Quality Control of Weather Radar Data Using Texture Features and a Neural Network

V Lakshmanan[1], Kurt Hondl[2], Gregory Stumpf[1], Travis Smith[1]

Abstract— **Weather radar data is subject to many contaminants, mainly due to non-precipitating targets (such as insects and wind-borne particles) and due to anamalous propagation (AP) or ground clutter. Although weather forecasters can usually identify, and account for, the presence of such contamination, automated weather algorithms are affected drastically. We discuss several local texture features and image processing steps that can be used to discriminate some of these types of contaminants. None of these features by themselves can discriminate between precipitating and non-precipitating areas. A neural network is used for this purpose. We discuss training this neural network using a million-point data set, and accounting for the fact that even this data set is necessarily incomplete.**

I. Introduction

From the point of view of automated applications operating on weather data, echoes in radar reflectivity may be contaminated. These applications require that echoes in the radar reflectivity moment correspond, broadly, to "weather". By removing ground clutter contamination, estimates of rainfall from the radar data using the National Weather Service (NWS) Weather Surveillance Radar-Doppler 1998 (WSR-88D) can be improved [1], [2]. A large number of false positives for the Mesocyclone Detection Algorithm [3] are caused in regions of clear-air return [4]. A hierarchical motion estimation technique segments and forecasts poorly in regions of ground clutter [5], [6]. Hence, a completely automated algorithm that can remove regions of ground clutter, anamalous propagation and clear-air returns from the radar reflectivity field would be very useful in improving the performance of other automated weather algorithms.

For a good review of the literature on ground clutter contamination, the interested reader is refered to [7]. Local neighborhoods in the vicinity of every pixel in the three weather radar moments were examined by [2] and used for automated removal of non-precipitating echoes. They achieved success by examining some local statistical features (the mean, median, and standard deviation within a local neighborhood of each gate in the moment fields) and a few heuristic features. [7] introduced the "SPIN" which is the fraction of gate-to-gate differences in a 11x21 local neighborhood that exceed a certain threshold (2dBZ in practice) to the total number of such differences. [2] introduced the "SIGN", the average of the signs of the gate-to-gate difference field within the local neighborhood. [7] used a decision tree to classify pixels into two categories – precipitation and non-precipitating while [2] used a fuzzy rule base using features that included the SPIN feature introduced by [7]. In addition to these elevation-based features, some vertical-profile features were also used – the maximum height of a 5dBZ echo was used by [7]. [2] discussed the use of vertical differences between the two lowest reflectivity scans.

Neural networks (NNs) have been utilized in a variety of meteorological applications. For example, NNs have been used for prediction of rainfall amounts by [8] and for identification of tornados by [9]. In fact, [10] attempted to solve the radar quality problem using neural networks. However, the performance of the neural network was no better than a fuzzy logic classifier (Kessigner, personal correspondence), and the neural network attempt was dropped in favor of the much more transparent fuzzy logic approach described in [2]. We propose some rationale for why our neural network approach achieves significantly better results than the network developed by [10] in Section III-A.

II. The Neural Networks

A. Inputs

Based on the extensive literature on descriptions of AP and ground clutter [7], we chose as inputs to the neural network the following: the data value, the mean, the median and the variance of each of the three moments (reflectivity, velocity, spectrum width)

[1]V Lakshmanan, Gregory Stumpf and Travis Smith are with the Cooperative Institute of Mesoscale Meteorological Studies (CIMMS), University of Oklahoma. [2]Kurt Hondl is with the National Severe Storms Laboratory, Norman, OK

at the lowest tilt of the radar. In addition, we took the same four values for the second lowest tilt of the radar. Finally, we computed some of the textural features that have been found to be useful in discriminating between precipitation and AP/GC. These were the SPIN [7], the gate-to-gate average square difference [2] and the SIGN [2]. We included the vertical gradient (difference between the reflectivities at the two lowest scans) as a separate input to the neural network.

