Central Archive at the University of Reading

Reading's research outputs online

A General Method for Tracking Analysis and Its Application to Meteorological Data

K. I. HODGES

Centre for Global Atmospheric Modeling and NERC Unit for Thematic Information Systems, University of Reading, Reading, England

(Manuscript received 17 May 1993, in final form 30 November 1993)

ABSTRACT

Recent interest in the validation of general circulation models (GCMs) has been devoted to objective methods. A small number of authors have used the direct synoptic identification of phenomena together with a statistical analysis to perform the objective comparison between various datasets. This paper describes a general method for performing the synoptic identification of phenomena that can be used for an objective analysis of atmospheric, or oceanographic, datasets obtained from numerical models and remote sensing. Methods usually associated with image processing have been used to segment the scene and to identify suitable feature points to represent the phenomena of interest. This is performed for each time level. A technique from dynamic scene analysis is then used to link the feature points to form trajectories. The method is fully automatic and should be applicable to a wide range of geophysical fields. An example will be shown of results obtained from this method using data obtained from a run of the Universities Global Atmospheric Modelling Project GCM.

1. Introduction

Climate prediction relies largely on parameterized general circulation models (GCMs). The models must be able to represent the current climate as accurately as possible to enable predictions about future climate to be made with confidence. Thus, means of analyzing the error between the simulated and observed climates are required.

Since the local climate at any particular place on the earth is an ensemble average of the phenomena affecting that region, the phenomena must be represented within the GCM as accurately as possible for the errors between the model climate and the current climate to be minimized; hence, the observed variability on a range of temporal and spatial scales must be accurately represented in the model. Not all scales of disturbance in the atmosphere, oceans, or land surface phenomena can be represented explicitly in a GCM, and so parameterizations are introduced to model phenomena on scales below which the GCMs can resolve.

The variability predicted by GCMs and, hence, the climatology are usually quite sensitive to these parameterizations. So, it is important to be able to measure the response of GCMs to various parameterizations and compare these with "real" data. This will indicate the parameterizations that provide the best representation of the current climate.

The traditional method of comparing data, makes use of temporal and/or spatial averages (for example, Gates 1992). However, the analysis of the transient characteristics may give more insight into the behavior of the atmosphere and the response of models to different parameterizations. An example of this type of analysis is given in Slingo et al. (1992). Their technique uses bandpass filtering of the variance to explore the spectral characteristics of data on the intradiurnal, synoptic, and intraseasonal time scales. A complementary method of analyzing the variability and performing comparisons may be achieved by using a synoptic objective analysis method. This is the approach that will be described in this paper. This will enable synoptic climatologies to be constructed and compared efficiently for different models and satellite data. The fact that the technique is fully automatic will provide greater confidence in performing comparisons between climatologies than a manual subjective approach.

It has only been in the past few years that semi- and fully automated methods have been used for the direct identification of phenomena for the objective validation of models and for comparisons with the plethora of satellite data that are available. The main application area has been in meteorology, although there is no reason why similar techniques cannot be used in oceanography. Several authors have attempted this difficult task. König et al. (1992) have attempted the objective identification of cyclones by searching for correlations between the relative minima in the 1000-mb geopotential height field and maxima in the 850-mb vorticity field for data from several GCMs. Also, Lambert (1988) used an automated method of determining cy-

clone densities to explore cyclone distributions in the Northern and Southern Hemispheres from GCM data. Both sets of these authors used the method of counting lows within a unit area in a specified time period.

A better means of exploring the variability is to track interesting phenomena as they develop and to examine their growth and decay and their spatial distribution. This approach to exploring the variability of the atmosphere is not new, with early attempts being performed manually by Streten and Troup (1973), for example. More recently, several authors have used tracking techniques to analyze the synoptic variability in the Tropics and extratropics. Akyildiz (1984) performed a tracking analysis of cyclones using European Centre for Medium-Range Weather Forecasts (ECMWF) forecast data for the 1981/82 winter for the North Atlantic and constructed statistics of cyclone speed, central pressure, geographical size, and life span. A similar study has been performed by Reed et al. (1986) using ECMWF analyses for the Tropics that identified and tracked African easterly waves. However, the identification and tracking has apparently still been performed manually in these studies. Also, the geographical regions explored and the number of phenomena followed tend to be limited.

