Tracking Weather Storms Using 3D Doppler Radial Velocity Information

X. Tang[1], J.L. Barron[1], R.E. Mercer[1], and P. Joe[2]

[1] Department of Computer Science
University of Western Ontario
London, Ontario, Canada, N6A 5B7
ngai@csd.uwo.ca,barron@csd.uwo.ca
[2] King City Radar Station
Atmospheric Environmental Services
4095 Dufferin St., Toronto, Ontario, M3H 5T4
Paul.Joe@ec.gc.ca

Abstract. 3D Doppler Radar is a key forecasting tool for severe weather storms and automatic tracking of these deformable objects is of great interest to meteorologists. Currently, we use the notion of a 3D ellipsoid to represent the uncertainty of a storm's 3D center in the Doppler precipitation data and have described elsewhere an algebra for its use in storm tracking. In this paper, we describe how we use 3D velocity data computed by an iterative 3D least squares/regularization algorithm to construct a velocity compatibility function that uses a prediction/verification calculation to rate the "goodness" of potential storm matches. We also use the hypothesis that the orientation of matched storms should change smoothly to obtain improved matching.

Keywords: 3D Optical Flow, 3D Doppler Radial Velocity, 3D Doppler Storm Tracking

1 Introduction

Recently, 3D Doppler precipitation reflectivity and radial velocity data has become available [3]. This data allows 3D tracking of severe storms, which until now has been done manually, to be performed nearly automatically. Fast and accurate storm tracking could significantly reduce the amount of damage to persons and property caused by these storms. Figure 1 shows the 3D structure of such data. There are 15 elevations of precipitation reflectivity (usually rain) and radial velocity (the velocity radially towards or away from the radar) data acquired on conic surfaces with various azimuth angles. We use the radial velocity vectors to compute 3D full velocity in a 3D least squares/regularization framework [1, 2] and show how these velocities can be used in a prediction/verification framework to design a new compatibility function, C_v, that is used to provide positive/negative evidence for a potential disparity in a relaxation labelling tracking algorithm [5, 6].

J. Bigun and T. Gustavsson (Eds.): SCIA 2003, LNCS 2749, pp. 1038–1043, 2003.

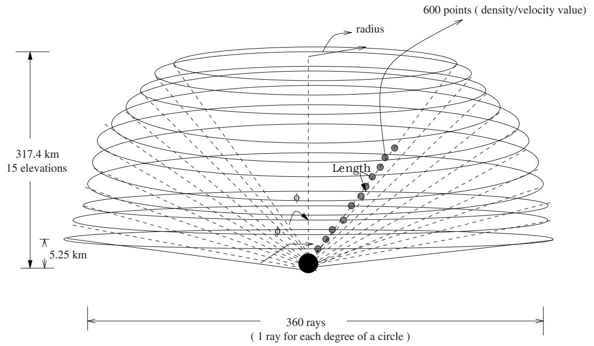


Fig. 1. The structure of 3D Doppler radar data. The length of a cell in each dataset is 1km and there are 15 elevations ranging from a minimum angle, ϕ_{min} (cone angle), of 58° to a maximum angle, ϕ_{max},] of 89.5°. The height of the rays range from a minimum of 5.25km to a maximum of 317.4km. The cone radii range from 508.84km to 599.97km.

2 3D Optical Flow From 3D Doppler Radial Velocity Fields

Chen et al. [1] presented a least squares 3D velocity calculation using Doppler radial velocities. The 3D motion constraint equation can be written as:

$$Ur_X + Vr_Y + Wr_Z = V_r \tag{1}$$

where $\boldsymbol{V} = (U, V, W)$ is the 3D velocity, $\hat{r} = (r_X, r_Y, r_Z)$ is the unit radial velocity direction (the direction of the velocity along a ray to or from the radar) and V_r is the radial velocity magnitude. This is one equation in 3 unknowns $\boldsymbol{V} = (U, V, W)$. If we assume a constant 3D velocity in an $N = n \times n \times n$ neighbourhood we can set up a linear system of equations:

$$\begin{bmatrix} r_{X_1} & r_{Y_1} & r_{Z_1} \\ r_{X_2} & r_{Y_2} & r_{Z_2} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ r_{X_N} & r_{Y_N} & r_{Z_N} \end{bmatrix} \begin{bmatrix} U \\ V \\ W \end{bmatrix} = \begin{bmatrix} V_{r1} \\ V_{r2} \\ \cdots \\ \cdots \\ V_{rN} \end{bmatrix}, \tag{2}$$