In addition to these discriminants described in the literature, we considered a few others:

1. The maximum vertical reflectivity, over all the elevations.
2. The maximum reflectivity in the local neighborhood.
3. A weighted average of the reflectivity values over all the elevations where the weight of each data point is given by the height of that pixel above the radar. This takes into account the entire vertical profile instead of just the first two elevations.
4. The sum of all the heights at which an echo exists (reflectivity value greater than 0 dBZ) at the pixel.
5. The homogeneity of the reflectivity field defined as:

$$hom_{xy} = \frac{\sum_{i \epsilon N_{xy}} \frac{1}{1+(\frac{I_{xy}-I_i}{\overline{I_{xy}}})^2}}{card(N_{xy}) - 1} \quad (1)$$

where N_{xy} is the set of valid pixels (I_i) in the neighborhood, N_{xy}, of the pixel at (x, y) in the image, I_{xy} is the pixel value and $card(N_{xy}$ is the number of such neighbors.

6. Echo-size defined as the fraction of neighbors whose values are within 10dBZ of this pixel's reflectivity value.
7. Fraction of inflection points with inflections at 5,10 and 15dBZ thresholds. An inflection point is defined similar to the SPIN [7] except that the inflection is defined not in a polar neighborhood, but along the entire radial until that point.
8. Echo-top height defined as the maximum height of reflectivity above a certain threshold. We used both 5dBZ and 10dBZ thresholds.
9. To decorrelate the data value from the mean and median, the difference between the data value and the local mean was used.

B. Computation of Inputs

Velocity data can be range-folded (aliased). In the WSR-88D, at the lowest tilt, the velocity scan has a shorter range than the reflectivity one. We therefore divided the training pixels into two groups – one where velocity data were available and another where there was no Doppler velocity (or spectrum width) information. Thus, two separate neural networks were trained. In real-time operation, the appropriate network was invoked for each pixel depending on whether there were velocity data at that point. All the neural network inputs were scaled such that each feature in the training data exhibited a zero mean and a unit variance when the mean and variance are computed across all patterns.

Histograms of a few selected features are shown in Figure 1. It should be noted that these features are not linear discriminants by any means – it is the combination of features that gives the neural network its discriminating ability. The histogram of Figure 1d illustrates the result of several strategies we adopt during the training, so that higher reflectivities are not automatically accepted.

C. Network Architecture

We used a resilient backpropagation neural network (RPROP) as described in [11]. There was one hidden layer. Every input unit was connected to every hidden unit, and every hidden unit to the output unit. In addition, there was a short-circuit connection from the input units directly to the output unit, to capture any linear relationships i.e. the network was "fully connected" and completely "feed-forward". Every hidden node had a "tanh" activation function, chosen because of its signed range. The output unit had a sigmoidal activation function: $g(a) = (1 + e^{-a})^{-1}$ so that the outputs of the networks could be interpreted as posterior probabilities [12]. Each non-input node had, associated with it, a bias value which was also part of the training.

The error function that was minimized was a weighted sum of the cross-entropy (which [12] suggests is the best measure of error in binary classification problems) and the squared sum of all the weights in the network:

$$E = E_e + \lambda \Sigma w_{ij}^2 \quad (2)$$

The first term is a variation of the cross-entropy error suggested by [12] and is defined as:

$$E_e = -\sum_{n=1}^{N} c^n (t^n ln y^n + (1 - t^n) ln(1 - y^n)) \quad (3)$$

where t^n is the target value of the nth training pattern (0 if non-precipitating and 1 if precipitating) while y^n

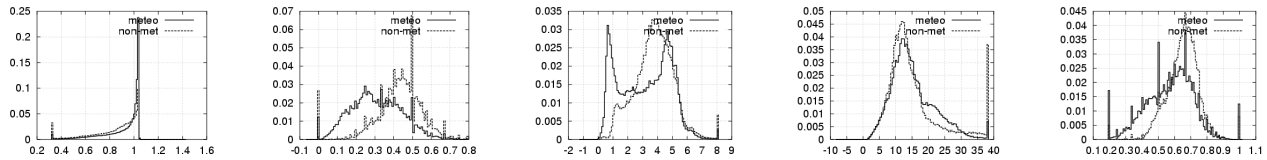


Fig. 1. Histograms of selected features on the training data set, after the features have been normalized to be of zero mean and unit variance. (a)Homogeneity (b) Radial inflections (c) Mean spectrum width (d) Mean reflectivity (e) SPIN. Note in (d) that, as a result of careful construction of the training set and selective emphasis, that the mean reflectivity histograms are nearly identical – this is not the apriori distribution of the two classes since AP is rare, and clear-air return tends to be smaller reflectivity values.

