DETECTING AND TRACKING SEVERE STORMS IN 3D DOPPLER RADAR IMAGES

Robert E. Mercer, John L. Barron
{**mercer,barron**}**@csd.uwo.ca**
Dept. of Computer Science
University of Western Ontario
London, Ontario, Canada, N6A 5B7

Paul Joe
Paul.Joe@ec.gc.ca
King City Radar Station
Meteorological Service of Canada
Toronto, Ontario, Canada, M3H 5T4

Abstract: We describe our project for detecting and tracking severe weather in Doppler radar datasets. This 3D detection and tracking algorithm is posed in a relaxation labelling framework using compatibility functions that are based on the notions of fuzzy storms and fuzzy algebra using 3D reflectivity and radial velocity data: (1) the Doppler reflectivity data is used to detect and track storms as deformable 3D objects and (2) the Doppler radial velocity data to compute 3D optical flow to predict a 3D storm's inter-frame motion.

Keywords: Severe Weather Storm Detection and Tracking, 3D Doppler Reflectivity (Precipitation Density), 3D Radial Velocity, Relaxation Labelling, 3D Fuzzy Algebra, 3D Optical Flow via Least Squares and Regularization

1 Introduction

We use both 3D Doppler reflectivity (precipitation density) [MBCB02] and radial velocity [BMCJ05] data to detect and track storms in a relaxation labelling framework using a "fuzzy" algebra [MBCB02]. We first present fuzzy algebra and fuzzy storms in Section 2, then present the calculation of 3D optical flow from radial velocities in Section 3, next present the relaxation labelling algorithm that integrates this data via compatibility factors in Section 4 and finally present some experimental results in Section 5.

2 3D Fuzzy Storms

Severe weather storms are not rigid and therefore can't be tracked using their center of masses or contour outlines. Our tracking algorithm uses the notion of "fuzzy" storms to capture the uncertainty in a storm's location.

Our storm detection program groups connected sets of precipitation reflectivity voxels above a threshold (30dBZ here) into potential storms using a floodfill algorithm. Each voxel has coordinates denoted as (x_1, x_2, x_3) with (μ_1, μ_2, μ_3) denoting the center of mass of a storm. We treat each set of 3D Doppler storm voxels as a 3D multivariate normal distribution [DHS02] and compute a 3×3 covariance matrix Σ (symmetric positive and semi-definite), where the $(i, j)^{th}$ element, σ_{ij}, is computed as:

$$\sigma_{ij} = \sqrt{\frac{\sum_{x_i, x_j \in R}(x_i - \mu_i)(x_j - \mu_j)}{|R| - 1}}, \tag{1}$$

where $|R|$ is the number of storm voxels in the ellipsoid. We compute the eigenvalues, λ_i, and the corresponding eigenvectors, $\hat{e}_i$ of Σ and use each eigenvector as one of the ellipsoid axes and $\sqrt{\lambda_i}$ as the corresponding radii. We use these ellipsoids to represent hypothesised fuzzy storms in the 3D Doppler precipitation reflectivity data.

The definitions for ellipsoidal fuzzy algebra are basically the same as for spherical fuzzy algebra [MBCB02], except where changes are needed when ellipsoids instead of spheres are used.

Definition 1. *A* **3D fuzzy point** $E\langle c, r\rangle$ *is defined as an ellipsoid with center* $c = (x, y, z)$, *three radii* $r = (r_x, r_y, r_z)$ *and three mutually orthogonal direction vectors* $e = (\hat{e}_x, \hat{e}_y, \hat{e}_z)$.

The point can be anywhere in the ellipsoid including the center. Two fuzzy points $E_1\langle c_1, r_1, e_1\rangle$ and $E_2\langle c_2, r_2, e_2\rangle$ are identical if and only if $c_1 = c_2, r_{x_1} = r_{x_2}, r_{y_1} = r_{y_2}, r_{z_1} = r_{z_2}, e_{x_1} = e_{x_2}$, $e_{y_1} = e_{y_2}$ and $e_{z_1} = e_{z_2}$.

Definition 2. *A* **fuzzy vector** *from a fuzzy point* E_1 *to another fuzzy point* E_2 *is defined as the infinite set of all displacement vectors from points in* E_1 *to points in* E_2*, see [MBCB02].*

Definition 3. *The* **fuzzy length** *or* **fuzzy magnitude** *of a fuzzy vector* $\overrightarrow{E}$ *is a set of lengths or magnitudes of all vectors in* $\overrightarrow{E}$ *and is defined as* $\|\overrightarrow{E}\|$.