which can be written as $A_{ls\,N\times 3}\boldsymbol{V}_{3\times 1} = B_{N\times 1}$, where A_{ls} has entries r_{X_i}, r_{Y_i} and r_{Z_i} in the i^{th} row and B has entry V_{r_i} in the i^{th} row. We can solve for $\boldsymbol{V}$ in the least squares sense as $A_{ls}{}^T_{N\times 3}A_{ls\,N\times 3}\boldsymbol{V}_{3\times 1} = A_{ls}{}^T_{N\times 3}B_{N\times 1}$, where $A_{ls}{}^T A_{ls}$ is a 3×3 symmetric real matrix (all eigenvalues are real and positive). If the smallest eigenvalue $\lambda_1 \geq \tau$ (a threshold), we assume we have a reliable velocity estimate. 3D line and plane normal velocities can also be computed [4], however they seem not to have a use for 3D storm tracking. Least squares velocities have proved to be prone to noise, not least because the numeric data is collected as character data.

Since the standard Horn and Schunck-like approach to compute regularized 3D flow, $\boldsymbol{V} = (U, V, W)$, from 3D radial velocities fails, as radial velocity is already varying smoothly everywhere and satisfies the motion constraint equation trivially, we force the regularization to give a smooth non-trivial full velocity field close to the true full velocity by using the computed least squares flow as a third consistency constraint in the minimization:

$$\begin{aligned} \iiint \quad & \underbrace{(\boldsymbol{V} \cdot \hat{r} - V_r)^2}_{Motion\ Constraint\ Equation} + \\ & \alpha^2 \underbrace{\left(U_X^2 + U_Y^2 + U_Z^2 + V_X^2 + V_Y^2 + V_Z^2 + W_X^2 + W_Y^2 + W_Z^2\right)}_{Smoothness\ Constraint} + \\ & \alpha^2 \underbrace{\left(U_X^2 + U_Y^2 + U_Z^2 + V_X^2 + V_Y^2 + V_Z^2 + W_X^2 + W_Y^2 + W_Z^2\right)}_{Smoothness\ Constraint} + \\ & \beta^2 \underbrace{\left((U - U_{ls})^2 + (V - V_{ls})^2 + (W - W_{ls})^2\right)}_{Least\ Squares\ Velocity\ Consistency\ Constraint} \partial X \partial Y \partial Z, \end{aligned} \tag{3}$$

where $\boldsymbol{V}_{ls} = (U_{ls}, V_{ls}, W_{ls})$ is computed least squares 3D velocity. The idea here is to compute a smooth regularized velocity compatible with the local least squares velocities. α and β are Lagrange multipliers that weight the constraint's importance. We use values $\alpha = 5.0$ and $\beta = 1.0$ for the experimental results reported here. The Gauss Seidel iterative equations can be written [3] as:

$$\begin{bmatrix} U^{n+1} \\ V^{n+1} \\ W^{n+1} \end{bmatrix} = \begin{bmatrix} (r_1^2 + \alpha^2 + \beta^2) & (r_1 r_2) & (r_1 r_3) \\ (r_1 r_2) & (r_2^2 + \alpha^2 + \beta^2) & (r_2 r_3) \\ (r_1 r_3) & (r_2 r_3) & (r_3^2 + \alpha^2 + \beta^2) \end{bmatrix}^{-1} \begin{bmatrix} (\alpha^2 \bar{U}^n + \beta^2 U_{ls} + V_r r_1) \\ (\alpha^2 \bar{V}^n + \beta^2 V_{ls} + V_r r_2) \\ (\alpha^2 \bar{W}^n + \beta^2 W_{ls} + V_r r_3) \end{bmatrix}. \tag{4}$$

3D regularized velocity fields are computed for each Doppler dataset and the velocity nearest a storm's center of mass is used to compute that storm's displacement.

3 Velocity Compatibility Function

Given a storm's velocity, we can predict where a storm should move in the next dataset. Figure 2 shows how the predicted and actual displacements of a storm might overlap. The **velocity compatibility** function C_v is:

$$C_v\left(\overrightarrow{SC_j SC_{j+1}}, \overrightarrow{SC_{j+1} SC_{j+2}}\right) = \frac{f_v(\overrightarrow{SC_j, SC_{j+1}}) + f_v(\overrightarrow{SC_{j+1}, SC_{j+2}})}{2}, \tag{5}$$

Suppose $SC_j = (C_{xj}, C_{yj}, C_{zj})$ is the center of the j^{th} fuzzy storm from one Doppler radar dataset and $SC_{j+1} = (C_{x(j+1)}, C_{y(j+1)}, C_{z(j+1)})$ is the center of