is the actual output of the neural network for that pattern input. N is the total number of patterns. The cost, c^n, captures the importance of that pattern. The second, square weights, term attempts to reduce the size of the weights, and thus improves generalization [13]. The relative weight, λ, of the two measures is computed every 50 epochs within a Bayesian framework with the assumption that the weights and the errors have Gaussian distributions, such that the ratio of their variances gives a measure of how much to decay the weights [14], [12]. We started by weighing the sum-of-weights twice as much as the cross-entropy term ($\lambda = 2$), updated λ based on the distribution of the weights and errors every 50 epochs and stopped the learning process at 800 epochs. We chose the final weights of the network from the epoch at which the *validation entropy error* was minimum, as will be discussed shortly.

The with-velocity network had 22 inputs, 5 hidden nodes and one output while the reflectivity-only network had 16 inputs, 4 hidden nodes and one output.

C.1 Validation

A validation set can ensure a network's generalization, typically through the use of early stopping methods [12]. In the neural network literature, a validation set is also utilized to select the architecture of the neural network [15]. We used a validation set that consisted of features derived from three volume scans that exhibited AP, convection and clear-air return.

We trained each network with three different numbers of hidden nodes. For each training run, we picked the result of training at the epoch that the validation error was at its minimum (See Figure 2). Thus, we used the validation set, both to determine when to stop, and to pick the final architecture of the neural network. Other than to choose the number of hidden nodes, we did not consider any alternate network topologies since, in theory at least, a single hidden layer is enough to interpolate any continuous function to arbitrary accuracy [12].

We used a testing set, independent of the training and validation sets, as decribed in Section III, and it is this independent set that the results are reported on.

D. Training

Eight volumes of WSR-88D data were selected. They covered a wide variety of weather and no-weather scenarios. A human interpreter examined these volume scans and drew polygons using the WDSS-II display [16] to select "bad" echo regions. An automated procedure used these human-generated polygons to classify every pixel into the two categories (precipitating and non-precipitating).

The data we have is not representative of true apriori probabilities, since each of the scenarios is a rare event. Patterns are assigned different importance factors c^n (See Equation 3). It is easy to see that if the cost factors are positive integers, the cost factor can be moved out of the error equation, by simply repeating the n^{th} pattern $c^n - 1$ times. In addition to assigning different costs, we also wished to train the network with approximately the same number of patterns in both classes. Because our dataset is necessarily incomplete, we repeat the patterns so as to have a balanced distribution of patterns at every reflectivity value. In the velocity network (a proxy for pixels close to the radar), precipitating echoes are repeated $d/20$ times while non-precipitating echoes are repeated $d/10$ times where d is the reflectivity value. Thus, AP with high-reflectivity (examples of which are hard to find when training with very few radar volumes) is emphasized as are strong reflectivity cores. In the no-velocity network, non-precipitating echoes are repeated $3d/5$ times. As can be seen from Equation 3, the repeating of patterns has the same effect as imposing a cost factor to each pattern. We are, in essence, assigning a higher cost to misclassifying high-dBZ pixels than to misclassifying low-dBZ pix-

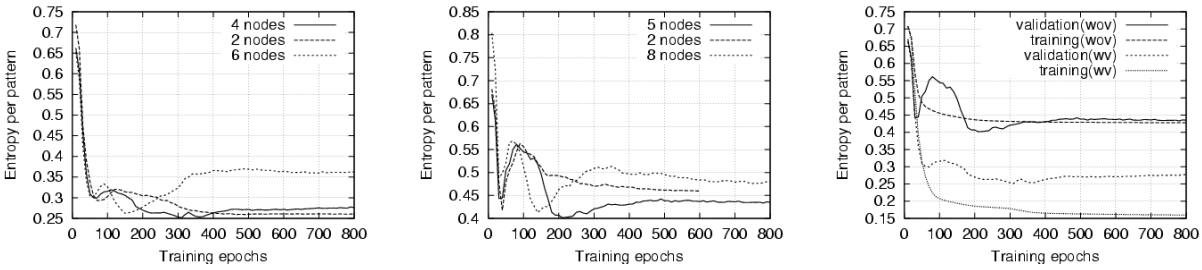


Fig. 2. Using a validation set to decide when to stop the training, and to decide on the number of hidden nodes. The y-axis is E_e/N – see Equation 3. (a) Validation error when training the without-velocity neural network. Final choice was 4 hidden nodes and the weights from the 310^{th} epoch. (b) Validation error when training the with-velocity neural network. Final choice was 5 hidden nodes and the weights from the 210^{th} epoch. (c) Training error vs Validation error for the final choices of hidden nodes. Note that the training error continues to decrease but the validation error starts to increase after a while, showing that the training is becoming counter-productive.

els. The histogram in Figure 1d shows the effect of this selective emphasis.

