IDENTIFYING AND TRACKING STORMS IN SATELLITE IMAGES

V. Lakshmanan[1,2]*, R. Rabin[1,3], V. DeBrunner[2]
[1]National Severe Storms Laboratory, Norman OK
[2]University of Oklahoma, Norman OK
[3]University of Wisconsin, Madison WI

Abstract

Both the correlation methods and cell-identification methods currently used to identify and track individual storms in remotely sensed weather images have drawbacks.

A hybrid method is introduced wherein we use a segmentation-based approach to identify storms in images and arrange the identified storms heirarchially. A genetic-algorithm method of matching these segmented regions across frames is also introduced. We show how this method can easily handle splits and merges of storm cells. Since it deals with all scales, this method should, theoretically, overcome the drawbacks inherent in the cross-correlation and cell-tracking methods.

1. INTRODUCTION

Identification and tracking of storm cells in weather imagery has been an extensively studied problem (Johnson et al., 1998). There are two broad approaches – one is to use a correlation-based approach that attempts to globally minimize the movement of large storm envelopes (Wolfson et al., 1999). The second is a cell-based approach where storm cells are identified and matched across frames (Johnson et al., 1998).

The correlation-based approach is superior when tracking large storms but handles small features poorly. In many cases, the small scale features are actually discarded, by smoothing the image before tracking. The cell-based approach is subject to complications arising from storm evolution. Therefore, storms reconstructed from tracked cells will have inaccurate envelopes.

In this paper, we attempt to combine the two approaches. Each image in the sequence is segmented into regions of approximately equal temperature, the underlying idea being that contiguous areas of equal temperature belong to the same storm. These regions are then arranged in a heirarchial structure so that small cells are contained in progressively larger (and, usually, warmer) storm envelopes. We then track these heirarchial structures across frames. By placing the identified storms into these tree structures, we obtain a constraint on the movement and evolution of smaller storm cells while at the same time retaining the identification capability that small cells provide.

Although the method described in this paper has been implemented on satellite images only, we expect that the technique can be applied to weather radar images of radar reflectivity or Vertical Integrated Liquid also. The pixel values in an infrared image are related directly to the temperature of the clouds in the image. For scientific clarity, we will refer to pixel values in degrees Kelvin; the absolute temperature values used in this paper were chosen with the physics of thunderstorms in mind.

2. PREPROCESSING

The algorithm deals with the infrared window channel (11μ) of GOES satellite imager data that have been

*lakshman@nssl.noaa.gov

Figure 1: The satellite infrared image contrast enhanced after thresholding.

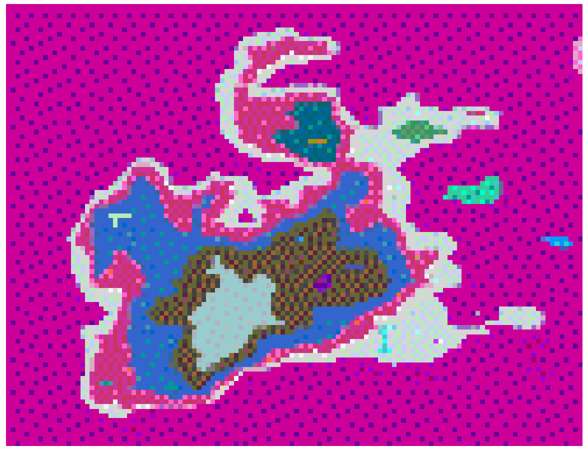

Figure 2: Features identified from the satellite infrared image of Figure 1.

mapped such that each pixel represents a constant area in latitude and longitude. The images are 1000x1000 with each pixel representing 0.02 degrees of latitude and 0.02 degrees of longitude. Images of the sequence are available at fifteen minute intervals.

a. IMAGE ENHANCEMENT

First, the image is smoothed by computing a moving 3x3 average. Then, the histogram of the image is computed and the smallest temperature threshold that will retain at least the coldest 1% of the pixels in the image is found. We do this rather than pick the absolute minimum in the image so that our statistic is more robust. The adaptive threshold is constrained to be no higher than 255K. Because we later quantize the levels in measurement space in multiples of 2K, we also constrain the threshold to be an odd number of degrees Kelvin. An approximately 100x120 cropped region of one of the frames after thresholding and contrast-enhancing is shown in Figure 1.

