Tracking Severe Weather Storms in Doppler Radar Images

D. Cheng,[1] R. E. Mercer,[1] J. L. Barron,[1] P. Joe[2]

[1] Department of Computer Science, Middlesex College, University of Western Ontario, London,
Ontario N6A 5B7, Canada

[2] King City Radar Station, Atmospheric Environmental Services, 4905 Dufferin St., Toronto,
Ontario M3H 5T4, Canada

ABSTRACT: We describe an automatic storm-tracking system to help with the forecasting of severe storms. In this article, we present the concepts of fuzzy point, fuzzy vector, fuzzy length of a fuzzy vector, and fuzzy angle between two nonzero fuzzy vectors, that are used in our tracking algorithm. These concepts are used to overcome some of the limitations of our previous work, where fixed center-of-mass storm centers did not provide smooth tracks over time, while at the same time, their detection was very threshold sensitive. Our algorithm uses region splitting with dynamic thresholding to determine storm masses in Doppler radar intensity images. We represent the center of a hypothesized storm using a fuzzy point. These fuzzy storm centers are then tracked over time using an incremental relaxation algorithm. We have developed a visualization program using the X11 Athena toolkit for our storm visualization tool. The algorithm was tested on seven real radar image sequences obtained from the Atmospheric Environment Service radar station at King City, Ontario, Canada. We can obtain storm tracks that are long and smooth and which closely match an expert meteorologist's perception. © 1998 John Wiley & Sons, Inc. Int J Imaging Syst Technol, 9, 201–213, 1998

I. INTRODUCTION

Because of the devastation caused by severe storms, the forecasting of severe storm movement is one of the most important tasks facing meteorologists. To help with this task, we have developed an automatic storm-tracking system. Tracking past storm movement is a prerequisite to forecasting future storm movement and minimizing property damage.

Physically, a storm is an area of updraft and downdraft turbulence of air and water particles. In the updraft, a rain cloud is formed from the collision of water particles. When the water particles are too large to be sustained by the updraft, they fall as precipitation in the form of snow, rain, or hail. The lifecycle of an average storm containing a single storm cell usually ranges from 10 to 20 min. On the other hand, severe storms having multiple storm cells tend to have a much longer lifespan of up to a few hours, and therefore have the potential to cause significant damage.

II. OUR PROBLEM

Since 1985, the Cloud Physics Research Division of Atmospheric Environment Service (AES) of Canada has been developing a Doppler radar system to detect severe storms such as thunderstorms and tornadoes. The Doppler radar generates intensity and radial velocity images, examples of which are shown in Figures 1 and 2. These images are preprocessed to remove pixels corresponding to lines, bars, and writing. Intensity interpolation is used to represent the values of removed pixels. Typically, there are 0–20 potential severe storms per image.

The recognition and tracking of storms in these radar images is currently performed manually by human experts, and the task is time-consuming. To improve the efficiency and quality of weather forecasting, AES is interested in developing an automatic storm-tracking system for use in their operations. Toward this end, we have developed a tracking program with visualization capabilities that uses a hypothesize and verify model to detect storms in radar images and construct storm tracks. We first hypothesize storm masses in the Doppler radar intensity images. Then we verify the correctness of these hypothesized storms by tracking them over time. If a hypothesized storm can be tracked over a desired number of frames, we conclude that the storm is a valid storm and we record its track. When all potential storms are verified, a set of valid storm tracks will be taken as outputs.

The hypothesis of storm masses in Doppler radar intensity images involves low-level computer vision techniques such as image segmentation and feature extraction. On the other hand, the tracking of hypothesized storm features is basically a correspondence problem in high-level computer vision.

A. Aliasing of Data.

The Doppler radar intensity image sequences that we have used as experimental data are actual data obtained from the AES radar station at King City, Ontario. Because of the operational nature of the radar at the King City radar station, an intensity/velocity image is generated every 10 min. During the 10-min interval, storms can appear or disappear. Moreover, other storm properties such as size, intensity, and shape can change significantly. Under such a low sampling rate, the data can easily be aliased. If the matching of storms in image sequences uses storm properties as a measure of correspondence, or if the storms are not allowed to remain unmatched, this data aliasing can directly affect the accuracy and performance of the storm-tracking system.

B. The Merging and Splitting of Storms.

Through studying the storm movements in several Doppler intensity image se-

Correspondence to: R. E. Mercer or J. L. Barron

CCC 0899–9457/98/040201-13

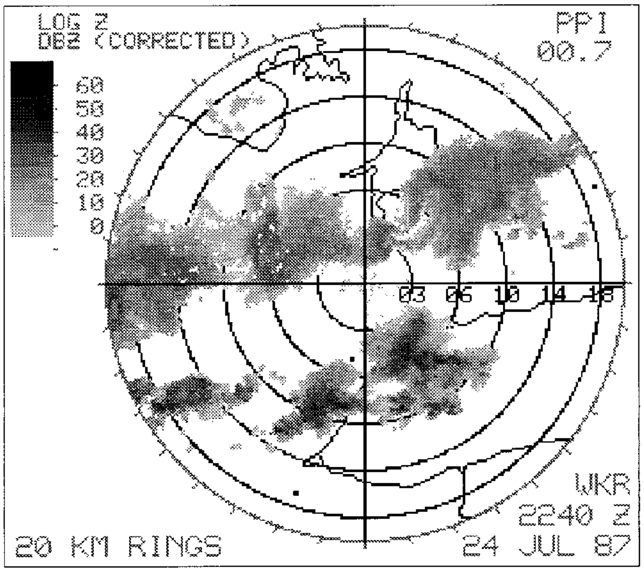


Figure 1. Doppler radar intensity image.

quences, two important phenomena are observed: A single storm may split into several separate storms, and two or more storms may merge into one storm in the next image. We have developed the concept of pseudo-storms to accommodate this behavior.

III. LITERATURE SURVEY

In this section, we briefly review some of the relevant literature on (storm) tracking. We especially emphasize how such work has influenced the design of our algorithms.

A. Correspondence. In studying the motion of an object in an image sequence, one of the basic tasks is to determine the correspondence between the representations of the same object in different images. This process is commonly referred to as the correspondence problem. Moreover, most tracking techniques are based on smoothness of motion, which assumes that the motion of an object cannot change abruptly.

Barnard and Thompson [1] used disparity analysis to determine the correspondence between a set of feature points selected from a stereogram, where a pair of images are taken simultaneously by laterally separated cameras from different positions. A disparity, represented as a vector, is constructed from a feature point in one image to all potentially matching points in the other. Note here that the possibility that a feature point is not matched is allowed (the no-match disparity). Based on the local continuity assumption of disparity, the certainty of each disparity at one feature point is iteratively refined by a relaxation-labeling algorithm. The refinement stops once a preset number of iterations is reached or some convergence criterion is satisfied. For each feature point in one image, the disparity with the highest certainty is selected as the best match.

Sethi and Jain [2] used a path coherence technique to obtain a trajectory of an object. The motion of the object is assumed to be smooth. They used a sequence of frames, instead of just two frames at a time, to determine correspondence. Trajectories are initially set up using the nearest-neighbor match, and the matches are iteratively refined using a greedy exchange algorithm which maximizes the path coherence. One of the difficulties with the use of multiple frames to establish correspondence is the problem of occlusion. Salari and Sethi [3] proposed a solution using phantom feature points. Incomplete tracks are padded out with these points, which are basically used as a computational convenience in a modified greedy exchange algorithm.

The path coherence technique was later generalized to property coherence by Sethi et al. [4]. They applied the property coherence method to track line tokens over multiple frames. They defined the property space for line tokens as a five-dimensional space consisting of the midpoint coordinates of the line token, the line length, the line orientation, and the directed distance of the line. Correspondence between line tokens was then based on these five attributes. Krezeski [5] also used property coherence in a temporal relaxation algorithm to track storms in Doppler radar image sequence (see below).

Lai [6] introduced an algorithm for tracking multiple features in an image sequence. The method uses relaxation labeling to iteratively deduce the disparity of each possible match. A constraint-aided exchange scheme is proposed to avoid recovering wrong correspondences originating from the occlusion of feature points.

B. Tracking Deformable Objects. Leymarie [7] proposed a method for tracking deformable objects, such as living cells. He used an active contour model, also commonly called a snake model, to represent the deformable shape. A snake, introduced by Kass et al. [8], is an energy-minimizing spline guided by external constraint forces and influenced by image forces that pull the spline toward features such as lines, edges, and subjective contours. User interaction is required to initialize the first frame of the tracking sequence. His algorithm provides good results when the deformation and movement of cells are small in consecutive frames (a constraint not satisfied by our data).