With the advent of high-speed computers, automated methods have been developed to perform objective analyses using tracking. Indeed, cloud tracking from satellite data to determine cloud-motion winds for use in operational forecasting was one of the earliest automated applications of tracking in meteorology. A variety of methods has been used. Endlich et al. (1971) tracked brightness centers. Their method found these centers objectively using a pattern recognition technique and tracked them using a matching technique. Leese and Novak (1971) used a cross-correlation technique to determine cloud motion characteristics for operational purposes. Both of these methods had some success in determining the velocity of cloud clusters, although problems were encountered due to occlusion of one cloud layer by another. (The term *occlusion* is used in dynamic scene analysis to mean the total or partial disappearance of an object in part of the frame sequence due either to poor detection or variable lighting or because of concealment by another object).

An automated detection and tracking method for cyclones has been developed by Williamson (1981). This technique identified anomalies in the 500-mb geopotential height field of ECMWF forecasts and National Meteorological Center (NMC) analyses for the Northern Hemisphere. The tracking was performed by minimizing a function of $6n$ variables, where n is the number of detected features. However, this technique together with the techniques of Akyildiz (1984) and Reed et al. (1986) are essentially forecast validation tools rather than global climate analysis tools.

More recently, Le Treut and Kalnay (1990) used an ad hoc method of finding and tracking cyclones in pressure data from the ECMWF, while Murray and Simmonds (1991) used a sophisticated automated method that extracts low pressure centers (feature points) from interpolated GCM data using a local optimization scheme. These feature points are formed into tracks by calculating the probability of association (correspondence problem) between the feature points. Both of these studies used the pressure field to determine the occurrence and position of cyclonic activity. Their methods worked well for the extratropics where cyclonic activity can be associated with strong variations in the pressure field. They managed to track the cyclones reasonably well and to derive statistics for the cyclogenesis, cyclolysis, and distribution of cyclones.

The motivation for the development of the technique described here is to aid the analysis of data obtained from the UGAMP GCM and to assist in the validation of the model and parameterizations. It was thus decided that a general, fully automated, technique was required that would be applicable to a wide range of datasets. The method described in this paper makes use of techniques usually associated with image processing and dynamic scene analysis. Although some image processing techniques have found their way into meteorological problems before, the techniques used here are more up-to-date and allow a more general method to be developed.

2. Overview of the method

The basis of the method presented here is the identification of objects in suitable meteorological images and/or fields. These objects might be regions around extrema in a vorticity or pressure field for model data, or clouds identified by their brightness or by regions around minima in the outgoing longwave radiation (OLR) for remotely sensed data. The objects can be extracted from time sequences of the field by scene segmentation. This entails the thresholding of the data into object and "background" points or pixels and then the partitioning of the object points into distinct objects, which alone are stored for further processing to find feature points. This identification of objects means that only a small subset of the raw data needs to be stored and explored for suitable points to track.

Tracking is a feature- (token) based method of motion analysis; therefore, suitable features must be identified. This is performed by the feature detection part of the algorithm that identifies suitable points—for example, centroids or local extrema within each object. The objects do not need to have a single feature point; for complex fields where objects coalesce and break apart, multiple feature points are common. Each token thus consists of the position of the feature point and, in the application reported here, the strength measured as

either an average over an object or as a point value (local maximum or minimum). Other data could also be part of a token—for example, information on the size and shape of an object—but it is not considered in this paper. The segmentation and feature point detection part of the algorithm reads and processes each frame (data for each time step) sequentially and stores the object and feature point data for use by the tracking algorithm.

The main problem that tracking algorithms must overcome is called the correspondence problem. [See Aggarwal and Nandhakumar (1988) for a review of tracking techniques.] This is where the correspondence between features obtained in consecutive frames needs to be determined. This is a nontrivial problem, particularly as objects can appear and disappear within a sequence and be occluded by other objects. The technique described in this paper was originally developed for the monocular dynamic scene analysis of video footage. However, there is no reason why the same, or similar, techniques cannot be applied to analyze sequences of data obtained from numerical models or sequences of satellite images.