3. IMAGE SEGMENTATION

At this stage, we are dealing with the thresholded image. The method of segmentation is very simple. The measurement space (between the chosen threshold and 255K) is divided into 2K increments. Then, the sets of 8-connected pixels that are in the same 2K bin in the measurement space are identified. Each such set is considered a feature that is to be tracked. Extremely small (less than $5km^2$) features are discarded. Features identified from Figure 1 are shown in Figure 2.

a. REPRESENTATION OF SEGMENTS

Instead of representing each of the identified regions as a collection of pixels, each region is represented by the smallest irregular convex octagon whose sides make 0, 45, 90 ..., 315 degrees with the horizontal axis and contain the region in its entirety. This is a good representation for small convex regions (see Figure 3).

However, for regions, especially large ones that happen to have a few concave boundary sections, the representation is not very good. This representation is easy to compute (just compute the minimum and maximum values for x, y, x+y and x-y for each pixel in the segment). In addition, this representation makes it easy to compute relationships such as whether one region contains another. The quality of the convex-octagon approximation (defined as ratio of the number of pixels in the region to the area of its octagon representation) captures two attributes of a region: its shape and its orientation.

During the segmentation process, we compute several

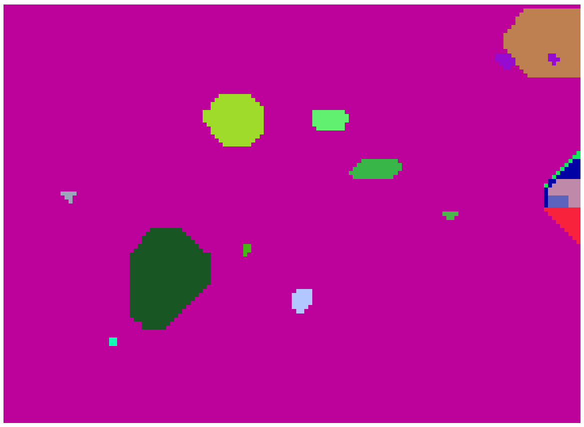

Figure 3: The regions of Figure 2 that meet certain size criteria.

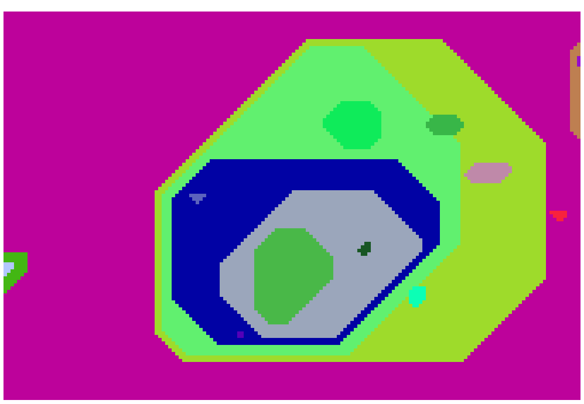

Figure 4: The representation of the regions identified in Figure 2.

attributes of each region, including its centroid, average value and quality of convex-octagon approximation. We also find out several relationships (contains, adjacent, is contained by, etc.) amongst pairs of identified regions.

An interesting exercise is to find out the retained structure in image space. One way to represent that knowledge is to regenerate an image from the internal representation of the regions. This is not perfect since we can associate only one pixel with a region; the convex-octagon representation will lead to overlap when the regions are not convex.

Some of the obvious benefits of the representation can be seen from the regenerated image shown in Figure 4. It would be easy to find such features as anamolies (warm regions inside cold ones) or to find out the relationship of features with respect to each other (for example, overshooting tops near the back edge of a thunderstorm anvil). In particular, we can use this structured information in our tracking algorithm.

4. TRACKING SEGMENTS

Following Dixon (1994); Wiklund and Granlund (1987), we approach tracking as a global optimization problem. We wish to find a one-to-one mapping from the set of segments in the previous frame to the set of segments in this frame such that a global cost function is minimized.