C. Automatic Storm Tracking. Automatic storm tracking can be considered a practical example of object tracking. In 1979,

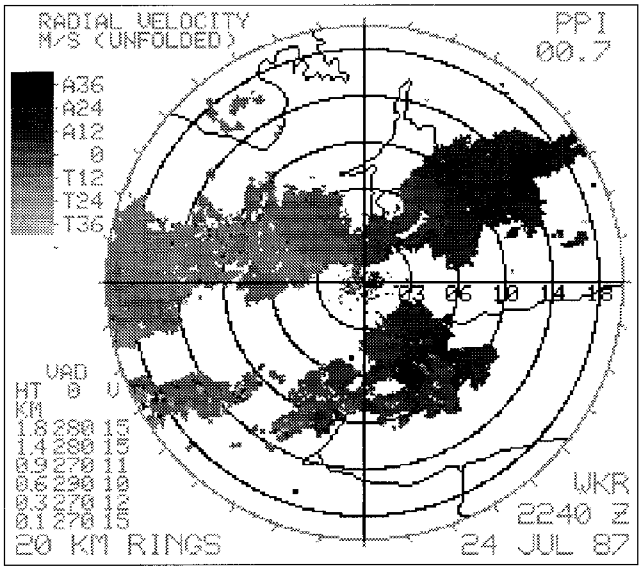


Figure 2. Doppler radar radial velocity image.

Bjerkaas and Forsyth (''Real-time automated tracking of severe thunderstorms using Doppler Weather Radar,'' Technical Note, Air Force Geophysics Laboratory, Hanscom AFB, 1979, unpublished) developed a simple storm-tracking system to assist meteorologists. The system used a nearest-neighbor algorithm to match storms in adjacent images. It is claimed to have a forecast error rate lower than the mean rate achieved in regular operations.

More recently, Hodges [9] has shown a practical use of the dynamic scene analysis techniques developed by Sethi [2,3] on data generated by general circulation models. His system achieved promising results in tracking some meteorological activities.

D. Zhang and Krezeski et al.'s Algorithms. The early work done in this project on automatic storm tracking can be found in a master's thesis by Zhang [10] and in a paper by Krezeski et al. [5]. Here, we highlight a few features in their work, as they form the basis of our current work.

Based on a merge and split algorithm developed by Horowitz and Pavlidis [11], Zhang [10] devised a similar region growing algorithm to identify storm masses in a sequence of Doppler radar intensity images. An image is postprocessed into a square digital image which is initially divided into a user-defined number of square regions or subnodes. A subnode $S = \{p_1, p_2, \ldots, p_n\}$ can be considered as a set of n pixels, where each pixel p_j represents a gray intensity level. Subnodes are then merged or split according to rules associated with a global threshold $T_{\bar{z}}$:

- A subnode S is split into four smaller, equal-sized square subnodes if the difference between the maximum and the minimum pixel values of the subnode is greater than the threshold $T_{\bar{z}}$:

$$max(S) - min(S) > T_{\bar{z}}$$

- Four adjacent subnodes, S_1, S_2, S_3, and S_4, are merged into a single subnode if the difference between the maximum and the minimum pixel values of the four subnodes is less than the threshold $T_{\bar{z}}$:

$$max(\bigcup_{j=1}^{4} S_j) - min(\bigcup_{j=1}^{4} S_j) \leq T_{\bar{z}}$$

The merge and split operations are mutually exclusive. That is, a subnode which is split cannot be merged, and vice versa. When no subnode can be further split or merged, the original image is completely segmented into many different square subregions which are represented in a pyramid data structure. Then, a grouping operation is performed to group adjacent regions to form an irregularly shaped region. A region whose size and average intensity are above two other thresholds is hypothesized to be a storm. The center of mass of an hypothesized storm region is then considered to be a hypothesized storm center.

By modifying Barnard and Thompson's spatial relaxation-labeling algorithm [1], Zhang developed a temporal relaxation algorithm to track storm centers represented by normal Euclidean points over time. The algorithm is based on the smoothness assumption of storm movement. Initially, disparities between the storm centers in adjacent images are constructed and each disparity is associated with a certainty value. The certainty value is then modified by relaxation on distance and angle compatibility. In the end, disparities with the highest certainty values are chosen as the best matches.

To handle the merging and splitting of storms, Zhang's algorithm allows single storms to be matched to several storms in the immediately previous or following images. This approach was less successful than desired because her tracking algorithm could incorrectly match premerged, single, small storms to a larger, single storm.

Krezeski et al. [5] added property coherence and the concept of pseudo-storms to improve Zhang's tracking algorithm. Their algorithm yielded some promising experimental results. Property coherence [4] allows multiple features of a storm to be tracked over time in addition to the location of the storm center. Five additional storm properties they considered were average intensity, storm size, velocity variance, storm shape and orientation, and convexity. The velocity variance property is the only use made of the Doppler velocity in our work to date. Areas of high velocity variance corresponded to storms.

Krezeski et al. [5] used the same merge and split algorithm as Zhang [10] to segment a radar intensity image and to hypothesize storm masses. Minor modifications are made to the algorithm to extract additional storm properties. A disparity is constructed from a storm in one image to a storm in the other if the distance between their storm centers is within a certain threshold, T_d. Connections between two adjacent disparities are established if the angle between the two is less than another threshold T_θ. A certainty value is associated with each disparity and is determined by a set of compatibility values. Initially, the compatibility between adjacent disparities is given by:

$$C_0(d_1, d_2) = w_6 c_d(d_1, d_2) + w_7 c_\theta(d_1, d_2) + \sum_{k=1}^{5} w_k \left(\frac{f_k(d_1) + f_k(d_2)}{2} \right)$$

where $w_1, w_2, \ldots, w_7$ are normalized weights which sum to 1, c_d is the length compatibility function, c_θ is the angle compatibility function, and f_k is the storm property function for property p_k, where $k = 1, 2, \ldots, 5$. For each property p_k, the property function computes the compatibility value of that property for each disparity d corresponding to two temporally connected storms, s_1 and s_2, in adjacent images:

$$f_k(s_1, s_2) = 1 - \left(\frac{|p_k(s_1) - p_k(s_2)|}{\max(p_k(s_1), p_k(s_2))} \times \frac{M_k}{M_k - m_k} \right),$$

where M_k and m_k denote, respectively, the maximum and the minimum values of property p_k of all storms s. The certainty value of each disparity is later refined iteratively by relaxation.

To handle situations where storms merge and split, Krezeski et al. introduced the notion of a pseudo-storm. Their idea is based on the work by Einfalt et al. [12], who suggested that storms which are close together could be considered as a single entity for comparison with other storms in adjacent images. Although the algorithm of Krezeski's et al. with pseudo-storm is still under development, initial experiments show that the merging and split-

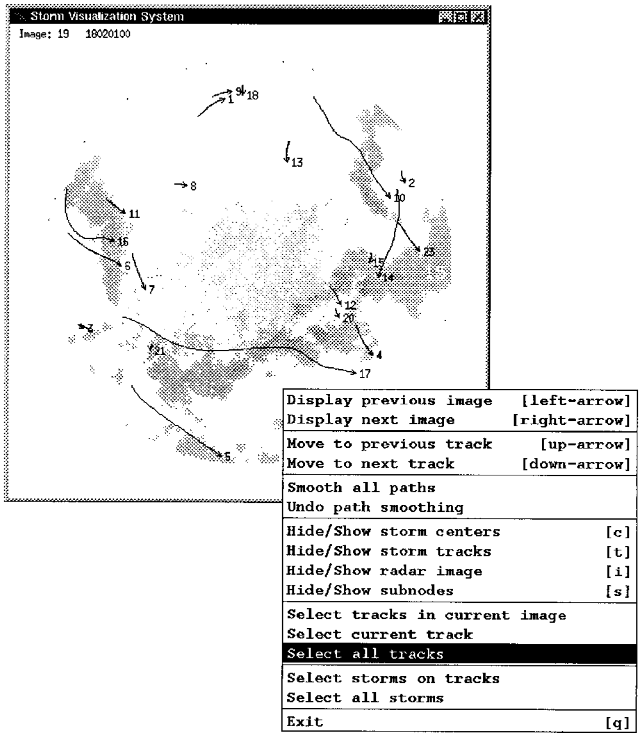

Storm Visualization System
Image: 19 18020100
Display previous image [left-arrow]
Display next image [right-arrow]
Move to previous track [up-arrow]
Move to next track [down-arrow]
Smooth all paths
Undo path smoothing
Hide/Show storm centers [c]
Hide/Show storm tracks [t]
Hide/Show radar image [i]
Hide/Show subnodes [s]
Select tracks in current image
Select current track
Select all tracks
Select storms on tracks
Select all storms
Exit [q]

Figure 3. Screen shot of our storm visualization program.

ting of storms can be handled nicely by matching real storms to pseudo-storms in adjacent images [5].

Krezeski et al.'s result showed some problems which can be attributed to the deformability and changing intensity values in the storm. The representation of the location of a storm in an image (which is required by the relaxation algorithm) is a point. The method used to determine this point (center of mass calculation) is sensitive to the properties of the storm (deformability and changing intensities). Although the storm moves in one direction, the algorithm can actually record a movement in the opposite direction owing solely to the somewhat arbitrary location of the center of mass of the storm in the subsequent image. This undesirable movement of the point representing the storm affects the performance of the relaxation algorithm. To obtain realistic results with the methodology described in Krezeski et al., we had to set thresholds in the relaxation algorithm at levels that sometimes allowed incorrect tracks to be generated.