A brute force approach to the correspondence problem is combinatorially intractable; that is, considering all possible trajectories for the feature points would be computationally prohibitive. Thus, constraints are usually introduced to make the problem tractable. Early attempts were based on an assumption that moving objects remained rigid (their physical shape remained unchanged), which is clearly inappropriate for the problem here.

Several new methods have been developed in an attempt to solve this problem. The related methods of Sethi and Jain (1987), Salari and Sethi (1990), and Rangarjan and Shah (1991) use assumptions about the smoothness of the motion. This enables them to construct a cost function that is minimized to produce the smoothest possible tracks. These assumptions are essentially constraints on the expected motion; therefore, if a sudden change in motion occurs, these techniques will not detect it and the wrong correspondence may be assigned. However, they have the advantage of being able to deal with a large number of moving objects (represented by feature points). Also, for the application that will be considered here, sudden changes in motion are not expected, provided the frame sampling rate is sufficient.

The tracking technique adopted here is that of Salari and Sethi (1990). It is more flexible than the method of Rangarjan and Shah (1991) because it can cope with the appearance and disappearance of objects. Also, the method of Rangarjan and Shah (1991) depends on some initial correspondence being available in the first few frames. The determination of this initial correspondence may be a complex task, particularly if objects can appear at anytime in the frame sequence, although it is possible if the objects exist for the whole sequence, an unlikely situation for a meteorological application.

The full algorithm for object identification and tracking has been developed as a two-dimensional method. It has been coded in the ANSI C programming language (Kernighan and Ritchie 1988), making full use of recursion and the freedom to define data types.

3. Segmentation

In image processing, segmentation is the technique of thresholding an image into distinct regions (usually of similar intensities). In this paper it refers to the thresholding of data points into object and background points, connected component labeling (CCL) [the separation of object points into distinct objects by labeling (Rosenfeld and Kak 1976)], and region growing. The CCL is based on the use of the quad-tree data structure (Samet 1989), which provides an efficient method of examining data at different resolutions, although at the possible expense of extra storage. The region growing algorithm is another method by which object points are assigned to distinct objects by searching for object points that have not been previously labeled. This is required to search for parts of objects outside the region of interest for track continuity purposes and will be described later.

The segmentation identifies regions in a scene or frame by giving all points, or pixels, within that region a unique label. Point and pixel are interchangeable in this discussion, and so from now on only the term *point* is used. The methods described here assume that the scene, or field, is defined on a regular or semiregular (parallel grid lines not necessarily uniformly spaced) rectangular grid. More will be said of this later when different types of projections are discussed. The stages involved in the segmentation part of the method will be described in turn. While parts of this work are in common use in the analysis of spatial data, all will be described here for completeness.

The segmentation is initiated when the user chooses the region of interest in the frame and a threshold value. Although the whole globe can be considered, in practice a region of interest is chosen to speed up the computation. The region chosen must satisfy the area criterion: it must be $2^n \times 2^m$ grid points in dimension. A binary map is then produced of the region of interest based on the chosen threshold, so that points with intensities greater than the threshold are considered as object points, and points with values below this threshold are considered as background points. If the region of interest does not fulfill the area criterion, the binary map is extended with zeros so that the criterion is fulfilled. The thresholding is defined mathematically as

$$b_{i,j} = \begin{cases} 1, & f_{i,j} \geq T \quad i = N_x, \cdots, N_x + 2^n - 1; \\ 0, & f_{i,j} < T \quad j = N_y, \cdots, N_y + 2^m - 1, \end{cases}$$
$$(1)$$

where $b_{i,j}$ is the binary field value, $f_{i,j}$ is the actual field value, T is the value of the threshold, and N_x and N_y represent a corner node of the region of interest. More sophisticated thresholding—for instance, that based on the intensity histogram, will be considered in the future.

