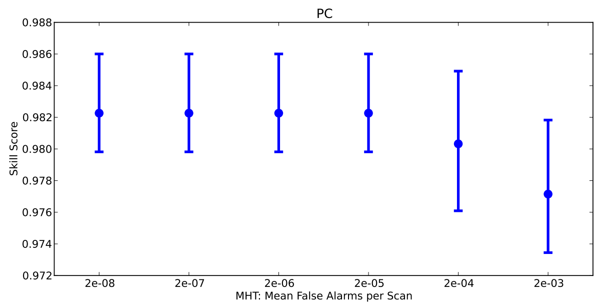

FIG. 5. MHT tracking scores with respect to changes in the mean rate of false alarms parameter. The 95% confidence interval for the average PC value is depicted by the error bars.

NTPS. From this, it is concluded that a MFAPS value of 0.00002 should be sufficiently smaller than the NTPS and allow for adequate performance.

Finally, the λ_x parameter was examined. There is not much documentation for this parameter and exactly what it controls in MHT. The current understanding is that this parameter is the probability of track termination at any given scan, but multiplied by 100 (unlike the other probability parameters). Further examination of the source code is needed to determine its exact nature. Using a range of values from 5.0 to 95.0 for λ_x revealed a curve that increased slowly from around 0.95% to just above 0.965%. Figure 6 depicts this relationship. While further examination of larger values of λ_x is warranted, a value of 30.0 appears to be sufficient for this type of scenario simulation.

Utilizing the knowledge gained from these parameter optimization analyzes, the MHT algorithm can now be used to track features in real-world radar data. Due to the lack of an objective truth for reference, it is not possible to quantify the performance of the tracker in the same manner as before in this study. However, a visual inspection of the results should be sufficient to determine if results are visually coherent.

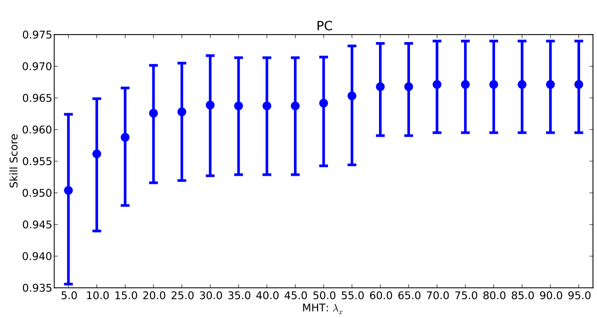

FIG. 6. MHT tracking scores with changing values of the λ_x parameter. The 95% confidence interval for the average PC value is depicted by the error bars.

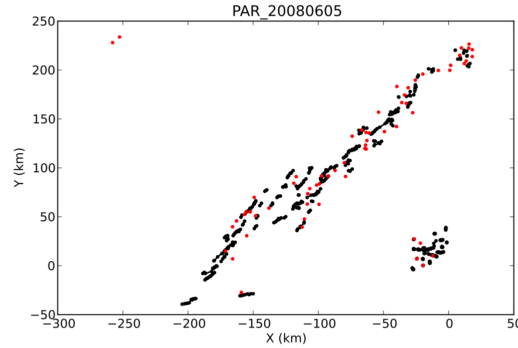

FIG. 7. Tracking results from applying MHT to features identified in a 17-minute radar reflectivity time series. Radar data from the NWRT PAR site in Norman, Oklahoma on June 5$^{\text{th}}$, 2008 between 2343 and 0000 GMT. Black lines are the track associations, while the red dots are the un-associated features.

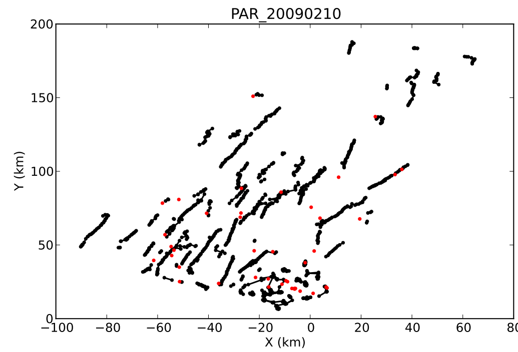

FIG. 8. Tracking results from applying MHT to features identified in a 22-minute radar reflectivity time series. Radar data from the NWRT PAR site in Norman, Oklahoma on February 10$^{\text{th}}$, 2009 between 2142 and 2204 GMT. Black lines are the track associations, while the red dots are the un-associated features.

Figures 7 and 8 depict the tracking results from the MHT algorithm applied to feature data derived from radar reflectivity data. The radar reflectivity data are from the NWRT PAR site in Norman, Oklahoma. The feature data were created from the Strong Point Analysis (SPA) (Root et al. 2011b) image clustering algorithm. The tracks depicted in these figures do not appear to be spurious and seem to be coherent.

4. Conclusions

This project provided an initial in-depth look at the MHT algorithm as a potential storm tracking algorithm. By using ZigZag to simulate storm tracks, we were able to perform controlled testing and analysis of MHT's performance with respect to its parameters. It appears that

MHT regularly produced results that were more than 90% accurate regardless of the values of the parameters, and were more the 95% accurate for most parameter values. All four parameters appear to be very stable over wide ranges of values. The only major parameter space constraint that was discovered was that MFAPS should be kept smaller than NTPS.

Further analysis of these parameters under a wider variety of scenarios should be done to determine how much (or how little) tuning is needed for MHT to produce results that are superior to the current generation of tracking algorithms. If a given set of parameters can lead to sufficiently accurate results from MHT regardless of the tracking scenario provided, then its value as a storm tracking algorithm increases. Indeed, initial examination of the applicability of MHT to real-world radar data appear promising.

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.

Murty, K. G., 1968: An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, **16 (3)**, 682687.

Root, B., M. Yeary, and T. Yu, 2011a: Generalized testing and analysis of storm cell tracking algorithms. *27th Conference on International Interactive Information and Processing Systems*, Amer. Meteor. Soc., Seattle, WA.

Root, B., T. Yu, and M. Yeary, 2011b: Consistent clustering of radar reflectivities using strong point analysis: A prelude to storm tracking. *IEEE Geoscience and Remote Sensing Letters*, doi:10.1109/LGRS.2010.2070787.

Yilmaz, A., O. Javed, and M. Shah, 2006: Object tracking: A survey. *ACM Comput. Surv.*, **38 (4)**, 13.