IV. FUZZY STORM CENTERS

To diminish the effect of the arbitrariness of storm location, we have modified the representation of the center of a storm from an Euclidean point to a fuzzy point [13]. A fuzzy point is a circle whose inner region represents the uncertainty of the location of a targeted point. Since the fuzzy point represents the uncertain location of the storm center, we refer to the fuzzy point as the fuzzy storm center. Our current work [13,14] uses fuzzy points, fuzzy vectors, fuzzy lengths of fuzzy vectors, and fuzzy angles between two nonzero fuzzy vectors in the relaxation framework to produce more realistic storm tracks. Note that our interpretation of a fuzzy point is not based on the theory of fuzzy sets [15]; rather, it is in the spirit of fuzzy geometry [16,17].

A. Fuzzy Point Algebra. A fuzzy point $P = \langle c, r \rangle$ is a circle with center $c = (c_x, c_y)$ and radius r. It represents a region where a targeted point can arbitrarily be located. We denote the set of all fuzzy points as $\mathbf{P}$. We define the distance δ between two fuzzy points $P_1 = \langle c_1, r_1 \rangle$ and $P_2 = \langle c_2, r_2 \rangle$ as:

$$\delta(P_1, P_2) = \|c_1 - c_2\|_2 + |r_1 - r_2|,$$

where $(\mathbf{P}, \delta)$ forms a metric space. If we consider P as a subset of $\mathbf{R}^2$, then a fuzzy vector $\overrightarrow{P_1P_2}$ from fuzzy point P_1 to fuzzy point P_2 is the set of all displacement vectors from a point in P_1 to a point in P_2. The fuzzy length of this fuzzy vector is then defined as the set of lengths of all displacement vectors in $\overrightarrow{P_1P_2}$. This set can be represented as the real interval $[d_{\min}, d_{\max}]$, where $d_{\min}$ and $d_{\max}$ are, respectively, the least and greatest distance between any two points in the circles representing the fuzzy points P_1 and P_2. Any fuzzy vector with fuzzy length $[0, d]$, $d \geq 0$ is considered as a zero fuzzy vector. The fuzzy angle sub-

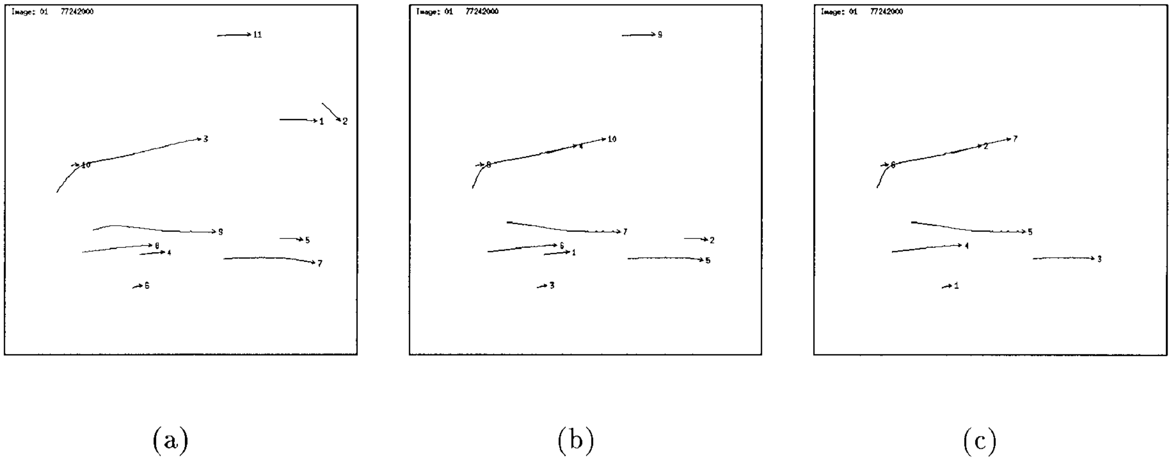


Figure 4. Verified storm tracks for series 77 images for (a) T_d = 10.0 and T_{sc} = 0.0; (b) T_d = 10.0 and T_{sc} = 0.5; and (c) T_d = 5.0 and T_{sc} = 0.5. Tracks in the 77 series image indicate that the storms image are moving from left to right.

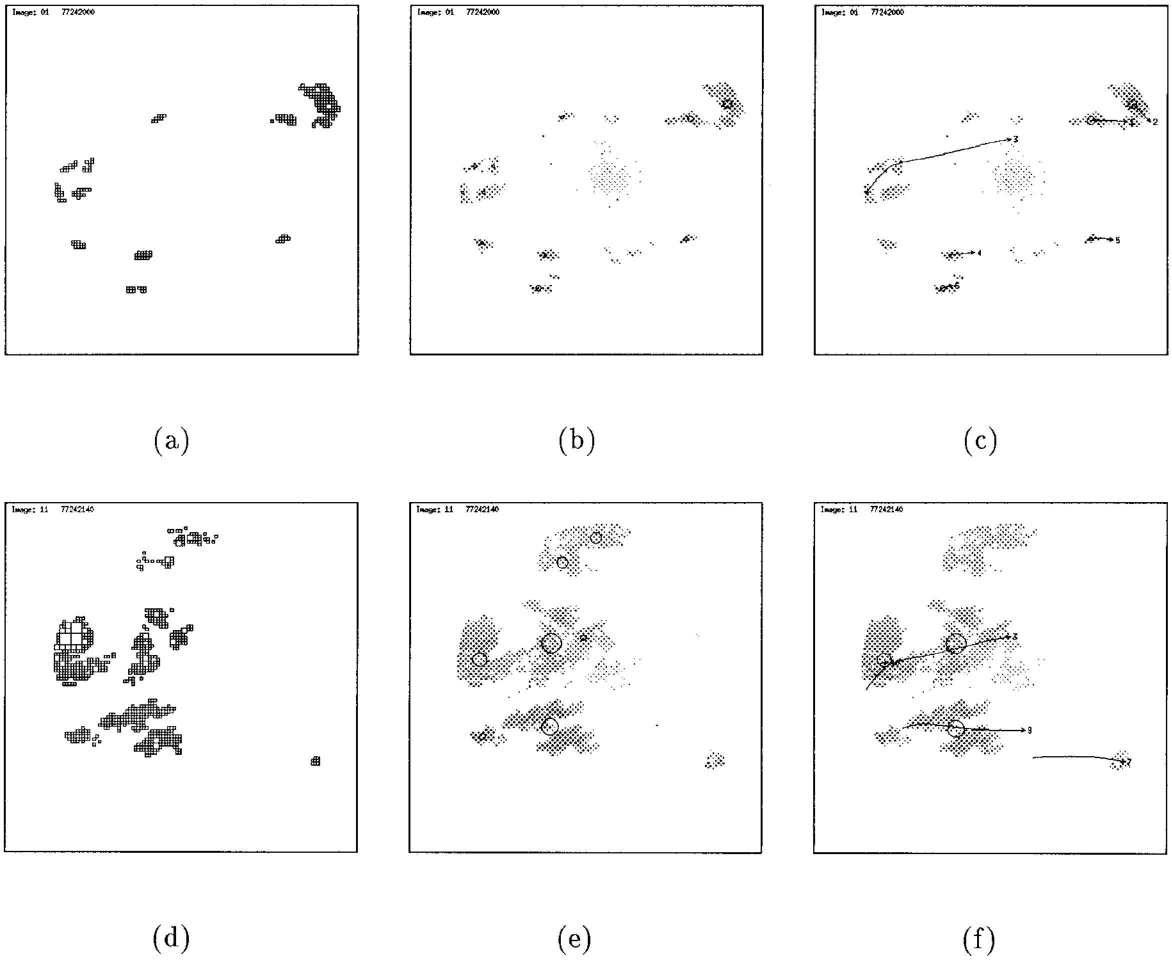


Figure 5. (a,d) The split regions of the first and 11th images, respectively, in the 77 series; (b,e) the storms hypothesized (denoted by circles) for these images: 11 for image (b) and eight for image (e); (c, f) the storm tracks for the first and 11th images which include a subset of all possible storm hypothesized in (b) or (e). $T_d = 10.0$ and $T_{sc} = 0.5$ for both sets of tracks.

tended by a nonzero fuzzy vector $\overrightarrow{P_2P_3}$ relative to a nonzero fuzzy vector $\overrightarrow{P_1P_2}$ is defined as the set of angles subtended by any displacement vector in $\overrightarrow{P_2P_3}$ relative to a displacement vector in $\overrightarrow{P_1P_2}$ having touching heads and tails, respectively. The angle between the two displacement vectors, θ, can be determined by the dot product. Like the fuzzy length of a fuzzy vector, the fuzzy angle between two nonzero fuzzy vectors is a real interval $[\theta_{\min}, \theta_{\max}] \subseteq [0, \pi]$. However, determining the two end points $\theta_{\min}$ and $\theta_{\max}$ is not trivial. Currently, we use an $O(n)$ search algorithm to approximate the two end points.

B. Storm Hypothesis. A region-splitting algorithm with dynamic thresholding is used to determine storm masses in a radar intensity image [13,14]. This algorithm replaces the one in [5] that uses Horowitz and Pavlidis' merge-split algorithm [11]. The merge-split process in [5] is determined by a threshold on the difference between the maximum and minimum intensity levels of adjacent regions. We have observed that this criterion is sensitive to outliers in the data. It also requires the appropriate threshold to be manually chosen for each image. In our new algorithm, we use the standard deviation s of the average intensity level of a region to govern the splitting process. We find that thresholding using s is very robust and no user interaction is required.