Consider two fuzzy points E_1 and $E_2 \in \mathbf{E}$ (the set of all fuzzy vectors). The displacement vector from a point in E_1 to any point in E_2 can be defined as $\overrightarrow{E_1E_2}$ since a fuzzy point is only a set of the Euclidean points in three dimensions. The set of all such vectors is the fuzzy vector from E_1 to E_2, i.e.

$$\overrightarrow{E_1E_2} = \left\{ \overrightarrow{e_1e_2} \mid e_1 \in E_1 \textit{ and } e_2 \in E_2 \right\}. \tag{2}$$

with fuzzy magnitude:

$$\|\overrightarrow{E_1E_2}\| = \left\{ \|\overrightarrow{e_1e_2}\| \mid \overrightarrow{e_1e_2} \in \overrightarrow{E_1E_2} \right\}. \tag{3}$$

We can express d_{min} and d_{max} given by the variables c_1, r_1, e_1 and c_2, r_2, e_2 as:

$$d_{min} = \min\left\{ \|\overrightarrow{e_1e_2}\| \mid \overrightarrow{e_1e_2} \in \overrightarrow{E_1E_2} \right\} \text{ and} \tag{4}$$

$$d_{max} = \max\left\{ \|\overrightarrow{e_1e_2}\| \mid \overrightarrow{e_1e_2} \in \overrightarrow{E_1E_2} \right\}. \tag{5}$$

A fuzzy magnitude is then the interval $[d_{min}, d_{max}]$.

Definition 4. *The* **fuzzy angle** *subtended by a non-zero fuzzy vector* $\overrightarrow{Q}$ *relative to another non-zero fuzzy vector* $\overrightarrow{P}$ *is defined as the set of angles subtended by any displacement vector* $\overrightarrow{q}$ *in* $\overrightarrow{Q}$ *relative to another displacement vector* $\overrightarrow{p}$ *in* $\overrightarrow{P}$ *having a touching head and tail, respectively. The set can be denoted as* $\langle \overrightarrow{P}, \overrightarrow{Q} \rangle_\theta$.

Consider the three fuzzy points: $E_1 = \langle c_1, r_1, e_1\rangle, E_2 = \langle c_2, r_2, e_2\rangle$ and $E_3 = \langle c_3, r_3, e_3\rangle$. We can pick any point e_1 in E_1, e_2 in E_2 and e_3 in E_3 to from a pair of displacement vectors $\overrightarrow{e_1e_2}$ and $\overrightarrow{e_2e_3}$. The angle between these two vectors can be calculated by the dot product of the two vectors:

$$\cos\theta = \frac{\overrightarrow{e_1e_2} \cdot \overrightarrow{e_2e_3}}{\|\overrightarrow{e_1e_2}\|_2 \, \|\overrightarrow{e_2e_3}\|_2}. \tag{6}$$

We can define the minimum and maximum angles as: $\theta_{min} = \min\langle \overrightarrow{E_1E_2}, \overrightarrow{E_2E_3} \rangle_\theta$ and $\theta_{max} = \max\langle \overrightarrow{E_1E_2}, \overrightarrow{E_2E_3} \rangle_\theta$. The fuzzy angle is then the interval $[\theta_{min}, \theta_{max}]$.

We currently compute the fuzzy distances and angles by brute force; we just enumerate all distances and angles between the voxels in the ellipsoids.

3 3D Optical Flow

Our approach for computing 3D Least Squares and 3D Regularized optic flow follow the 2D optical flow methods proposed by Lucas and Kanade and Horn and Schunck [BFB94].

We use a 3D Lucas and Kanade like least squares calculation by using a radial velocity constraint equation $\vec{V} \cdot \hat{r} = V_r$ (the radial velocity is the projected 3D velocity $\vec{V} = (U, V, W)$ in the radial direction at each voxel) and assuming constant local velocity, to obtain a $n \times n \times n$ linear system of equations (at each voxel $\hat{r}$ and V_r are usually different but $\vec{V}$ is always the same) to compute the least square velocity, $\vec{V}_{ls}$ at each voxel [BMCJ05].

We then use a 3H Horn and Schunck like regularization to obtain a smooth full velocity field that is close to the true full velocity and roughly satisfies the radial motion constraint at each voxel and is similar to the least squares $\vec{V}_{ls}$ at that voxel. The regularization term is:

$$\int\int\int \underbrace{(\vec{V} \cdot \hat{r} - V_r)^2}_{\textit{Radial Velocity Constraint}} +$$

$$\alpha^2 \underbrace{(U_X^2 + U_Y^2 + U_Z^2 + V_X^2 + V_Y^2 + V_Z^2 + W_X^2 + W_Y^2 + W_Z^2)}_{\textit{Smoothness Constraint}} +$$

$$\underbrace{\beta^2((U - U_{ls})^2 + (V - V_{ls})^2 + (W - W_{ls})^2)}_{\textit{Least Squares Velocity Consistency Constraint}} \partial X \partial Y \partial Z, \qquad (7)$$

