METEOROLOGICAL APPLICATIONS
Meteorol. Appl. **22**: 694–704 (2015)
Published online 6 August 2015 in Wiley Online Library
(wileyonlinelibrary.com) DOI: 10.1002/met.1501

Royal Meteorological Society

The advanced algorithm for tracking objects (AALTO)

George Limpert,[a,*] Adam Houston[a] and Noah Lock[b]
[a] *Department of Earth and Atmospheric Sciences, University of Nebraska-Lincoln, NE, USA*
[b] *Weather Decision Technologies, Inc., Norman, OK, USA*

ABSTRACT: The advanced algorithm for the tracking of objects (AALTO) constructs tracks from objects, such as thunderstorms or mesocyclones, detected by multiple weather radars at irregular time intervals. It is important to have high accuracy in tracking thunderstorms to generate skilful forecasts and high-quality climatologies and, fundamentally, to ensure that any derived product from tracks captures only that particular storm, and in its entirety. AALTO incorporates many of the best practices of existing tracking algorithms and techniques employed by meteorologists in constructing tracks. AALTO differs from existing algorithms designed to track meteorological phenomena that manifest in radar data in the following ways: (1) AALTO is designed to track objects from multiple radars, enabling analysis over a larger domain than if a single radar was used; (2) improved tracking is realized through improved initial motion estimates and directional thresholding and (3) AALTO looks both at the track history and at the subsequent possible positions along a track when constructing the best possible tracks, mimicking the approach that would be taken by a human meteorologist. Verification was done using metrics that were objectively determined to distinguish between good and degraded tracks; a description of the approach to determine the appropriate metrics is presented. An overview of the AALTO tracking procedure and an example case are presented in this study.

KEY WORDS forecasting; remote sensing; verification; hazards

Received 16 May 2014; Revised 14 November 2014; Accepted 22 January 2015

1. Introduction

Automated tracking of meteorological phenomena such as thunderstorms and mesocyclones is useful in many applications such as developing climatologies and nowcasting severe weather. Although existing algorithms designed to detect thunderstorms, mesocyclones and tornado vortex signatures, such as thunderstorm identification, tracking, analysis and nowcasting (TITAN; Dixon and Wiener, 1993); storm cell identification and tracking (SCIT; Johnson *et al.*, 1998); the National Severe Storms Laboratory (NSSL) Tornado vortex signature detection algorithm (TDA; Mitchell *et al.*, 1998); the NSSL mesocyclone detection algorithm (MDA; Stumpf *et al.*, 1998); w2segmotion (Lakshmanan *et al.*, 2007, 2009); Thunderstorm observation by radar (ThOR; Barjenbruch and Houston, 2006; Barjenbruch, 2008; Lahowetz *et al.*, 2010) and enhanced-TITAN (E-TITAN; Han *et al.*, 2009) include a tracking component, many recommendations to improve tracking in legacy algorithms have been proposed. For example, Johnson *et al.* (1998) propose that SCIT should incorporate different estimates of initial storm motion throughout the domain and impose limits upon the ability of tracks to change direction. Moreover, many existing algorithms (e.g., Johnson *et al.*, 1998; Mitchell *et al.*, 1998; Stumpf *et al.*, 1998) limit their tracking to within a single-radar domain rather than a larger domain. This study describes a new algorithm, the advanced algorithm for tracking objects (AALTO) that combines many of the best features of existing tracking algorithms while addressing many of the known limitations of legacy tracking algorithms. The key features of AALTO include:

- variable search radii for new objects along a track;
- directional thresholding, and
- ability to incorporate object detections from multiple radars into tracks.

Verification of AALTO was done both by disabling various components of the algorithm to show their impact and through a direct comparison with SCIT. A variety of approaches has been suggested to quantify the performance of tracking algorithms including those proposed by Lakshmanan and Smith (2010) and Reed and Trostel (2012). In this study, a novel approach to identifying appropriate verification statistics is presented by comparing quality tracks with those that have been systematically degraded, but still exhibit some skill, to demonstrate that the proposed verification metrics can distinguish between tracks of differing quality in a statistically significant manner (Table 1). Tests of AALTO demonstrate that AALTO objectively verifies well against current tracking algorithms such as SCIT. Additionally, AALTO provides guidance to the user, which can be used to improve tracking quality. Section 2 is an overview of the tracking principles in AALTO, compared with established algorithms. Section 3 describes the implementation of AALTO and Section 4 presents a performance analysis focusing on the sensitivity of track accuracy to the search area and a comparison between AALTO and SCIT.

2. Algorithm design

AALTO was developed to incorporate practices from radar-based tracking algorithms such as TITAN (Dixon and Wiener, 1993), SCIT (Johnson *et al.*, 1998), the NSSL MDA (Stumpf *et al.*, 1998), w2segmotion (Lakshmanan *et al.*, 2007, 2009), ThOR (Barjenbruch and Houston, 2006; Barjenbruch, 2008; Lahowetz *et al.*, 2010), E-TITAN (Han

* Correspondence: G. Limpert, University of Nebraska-Lincoln, 214 Bessey Hall, Lincoln, NE 68588, USA. E-mail: george.limpert2@huskers.unl.edu

Table 1. Verification statistics for a variety of AALTO configurations using data from the 2 January 2006 event.

	Strength error (kg m^{-2})	Straightness error (km)	Median duration (s)	Mean duration (s)
Default	6.70	4.80	1249	2144
Degraded − prioritization	7.49	5.70	1520	2261
Degraded − motion history + prioritization	7.67	6.24	1519	2266
No look ahead	6.72	4.81	1217	2131
Degraded − motion history + prioritization + no look ahead	7.70	6.78	1518	2314
40 min look ahead, e-fold 600 s	6.69	4.83	1216	2163
70 min look ahead, e-fold 900 s	6.64	4.90	1216	2175
No search angle	6.76	5.25	1519	2218
SCIT	6.34	4.84	912	1693

SCIT, storm cell identification and tracking.

et al., 2009), multiple hypothesis tracking (MHT; e.g., Lakshmanan and Smith, 2010; Scharenbrouch *et al.*, 2010; Root *et al.*, 2011; Lakshmanan *et al.*, 2013; Miller *et al.*, 2013) and satellite-based tracking algorithms (Zahraei *et al.*, 2012; Sieglaff *et al.*, 2013). Each of these algorithms adopts different approaches to forecasting object motion, searching for new objects and identifying the best object to use in continuing a track.

2.1. Object identification across multiple radars

Both SCIT and MDA are single-radar algorithms, not incorporating data from nearby radars that may overlap coverage. This approach does not make the best use of data in areas with numerous radars such as multiple WSR-88Ds and terminal Doppler weather radars (TDWRs) and perhaps other radars such as the UMass collaborative adaptive sensing of the atmosphere (CASA; Junyent *et al.*, 2010) radars. The single-radar limitations of SCIT and MDA are often avoided by first creating a mosaic of the product being analysed, such as radar reflectivity (the proportion of energy backscattered by targets such as hydrometeors) or radial velocity (the component of the mean motion of the targets towards or away from the radar), using an algorithm such as w2merger (Lakshmanan *et al.*, 2006). However, the identification of objects in mosaicked reflectivity or radial velocity data requires new multi-radar algorithms that are likely based on local maxima or minima (Lakshmanan *et al.*, 2009), whereas the present applications of AALTO have used objects identified by legacy algorithms such as SCIT (Johnson *et al.*, 1998) and MDA (Stumpf *et al.*, 1998).