A square image S is initially at level 1 of a quad-tree and is divided into four equally sized subnodes, S_1, S_2, S_3, and S_4, which correspond to the northwestern, northeastern, southeastern, and southwestern regions respectively.

For each subnode S_n, where $n \in \Lambda^+$ and $\Lambda = \{1, 2, 3, 4\}$, we compute the mean, $\bar{z}$, and the standard deviation, s_z, of the intensity levels of all pixels in S_n. The splitting criterion of the subnode is based on the value of s_z. The value of s_z indicates how well the value of $\bar{z}$ represents the average intensity level of all pixels in S_n. We currently set a threshold T_s on s_z to 10.

- If s_z is large, as indicated by $s_z \geq T_s$, then the intensity levels of most pixels in S_n differ significantly from $\bar{z}$, and therefore, the subnode S_n is split into four smaller subnodes, S_{n1}, S_{n2}, S_{n3}, and S_{n4}, at the next level.
- On the other hand, if s_z is small, as indicated by $s_z < T_s$, then we consider $\bar{z}$ to be representative of the intensity levels of most pixels in S_n and we compare the value of $\bar{z}$ to the dynamic threshold $T_{\bar{z}}$. $T_{\bar{z}}$ is based on the mean, $\bar{z}'$, and standard deviation, s_z', of the intensity levels of a subset of pixels in the image with intensity levels greater than $z_{\min} = 16$. It is computed as $T_{\bar{z}} = \bar{z}' + ks_z'$, where $k = -0.5$. If we

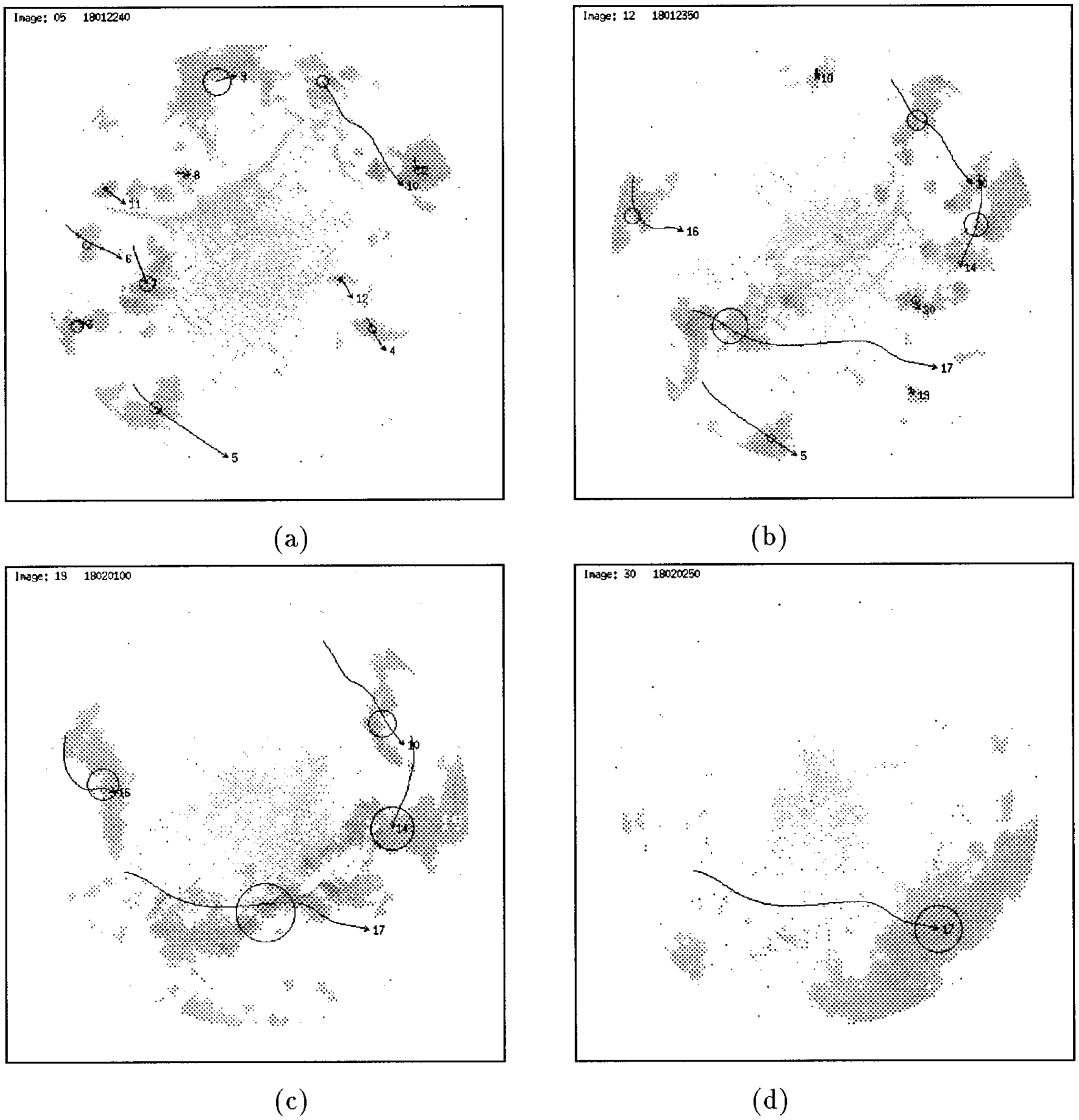


Figure 6. The storm tracks for the (a) fifth, (b) 12th, (c) 19th, and (d) 30th images of the 18 series. T_d = 5.0 and T_{sc} = 0.6.

have $\bar{z} \geq T_{\bar{z}}$, then the subnode S_n will be considered as a part of a potential storm and will be marked for further processing.

The above splitting process continues recursively until no subnode can be further split. Neighboring subnodes that are marked for further processing will be grouped into a single region,

$$R = S_{k_1} \cup S_{k_2} \cup \cdots \cup S_{k_n}$$

if they are connected. We say that a subnode S' is connected to a subnode S with a distance d if all of the following conditions hold,

$$l_y + d \geq u'_y$$

$$u_y - d \geq l'_y$$

$$l_x + d \geq u'_x$$

$$u_x - d \leq l'_x$$

where (u_x, u_y) and (l_x, l_y) denote the upper-left and lower-right corners of S, respectively; (u'_x, u'_y) and (l'_x, l'_y) denote the upper-left and lower-right corners of S', respectively; and d is a threshold currently set to two pixels.

C. Construction of Fuzzy Storm Centers. After region splitting, we have a set of regions $\{R_1, R_2, \ldots, R_n\}$ in the Doppler radar intensity image which are hypothesized as storms. To repre-

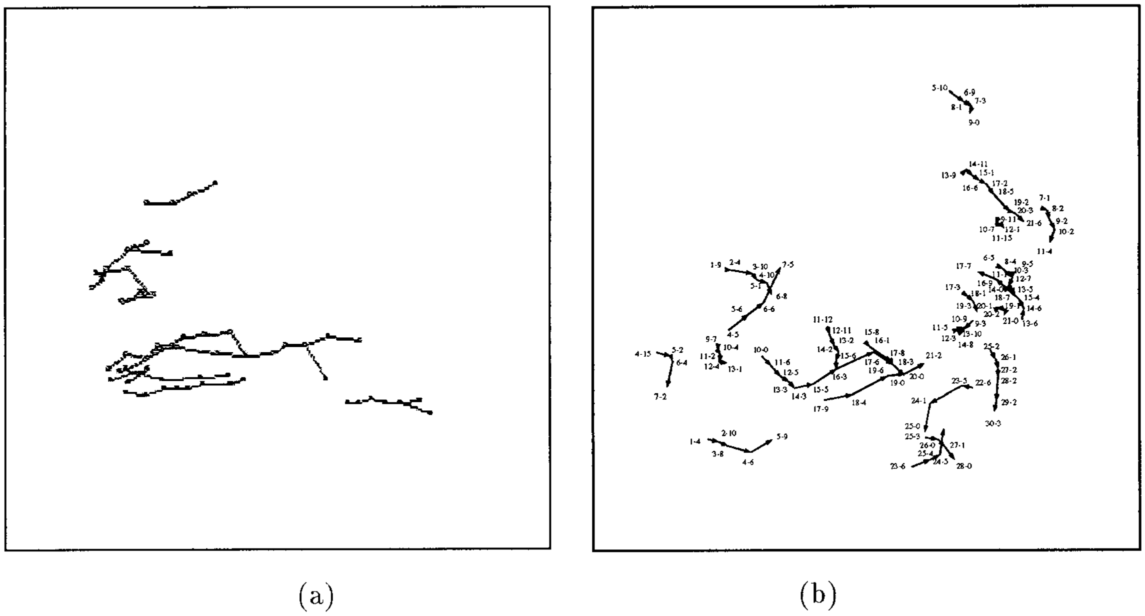


Figure 7. (a) Zhang's tracks for the 77 series and (b) Krezeski et al.'s tracks for the 18 series.

sent the location of each storm region, R_j, using a fuzzy point $P = \langle c, r\rangle$, we first compute the weighted averages, $\bar{x}$ and $\bar{y}$, of the centers of all subnodes forming the region, in the x- and y-direction, respectively; and also the corresponding standard deviations, s_x and s_y. Then, the center of the fuzzy point P is taken as $c = (\bar{x}, \bar{y})$; and the radius of the fuzzy point is determined by $r = k_r \max(s_x, s_y)$, where k_r is a parameter to control the size of the fuzzy point. We currently use $k_r = 0.5$. We interpret the above construction as fitting a quasi-circular Gaussian surface through the storm region. Since the data points in the storm region do not necessarily spread around the peak of a Gaussian surface, we could not employ a least-square method to perform a Gaussian fit.