The binary map for the region of interest is now converted into a hierarchy of levels, called the "data hierarchy," which are linked together as a quad tree. A quad tree is a linked hierarchical data structure based on the recursive decomposition of the region of interest, and it allows the data to be examined at varying resolution. The best way to describe the data hierarchy and quad tree is by illustration. Consider a $2^3 \times 2^3$ binary map with the value 1 representing an object point, and the value 0 a background point, as shown in Fig. 1. By combining quads, as shown in (2), a hierarchy of levels is obtained (the data hierarchy), where the value -1 represents a combination of 1's at one level and 0's at lower levels, and the superscript l is the level

$$q_{i,j}^0 = b_{i,j}$$

$$\left.\begin{array}{r} Q_{i,j}^l = q_{2i-1,2j-1}^{l-1} + q_{2i-1,2j}^{l-1} + q_{2i,2j-1}^{l-1} + q_{2i,2j}^{l-1} \\[4pt] \forall i,j : 1 \leq i \leq 2^{n-l} \quad \text{and} \\[4pt] 1 \leq j \leq 2^{m-l}; \quad l > 0 \\[4pt] \text{if} \quad q_{2i-1,2j-1}^{l-1}, q_{2i-1,2j}^{l-1}, q_{2i,2j-1}^{l-1}, q_{2i,2j}^{l-1} \geq 0 \\[4pt] q_{i,j}^l = \begin{cases} 1, & \text{if } Q_{i,j}^l = 4 \\ 0, & \text{if } Q_{i,j}^l = 0 \\ -1, & \text{otherwise} \end{cases} \\[4pt] \text{else} \quad q_{i,j}^l = -1 \end{array}\right\} . \quad (2)$$

The hierarchy of levels for the example are as shown in Fig. 2. Note that for a square region $2^n \times 2^n$, and then $l_{max} = n$. For a rectangular region, this method needs to be modified. For this situation, eventually a level is reached that is a single row or column. Every time this occurs, a row or column of 0's is incorporated

```
0 0 0 0 0 1 1 1 0
0 0 1 0 0 0 1 1 1
1 1 1 1 0 0 1 1
1 1 1 0 0 0 0 0
0 0 0 0 1 1 1 1
0 0 0 0 1 1 1 1
0 0 0 0 1 1 1 1
0 0 0 0 1 1 1 1
```

level 0

FIG. 1. An example binary map to be used for an illustration of the construction of a data hierarchy and quad tree.

```
0 -1 -1 -1
1 -1  0 -1      -1 -1    -1
0  0  1  1       0  1
0  0  1  1
```

level 1　　　　level 2　　level 3

FIG. 2. Data levels of data hierarchy formed by combining quads for the example binary map in Fig. 1.

until a level is reached that has only a single value. For this more general situation of a $2^n \times 2^m$ region, $l_{max} = \max(n, m)$.

The levels of the hierarchy are linked to form a quad-tree data structure by noticing that for nodes of value 1 or 0 all their offspring at lower levels have the same value. Hence, once one of these nodes is labeled, all the offspring necessarily have the same label. The quad tree is constructed by linking the nodes and terminating each branch on the highest node with value 1 or 0. The quad-tree representation for the example looks like that shown in Fig. 3, with the tree rooted at level 3.

For the quad-tree representation, the node ordering is important and must be carried out consistently. In fact, if the quad tree is ordered the same way as the block decomposition, the numerical labels indicate the order in which the nodes are visited traversing down the tree. In the example shown, the tree nodes are ordered from left to right (northwest, northeast, southwest, southeast): this is the scanning order. Other scanning orders can be considered but will depend on the application of the quad tree. The position of a node in the tree is determined uniquely by the path from the root to the node—for example, (northwest, southwest) for node 6 of Fig. 3. This is often encoded into a locational code. However, for a pointer-based quad tree this is taken care of by using pointers to node records.