AALTO adopts an alternate approach to avoid the single-radar limitation: instead of merging the data used to identify objects, the objects are identified independently by individual radars and then synthesized into a single composite (Stumpf *et al.*, 2003). AALTO does not require the development of new multi-radar algorithms, but tracks objects, represented by the latitude and longitude of 3D centroids, identified by the existing single-radar algorithms such as those used by the National Weather Service in the United States for severe weather detection based on WSR-88D data. There is a number of challenges to such an approach including overlapping areas of radar coverage resulting in multiple detections of the same object and the lack of synchronization in radar scan times, resulting in portions of the domain being updated at different times than other regions. The approach used in AALTO to resolve these challenges will be discussed below.

2.2. Tracking

The tracking logic in both MDA and SCIT is largely identical: object motion is extrapolated, based on either a prior motion or a first guess for new objects, and new objects to create or continue a track are sought within a set radius from the forecast position. The first guess of object motion for a new track depends on either a forecast storm motion that is based on upper-air data or the mean motion of all existing tracks within the domain. Implicitly, this assumes that the mean motion of tracks should be similar throughout the domain, and thus an average of existing tracks should be representative enough to suffice as a first guess for new tracks. This is acknowledged as a limitation of SCIT by Johnson *et al.* (1998), and it could decrease the accuracy of tracking if the assumption is violated. This limitation will be particularly severe for domains covered by multiple radars. AALTO avoids this limitation by using the storm motion estimate nearest to the time and position of an object to produce the best initial estimate of storm motion.

For established tracks, the existing algorithms rely on prior object motion to extrapolate the position of an object and create a forecast position. SCIT uses the previous 10 forecast positions (Johnson *et al.*, 1998), whereas MDA maintains a history of the previous 30 min (Stumpf *et al.*, 1998). AALTO defaults to using the motion history over the previous 30 min of the track. If the length of the time interval used to determine previous object motion is too short, then object tracks are likely to be overly influenced by erroneous variations in object velocity, subsequently referred to in this paper as jitter. This jitter is a consequence of the inability to determine the position of the object exactly due to the radar beam width and height and the difficulty in an algorithm assigning a point to an object with area and mass. These issues tend to be exacerbated when multiple radars are involved. However, if the history interval is too large, it will restrict the ability of the algorithm to detect actual changes in the bearing of the object. If the track is younger than 30 min, a combination of the initial motion estimate and the prior motion of the track is used.

To continue an object track forward in time, algorithms such as SCIT search for the closest object within a specified radius of the forecast position of the track. The search area is not further restricted, though Johnson *et al.* (1998) recommend directional thresholding, which is restricting the ability of a track to make a large change in direction from the recent motion of the track. Because AALTO is designed to track data at irregular intervals, a single search radius is not appropriate. AALTO also implements a form of directional thresholding, based on the age and predicted motion of a track.

AALTO calculates the search radius using (1) a linear relationship between search radius (r) and the time since the last update to the track (Δt) that accounts for the sensitivity of optimal search radius to data interval and (2) a prescribed minimum search radius ($r_{\min}$) that accounts for the effects of jitter on object tracks:

$$r = r_{\min} + \Delta t r_{\mathrm{inc}} \tag{1}$$

The calculated search angle (a) is based on the forecast distance (d) expected to be travelled by an object in the elapsed time since the previous update to the track and the degree to which the forecasted track motion is based on track history versus initial motion estimates:

$$a = \begin{cases} \max\left(a_{\max} + i, 180\right); d \le d_{\min} \\ \max\left(a_{\min} + i + \left(a_{\max} - a_{\min}\right) e^{\frac{d_{\min} < d}{f}}, 180\right); d > d_{\min} \end{cases} \tag{2}$$

where $a_{\min}$ and $a_{\max}$ are the minimum and maximum search angles, respectively, and i is the angle expansion used to account for uncertainty in track extrapolation. If the forecast distance travelled is less than $d_{\min}$, the maximum search angle, for a given value of i is used. For $d > d_{\min}$, the search angle decreases exponentially to a value of $a_{\min} + i$, with an e-fold forecast distance, denoted f. This approach to directional thresholding allows for larger changes in direction if the effects of jitter are bigger compared with the actual distance the track would have moved since the last update. The angle expansion, i, is included because estimates of track motion are least certain when they are based fully on the initial estimate in the early stages of a track and are most certain when fully transitioned to being based on track history. As such, track motions that are based on initial estimates are allowed an additional 15° change in direction, decreasing linearly to zero when fully based on track history:

$$i = \begin{cases} 0; t \ge h \\ \frac{15(h-t)}{h}; t < h \end{cases} \tag{3}$$

where h is the time when the track motions are based entirely on track history. Figure 1 shows examples of the shape of possible search areas.

Algorithms such as SCIT that establish coherence between objects at only two successive times may perform worse than tracking algorithms that consider additional times. A human tracking objects manually would likely look at a few previous volume scans and a few future volume scans to find the next object on the track. While this approach is most appropriate when operating on archived data, it could be used in nowcasting to retroactively refine real-time tracks. This approach that considers future positions of the object in addition to the past has been adopted in some automated tracking algorithms including MHT (e.g. Reid, 1979; Cox and Hingorani, 1996; Blackman, 2004; Lakshmanan and Smith, 2010; Scharenbrouch *et al.*, 2010; Root *et al.*, 2011; Lakshmanan *et al.*, 2013; Miller *et al.*, 2013) and ThOR (Barjenbruch and Houston, 2006; Barjenbruch, 2008; Lahowetz *et al.*, 2010) and is employed in AALTO. Future positions are examined by building a search tree in which each node represents the position of an object along a candidate track, which refers to a possible continuation of the existing track.

As in ThOR, AALTO develops all possible candidate tracks and then traverses the tree to find the track with the lowest mean position error, defined as the distance between the forecast position and the actual position along the track. For higher-level branches, all possible positions within an interval of time, defaulting to 12 min, are considered in order to extend the branch. Assuming a nominal 5–6 min time interval between volumes, a 12 min window allows tracks to skip times without nearby objects. This has been found to produce straighter candidate tracks, in which position errors at future times are primarily due to changes in object displacement and not due to jitter. The first-level branch that contains the selected candidate track is chosen as the continuation of the track and is given a weight of 1. Each branch thereafter is exponentially weighted less, with an e-folding time of 300 s. Building and searching the tree is an exponentially complex problem, potentially increasing without bound if the tree is allowed to grow unchecked. However, it is unlikely that the highest branches (track segments connecting objects at times well into the future) will have a significant impact on which first-level branch to select. The influence of higher-level branches scales inversely with level; it is unlikely that a human meteorologist would give the same weight to a tenth-level branch as the second-level branch. AALTO implements this by decreasing the weighting of the position errors exponentially from higher-level branches. When the weight drops below a user-specified threshold, defaulting to 0.05, higher-level branches are no longer added to the search tree. The combination of these default parameters causes AALTO to look ahead ~15 min, or three WSR-88D volume scans. Analysis of test cases revealed that when either the lookahead was disabled or the e-fold time was increased and minimum weight was decreased, resulting in a deeper search, negligible changes in track straightness were noted, with the apparent best values, though without statistically significant differences, being for the default parameters.

As in TITAN and MHT, AALTO performs global optimization to minimize the cost at each time step. Cost is the distance from the forecast position of the object to the actual position. TITAN, MHT and AALTO seek to minimize the overall cost of connecting all tracks to new objects at each time step, rather than merely minimizing the cost for one track at a time. In AALTO, all possible tracks at a given analysis time are considered and the track with the closest forecast position to any available object is given the first opportunity to select an object as a continuation of the track, similar to that described by Lakshmanan and Smith (2010), with object strength breaking any ties. Multiple options were considered in prioritizing the tracks, including allowing 'stronger' tracks (e.g. higher vertically integrated liquid, higher mesocyclone strength index, stronger gate-to-gate shear) to select first, older tracks to select first, and randomizing the order in which tracks are selected.