V. INCREMENTAL RELAXATION-LABELING ALGORITHM

Once a set of storms has been hypothesized for the radar intensity image sequence, the correctness of these storms can be verified by tracking them over time. Our tracking algorithm is based

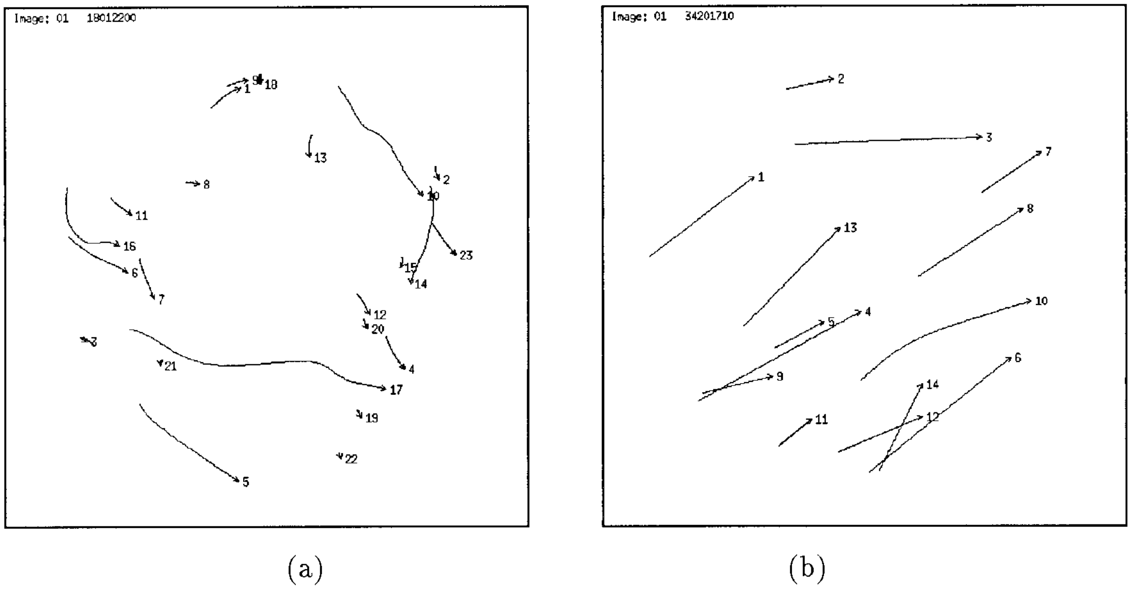


Figure 8. The verified tracks for (a) the 18 series and (b) the 34 series. $T_d = 5.0$ and $T_{sc} = 0.6$ for the 18 series; and $T_d = 5.0$ and $T_{sc} = 0.45$ for the 34 series.

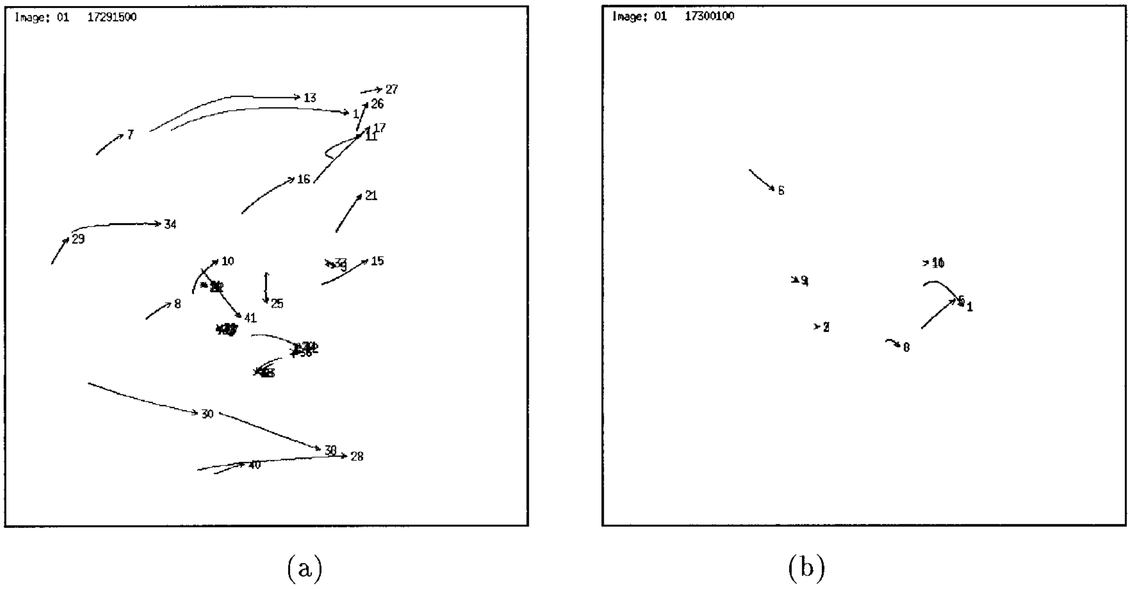


Figure 9. The verified tracks for (a) the 17a series and (b) the 17b series. $T_d = 5.0$ and $T_{sc} = 0.75$ for both series.

on Krezeski et al.'s temporal relaxation algorithm with property coherence [5]. We have selected the size of a fuzzy storm center as a property to be coherent over time.

Let S_k be a hypothesized fuzzy storm center in the kth image. A disparity represented by a fuzzy vector $\overrightarrow{S_jS_{j+1}}$ is constructed from S_j to S_{j+1} if the infimum of the fuzzy length of the fuzzy vector is less than a threshold, T_d, which is set to a default value of 10 pixels; and concurrently, the two fuzzy storm centers have compatible sizes. We define a property function, f_s, to measure the size compatibility of two fuzzy storm centers, $S_1 = \langle c_1, r_1\rangle$ and $S_2 = \langle c_2, r_2\rangle$, as:

$$f_s(S_1, S_2) = \begin{cases} 1 - \dfrac{|r_1 - r_2|}{\max(r_1, r_2)} & \text{if } r_1 > 0 \text{ or } r_2 > 0, \\ 1 & \text{otherwise.} \end{cases}$$

A size compatibility threshold, T_{sc}, is set to 0.5. Note that if T_{sc} is set to 1, then a disparity will be constructed between two fuzzy storm centers only when they have exactly the same size. On the other hand, if T_{sc} is set to 0, then the size compatibility criterion is effectively removed. We measure the partial compatibility between two adjacent disparities using a weighted sum of three components: length compatibility C_d, angle compatibility C_θ, and size compatibility C_s. The overall compatibility function is defined as:

$$C = w_dC_d + w_\theta C_\theta + w_sC_s,$$

where w_d, w_θ, and w_s are normalized weights such that $w_d + w_\theta + w_s = 1$. We currently use $w_d = 0.2$, $w_\theta = 0.2$, and $w_s = 0.6$. Values for these and other weight coefficients (below) were chosen by empirical observation. Two adjacent disparities are connected together if their compatibility value is greater than a threshold:

$$C(\overrightarrow{S_jS_{j+1}}, \overrightarrow{S_{j+1}\,S_{j+2}}) > T_c,$$

where T_c is currently 0.2. When all qualified adjacent disparities have been linked together, the certainty of each disparity is refined iteratively by relaxation on the overall compatibility among its adjacent disparities. Consider a disparity $d = \overrightarrow{S_jS_{j+1}}$. The initial certainty of the disparity, denoted as $p_0(d)$, is set to $f_s(S_j, S_{j+1})$. During each iteration, we apply both spatial and temporal consistency constraints to compute the supporting and contradictory evidence of the disparity using the compatibility values. Let E_s and E_c denote the supporting and contradictory evidence, respectively. Let n_s and n_c denote the number of supporting disparities and the number of contradictory disparities, respectively. These four quantities are reset at the start of each iteration.