Since the quad-tree representation is required for the CCL method, it must be structured in a way that is consistent with the CCL method. The usual and most efficient representation is as a linked tree structure using pointers (see Knuth 1973). This is achieved by using a data record of type *image node*. These data records are stored in raster order for each level, so that each point at each level of the data hierarchy is represented by an image node data record in memory. The image node record will contain a field with the node type identifier (1, 0, or -1), a pointer to the parent node, pointers to the four offspring nodes, and a pointer to an equivalence class, which will be discussed shortly. For the root node of the quad tree, the pointer to parent actually points to "null" (points to the zero memory address). For 1 and 0 (zero) nodes, the pointers to offspring point to "null"; these are called the leaf nodes. Thus, the quad tree representation of the data hierarchy as used here is defined by the links set using the pointers to parent and offspring. If large im-
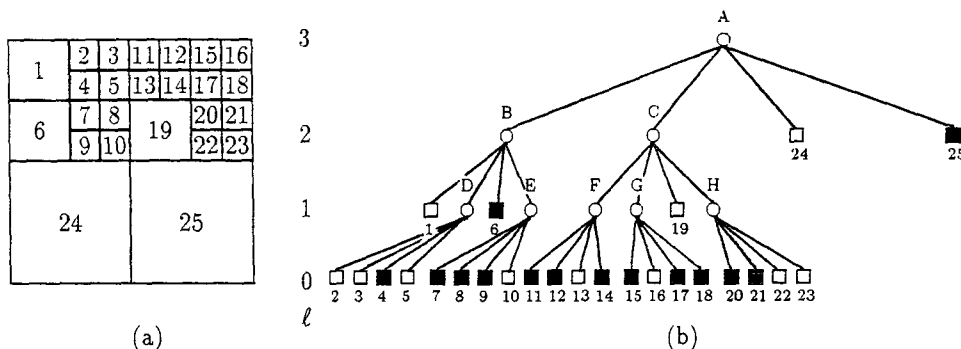
FIG. 3. (a) The maximal block representation for the example image. The larger the block, the higher the corresponding node that resides in the quad-tree data structure, and (b) the quad-tree representation of the image. Level 3 is the root, and ■ = 1, □ = 0, and O = −1.

ages or very high density grids are used, then this method of quad-tree construction may need to be revised and more efficient methods of converting a raster image to a quad tree may need to be implemented.

The equivalence class record is used to identify equivalences between tree node labels. The equivalence class is stored as a record with two fields: the first field is the component label for the class, not to be confused with the node-type identifier, and the second is a pointer that can point to the equivalence class record of another node. In other words, it defines an equivalence relationship between tree node labels. The equivalence class representation is another form of tree (see Knuth 1973). Processing the equivalences partitions a set of elements into distinct sets, where the elements are the labels of quad-tree nodes of type 1 and the sets represent the objects.

The CCL algorithm performs the operations of labeling adjacent nodes and assigning equivalences, followed by propagating the equivalences through the quad tree so that all points belonging to a distinct object are uniquely labeled. The process starts at the root node and traverses down the tree searching for object nodes. For each unlabeled object node encountered, a unique component label is assigned. Then all its adjacent object nodes are found in the direction specified by a search template. This is performed recursively for each connected object node that is found. The adjacency search is performed according to the search templates

FIG. 4. Search templates used by the CCL algorithm for (a) four-connected CCL and (b) eight-connected CCL.

shown in Fig. 4 using the neighbor finding techniques given by Samet (1989). These are quite involved, and so only a brief description is given here. The basis of the neighbor finding techniques is to ascend the tree from the current node to the nearest common ancestor (NCA). This is the first node at a higher level in the tree reached by an offspring not related to the direction of search. For example, in Fig. 3, to find a western neighbor of tree node 7, the tree is ascended to the NCA, node B (path northwest, southeast). The tree is then descended along the path, which is the mirror of the ascent path, so, using the example from node B, the tree is descended to node 6 (path southwest). Further descent is impossible because node 6 is a leaf node. Any adjacent object nodes found are given the same unique label. If an adjacent node is found that has already been labeled, then the equivalence is noted using the pointer in the equivalence class record. Other operations are performed to compress the path to the head of the class. After completion of the first tree traversal, the quad tree is traversed a second time, updating the labels based on the generated equivalences.