To evaluate the performance of these prioritization methods, each method was applied to five cases using both SCIT and MDA detections. To quantify the performance, a correct match was defined as a track selecting an object such that no other tracks had a forecast position closer to that object. Similarly, an incorrect match was defined as a track selecting an object that would have been closer to the forecast position of a track selecting later in the sequence. For both object types, the random prioritization resulted in ~50% correct matches. Both the strength and longevity prioritizations resulted in roughly 70% correct matches. Prioritizing based on longevity appeared to introduce a bias in track durations. The current optimized approach results in at least 99% correct matches. The significance of prioritization is underscored by the observation that nearly 30% of tracks terminated when using strength-based prioritization because an object that would have been within the search radius of the track

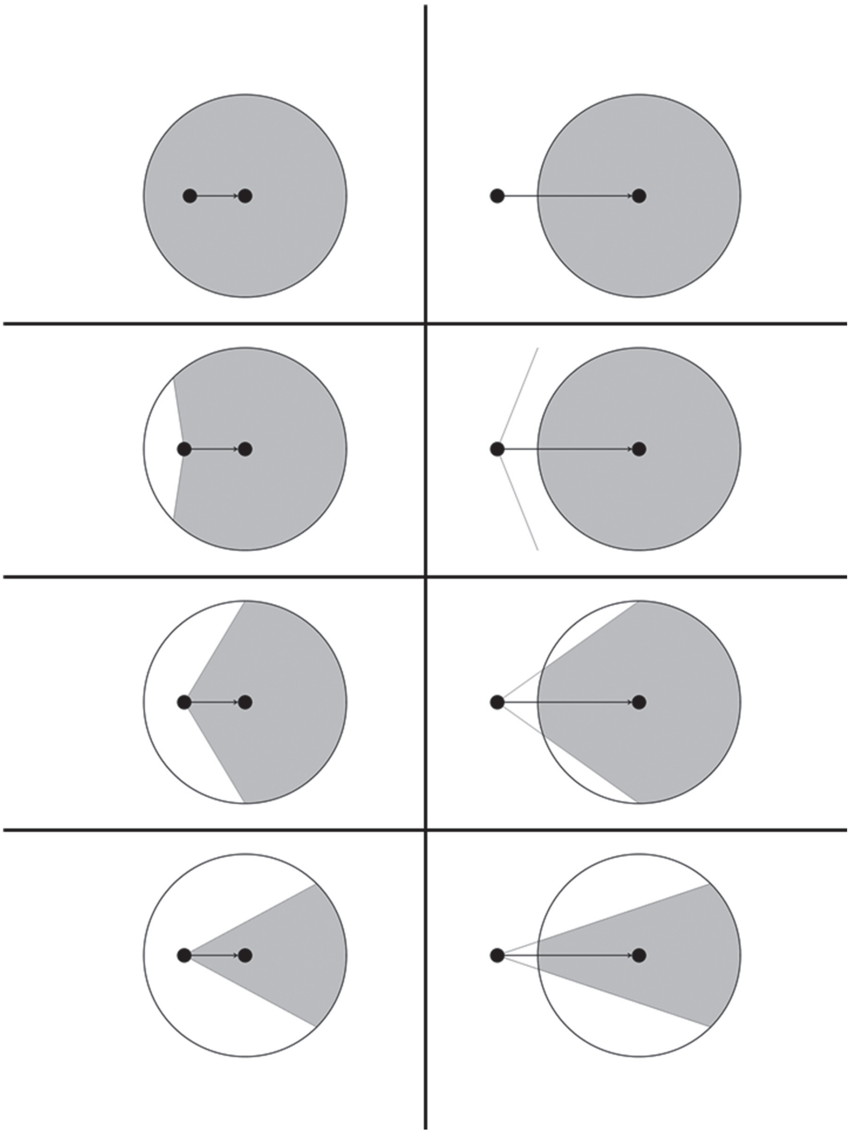

Figure 1. The possible configurations for search areas. The grey area shows the actual search area. The left column shows a situation in which the initial position is within the search area. The right column shows a situation in which the initial position is outside the search area. The top row shows a situation in which there is no restriction on search angle. Progressively smaller search angles are shown towards the bottom. The search angle is narrowed for faster moving objects for tracks that have been updated less recently.

had already been claimed by another track. Furthermore, the current optimization approach produced a statistically significant result of straighter tracks with fewer changes in direction than those produced by randomized prioritization. The best tracking performance was achieved when AALTO finds the track with the minimum distance from its forecast position to any currently detected objects and allows this track the next opportunity to select from the pool of available objects.

3. Implementation

The overall processing by AALTO is shown in Figure 2. The first step in the algorithm is to develop initial object motion estimates. Flexibility is permitted in how these estimates are derived, allowing for both gridded and irregularly spaced data sets. This allows for versatility in estimating the motion of objects initially while maintaining the best practice of using spatiotemporally heterogeneous initial motion estimates. Next, AALTO ingests the objects to be tracked. AALTO permits any number of fields to be associated with each object, all of which are preserved through the algorithm and written in the output. However, a few fields are required in the input: latitude, longitude, a timestamp and a strength parameter. For circulations, an appropriate strength parameter might be shear or rotational velocity. For thunderstorm objects, vertically integrated liquid (VIL) or maximum reflectivity could be appropriate. If the data set is comprised of data from multiple radars, an identifier for the radar and the distance from the radar are also required parameters, which are necessary for filtering collisions, as defined below. Because radars are not synchronized to scan at the same time, and volume coverage patterns require different amounts of time to complete a volume scan, objects in the database appear at irregular intervals. The objects are matched with the nearest initial motion estimate in time and space as they are ingested.

Multiple radars may scan the same object at precisely the same time (to the granularity of a second), meaning that one object would be counted two or more times at the same time step.