- To apply the temporal consistency constraint, for each adjacent disparity d_t to d of the form $d_t = \overrightarrow{S_{j-1}S_j}$ or $d_t = \overrightarrow{S_{j+1}S_{j+2}}$, we compute the compatibility at the kth iteration ($k > 0$) between the two disparities as:

$$C_k(d, d_t) = w_1C(d, d_t) + w_2\left(\frac{p_{k-1}(d) + p_{k-1}(d_t)}{2}\right)$$

where w_1 and w_2 are normalized weights that sum to 1. We currently use $w_1 = 0.4$ and $w_2 = 0.6$. If $C_k(d, d_t) > T_k$ (we use $T_k = 0.6$), then we add $p_{k-1}(d)$ to E_s and increment n_s by 1. Otherwise, we add $p_{k-1}(d)$ to E_c and increment n_c by 1.
- To apply the spatial consistency constraint, for each disparity d_s which has the same head storm or tail storm as d, if $p_{k-1}(d) \geq p_{k-1}(d_s)$, then we add $p_{k-1}(d)$ to E_s and increment n_s by 1. Otherwise, we add $p_{k-1}(d)$ to E_c and increment n_c by 1.
- The certainty of the disparity d at the kth iteration is modified by:

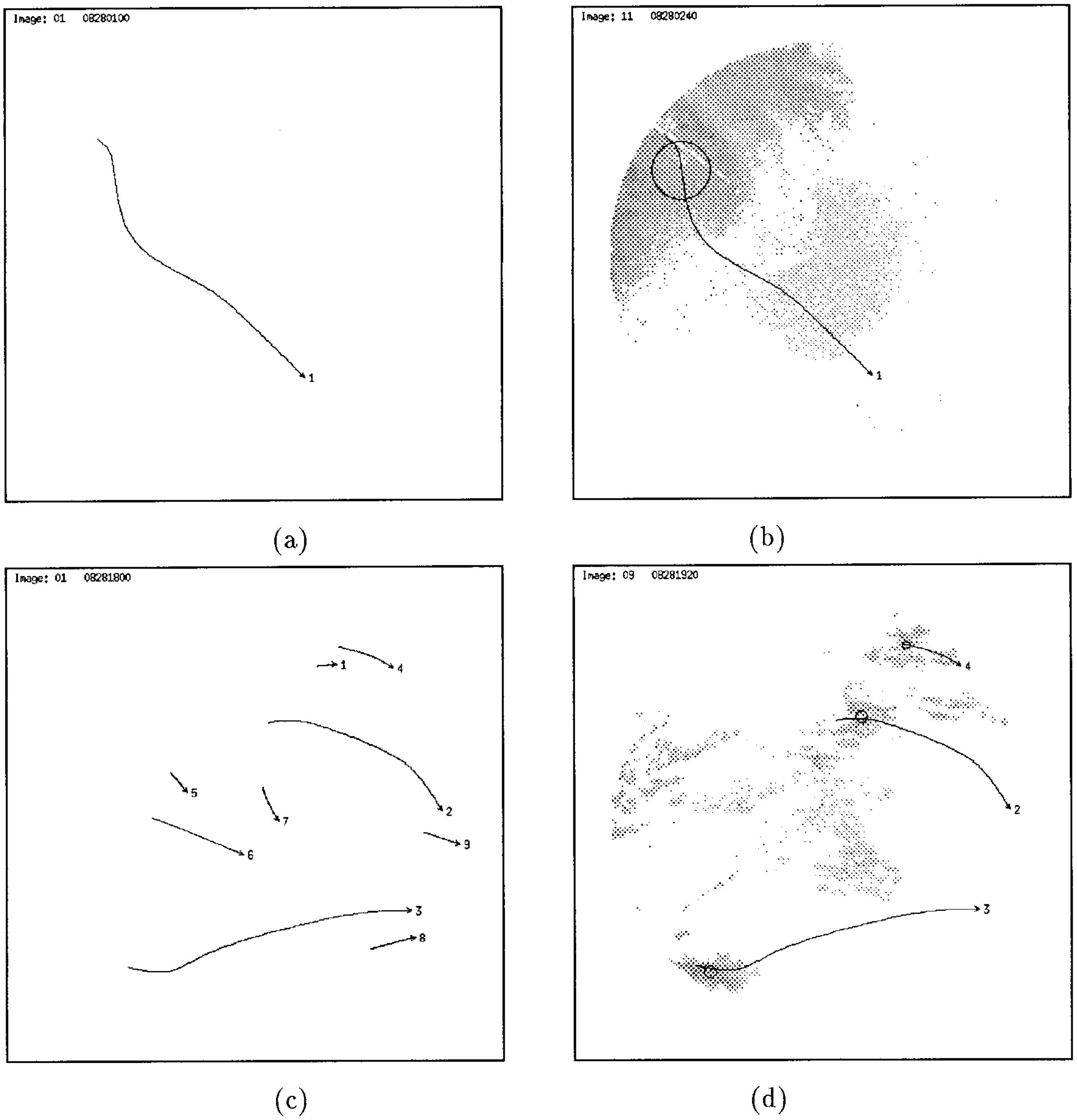


Figure 10. (a,c) Verified tracks, and (b,d) tracks for the 11th and ninth images of the 08a series and 08b series sequences, respectively. In both cases, $T_d = 10.0$ and $T_{sc} = 0.5$.

$$p_k(d) = \begin{cases} \frac{1}{2}\left(1 + \frac{w_sE_s - w_cE_c}{w_sE_s + w_cE_c}\right) & \text{if } E_s \neq 0 \text{ or } E_c \neq 0, \\ 0 & \text{otherwise;} \end{cases}$$

where w_s is the weight of the supporting evidence and is computed as $w_s = n_s/(n_s + n_c)$, and w_c is the weight of the contradictory evidence and is computed as $w_c = n_c/(n_s + n_c)$.

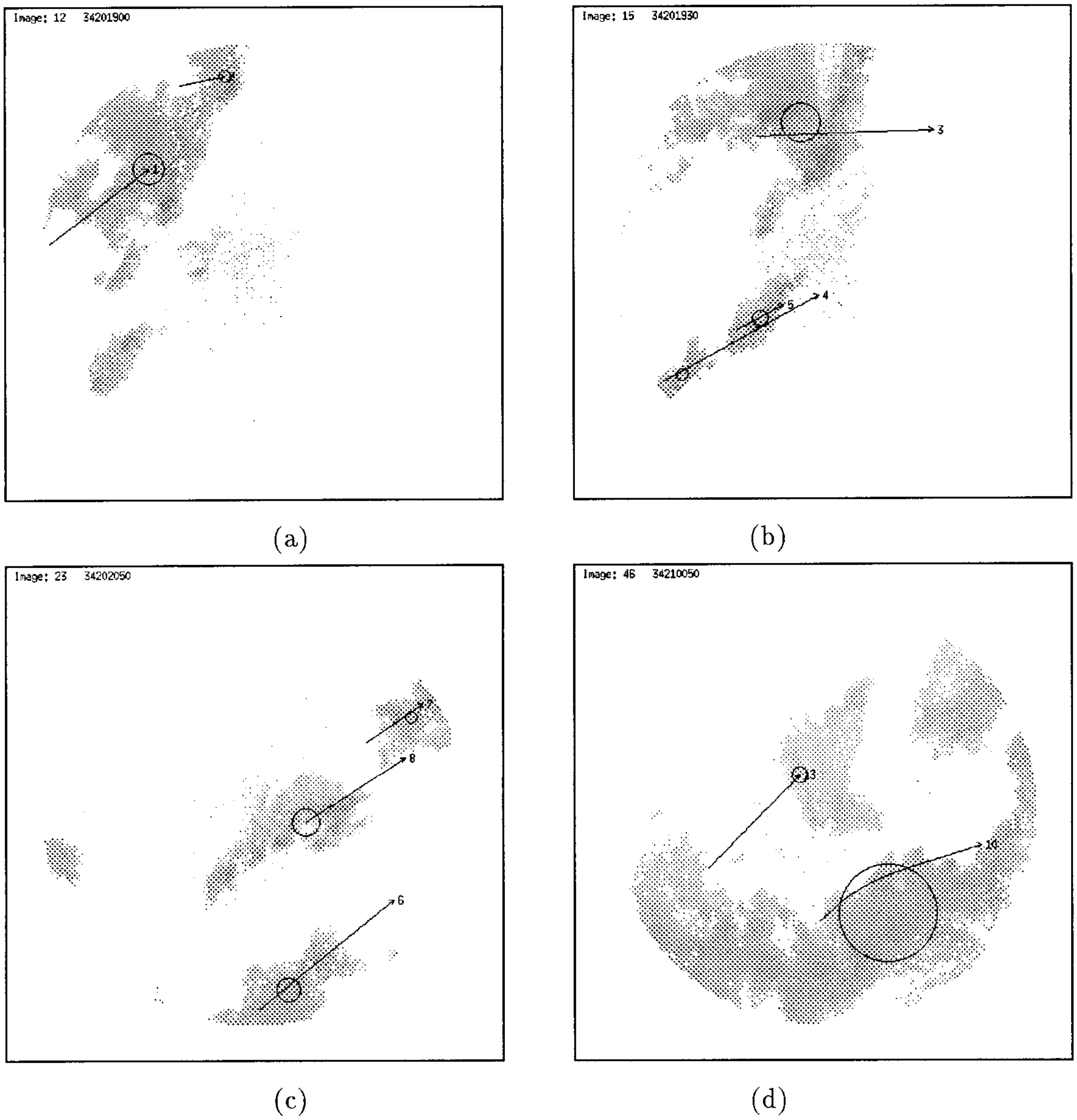
- The iterative process stops at the kth iteration when the certainty of each disparity has converged to the desired level of confidence (ϵ), say, to n decimal places (we use $n = 6$):

$$\epsilon = |p_k(d) - p_{k-1}(d)| < 10^{-n},$$

for each disparity d or the maximum number of iterations has been reached: $k \rightarrow T_k$, where T_k is currently set to 20.