For a 2D domain, two types of connectivity need to be considered: four connected and eight connected. For the four-connected case, adjacency is determined along the four sides of the image element (cf. Fig. 3a). For the eight-connected case, adjacency is determined along the four sides and the four vertices of the image element. The CCL algorithm achieves this by using the search templates shown in Fig. 4. If a different scanning order is used, then these templates will also be different. So that an eight-connected search can be performed, the CCL algorithm of Samet (1989) has been modified to include his algorithm for a vertex adjacency search.

Finally, the tree is traversed a third time, starting at the root node. For each component label that is encountered, a data record is created of type *object*. This will contain three fields, a pointer to the object point
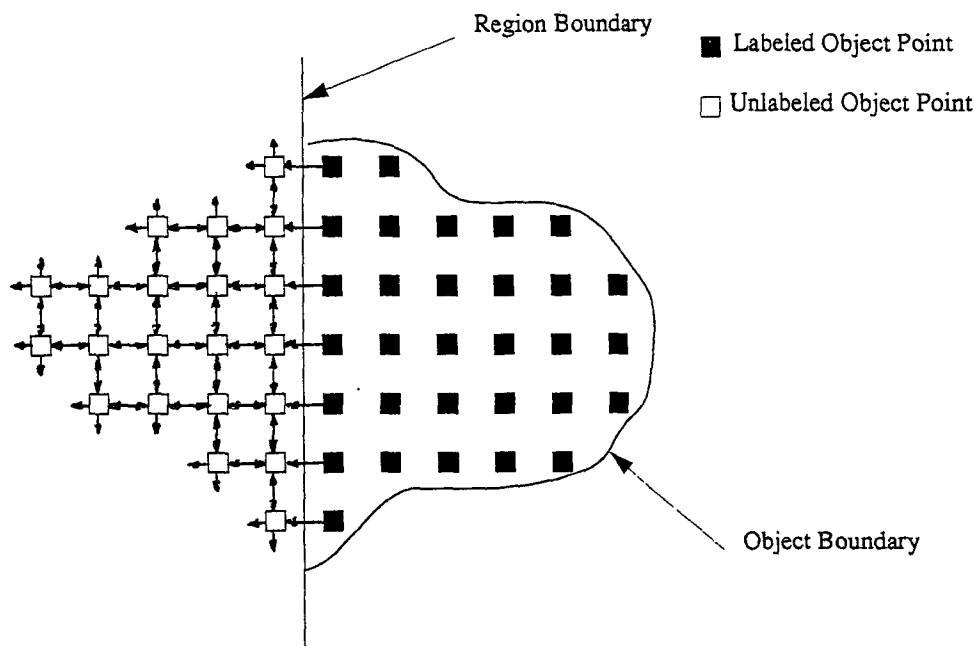
FIG. 5. Schematic of the boundary search procedure for a four-connectivity search. The unlabeled points beyond the region boundary but that belong to the object are labeled as they are encountered along the search directions indicated.

records, the number of object points, and the object label. All object leaf nodes encountered are extrapolated to level 0 of the data hierarchy (unless the node is already at level 0). This extrapolation is required because leaf nodes have no pointers to nodes at lower levels. As an example, consider the quad tree node labeled 25 in Fig. 3. This represents the lower right-hand quadrant of the image and resides at level 2 of the quad tree. The extrapolation to level 0 assigns the same label to all the image or grid points that reside at level 0 in the region represented by this block. The extrapolation to all points at level 0 related to a node at level $l > 0$ is easily achieved using the level index and the node position at that level. For the general case, if a node is at level $l > 0$ at position (i, j), then all nodes at level 0 at positions (m, n), $2^l(i - 1) + 1 < m < 2^l i$, $2^l(j - 1) + 1 < n < 2^l j$, are related. These points are added to the relevant object record. The object point data are stored in records of type point, which will be accessed by means of the pointer in the object record. An object point record has three fields: two for the coordinates, and one for the image intensity value at that point.

The next stage of the segmentation is to determine points in the image that lie outside the region of interest but that belong to objects partially in the region. If this procedure is not performed, feature points can bunch up on the boundaries of the region of interest, distorting the apparent object motion as represented by the tracks.