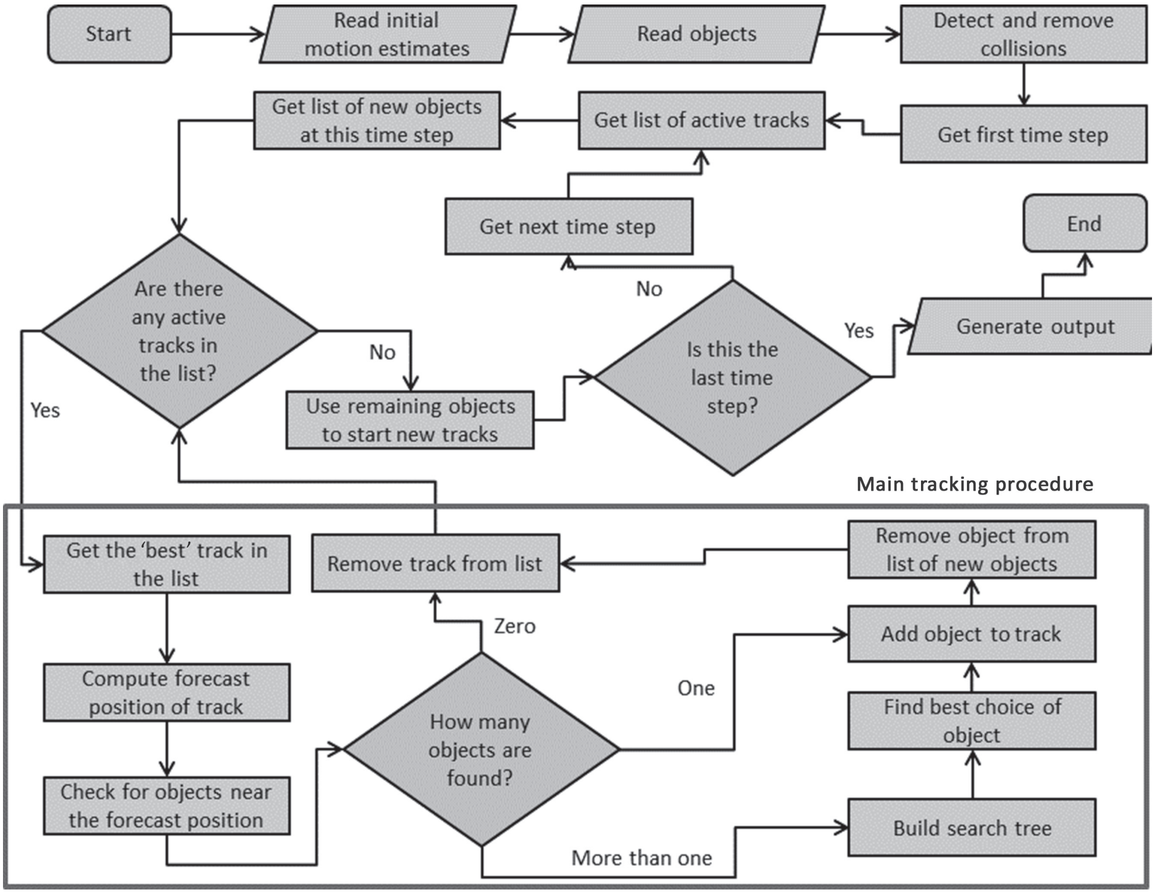

Figure 2. The flowchart showing the general steps in the advanced algorithm for the tracking of objects (AALTO).

Multiple instances of the same object at the same time step could adversely affect the results of the tracking. While this rarely occurs, it was found in some of the data sets used to test AALTO. Such an instance is referred to as a collision, which AALTO detects and resolves once all the objects have been ingested. At any particular time step, if two objects are detected by different radars and are within a specified radius of each other, defaulting to a distance of 5 km, it is considered to be a collision. Collisions are resolved by retaining the object with the smallest distance to its detecting radar. All objects colliding with this particular object are deleted from the data set. This step is repeated for each time step until all collisions have been resolved.

At any particular time step, all active tracks are examined, beginning with the closest match to available objects and iterating through the remaining tracks. Active tracks are all tracks that have been updated within a specified amount of time, defaulting to 12 min (slightly longer than two typical WSR-88D volume scans). The track motion is estimated by a weighted combination of prior motion along the track and the initial motion estimates provided to AALTO, producing a forecast location for the object being tracked. The region around the forecast position is searched for objects that are candidates to continue the track. This region is restricted based on the proximity to the forecast position and the change in bearing between the estimated motion and the actual motion from the last track position to the candidate track. To account for the possibility that an object identification algorithm misses an object at a given time or calculates an object centroid that is removed from the correct track, AALTO considers all possible object positions within 12 min from the time of the latest position on the candidate track.

If more than one candidate object is found within the search area, additional processing is done to determine which object is most suitable for the continuation of the track. Each of the tracks to candidate objects is treated as a first-level branch. From these branches, the procedure is repeated recursively, adding higher-level branches for the duration the track would remain active. Tracks are added until the exponentially decaying branch weight passes below a threshold value, as described in the previous section. The exponentially decaying branch weights are used to calculate the weighted average error, as defined by the distance between the forecast position and the position of each object, for each candidate path. The first-level branch associated with the candidate path with the minimum weighted error is selected to continue the track, but the rest of the candidate path is discarded. An illustration of the concept of a search tree is shown in Figure 3.

Any objects that have not been added to existing tracks at the end of a time step are used as the first position in new tracks. The process is repeated for each time step until the end of available data. The user has the option of choosing whether to retain objects that were never connected in tracks, the default behaviour or to omit them from the output.

4. Performance analysis

4.1. Sensitivity to search area

The performance of AALTO is examined by first assessing the sensitivity of track accuracy to the search area. These results

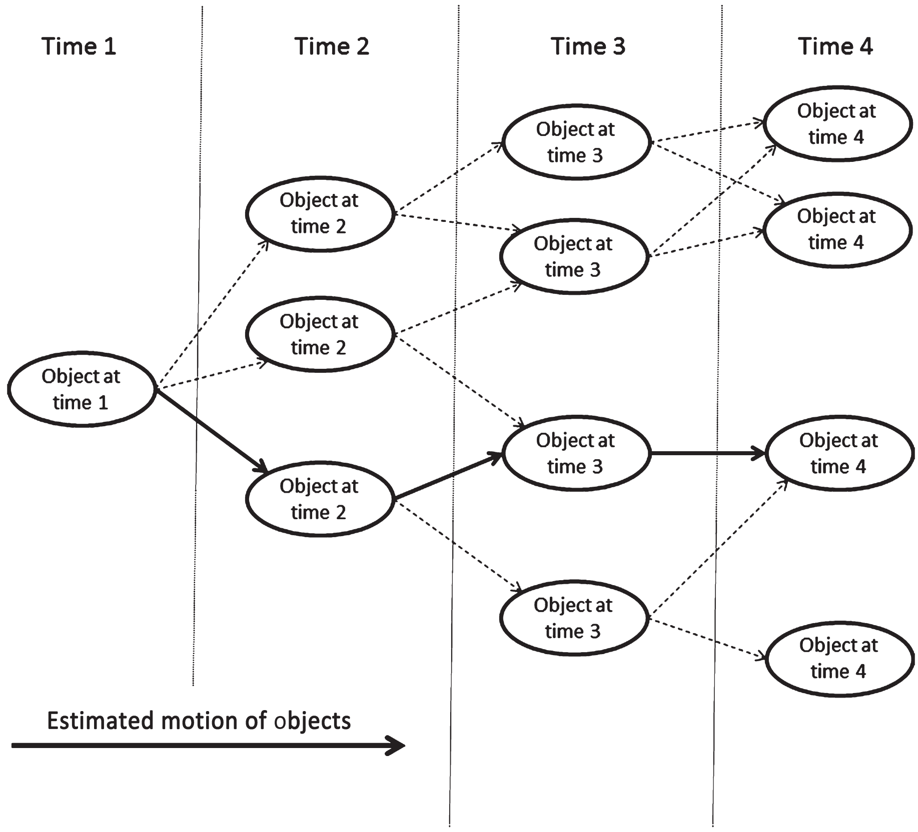

Figure 3. The approach of building trees to find the best path is shown. Although the middle object at time 2 is closer to the forecast position for that time, the bottom object is chosen because it reduces the forecast error at later times and results in an overall smaller error.

offer general guidance for good tracking practices. Figures 4 and 5 show the distribution of distances from the forecast position for the 2854 SCIT detections from a single radar during a thunderstorm event in the central United States (Figure 6(a)). Figure 4 shows a clear separation between the distribution of the closest (most accurate) objects and other objects. Specifically, the first (closest) object is typically within a few kilometres of the forecast position, peaking in the 1–2 km bin and decreasing significantly afterwards. However, the second object peaks in the 12–13 km bin, with few objects closer than 7 or 8 km. This result supports the AALTO logic incorporating a variable search radius to continue tracks. Figure 5 shows that the distance from the forecast position to the closest (most accurate) object increases with time since the last update to the track (Δt). If the track has been updated within 5 min most of the closest objects are within a few kilometres of the forecast position, but the distance is much larger for an interval of 10 min. Thus, as reflected in the expression for search radius (1) used in AALTO, it is necessary to increase the radius with time since the last update to the track. Root *et al.* (2009) first identified a limitation of SCIT that if an object is not detected at an intermediate position along a track, it will incorrectly split the track into two tracks. Allowing a variable amount of time between detections along a track to address this known limitation of SCIT requires the implementation of a variable search area.