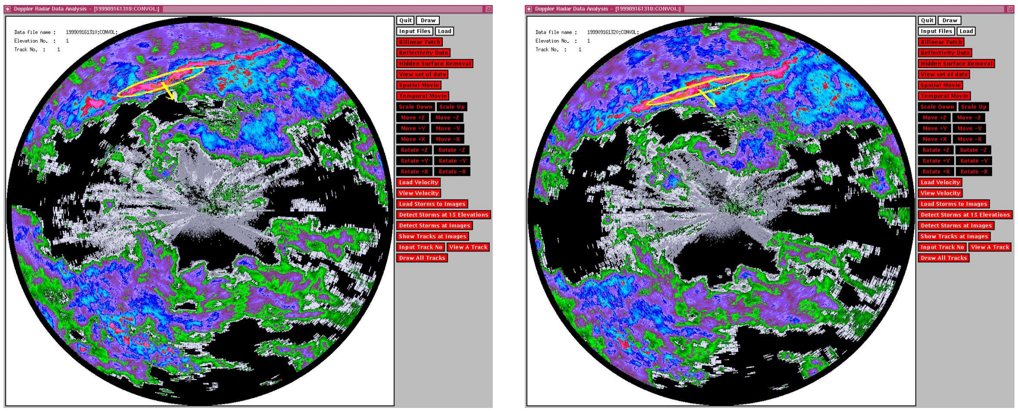
where $U_X, U_Y, U_Z, V_Z, V_Y, V_Z, W_X, W_Y$ and W_Z are the 1^{st} order velocity derivatives along the X, Y and Z dimensions. We used the Gauss Seidel method on the Euler-Lagrange equations of Equation (7) to minimal this functional [BMCJ05].

4 3D Tracking via Relaxation Labelling

Fuzzy storms are then tracked over time using an incremental relaxation labelling algorithm with compatibility factors that use fuzzy storm quantities and 3D optical flow. We used a modified version of Barnard and Thompson's relaxation labelling algorithm [BT80], that uses spatio-temporal smoothness constraints. We added *property coherence* to the algorithm which allows multiple properties of a storm (and not just disparities) to be tracked over times [MBCB02]. The storm properties used are: **size, magnitude displacement, angle, orientation** and **velocity**. Given hypothesized storms along a potential storm track we defined functions (using our fuzzy algebra) to specify how similar a storm's size, displacement magnitude, displacement angles (in 3 adjacent frames), ellipsoid orientation and center of mass velocity are to other storms. The total compatibility between two adjacent potential storms is computed using a weighted sum of these factors. These weight values are determined empirically. Two adjacent disparities are connected together if their compatibility value is greater than a threshold. When all qualified adjacent disparities have been linked together, the certainty of each disparity is refined iteratively by a relaxation labelling algorithm [BT80] that uses both supporting and contradictory evidence provided by the overall compatibility among its adjacent disparities [MBCB02].

5 Experimental Results

Each NEXRADI dataset consists of 15 elevations of precipitation density (reflectivity) and radial velocity of moving precipitation reflectivity data. At each elevation the data consists of 360 rays of reflectivity/radial velocity data (1 ray for each degree of a circle) and each ray consists of 600 individual reflectivity and radial velocity values. We present results for detecting and tracking storms in the 3D Doppler reflectivity data that was collected at the Kurnell Radar Station in Australia at intervals of 10 minutes on 1999 September 16. The name of each image file gives the date and time of the image. Figure 1 shows an oblong storm, represented as ellipsoidal fuzzy storms tracked in 4 images from this dataset. The vectors are the optical flow vectors at the storm's center of mass. We show that the predicted storm (via the center of mass velocity) and the actual storm in the next image overlapped by 90% or more [BMCJ05].



199909161310, elevation 1 199909161320, elevation 1

Figure 1: Velocity vectors (in yellow) at the center of mass of the second storm in 2 adjacent reflectivity images.

Figure 1 shows that the center of mass velocity points the direction of the storm's displacement between 2 adjacent storm. Similar tracking results show that the other properties (even without velocity) produce correct tracking results for all the data we have [MBCB02].

6 Conclusions and Future Work

We have shown an effective tracking algorithm that uses relaxation labelling to track a number of storm properties, including size, length, angle, orientation and velocity displacement. Current work includes tracking storms among overlapping Doppler radars, integrating Windprofiler and overlapping Doppler data together and detecting hook echoes (maybe indicating tornadoes) in severe weather events.

References

[BFB94] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *Int. Journal of Computer Vision*, 12:43–77, 1994.

[BMCJ05] J.L. Barron, R.E. Mercer, X. Chen, and P. Joe. 3d velocity fields from 3d doppler radial velocity. *Intl. Journal of Imaging Systems and Technology*, 15(3):189–198, 2005.

[BT80] S. T. Barnard and W. B. Thompson. "Disparity analysis of images". *IEEE Trans. Patterm Analysis and Machine Analysis*, 2(4):333–340, 1980.

[DHS02] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley and Sons, 2nd edition, 2002.

[MBCB02] R.E. Mercer, J.L. Barron, D. Cheng, and A. Bruen. Fuzzy points: Algebra and application. *Pattern Recognition (PR2002)*, 35(5):1153–1166, May 2002.