This is a particular problem if the required feature points are centroids (obtained by means of some averaging process over an object). If local extrema are required as feature points, it is still useful to perform the boundary search, so that an indication of whether an object is leaving or entering the region of interest can be determined from the tracking.

The boundary search algorithm is straightforward and begins by searching through the object point data for the current frame and noting those objects that have points on the boundary. A new field is introduced to the object record that tags the object as a boundary or interior object. A region growing technique is then used to extend the relevant objects beyond the boundaries. This technique is initialized by mapping the boundary objects onto a 2D array using the object labels. This array has dimensions sufficient to cover the region of interest and extends some user-defined integer number of grid points or pixels beyond the boundaries of the region of interest. A search is performed in all directions consistent with the required connectivity, starting from the boundary points of an object. This is illustrated in the schematic of Fig. 5 for a four-connectivity search. If a new adjacent point is discovered, which is determined by comparing its intensity value with the threshold, it is mapped onto the new array with the relevant object label and a new point record added to the relevant object record. If a new point is discovered that is adjacent to two points with different labels, then

the two relevant object records are merged and given the same label.

Finally, the segmentation is completed by filtering out all object records for objects below some user-defined size.

The segmentation process is carried out for all the frames required and a new record is created for each frame. This has a pointer to the object records and a field with the number of objects in the frame.

4. Feature point detection

The extraction of the feature points is probably the most awkward part of the algorithm being described, since the feature points must represent relatively smooth tracks if the tracking part of the algorithm is to work well. The method used will invariably be chosen on the basis of the type and quality of the data and the coordinate system. The methods described in the previous section depended on the data being defined on a regular, or semiregular, grid. If the data being analyzed are not in this form, then some form of interpolation onto a regular grid will be needed.

For the work reported in this paper, a simple cylindrical projection (*Plate Carée*) has been used. This is perfectly adequate for the region of the globe considered here—the tropics and midlatitudes—even though it is not conformal. Other, shape-preserving or conformal projections can also be used if higher latitudes need to be explored. Cylindrical conformal projections are particularly useful in this respect because they preserve not only the shape of phenomena but also the shape of the rectangular domain. For statistical analysis of results, a cylindrical equal-area projection may be useful (Canters and Decleir 1989).

Detection of suitable feature points will depend on whether the objects identified have any internal structure. For the type of fields likely to be encountered that are of geophysical interest, the local extrema are the most suitable feature points. For a field defined on a high-density grid—for example, a satellite image—the nodal local extrema (a grid point with a value greater than or equal to its neighboring grid points) may be good enough. However, for a sparse grid, the nodal extrema may not be sufficient for tracking, as the tracks produced would not look very smooth, showing a "staircase" effect.

The nodal maxima or minima are relatively easy to find for each object. This is achieved simply by comparing each object point with its neighbors. However, clusters of local maxima or minima may occur with points having the same local highest value. To get a minimal set of local extrema for each object, the points obtained for each object are passed to the CCL algorithm again to determine connected points. The centroid of these points is found for each cluster so that each cluster is represented by a single feature point.

The term *centroid* is used here as a generic term for all points determined by an averaging process, including weighted averages—for example, center of mass. If there is very little structure within an object, which is an unlikely event, some other method must be used to determine feature points.

For a sufficiently high-density grid, the nodal extrema should be perfectly adequate for tracking. However, for a coarse-resolution grid, "better quality" feature points should be obtained. "Better quality" is used here in the context of the tracking and implies that smoother tracks are possible by finding these better quality points. One way to do this would be to fit a surface to the grid points using an interpolation technique. Then, using the nodal extrema as starting points, a local optimization method could be used—for example, a conjugate gradient method—to find the local extrema more accurately. The drawback to this approach is that interpolation techniques and local optimization may be computationally expensive.

We intend to implement an algorithm that will produce a surface fit to the data that together with a local optimization technique will be used to find local extrema more accurately. The approach to this task will be based on the algorithm of Dierckx (1980), which produces either the interpolating surface or a least-squares approximation depending on the choice of a smoothness parameter. This may help also with the analysis of noisy images by producing a smoothed approximation.