A variable search angle is also necessary, in order to implement directional thresholding properly, as suggested initially by Johnson *et al.* (1998). Assuming that the error in detecting the precise position of an object is randomly distributed within a set distance, perhaps a few kilometres, the influence of that error on

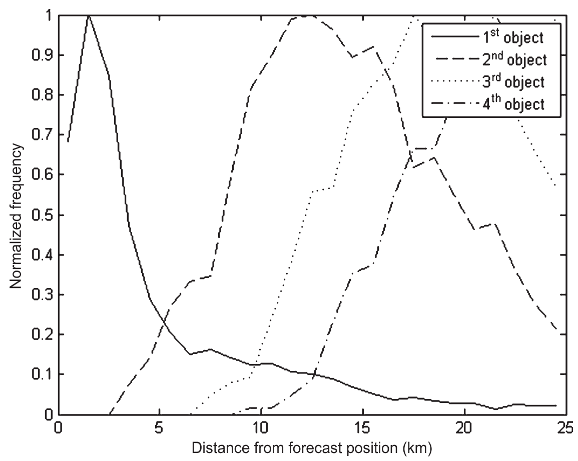

Figure 4. Distances and angles from the forecast position to objects from the 2 January 2006 event, using storm cell identification and tracking (SCIT) identifications. Distances are binned into increments of 1 km and angles are grouped by 5° increments.

the apparent direction of the object decreases as the object travels farther. It is likely that a small search angle, such as around a 45° directional change, will encompass most objects after a track has not been updated for a few minutes. However, this suggests that there ought to be a dependence on time or distance travelled in determining the search angle. Figure 7 shows the distribution of the closest detections relative to search angle for several typical distances and illustrates a decrease in the search angle needed to encompass a given percentage of detections as

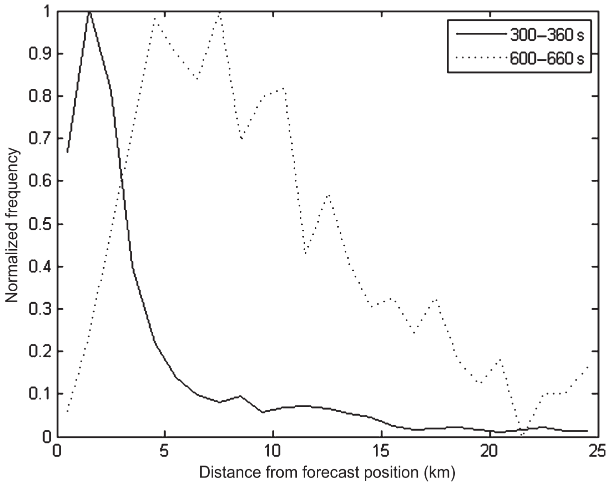

Figure 5. Distances from the forecast position to objects from the 2 January 2006 event, using SCIT identifications, depending on the time since the last update to the track. Distances are binned into increments of 1 km.

the forecast distance increases. This supports relating the search angle to the distance travelled by the object since its last position on the track.

Figure 8 shows an example of how the track age affects the search angle from the thunderstorm event used above, by plotting the age of the track versus the percentage of the closest objects within the angle. Newer tracks require larger search angles to incorporate the closest object. This justifies the inclusion of (3), which increases the search angle for newly identified tracks.

4.2. Comparison to SCIT tracks

In this section, AALTO thunderstorm tracks (processed using SCIT detections) are compared against SCIT tracks to quantify the differences in tracking and the differences in various AALTO settings. As noted by Lakshmanan and Smith (2010), construction of a truth data set is both difficult and subjective, especially when there are large numbers of objects. For that reason, it is best to determine a few criteria that describe the statistics of tracks produced by a good algorithm, and then perform an inter-comparison. Appropriately chosen metrics not only verify the skill of the algorithm but also identify the sources of errors and assist in improving the accuracy of tracking and prediction. Additionally, in a warning decision context, a forecaster may be presented with predictions from multiple algorithms. Objective real-time verification provides guidance as to which forecast products to use, hence the need to determine which metrics are most useful in evaluating algorithm performance. The metrics used are based loosely on those of Lakshmanan and Smith (2010).

One metric is the average duration of tracks. To some extent, longer tracks are better, but if tracks are too long, they likely contain multiple actual tracks that have been incorrectly merged into a single track. Thus, while longevity is potentially informative, it is generally not the best parameter. Similarly, it is not certain whether it is preferable for an algorithm to produce more tracks or fewer tracks.

Another potential metric is the stability of a strength parameter, such as VIL for thunderstorms. It is likely that, from one volume scan to the next, the strength parameter is likely to be relatively similar for a given track. For this work, this metric is calculated using VIL at each point along a given track as the difference between the actual VIL and the local mean VIL calculated using the VIL values at positions just before and just after the particular track position. The maximum error is reported for each track, unless the track only has two positions, and the mean value for all tracks is reported as the strength error. This method differs from the standard deviation proposed by Lakshmanan and Smith (2010) because of the suggestion of Reed and Trostel (2012) that it should not have a strong dependence on track duration.

Track straightness is another desirable characteristic of good tracks. While tracks may have small wobbles over short time durations and may curve at long durations, they are likely to be relatively straight at intermediate time intervals. For example, it might be reasonable to expect thunderstorms to travel in a mostly straight path for a 30 min interval. Thus, a line between positions that are 30 min apart can be used as a predictor of position within that interval, and deviations from the prediction are a measure of error. For this work, the metric used is the maximum change in position error between the forecast position and actual position at any time during 30 min intervals along a track. The maximum value for each track within the moving window is recorded for the track, and the mean of all tracks is reported as the straightness error.

Even though these metrics are somewhat intuitive, it is still desirable to demonstrate their usefulness and determine which is most important. The degradation of tracks should balance two competing, but important, principles: (1) there must be evidence that the measures should actually degrade tracking performance and should never improve tracking but (2) should still produce tracks that exhibit a reasonable degree of tracking skill. If the tracks are degraded too little, then no appropriate verification metrics will be identified. However, if the tracks are degraded too much, then any metric will appear to have value in demonstrating tracking skill. For this verification process, AALTO tracks were intentionally degraded by using random prioritization and increasing the motion history from 30 min to one day, causing AALTO to almost exclusively use the initial motion estimate, in this case based on the RUC model (Benjamin *et al.*, 2004) storm motion vectors. Both prior art and tests of AALTO demonstrate the need for global optimization in order to maximize tracking performance. However, despite the many known issues with SCIT, which lacks global optimization, the tracks produced do exhibit a useful degree of skill. The RUC storm motion parameter estimates the motion of right-moving supercells. It does not take into account the actual motion of storms as observed by radar, and is related to the motion of non-supercell storms, but offset in its direction. Although the RUC storm motion parameter offers some skill in forecasting storm motion, it is less than optimal in most situations. Both of the methods used to degrade the tracks meet the above criteria to clearly decrease the tracking performance, while still exhibiting enough skill to identify the best parameters for measuring tracking performance.