Once the relaxation process has converged, we construct a set of all longest tracks such that each disparity has a final certainty over a threshold T_p (we use $T_p = 0.85$). We choose a subset of these tracks, with the condition that no storm lies upon more than one chosen track.

We have changed the implementation of the algorithm from processing of a complete image sequence of known length (i.e., batch mode) to be in incremental mode. Given n images, the hypothesized storm disparities are relaxed. When an $(n + 1)$th image is added, the relaxation is restarted using the results for the first n images plus the hypothesized storms of the $(n + 1)$th image. We have observed empirically that the result is always



(a) (b) (c) (d)

Figure 11. The storm tracks for the (a) 12th, (b) 15th, (c) 23rd, and (d) 46th images of the 34 series. T_d = 5.0 and T_{sc} = 0.45.

the same as if all $n + 1$ images had been initially relaxed. The difference in computation speed between the two methods is insignificant, since relaxation converges within 10 iterations in either mode. The incremental algorithm allows us to view the current storm tracks as the data become available.

VI. AN X-WINDOW-BASED STORM VISUALIZATION PROGRAM

We have developed and implemented a storm visualization program using the standard X11 Athena toolkit. We have chosen the Athena toolkit for its portability across different UNIX platforms. Our program has been compiled and tested on several platforms including a Sun Sparcstation, an IBM RS/6000 workstation, and an Intel Pentium workstation and notebook. The main design goal of this storm visualization program was to allow the study of storm movements in radar intensity image sequences and to visualize fuzzy storm centers and storm tracks generated by our storm-tracking algorithms.

A. Storm-Track Visualization. The original algorithms by Zhang and Krezeski et al. used polylines to draw tracks connecting storm centers represented by Euclidean points. We find this method inappropriate for fuzzy storm centers because the actual location of the storm center can be anywhere inside the circumference of a fuzzy point. In addition, the zigzag look of a polyline fails to represent the smooth movement of a storm. Therefore, we have used cubic B-spline approximation [18] with storm center averaging to draw the storm tracks. We did not use cubic B-spline interpolation because this would involve solving a set of simultaneous linear equations, and we do not need the curve to pass through the center of a fuzzy point. Because we select the centers of the fuzzy points as the knot points of the spline

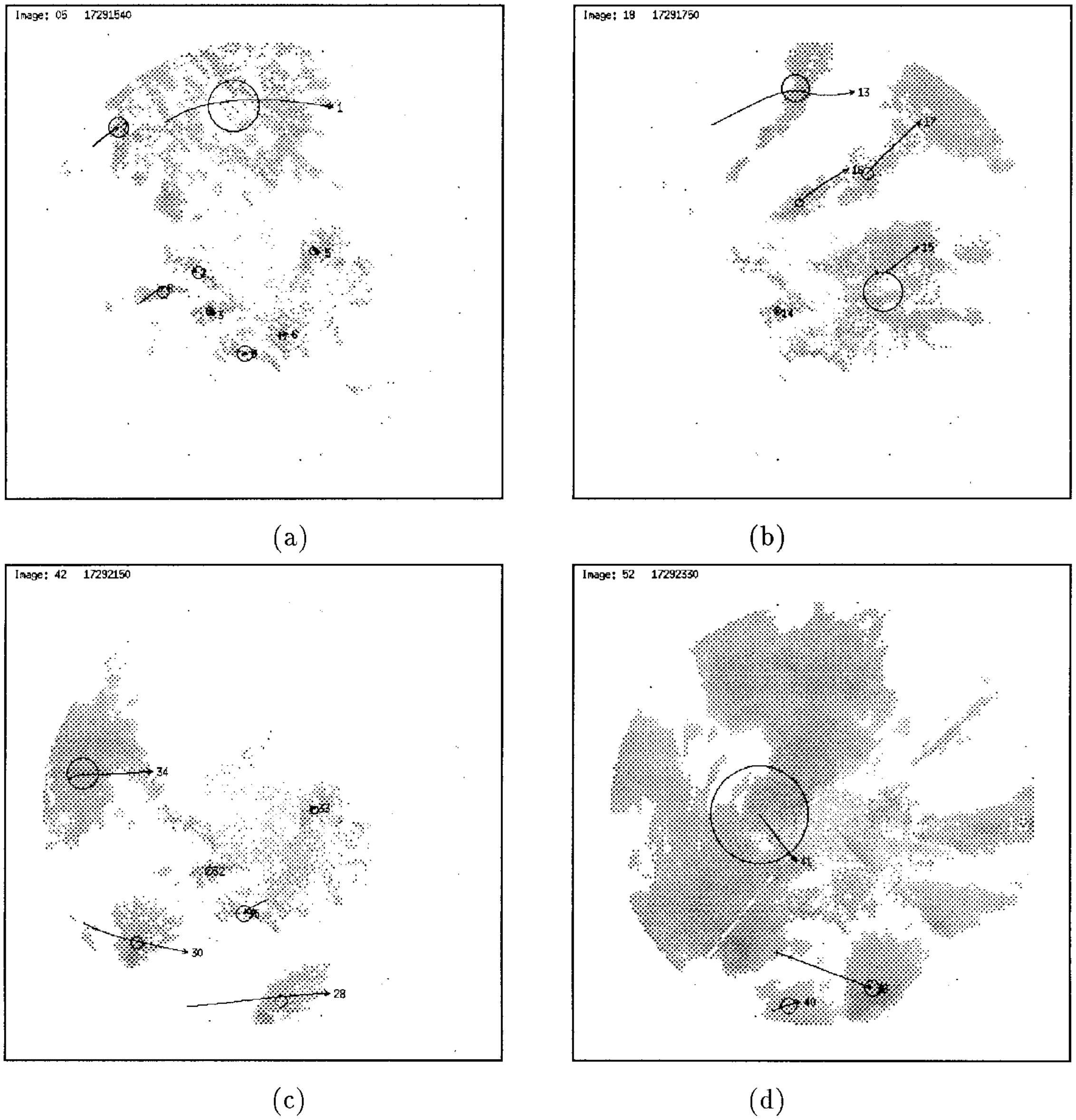


Figure 12. The storm tracks for the (a) fifth, (b) 18th, (c) 46th, and (d) 52nd images of the 17a series. $T_d = 1.0$ and $T_{sc} = 0.75$.

curve, the curves have many bends. To improve the smoothness of the spline curve, we used a simple smoothing filter to apply on the spline knots. For a set of spline knots P_{-1} to P_{n+1}, we construct a set of smoothed spline knots P'_{-1} to P'_{n+1} using a smoothing filter F: $P'_j = F(P_j) = \frac{1}{3}(P_{j-1} + P_j + P_{j+1})$, if $j = 1, 2, \ldots, n - 1$, and P_j otherwise. We then use this set of smoothed spline knots P'_j to plot the spline curve.

The interaction between the storm visualization program and a user is performed through a pop-up menu (see Fig. 3 for a screen shot of the program with the pop-up menu activated).

VII. EXPERIMENTAL RESULTS

In this section, we present results of our tracking algorithm. We have performed experiments on seven sequences from the radar station at King City, Ontario. These are denoted as the 77 series, the 18 series, the 08a series, the 08b series, the 34 series, the 17a series, and the 17b series [14]. We compare our results with those obtained from the algorithms of Zhang [10] for the 77 series and those obtained from the algorithm by Krezeski et al. [5] for the 18 series.

In our algorithm there are a number of thresholds to be set. For the region-splitting algorithm, $z_{\min}$ governs the background pixel intensity, K governs the dynamic threshold, and T_s governs the region-splitting operation. For all results in this article, we use the values $z_{\min} = 16$, $K = -0.5$, and $T_s = 10.0$. In the tracking algorithm, the construction of a disparity between two fuzzy storm centers in adjacent images is controlled by two thresholds: T_d governs the closest distance between two fuzzy storms and T_{sc} governs the size compatibility between two fuzzy storms. Usually T_s is 10, but occasionally it is tightened to 5 to remove unwanted storms. T_{sc} is usually set at 0.5. As we noted above, $T_{sc} = 0.0$ turns off size compatibility, while $T_{sc} = 1.0$ enforces strict size compatibility.

