Multiscale Storm Identification and Forecast

V Lakshmanan

April 11, 2003
Why?

Find a better way to identify storms from radar and satellite imagery.

Find a better way to estimate short-term motion.

Advect features (“image forecast”).
Storm identification methods

The operational way:

1. Use multiple thresholds

2. Count runs of values above a threshold along a radial

3. Associate adjacent radials

4. Use centroids as a proxy for the storms

5. Track the centroids using nearest-neighbor (SCIT) or linear optimization (Titan).
Short-term forecast methods

The common ways of forecasting storm locations include:

1. Find storm cores in each frame (volume scan) and then match cores across frames (e.g: WSR-88D/NSSL SCIT, NCAR Titan). Extrapolate change in position.

2. Use rectangular sub-grids and find maximum correlation within search radius. (e.g: TREC, MIT/LL Growth and Decay Tracker)

3. Use sub-grids ranging in size from entire image to small 16km x 16km grids and compute motion estimates, apply continuity criteria to derive wind-speed. (e.g: various wavelet methods, TREC method used a lot in hurricane tracking)
The storm core technique is suitable for small scale storms i.e. for short-range forecasts.

The large scale features and cross-correlation technique is suitable for longer forecasts, but with loss of detailed motion estimates. Also, assumption is that storms are of the scale of the sub-grid, not larger.

The multiscale estimation is suitable also for large scale forecasts.

All correlation techniques rely on the idea that changes between frames are small, but distinguishable.

The image template methods assume that all pixels with a grid are moving together.
Our hybrid approach

Motion estimates are made for groups of storms, but at various scales.

The motion estimate for a storm cell is the movement that minimizes the mean-absolute-error between the current frame and corresponding pixels in the previous frame.

The template is not a sub-grid of the image. It is the actual shape of the storm cell.

Rather than matching storm cells across frames, motion estimates are made via an image-analysis manner.
Major stages

1. Find storms at different scales.

2. Estimate motion at the various scales.

3. Forecast for different periods using motion at different scales.
Identifying storms

A K-Means clustering technique is used to identify components in vector fields such that the components found at different scales are nested.

You can think of this as an iterative contouring where we are slowly moving the contours by minimizing a cost function at each pixel on the boundary (local optimization).

The cost function incorporates all the elements of the vector: the actual data value, the local mean, etc.
Multiscale Segmentation

How do we get from the single-scale segmentation to a multiscale segmentation?

By steadily relaxing inter-cluster distances.
Hierarchical segmentation

The inter-cluster distances of all adjacent clusters (or regions) in the image are computed.

A threshold is set such that half the pairs fall below this threshold. If a pair of clusters differ by less than this threshold, they are merged and cluster means updated.

Continue until no two adjacent regions are closer in cluster space than the threshold. When this process is complete, we have the
next coarser scale of the segmentation.

Repeat this process until no changes happen.
Figure 1: (a) reflectivity image (b) Most detailed (c,d) coarser levels
Storm identification in radar imagery

This is the original radar reflectivity image from Fort Worth on Apr. 20, 1995.
These are the detected storms at the most detail:

These storms will be used as template for forecasting less than 30 min.
These are the detected storms at lesser detail:

These storms will be used as template for forecasting 60 - 90 min.
Storm identification in satellite imagery

This is the original satellite infrared window channel image on Oct. 9, 2001.
These are the detected storms at the most detail:
Using these storms as a template, the movement that minimizes the absolute-error between two frames is computed.

Usually, use consecutive frames, but arbitrary frames, say 5 minutes apart, can be used.
Stage 2: Motion Estimate

Motion estimation is done by moving a template of the identified cluster at the appropriate scale around in the previous image.

A matrix of mean absolute error at the different positions is obtained.
Why Mean-Absolute-Error?

It is much more common to use the mean squared error when doing correlation between frames. This is because these techniques were developed for targets (such as tanks) which do not change size/shape a lot.

The mean squared error, if minimized, generally gives us the conditional mean of the movement vector.

On the other hand, the mean absolute error (the L-1 norm, or the Minkowski-1 error) generally gives us the conditional median of the movement vector. Thus, the L-1 norm is less sensitive to outliers.