In Dixon (1994), tracking storms identified on radar imagery is treated as an optimal assigment problem that reduces to a linear programming problem with an optimal solution that may be obtained by the Hungarian method (Nering and Tucker, 1993). However, in this method, splits and merges are factored in after the linear programming problem has been solved. Doing so defeats the global optimization since treating two already tagged storms as having merged leaves at least one (sometimes both) the storms in the previous frame untracked. If we incorporate splits and merges, then this becomes a non-linear optimization problem.

Based on the method of Wiklund and Granlund (1987), a good, fast converging, though not optimally best solution can be found by computing the distances (defined in our case as an aggregation of several fuzzy sets that capture the closeness of the regions spatially, in size, and in value) between each pair of regions and ageing the distance leeway across frames. Each region is then initially associated with the closest object and a penalty assigned for repeated associations. Consecutively iterating over changes to the assignments that reduce the global cost function, and not reusing discarded assignments, yields a quickly convergent, though not optimal, process that is at least better than the initial assignment of “closest” regions.

Either of the above schemes should, ideally, identify splits and merges as completely new cells since the cost function which incorporates size should not allow any other cells to be identified. In practice, however, that is not the case – there are usually several cells of approximately equal size closeby The presence of these cells can throw the tracking off. In a global optimization process, especially the second where there is no way to recover from bad choices early on, mistakes cascade and result in very large errors.

a. GENETIC ALGORITHM

We can, however, improve our process if we can do two things: a) use the heirarchial structure to constrain solutions b) recover from bad choices made initially. In addition, we also realize that optimal solutions for parts of the heirarchial structure are very likely to be part of the optimal solution for the entire frame. (This is a relaxation of the topological rule that the shortest path from A to B via P involves the shortest paths from A to P and P to B, except that in this case, the optimal paths AP and PB are not independent and can not, therefore, be evaluated in isolation).

The heirarchial structure of the segmented regions in each frame can then be arranged so that the ”children” of a region immediately follow it. Then, if a part of this structure has been matched optimally with the heirarchial structure from the previous frame, it is likely that the part of the structure should be retained in the final optimal solution.

A genetic algorithm is a fast converging optimization technique that solves this exact problem (Goldberg, 1989). Our method to arrange the regions heirarchially and then to find all the cells (regardless of size) in the previous frame that are close to each cell in the current frame. Each of these cells, combinations of two or more of these cells, and a non-match are all termed possible tags for the cell. Thus, each node in the heirarchial tree built for the current frame has one of a finite number of possible tags. The optimization problem then reduces to finding the set of tags (for all the cells in the current frame) that reduces the overall cost function. As can be seen, splits are handled as combinations of cells in the current frame while merges are handled as combinations of cells in the previous frame. Both are handled in the context of global optimization itself.

The cost function rewards storms maintaining their position in the heirarchy, their temperature and their size. Thus, small features in one frame tend to be assigned to small features in another frame, and both are contained within larger features that have also been assigned to each other.

5. RESULTS

The method of identifying and tracking storms in satellite imagery described in paper was carried out on a sequence of GOES infrared images available at 15 minute intervals.

A portion of the trees generated from the images, as well as the assignment of tree nodes (features in the image) between frames, is shown in Figure 5. Each node in the diagram represents a region that has been identified. The location of that region is shown within the node and the horizontal arrow leading away from the node has the predicted location of the next node. In the first frame, there is no motion information available and therefore, the predicted location is the same as the current location. In later frames, the history of the region’s movement is used in prediction. On the top-right of each node, two numbers are marked. The first number captures the shape of the region while the second number indicates the size. One of the regions that the node contains is pointed to by the vertical arrow. The horizontal arrow leading from the node points to the node in the next frame that has been assigned to this one.

The predicted locations (of the centroid of the regions) is not a very good indicator of where the region is to be found in the next frame. However, the constraints posed by the shape, size and heirarchial structure allow the algorithm to correctly make assignments between frames.