Some input vectors can be classified very easily because they are obvious. To avoid CPU cycles both in the training stage, and in the running stage, we pre-classify such pixels. Such pixels are not presented to the neural network in training, and pixels that match these criteria are pre-classified the same way in run-time as well. We discard shallow, low-reflectivity echoes and accept fast-moving and high-topped echoes.

In addition to emphasizing some pixels and pre-classifying others, we remove a third set of pixels from training altogether. In effect, we move them to an "ignore" category. These pixels are not presented to the network. The ignored pixels are those pixels for which the echo size is less than 0.2. Because of the way the echo size is defined, small echo sizes are points associated with speckle and are at the boundaries of storms where spatial statistics such as the mean and variance break down. To avoid the network expending cycles on these pixels, whose correct classification is not of paramount interest, these pixels are not part of the training at all.

In the process of training the networks, some of the computed inputs were removed and the neural network re-optimized. The probability of detection of precipitating echoes and the false alarm rates for both networks (with-velocity and reflectivity-only) were noted. If removing the input had no significant negative effect on the four statistics, the input was permanently removed.

Using this process, it was found that retaining just the mean and variance in the local neighborhood was enough – use of the median did not improve the capability of the neural network to learn the data, as measured by the probability of detection of precipitation, and the false alarm rate. We also found that the use of the maximum in the local neighborhood hurt trainability.

This pruning was not done in a rigorous manner. In particular, the numerous textural features were not pruned. We did not experiment with varying the set of features used for each moment – it is likely that we can use a different subset of features for the velocity than for the spectrum width, for example. Examination of the histograms did not yield many insights, since it is likely that it is a combination of features that possesses actual discrimination ability.

The final set of features used in the network for which results are reported were:

1. Lowest scan of velocity, spectrum width and the second lowest scan of reflectivity: local mean, local variance, difference between the data value and the mean
2. The lowest scan of reflectivity: local mean, local variance, difference between the data value and the local mean, REC Texture [2], homogeneity, SPIN [7], number of inflections at a 2dBZ threshold, SIGN [2], echo size.
3. Vertical profile of reflectivity: maximum value, weighted average, difference between data values at the two lowest scans, echo top height at a 5dBZ threshold.

During the process of training, we also discovered that one of the training cases was essentially untrainable. Rather than increase the complexity of the network, and risk a poor generalization, we chose to omit part of this data case from the training. The original reflectivity data, and the trained network's output on

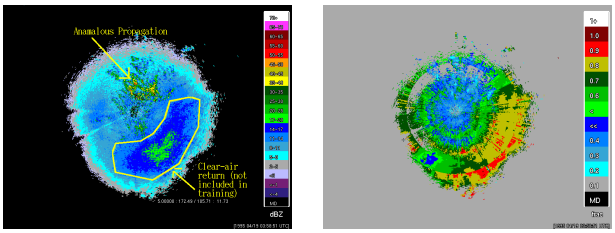


Fig. 3. Lowest scan of reflectivity from the KFWS radar at 1995/04/19 03:58:51UTC and the resulting classification of a network that included this data set in its training regimen. The network can learn to distinguish the AP, but not the clear-air return to the south-east of the radar. The unlearnable part of this volume scan (shown by the polygon) was removed from the training of the neural network.