The proposed verification statistics were calculated for both the default and the degraded tracks, and compared between the two using a two-tailed t-test for statistical significance (Table 1). The choice of an event also potentially has an impact on the parameters that will be selected. Based on the above criteria, many of the tracks from both the good and the degraded sets will overlap. The event that is chosen should generate enough tracks such that a sufficiently large number of tracks differ between the two data sets. However, the event should be small enough such that small differences in the verification parameters are not

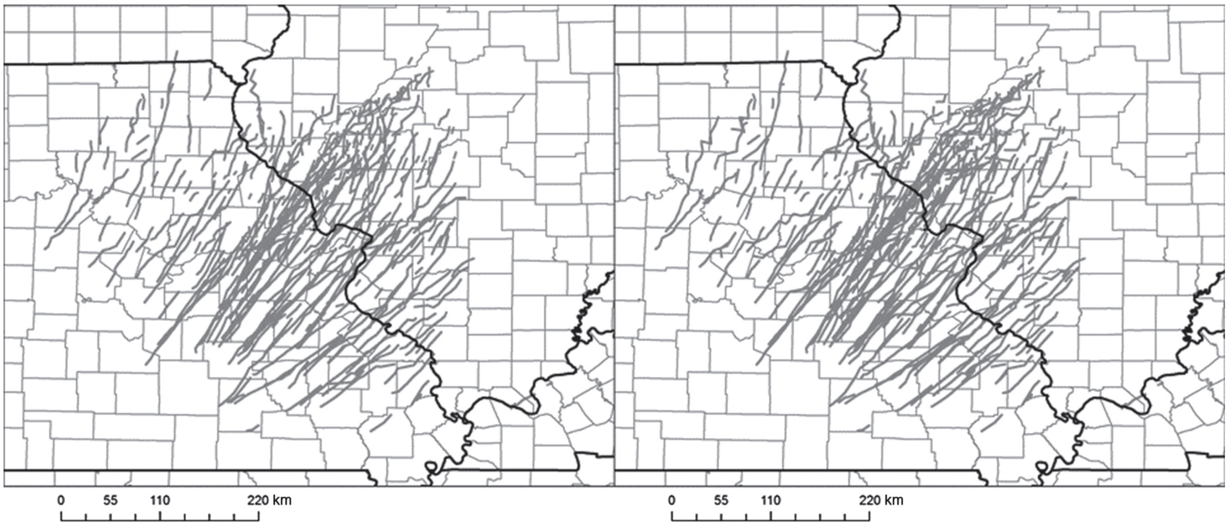

Figure 6. AALTO tracks from the 2 January 2006 thunderstorm event. (a) Default settings and (b) degraded tracks.

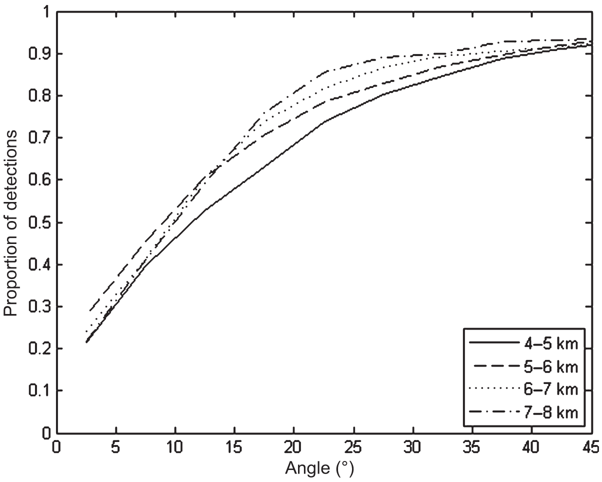

Figure 7. The relationship between the search angle and the proportion of detections of the closest object encompassed within the search area for the 2 January 2006 event. The lines represent different forecast distances. Although some points were outside the 4–8 km range, there were not enough detections for this case at those ranges to accurately represent the distribution, so they were omitted.

statistically significant purely due to the size of the data set. In practice, a data set with a few hundred tracks is likely an appropriate choice.

The same thunderstorm event used previously (Figure 6) was examined using the verification metrics on all tracks to determine if any of the metrics showed statistically significant differences. There were 328 tracks using the default AALTO settings and 332 tracks using the degraded settings. Although there are subtle differences between the tracks, the AALTO tracks from the default settings and those that were intentionally degraded visually appear very similar, such that even the degraded tracks would appear, without careful inspection, to be good tracks. Although the tracks generally appear similar (Figure 6), small differences in track straightness can be noticed upon careful inspection. However, in the absence of verification statistics and a set of tracks to compare against, it is unlikely that the degraded tracks would readily be identified as inferior, supporting the use of statistics to distinguish between good tracks and those that have been degraded. Track duration did not appear to be a useful metric, as intentionally degraded tracks actually were longer than those that were presumed better, and that performed better in other metrics, yielding a p-value of 0.479. Although there are differences in track strength errors, they were marginally significant for this data set, with a p-value of 0.034. The track straightness errors were, however, statistically significant, with a p-value of 0.0000121, and seemed to provide the best discriminator between good tracks and poor tracks.

It is apparent from the statistics that both prioritization and the impacts of using prior motion to forecast future track positions are important in the tracking. Looking ahead to future track positions generally seems to have little impact when the other tracking optimizations are applied. However, when the degraded tracks are compared against the same settings but without looking ahead to future positions, there is a clear difference in track straightness. This suggests that looking ahead may have value to improving the otherwise poor tracking. Thus, the idea of building candidate tracks likely has some merit in improving tracking performance, but is not particularly evident when the tracking algorithm is otherwise well designed and configured. Evidence from testing AALTO strongly suggests that the closest object to the forecast position within the search area is nearly always the correct object. This is based on the frequency at which the first-level branch closest to the forecast position is not the branch that is selected after building the candidate tracks. At most, this occurs around 10% of the time, but is generally closer to 2–5%. It is difficult to envision that a process occurring quite infrequently could have a significant impact on the verification statistics, which likely explains this result.

Although SCIT tracks provide similar levels of straightness and strength errors, this comes at the expense of much shorter tracks. Both the mean and the median length of the SCIT tracks are much shorter than those of the AALTO tracks. However, the longer AALTO tracks do not seem to significantly degrade the quality of the tracks based on the straightness and strength errors. For this reason, it seems that SCIT is prematurely terminating many tracks that ought to continue. This assertion is supported through visualization of AALTO and SCIT tracks (Figure 9). There are likely two reasons for this: (1) SCIT is configured to terminate a track if a storm is not identified at a particular time, even if it appears both at the previous and the next time (Root *et al.*,

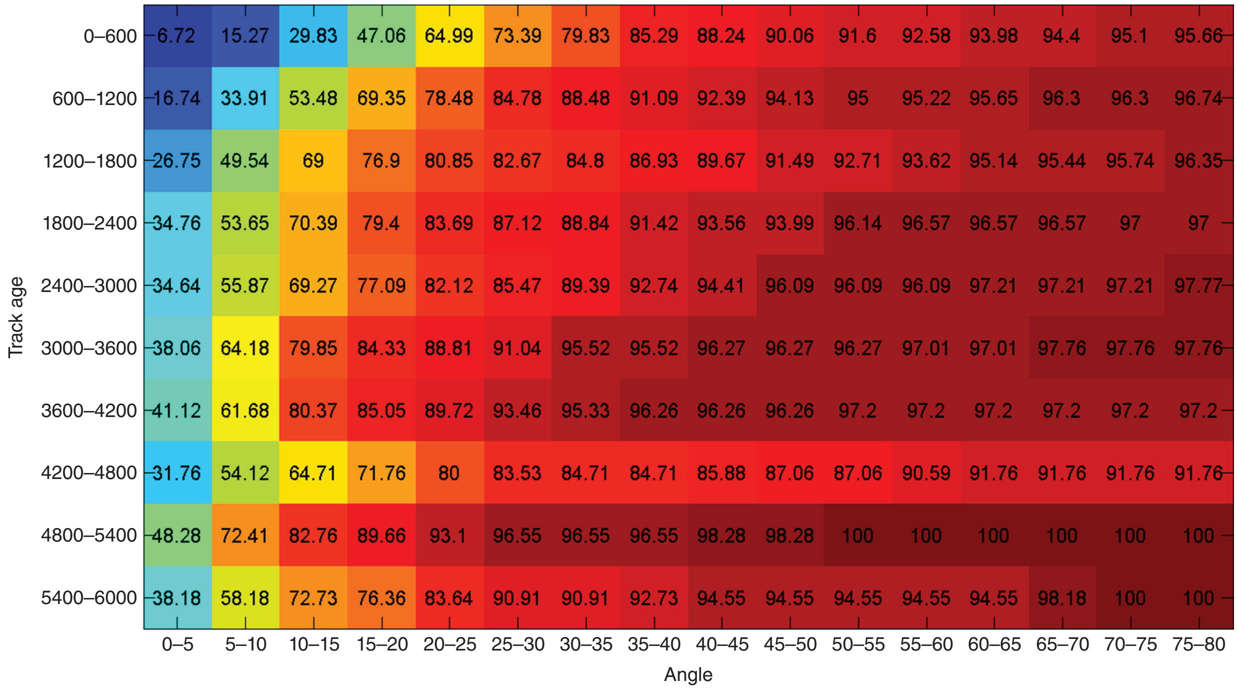

Figure 8. A heat map showing the percentage of objects as a function of track age in seconds and search angle, demonstrating the need to expand the search angle for newer tracks.