As expected, larger features are tracked for longer periods of time. The structure of the contains-relations is retained. The errors in tracking storm cells are not so large that smaller storms get missed. The size of the regions jumps around quite a bit, suggesting significant drawbacks in the segmentation technique but the quality of convex-approximation ratio remains pretty much fixed. (Again, this is a function of shape and orientation, not of size).

Current research is geared toward examining different segmentation techniques, large scale filtering applications to better constrain the movement of smaller cells and obtaining a faster converging and/or optimal solution to the global optimization problem.

References

Dixon, M.: , 1994: *Automated Storm Identification, Tracking and Forecasting – A Radar-Based Method*. Ph.D. thesis, University of Colorado and National Center for Atmospheric Research.

Goldberg, D.: , 1989: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Inc., 432 pp.

Johnson, J., P. Mackeen, A. Witt, E. Mitchell, G. Stumpf, M. Eilts, and K. Thomas: , 1998: The storm cell identification and tracking algorithm: An enhanced WSR-88D algorithm. *Weather and Forecasting*, **13**, 263–276.

Nering, E. and A. Tucker: , 1993: *Linear Programs and Related Problems*. Academic Press, Inc., Boston, 275-318 pp.

Wiklund, J. and G. Granlund: , 1987: Tracking of multiple moving objects. *Time-Varying Image Processing and Moving Object Recognition*, V. Cappellini, ed., Elsevier Science Publishers B.V., 241–250.

Wolfson, M., B. Forman, R. Hallowell, and M. Moore: , 1999: The growth and decay storm tracker. *8th Conference on Aviation*, Amer. Meteor. Soc., Dallas, TX, 58–62.

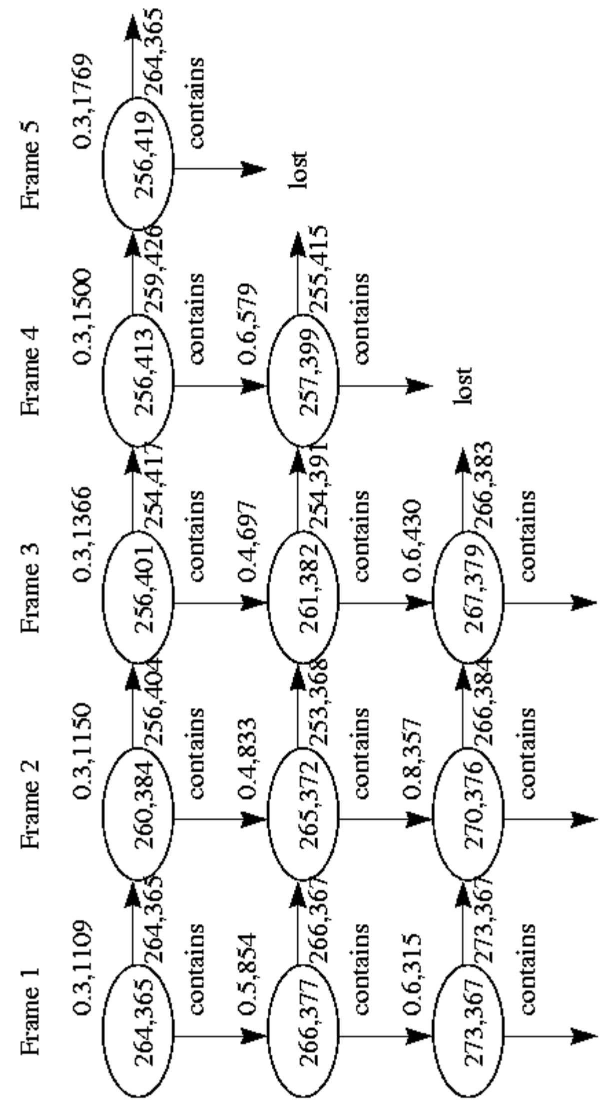

Figure 5: Parts of the trees generated for five of the frames in the sequence. The x, y pixel location of the centroid of each feature is shown inside the node. The predicted x, y pixel location of the centroid of the feature is shown under the arrow. One of the children of the node is also shown. The numbers on the top right of a node describe its quality of convex-approximation and the size of the region in pixels.