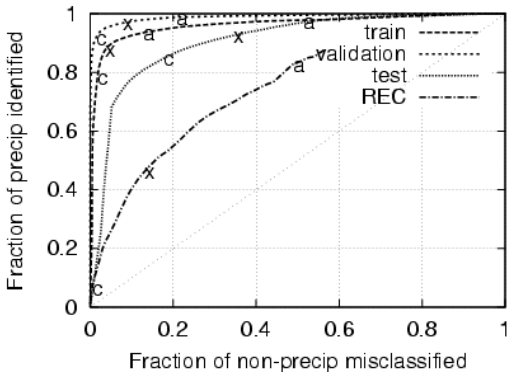


Fig. 4. A ROC curve showing the performance of the neural network on the training and testing data sets. Also shown, for comparision, is the performance of the Radar Echo Classifier. Three thresholds are marked on each of the curves – a indicates a 0.25 threshold, x a 0.5 threshold and c a 0.75 threshold. Classifiers with curves above the dashed diagonal can be considered skilled. The closer a classifier is squashed to the left and top boundaries of the graph, the better it is.

the data, are shown in Figure 3.

Finally, to improve the robustness of the local statistics being computed, we set all pixels in the reflectivity fields which could conceivably have had a radar return (those pixels with a height below 12km) which had a radar return below the noise threshold (and was therefore set to missing) to be zero. Thus, the only missing data values correspond to atmospheric regions which are not sensed by the radar at all.

Although the neural network computes the posterior probability that given the input vector, the pixel corresponds to precipitating echoes, adjacent pixels are not truly independent. Hence, the final 2D polar grid of posterior probabilities are mean filtered, and it is this mean-field that is used to perform quality control on the radar data. If the mean-field value is greater than 0.5, the pixel is assumed to have good precipitating data, and all elevations at that location are accepted. Bad data values are wiped out en-masse, although some researchers (e.g: [7]) use data from higher elevations in such cases.

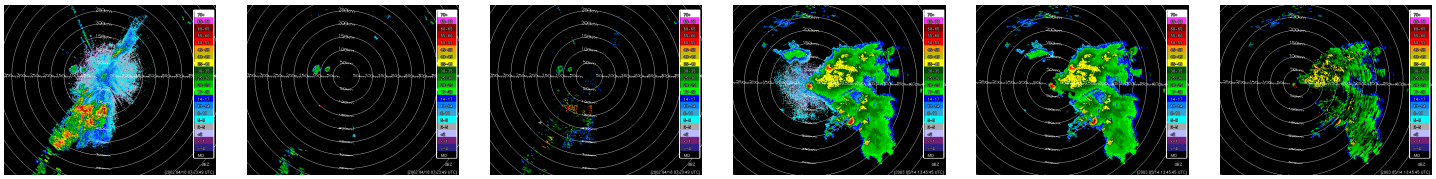
III. Results and Conclusions

A diverse set of volume scans of weather data were chosen and bad echoes marked on these volume scans by a human observer. The volume scans were processed using the trained neural network and using the Radar Echo Classifier [2]. Comparisions were made on a pixel-by-pixel basis of all pixels for which at least one of the elevations had a reflectivity value greater than zero dBZ. Performance is evaluated using the Receiver Operating Characteristics (ROC) curve [15], shown in Figure 4. In the ROC curve, the area under the curve can be taken as a measure of classifier skill (with areas above 0.5 showing considerable skill). Several thresholds are marked , so the sensitivity of classifier performance to the choice of threshold, as well as the effect of different thresholds may be gauged immediately.

A. Comparision with Cornelius

As mentioned in the introduction, [10] utilized a neural network to solve the same radar quality problem. The network developed in this paper has a significantly better performance. The reasons probably include:

1. The choice of error function: we minimized a combination of cross-entropy and square-weights whereas [10] minimized the mean absolute error, using the cross-entropy only for stopping criteria. The cross-entropy is a better measure of performance for a classifier [12] and the use of a weight-decay allows greater generalization [13].
2. Our use of a separate validation set to determine stopping criteria, whereas [10] did not, relying instead on two measures of performance on the same data set.
3. We used 4 or 5 hidden nodes, whereas [10] with a lesser number of inputs than our network used 15 hidden nodes.
4. Our use of nearly all pixels (other than the preclassified ones) in our data cases for training, whereas the pixels were chosen by hand or by random sampling by [10]. While a smaller selection improves training speed, the network is not trained on the full diversity of the data.
5. Our use of costs, (c_n in Equation 3) to direct the



a b c d e f

Fig. 5. Testing cases: (a) A data case with significant AP (b) Edited using the neural network (c) Edited using REC. Note that some very high-reflectivity AP values remain. (d) Typical spring precipitation (e) Edited using the neural network (f) Edited using REC. Note that quite a few good echoes have been removed from the stratiform rain region.

network to expend its training where the errors are less tolerable.