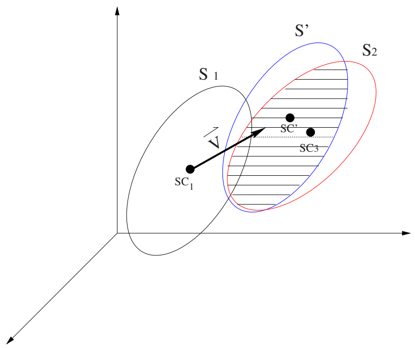


Fig. 2. SC_1 is the center of storm S_1. SC_2 is the same storm in next image. $\overrightarrow{V}$ is the full velocity of SC_1. If the internal time between two adjacent images in this image sequence is δt, we calculate $SC' = SC_1 + \overrightarrow{V}\delta t$. Then we can get the intersection volume between SC' and SC_2. We can use this volume to judge if two adjacent disparities should be connected together in a track.

the $(j+1)^{th}$ fuzzy storm from the second Doppler radar data collected after time δt and $\boldsymbol{V}_j = (V_x, V_y, V_z)$ is the full velocity of SC_j. We can calculate SC'_{j+1} as $SC_j + \boldsymbol{V}_j\delta t$ and compare it to SC_{j+1} as a test of the storm track goodness. A **velocity intersection** function $f_v(SC_j, SC_{j+1})$ can be defined as:

$$f_v(\overrightarrow{SC_j, SC_{j+1}}) = \frac{Intersect\_Volume(SC'_j, SC_{j+1})}{Min\_Volume(SC'_j, SC_{j+1})}. \tag{6}$$

3.1 Calculating the Intersection Volume of Two Ellipsoids

To calculate the intersection volume of two fuzzy storm ellipsoids (the predicted ellipsoid and the actual fuzzy storm ellipsoid), firstly, we need to obtain the predicted fuzzy storm ellipsoid center $SC'(C'_x, C'_y, C'_z)$ and radii r'_x, r'_y and r'_z as:

$$C'_x = C_{x1} + U \times t \times 0.06 \tag{7}$$
$$C'_y = C_{y1} + V \times t \times 0.06 \tag{8}$$
$$C'_z = C_{z1} + W \times t \times 0.06, \tag{9}$$

where (C_{x1}, C_{y1}, C_{z1}) is the center of the initial fuzzy storm and (C'_x, C'_y, C'_z) is the center of the predicted fuzzy storm, both after time interval t (in minutes); the 0.06 factor comes from the conversion of time into seconds from minutes and velocity (displacement) into kilometers from meters. It should be noted that an assumption that the size and shape of the hypothesized fuzzy storm remains constant within the time interval t is made in the computation of the predicted future positions of fuzzy storms (in general this is only approximately true).

Second, we use the equation of an ellipsoid to determine if the points in the predicted fuzzy storm are inside the actual fuzzy storm (or vice versa). So if

(X, Y, Z) is the center of a voxel that satisfies:

$$\frac{(X - C_x)^2}{r_x^2} + \frac{(Y - C_y)^2}{r_y^2} + \frac{(Z - C_z)^2}{r_z^2} \leq 1 \qquad (10)$$

for both the actual and predicted ellipsoids we count it as a part of intersection. A closed-form intersection would be desirable but is an open area of research (for example, in 3D video game playing software[3]). After checking if every voxel is in an ellipsoid, we can use the count of voxels in both actual and predicted storms as an estimate of the volume of the ellipsoid intersection.

4 Experimental Results

Table 1 shows that the velocity intersection values, f_v, are higher for fuzzy storms represented as ellipsoids than as spheres [3, 5, 6]. Figure 3 shows the 3D velocity values computed for the ellipsoid center of storm 2. This is the same dataset showing a large oblong storm moving from northeast to southwest [3]. The velocities are shown as white (yellow in colour) vectors and can be seen to point in the direction of the storm's displacement.