2009), and (2) SCIT does not prioritize the order in which tracks are allowed to select from the pool of available objects. As such, the SCIT tracks are track segments that have similar statistics to the AALTO tracks, but are not the full tracks. There were 328 AALTO tracks, leaving 112 SCIT detections as not being part of any track. In contrast, SCIT produced 346 tracks, with 315 detections being not part of any track, meaning that roughly 7% of the total SCIT detections were excluded from tracks by SCIT but not by AALTO. Because the verification statistics for the AALTO tracks are at least on par with the SCIT statistics, it is likely that the 203 detections excluded from tracks by SCIT actually do belong in tracks.

5. Discussion

AALTO is similar to w2segmotion, E-TITAN and MHT and implements many enhancements over operational tracking algorithms. These enhancements include the ability to incorporate detections of the same object from multiple radars into one track and an approach to minimize errors associated with the order in which existing tracks are processed. Furthermore, a robust approach to tracking verification is presented here that does not require the tedious and subjective manual generation of tracks. Although AALTO is designed to track meteorological objects well, the development of AALTO revealed a general insight into the characteristics of tracking algorithms most likely to significantly impact algorithm performance. It is likely that the parameters chosen by default for AALTO are valid for similar storm-scale meteorological phenomena, such as mesocyclones, and thus AALTO is expected to perform well in these situations.

The potential applications of AALTO to build a mesovortex climatology using MDA output across multiple radars and incorporating AALTO as part of an algorithm to identify, track and classify storms are included in a forthcoming paper. The latter application could be used both in real-time on radar data and to track and classify storms in the output of high-resolution models and ensemble forecasts. The inclusion of AALTO in such an algorithm would likely involve modifying AALTO to operate on 2D or 3D representations of storms rather than simply using centroids. Depending on the size of the domain, it may be necessary to modify the approach to building trees to reduce the complexity of the algorithm. Approaches to improve AALTO in this way are discussed later in this section.

It is clear that prioritizing the order in which tracks select from the available objects is important to producing good tracks, given the verification statistics and additional evidence from AALTO output. It appears that the best approach is to give priority to the track with the closest forecast position to any of the available objects (Lakshmanan and Smith, 2010). It is also clear that merely examining a set of tracks visually is insufficient to determine if the tracks are good. Even comparing the default parameters from AALTO with intentionally degraded tracks requires careful inspection to identify the differences between the two sets of tracks. Objective verification of the tracks, however, yields clear results as to which sets of tracks are better. This underscores the need for a robust verification scheme and offers some insight as to how to construct it. Verification metrics should be based upon reasonable assumptions about the expected behaviour of tracks, such as their expected straightness and the predictability of certain attributes such as strength. However, this is still insufficient to establish whether the metrics are actually useful. Tracks to be verified should be compared against an additional set of tracks that are known to be poor, such as intentionally degrading the performance of the algorithm. Degrading the tracking should be done based on factors that are supported by other data, such as the impacts of using the initial motion estimate or tracking prioritization. Comparing the verification statistics of the degraded tracks against the default settings demonstrates both the validity of the chosen metrics and the assumptions made in the tracking process.

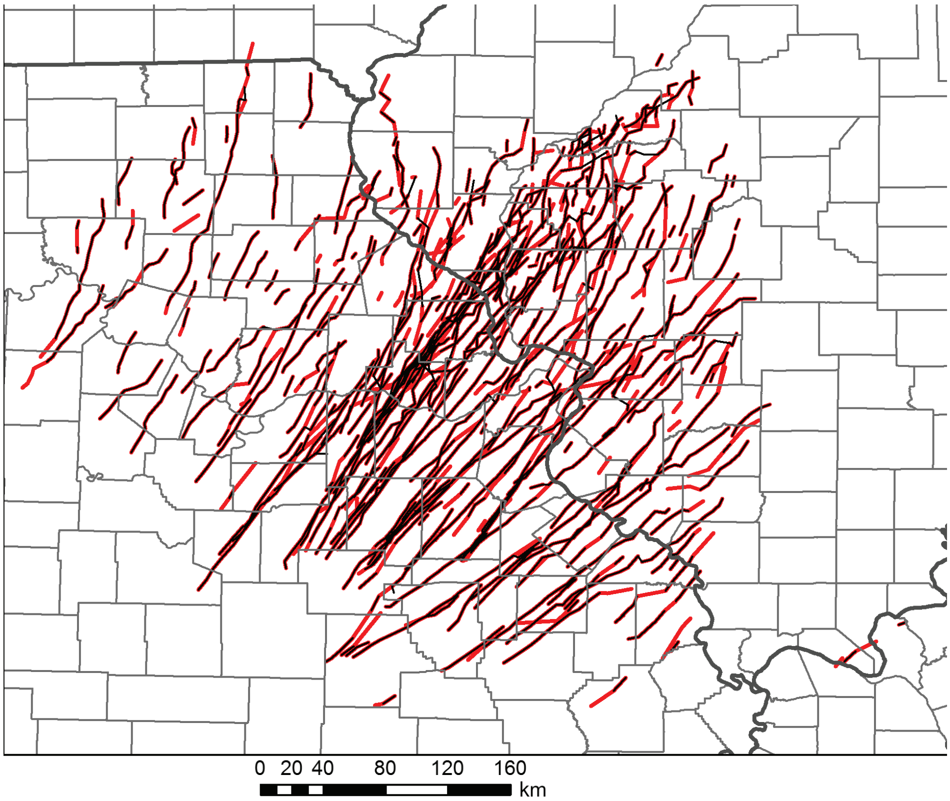

Figure 9. A comparison of SCIT (black) and AALTO (red) tracks from the 2 January 2006 case.

One interesting finding in the verification was that the strength parameter was not as useful as expected in discerning between good and poor tracks. It has been suggested that, in addition to a cost function based on the distance from a forecast position, a strength parameter might be useful in matching objects to existing tracks within a tracking algorithm, such as that done by Lack *et al.* (2010). If strength is of limited utility in discerning between good tracks and bad tracks, incorporating strength into a cost function may not significantly improve tracking.

Another important finding was the impact of looking ahead to future track positions. Although it appeared unlikely to have a significant impact based on how infrequently it affected the choice of a first-level branch, this is further supported by the verification statistics. Although it seems that looking ahead about 15 min, or approximately three levels, does result in a small improvement in tracking performance, it is much smaller than other enhancements in AALTO. In the data set used for testing, 15 min is approximately three volume scans, or a tree depth of three levels.