We will get outliers in motion estimation if the storm cell does not totally occupy the search space. In subgrid-based tracking, this happens all the time. In our cluster-based tracking, this happens when the cell
grows in size.
MAE Minimization

The field is minimized by weighting each pixel by how much it differs from the absolute minimum and finding the centroid.
Example

This is the east component of the motion at the most detailed scale. Note that for each storm template, we get a single motion estimate. Red is moving east. Not shown: the north-south component.
Motion Filling

So, we get motion estimates at all the places where we identified storms.

What about the rest of the image?
Gravitational Field

For every cluster, we have a motion estimate $u_i$.

At the pixel $xy$, the motion estimate $u_{xy}$ is given by

$$u_{xy} = \frac{\sum_i u_i w_{ixy}}{\sum_i w_{ixy}}$$  

(1)

where the weight of the $i^{th}$ region at the point $xy$ is given by:

$$w_{ixy} = \frac{N_i}{\|xy - c_i\|^2}$$  

(2)

where $N_i$ is the number of pixels in the $i^{th}$ region, $c_i$ its centroid and $\|$ denotes the Euclidean distance between the two points.
In practice, we threshold this so that regions too far away are not affected at all.
Temporal smoothing

We have estimated motion for a pair of frames. To enforce that the motion at a pixel is actually relatively smooth, we run a Kalman filter at every pixel.

Our Kalman filter is based on a constant acceleration/deceleration model.

This smooths out the velocity estimates.
Growth/Decay Estimate

For each storm template, we also get a growth/decay estimate. This is based on how much the average value inside the template changes.
The growth/decay field is also subject to the gravitational field and Kalman filter considerations.
Stage 3: Forecast

Using all of this information, we can forecast how the field is going to look ... The original reflectivity field:
The forecast is done:

1. **Forward:** project data forward in time to a spatial location given by the motion estimate at their current location and the elapsed time.

2. **At points not filled by the forward projection,** interpolate based on an inverse-square distance metric.
15 minute forecast

Clustering happened in the range 30-60dBZ, so that is what gets forecast:
60 minute forecast
Choice of scale of comparison

From now on, we’ll use the most detailed scale (scale 0) since that gives the most errors ...
Performance Measures: Bias

The bias of the forecast fields at different scales. The time axis is over 600 minutes (10 hours). Comparisons are with persistence and using local correlation at each pixel (without any regularization).
Performance Measures: CSI

Skill at predicting 30dBZ+ within a 5x5 spatial window.
Performance Measures: Mean Absolute Error

Mean absolute error for the 15 minute forecast compared with a persistence forecast.
CSI vs MAE

The CSI and MAE measure two different aspects of the forecast.

Why?
CSI vs MAE

The CSI and MAE measure two different aspects of the forecast.

Why?

The CSI measures effectiveness of forecasting storm location. The MAE takes into account actual reflectivity values and thus measures effectiveness of growth and decay.
How about longer forecasts?

As anticipated, skill scores decrease with time, but are still better than the alternatives studied.
15 and 30 minute forecast

15 minute forecast accuracy

30 minute forecast accuracy
Note that our CSI remains high even at 60 minutes. Our mean absolute error performance is poor, meaning that our growth-and-decay heuristics are not very good.
90 and 120 minute forecast
Satellite Water Vapor imagery

The same technique is illustrated on water vapor imagery from Feb. 28, 2003.
Satellite Water Vapor imagery

After tracking the clouds (215-235K) and making a 30min forecast:
Satellite Water Vapor imagery

After tracking the clouds (215-235K) and making a 60min forecast:
30 and 60 min forecast
Skill?

It is not clear that the technique has any skill, better than persistence on satellite data.

On radar data, the technique is much more skillful than persistence, and better than a simple correlation.

My first thoughts on why it doesn’t work so well on satellite data: the data is poor resolution both spatially (4km) and temporally (15min) so clustering is counter-productive.
One of the uses of the motion estimates here is to advect older data before combining data from multiple radars.

The motion estimates are computed on the merged grid itself, so the potential for positive feedback does exist.
Merger with time and distance weighting
Merger with advection as well
Conclusion

We can achieve better motion estimation and better image forecasts by clustering the image and estimating movement of those storm regions as a whole, at least on high-resolution radar data.

The growth-and-decay heuristic assuming a linear rate of growth does not work very well.