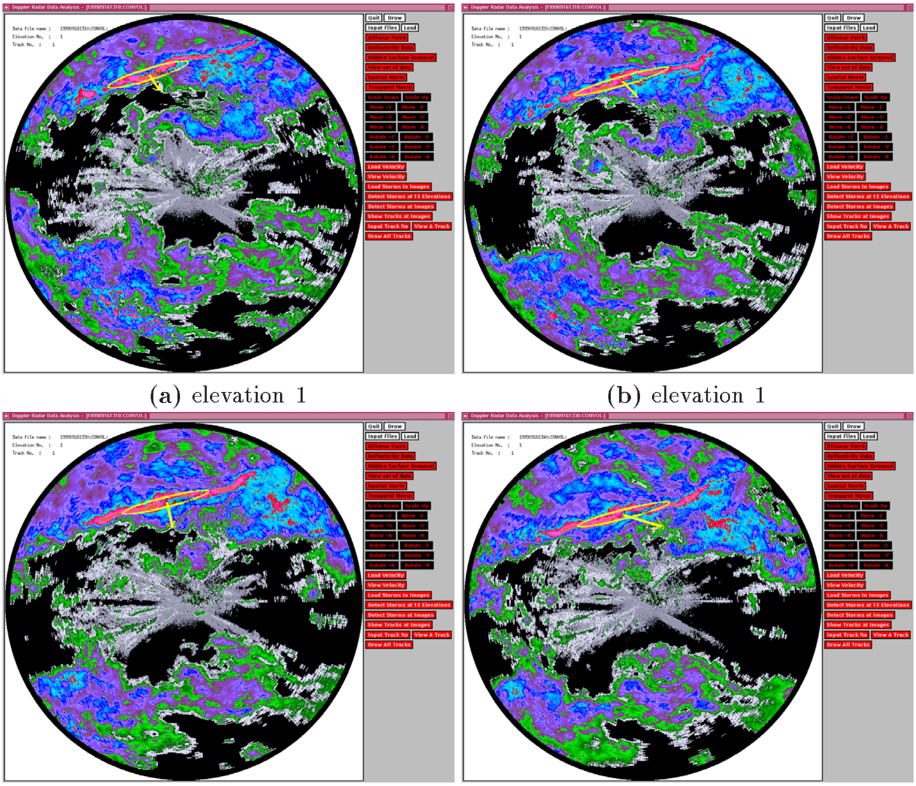
$Image_1$-$Image_2$	*Sphere*	*Ellipsoid*
199909161310-1320	57.42	87.62
199909161320-1330	65.08	90.17
199909161330-1340	67.90	95.00

Table 1. Velocity Intersection Values, f_v, of the Predicted and Actual Fuzzy Spherical Storms and Fuzzy Ellipsoidal Storms. The first column gives image sequence numbers, *Sphere* is the fuzzy storm intersection volume percent using spheres and *Ellipsoid* is the fuzzy storm intersection volume percent using ellipsoid.

5 Conclusions and Future Work

We can see that 3D optical flow is a good predictor of storm displacement. Also, ellipsoids better modelled the 3D shape of Doppler storms than spheres did. Our current research includes finding a closed form (or efficient) solution to the ellipsoid intersection problem. Future work includes integrating data from a wind profile radar with Doppler data, testing our velocity calculation on synthetic data generated by a physics-based (meteorological) storm model and tracking mesostorms, which are parts of a larger storm that move within the larger storm.
Acknowledgements The authors gratefully acknowledge support from NSERC (Natural Sciences and Engineering Research Council of Canada) operating grants and an AES (Atmospheric Environmental Services) subvention.

[3] www.magic-software.com



(a) elevation 1 (b) elevation 1

(c) elevation 1 (d) elevation 1

Fig. 3. Velocity at the center of mass of the second storm in the reflectivity images: (a) 199909161310, (b) 199909161320, (c) 199909161330 and (d) 199909161340.

References

1. X. Chen, J.L. Barron, R.E. Mercer, and P. Joe. 3d least squares velocity from 3d doppler radial velocity. In *Vision Interface (VI2001)*, pages 56–63, June 2001.
2. X. Chen, J.L. Barron, R.E. Mercer, and P. Joe. 3d regularized velocity from 3d doppler radial velocity. In *IEEE Intl. Conf. on Image Processing (ICIP2001)*, volume 3, pages 664–667, 2001.
3. W. Qiu, R.E. Mercer, and J.L. Barron. 3d storm tracking in 3d doppler precipitation reflectivity datasets. In *Irish Machine Vision and Image Processing conference (IMVIP2001)*, pages pp79–86, 2001.
4. H. Spies, B. Jähne, and J. L. Barron. Range flow estimation. *Computer Vision Image Understanding (CVIU2002)*, 85(3):209–231, March 2002.
5. X. Tang. Tracking 3d doppler weather storms using fuzzy ellipsoids and radial velocity. Master's thesis, Dept. of Computer Science, The University of Western Ontario, 2003.
6. X. Tang, R.E. Mercer, J.L. Barron, and P. Joe. Tracking 3d storms using fuzzy points represented as ellipsoids. In *submitted*, 2003.