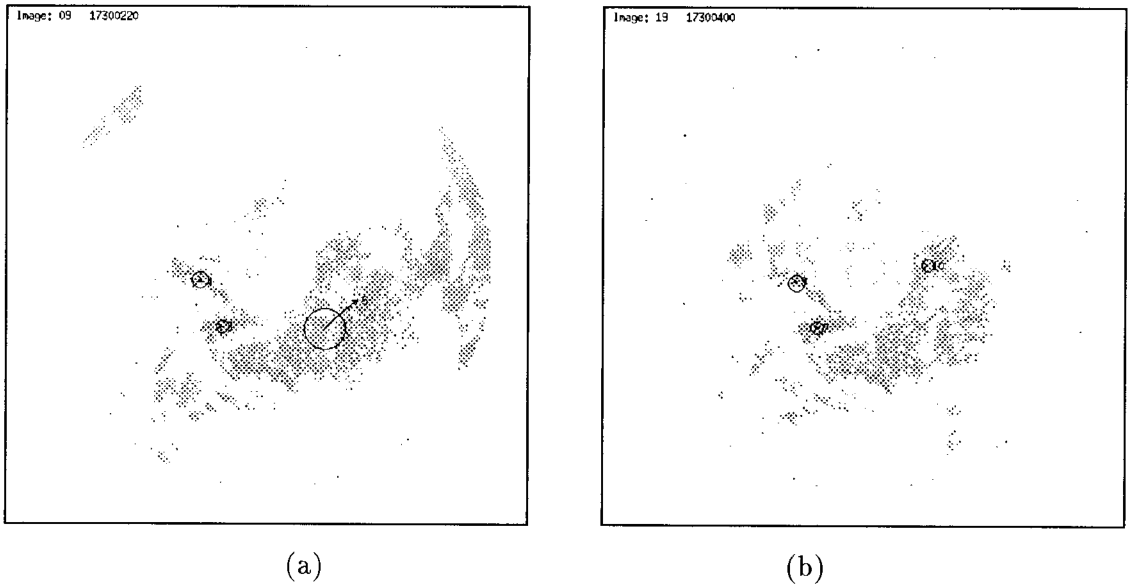


Figure 13. The storm tracks for the (a) ninth and (b) 19th images of the 17b series. T_d = 5.0 and T_{sc} = 0.75.

A. 77 Series. This sequence consists of 17 images of several storms moving left to right. A high-intensity blob first appears in the west and grows larger while moving east. In the meantime, additional smaller blobs emerge in the southwestern part of the image and eventually merge into one big blob. Figure 4(a–c) shows the complete set of storm tracks found in all the images for three different settings of T_d and T_{sc}. We see that the effect of tightening the thresholds is to eliminate some tracks but not to modify any existing tracks. We use the word ''verified'' to indicate those hypothesized storms that were found to belong to a track by our program. Figure 5 show two sets of storm data for the first and 11th images for this series. Shown are the actual split regions, the hypothesized storms, and the actual verified storms in the tracks for those images. Figure 7(a) shows the tracks computed using Zhang's algorithm. Obviously, our current tracks are superior to her tracks.

B. 18 Series. This sequence has 30 images consisting of complex storm movements. Storms are moving from the northeast to the southeast, and from west to east. In the end, both storm movements merge into one big storm moving southeast. This can be seen from the verified storm tracks shown in Figure 6(a–d), which show the verified storms tracks for storms in images 5, 12, 19, and 30. Figure 8(a) shows all the verified tracks for this sequence. T_d is 10.0 and T_{sc} is 0.6. Figure 7(b) shows the tracks computed using Krezeski et al.'s algorithm. Again, they are not as good. The tracks for this storm show the benefit of using fuzzy storm centers.

C. 08a Series. This sequence consists of 30 images of one large storm moving from the northwest to the southeast. Owing to its huge size, the distances between the center of masses of this storm in consecutive images can be huge. But because fuzzy storm centers grow with the size of storms, our algorithm is capable of tracking the storm with tight thresholds, and a long smooth track is the result. Figure 10(a,b) shows all the overall tracks and the track for the 11th image for this sequence. T_d was 10.0 and T_{sc} was 0.5.

D. 08b Series. This sequence consists of 36 images of a number of storms moving west to east. Figure 10(c,d) shows all the verified tracks for this image sequence and the tracks recovered for the ninth image of this sequence. Again, T_d was 5.0 and T_{sc} was 0.5.

E. 34 Series. This sequence consists of 55 images of a storm mass moving from southwest to northeast. The shape of the huge storm is changing so abruptly that its fuzzy storm center does not move uniformly in the same direction as the storm. However, the size of the fuzzy storm is very consistent in consecutive images and our algorithm is quite capable of producing the long, smooth, correct track it should. Figure 8(b) shows the overall tracks for this sequence, while Figure 11(a–d) shows the 12th, 15th, 23rd, and 46th images of this sequence with tracks in them. T_d was 5.0 and T_{sc} was 0.45. Again, the storm tracks for this sequence show how useful the concept of fuzzy storm centers is for tracking.

F. 17a Series. This sequence consists of 54 images. One interesting feature in this sequence is that blobs near the center of the images appear not to be moving. We do not consider these stationary blobs as storms because we suspect that these blobs are radar background noise or some atmospheric phenomena captured by the Doppler radar. Figure 9(a) show the verified tracks for this sequence, while Figure 12(a–d) shows four tracks for the fifth, 18th, 46th, and 52nd images of this sequence. T_d was 5.0 and T_{sc} was 0.75.

G. 17b Series. This sequence consists of 24 images and, like the tracks for the 17a series, there are some stationary blobs near the center of the images. This nonmotion could be used as a criterion for filtering out this noise from the image sequence.

Figure 9(b) show the verified tracks for this sequence, while Figure 13(a,b) shows two tracks for the ninth and 19th images. T_d was 5.0 and T_{sc} was 0.75.

VIII. CONCLUSIONS

By using fuzzy storm centers and relaxation labeling, we are able to obtain storm tracks that are long and smooth and which closely match human perception of a motion picture of a storm image sequence. Future plans include testing our algorithm on other types of tracking problems—for example, tracking clouds in satellite imagery or aerial views of oil/chemical spills in lakes and rivers. We also plan on extending this algorithm to 3D. Slices of Doppler image data at different radial angles (heights) are now available, and it should be possible to hypothesize and track 3D fuzzy storms, rather than 2D fuzzy storms as we do now. We believe that Doppler velocity data will be more useful in this extension.

REFERENCES

1. S. T. Barnard and W. B. Thompson. ‘‘Disparity analysis of images,’’ *IEEE Trans. Pattern Anal. Machine Intell.* **2,** 333–340 (1980).
2. I. K. Sethi and R. Jain. ‘‘Finding trajectories of feature points in a monocular image sequence,’’ *IEEE Trans. Pattern Anal. Machine Intell.* **9,** 56–73 (1987).
3. V. Salari and I. K. Sethi. ‘‘Feature point correspondence in the presence of occlusion,’’ *IEEE Trans. Pattern Anal. Machine Intell.* **12,** 87–91 (1990).
4. I. K. Sethi, Y. K. Chung, and J. H. Yoo. ‘‘Correspondence using property coherence,’’ *SPIE Appl. Artif. Intell. X Machine Vis. Robotics* **1708,** 653–662 (1992).
5. D. Krezeski, R. E. Mercer, J. L. Barron, P. Joe, and H. Zhang, ‘‘Storm tracking in Doppler radar images,’’ in *Proceedings of the International Conference on Image Processing (ICIP94)*, Vol. III, 1994, pp. 226–230.
6. J. Lai, ‘‘Tracking multiple features using relaxation,’’ *Pattern Recogn.* **26,** 1827–1837 (1993).
7. F. Leymarie and M. D. Levine. ‘‘Tracking deformable objects in the plane using an active contour model,’’ *IEEE Trans. Pattern Anal. Machine Intel.* **15,** 617–634 (1993).
8. M. Kass, A. Witkin, and D. Terzopoulos. ‘‘Snakes: Active contour models,’’ *Int. J. Comput. Vis.* **1,** 321–331 (1987).
9. K. I. Hodges. ‘‘A general method for tracking analysis and its application to meteorological data,’’ *Monthly Weather Rev.* **122,** 2573–2586 (1994).
10. H. Zhang, ‘‘Storm detection in radar images,’’ Master’s Thesis, Department of Computer Science, University of Western Ontario, 1991.
11. S. L. Horowitz and T. Pavlidis. ‘‘Picture segmentation by a tree traversal algorithm,’’ *J. ACM* **23,** 368–388 (1976).
12. T. Einfalt, T. Denoeux, and G. Jacquet. ‘‘A radar rainfall forecasting method designed for hydrological purposes,’’ *J. Hydrol.* **114,** 229–244 (1990).
13. D. Cheng, R. E. Mercer, J. L. Barron, and P. Joe, ‘‘Tracking fuzzy storm centers in Doppler radar images,’’ in *Proceedings of the International Conference on Image Processing (ICIP96)*, Vol. II, 1996, pp. 959–962.
14. D. Cheng, ‘‘Tracking fuzzy storm centers in Doppler radar images,’’ Master’s Thesis, Department of Computer Science, University of Western Ontario, 1996.
15. L. A. Zadeh. ‘‘Fuzzy sets,’’ *Inform. Control* **8,** 338–353 (1965).
16. A. Rosenfeld, ‘‘Fuzzy geometry: An overview,’’ in *Proceedings of the First IEEE Conference on Fuzzy Systems,* San Diego, 1992, pp. 113–117.
17. A. Rosenfeld. ‘‘Fuzzy plane geometry: Triangles,’’ *Pattern Recogn. Lett.* **15,** 1261–1264 (1994).
18. P. Burger and D. Gillies, *Interactive Computer Graphics* (Addison Wesley, New York), 1989.