AALTO employs a similar approach to MHT, but does not include pruning. Many implementations of MHT use pruning to reduce the number of branches, allowing MHT to search future positions at greater depth without requiring excessive computing resources (Reid, 1979). The reasoning behind pruning is that it is not necessary to search all the possible candidate tracks, but only those likely to influence the choice of which first-level branch to follow. Although pruning is not currently implemented in AALTO, doing so would allow searching candidate tracks at greater depth. There are two primary considerations to implement pruning in AALTO: how deep is it actually necessary to search and how to determine which branches can be eliminated. Determining an appropriate search depth could be done by examining the differences between the cost for first-level branches and terminating the branches when changes in the cost function become too small to influence the selection of a first-level branch. Pruning could be implemented by not building branches that are much worse than other established branches from the same node.

Finally, this study presents a discussion of how to choose appropriate verification statistics to measure the goodness of a tracking algorithm. Prior work has focused on which attributes should be qualitatively associated with good tracks and designing appropriate verification metrics to quantify these principles, including track duration, straightness and continuity of a strength parameter such as VIL. This study proposes systematically degrading good tracks as they are poorer, while still retaining some skill. The metrics are then applied to both the good and degraded tracks and are evaluated for statistical significance. Those metrics that demonstrate a statistically significant difference between the good and the degraded tracks are deemed to be most useful verification of tracking algorithms. In this study, track straightness was found to be the best parameter, while track strength continuity was not statistically significant enough, and track duration was not significant at all. This approach provides an objective means of determining which verification metrics are most useful to evaluate the skill of a tracking algorithm.

Acknowledgements

Computational resources were provided by the University of Nebraska Holland Computing Center. This work was supported by the National Science Foundation award AGS-0758509 and Air Force Office of Scientific Research grant FA9550-12-1-0412. The authors would also like to thank Steve Lack and an anonymous reviewer for their feedback.

References

Barjenbruch BL. 2008. A Technique for developing an objective climatology of supercell and non-supercell thunderstorms, University of Nebraska-Lincoln, 70 pp.

Barjenbruch BL, Houston AL. 2006. A technique for developing the ratio of supercell to non-supercell thunderstorms. Preprints. *23rd Conference on Severe Local Storms*, 6–10 November 2006, Saint Louis, MO, American Meteorological Society, P2.7.

Benjamin SG, Devenyi D, Weygandt SS, Brundage KJ, Brown JM, Grell GA, et al. 2004. An hourly assimilation/forecast cycle: the RUC. *Mon. Weather Rev.* **132**: 495–518.

Blackman SS. 2004. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerosp. Electron. Syst. Mag.* **19**: 5–18.

Cox IJ, Hingorani SL. 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**: 138–150.

Dixon M, Wiener G. 1993. TITAN: thunderstorm identification, tracking, analysis, and nowcasting – a radar-based methodology. *J. Atmos. Oceanic Techol.* **10**: 785–797.

Han L, Fu S, Zhao L, Zheng Y, Wang H, Lin Y. 2009. 3D convective storm identification, tracking, and forecasting: an enhanced-TITAN algorithm. *J. Atmos. Oceanic Techol.* **26**: 719–732.

Johnson JT, MacKeen PL, Witt A, Mitchell ED, Stumpf GJ, Eilts MD, et al. 1998. The storm cell identification and tracking algorithm: an enhanced WSR-88D algorithm. *Weather Forecast.* **13**: 263–276.

Junyent F, Chandrasekar V, McLaughlin D, Insanic E, Bharadwaj N. 2010. The CASA integrated project 1 networked radar system. *J. Atmos. Oceanic Techol.* **10**(1): 61–78.

Lack SA, Limpert GL, Fox NI. 2010. An object-orientated multiscale verification scheme. *Weather Forecast.* **25**: 79–92.

Lahowetz J, Houston A, Limpert G, Gibbs A, Barjenbruch B. 2010. A technique for developing a US climatology of thunderstorms: The ThOR Algorithm. Preprints. *25th Conference on Severe Local Storms*, 11–14 October 2010, Denver, CO, American Meteorological Society, 16B.1.

Lakshmanan V, Hondl K, Rabin R. 2009. An efficient, general-purpose technique for identifying storm cells in geospatial images. *J. Atmos. Oceanic Technol.* **26**: 523–537.

Lakshmanan V, Miller M, Smith T. 2013. Quality control of accumulated fields by applying spatial and temporal constraints. *J. Atmos. Oceanic Technol.* **30**: 745–758.

Lakshmanan V, Smith T. 2010. An objective method of evaluating and devising storm tracking algorithms. *Weather Forecast.* **29**: 721–729.

Lakshmanan V, Smith T, Hondl K, Stumpf GJ, Witt A. 2006. A real-time, three dimensional, rapidly updating, heterogeneous radar merger technique for reflectivity, velocity and derived products. *Weather Forecast.* **21**: 802–823.

Lakshmanan V, Smith T, Stumpf G, Hondl K. 2007. The warning decision support system–integrated information. *Weather Forecast.* **22**: 596–612.

Miller ML, Lakshmanan V, Smith T. 2013. New rotation track quality control techniques for a multi-year climatology. Preprints. *11th Conference on Artificial and Computational Intelligence and its Applications to the Environmental Sciences*, 6–10 January 2013 Austin, TX, American Meteorological Society, 2.5.

Mitchell ED, Vasiloff SV, Stumpf GJ, Witt A, Eilts MD, Johnson JT, et al. 1998. The National Severe Storms Laboratory tornado detection algorithm. *Weather Forecast.* **13**: 352–366.

Reed J, Trostel J. 2012. Evaluation of an improved storm cell identification and tracking (SCIT) algorithm based on DBSCAN and JPDA tracking methods. Preprints. *22nd International Lightning Detection Conference/4th International Lightning Meteorology Conference*, Broomfield, CO, Vaisala. http://www.vaisala.com/en/events/ildcilmc/Documents/Severe/Evaluation%20of%20an%20Improved%20Storm%20Cell%20Identification%20and%20Tracking%20%28SCIT%29%20Algorithm.pdf (accessed 13 February 2014)

Reid D. 1979. An algorithm for tracking multiple targets. *IEEE Trans. Automat. Control* **24**: 843–854.

Root B, Yeary M, Yu T-Y. 2009. Analysis and improvements to the WSR-88D storm cell tracking algorithm. Preprints. *34th Conference on Radar Meteorology*, 5–9 October 2009, Williamsburg, VA, American Meteorological Society, 10.14.

Root B, Yeary MB, Yu T-Y. 2011. Novel storm cell tracking with multiple hypothesis tracking. Preprints. *27th Conference on Interactive Information Processing Systems*, 23–27 January 2011, Seattle, WA, American Meteorological Society, 8B.3.

Scharenbrouch L, Magnusdottir G, Smyth P, Sterm H, Wang C-c. 2010. A Bayesian framework for storm tracking using a hidden-state representation. *Mon. Weather Rev.* **138**: 2132–2148.

Sieglaff JM, Hartung DC, Feltz WF, Cronce LM, Lakshmanan V. 2013. A satellite-based convective cloud object tracking and multipurpose data fusion tool with application to developing convection. *J. Atmos. Oceanic Techol.* **30**: 510–525.

Stumpf GJ, Smith TM, Manross KL, Gerard AE. 2003. Warning decision support system – integrated information (WDSS-II). Part II: Real-time test at Jackson Mississippi WFO. Preprints. *19th International Conference on Interactive and Information Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology,* 9–13 February 2003, Long Beach, CA, American Meteorological Society, P1.36.

Stumpf GJ, Witt A, Mitchell ED, Spencer PL, Johnson JT, Eilts MD, et al. 1998. The national severe storms laboratory mesocyclone detection algorithm for the WSR-88D. *Weather Forecast.* **13**: 304–326.

Zahraei A, Hsu K-I, Sorooshian S, Gourley JJ, Lakshmanan V, Hong Y, et al. 2012. Quantitative precipitation nowcasting: a Lagrangian pixel-based approach. *Atmos. Res.* **118**: 418–434.