In contrast, the additional local neighborhood and vertical profile features we used provide only a small, incrementatal benefit.

As can be readily seen, the neural network outperforms the fuzzy logic automated technique of [2], one of a a number of algorithms that perform similarly [17]. The first three images of Figure 5 show a case of significant AP/GC while the last three show a significant precipitation event. Looking at these images, it is possible to put the quantitative measures in context. We see that a lot of good data is misclassified by the Radar Echo Classifier. At the same time, the neural network makes its mistakes on lower reflectivity values, but gets higher reflectivity values (whether AP/GC or good data) correct more often. This is a consequence of how the network was trained.

IV. Acknowledgements

Funding for this research was provided under NOAA-OU Cooperative Agreement NA17RJ1227, FAA Phased Array Research MOU, and the National Science Foundation Grants 9982299 and 0205628.

References

[1] R. Fulton, D. Breidenback, D. Miller, and T. O'Bannon, "The WSR-88D rainfall algorithm," *Weather and Forecasting*, vol. 13, pp. 377–395, 1998.

[2] C. Kessinger, S. Ellis, and J. Van Andel, "The radar echo classifier: A fuzzy logic algorithm for the WSR-88D," in *19th IIPS Conference*, (Long Beach, CA), Amer. Meteo. Soc., 2003.

[3] G. Stumpf, C. Marzban, and E. Rasmussen, "The new NSSL mesocyclone detection algorithm: a paradigm shift in the understanding of storm-scale circulation detection," in *27th Conference on Radar Meteorology*, 1995.

[4] K. McGrath, T. Jones, and J. Snow, "Increasing the usefulness of a mesocyclone climatology," in *21st Conference on Severe Local Storms*, (San Antonio, TX), Amer. Meteo. Soc., 2002.

[5] V. Lakshmanan, R. Rabin, and V. DeBrunner, "Multiscale storm identification and forecast," *J. Atmospheric Research*, pp. 367–380, July 2003.

[6] V. Lakshmanan, *A Heirarchical, Multiscale Texture Segmentation Algorithm for Real-World Scenes*. PhD thesis, U. Oklahoma, Norman, OK, 2001.

[7] M. Steiner and J. Smith, "Use of three-dimensional reflectivity structure for automated detection and removal of non-precipitating echoes in radar data," *J. Atmos. Ocea. Tech.*, no. 19, pp. 673–686, 2002.

[8] C. Venkatesan, S. Raskar, S. Tambe, B. Kulkarni, and R. Keshavamurty, "Prediction of all india summer monsoon rainfall using error-back-propagation neural networks," *Meteorology and Atmospheric Physics*, vol. 62, pp. 225–240, 1997.

[9] C. Marzban and G. Stumpf, "A neural network for tornado prediction," *J. App. Meteo.*, vol. 35, p. 617, 1996.

[10] R. Cornelius, R. Gagon, and F. Pratte, "Optimization of WSR-88D clutter processing and AP clutter mitigation," tech. rep., Forecast Systems Laboratory, Apr. 1995.

[11] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in *Proc. IEEE Conf. on Neural Networks*, 1993.

[12] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford, 1995.

[13] A. Krogh and J. Hertz, "A simple weight decay can improve generalization," in *Advances In Neural Information Processing Systems* (S. H. Moody, J. and R. Lippman, eds.), vol. 4, pp. 950–957, Morgan Kaufmann, 1992.

[14] D. J. C. MacKay, "A practical Bayesian framework for backprop networks," in *Advances in Neural Information Processing Systems 4* (J. E. Moody, S. J. Hanson, and R. P. Lippmann, eds.), pp. 839–846, 1992.

[15] T. Masters, *Practical Neural Network Recipes in C++*. San Diego: Morgan Kaufmann, 1993.

[16] K. Hondl, "Current and planned activities for the warning decision support system-integrated information (WDSS-II)," in *21st Conference on Severe Local Storms*, (San Antonio, TX), Amer. Meteo. Soc., 2002.

[17] M. Robinson, M. Steiner, D. Wolff, C. Kessinger, and R. Fulton, "Radar data quality control: Evaluation of several algorithms based on accumulating rainfall statistics," in *30th International Conference on Radar Meteorology*, (Munich), pp. 274–276, Amer. Meteo. Soc., 7 2001.