For the current work we have used an ad hoc method to find the centroids of regions within each object; these have been sufficiently accurate for use with the tracking algorithm provided the tracking algorithm parameters are chosen wisely. The method for finding the centroids could form the basis of an iterative scheme, but in practice a single iteration has been found to be sufficient. The approach is to use the nodal extrema, or the points derived from them by the CCL algorithm, as initial feature points and to allocate object points to the nearest initial feature point for each object, using the Euclidean distance measure. The centroid is then computed for each cluster; a strength is also computed. Although the nodal extrema value could have been used for the strength, instead the strength is computed by simply averaging the intensity or the field value for the object points for each cluster. This is then a measure of the mean field around the nodal extrema. For objects that have only a single extremal point, the centroid can be calculated directly without the cluster analysis. This is a form of clustering algorithm but is computationally very cheap.

Each object is processed for feature points and a new field is introduced into each object record to store the feature point data. This field is a pointer to a data record of type *features* that contains two fields: the number of feature points for each object and a pointer to the fea-

ture point records. The feature points for each object are stored in records that have four fields for a 2D domain: two for the coordinates, one for the strength of the feature point, and one for the track number on which the feature point lies, thus allowing cross referencing of objects and tracks.

An example of the result of the segmentation and feature point detection procedures are shown in Fig. 6 for a relative vorticity field from the UGAMP GCM. The task remaining is now to determine the correspondence between the feature points in consecutive frames.

5. Tracking

The method used to determine the correspondence between the feature points has been adapted from the algorithm of Salari and Sethi (1990), which was based on an earlier study of Sethi and Jain (1987). This method consists of a constrained optimization of a cost function. It is combinatorially intractable to consider all tracks that might be constructed from the feature points; hence, constraints are introduced. The main constraint is termed *path coherence: the motion of objects cannot change discontinuously.* The validity of this assumption depends on the time interval between each frame. This constraint essentially states that the velocity is expected to change smoothly for each object throughout the time sequence. It is also assumed that if the motion is smooth on the sphere, then any projection of the motion will also be smooth, albeit possibly distorted. This is certainly the case with the simple cylindrical projection used at present.

The version of this technique explored by Sethi and Jain (1987) was applied to simple test problems, where, in most cases, the number of feature points was conserved throughout the various time sequences, although a limited form of occlusion was considered. The cost function, which will be described shortly, was minimized using an iterative technique that involved swapping points between the tracks to improve the local smoothness. The local smoothness forms an integral part of the cost function and is measured using a function of the vector displacement over three frames for each track. Of course, in the situations encountered in meteorology, or oceanography, for any extended time sequence, phenomena can be expected to appear and disappear. To deal with this kind of situation, the technique expounded by Sethi and Jain (1987) has been extended by Salari and Sethi (1990) with the introduction of "phantom" feature points. These are essentially used as a computational convenience. Incomplete tracks are padded out with these points, which do not need to have any positional or intensity information defined for them. They allow the total number of "real" tracks to be varied; that is, they enable phenomena to appear and disappear and allow tracks to be broken up or merged depending on the local smoothness.

Their use will be described in more detail in the following.

The tracking algorithm is initiated by linking all the feature points based on the nearest neighbor distance between points in subsequent frames so that each feature point is assigned to a track and so that the displacement between feature points on tracks is within some user-defined upper-bound displacement d_{max}. The process of track initialization starts in the first frame by assigning the k feature points to k tracks. Then, points in the next frame are assigned to these tracks based on the displacement distance, which should be less than or equal to d_{max}. Any existing tracks that do not have a point assigned from this frame are assigned a phantom point, while any points that are not used are assigned to new tracks that are padded with a phantom point in the preceding frame. This process is repeated for subsequent frames until all the feature points are assigned to tracks.

The tracks are stored in a record that has two fields: the number of true feature points in the track and a pointer to records, which identify the frame, the object, and the feature point. A phantom feature point is identified by a negative integer value in the feature point field. Also, as part of the initialization process, a set of tracks is formed that is composed only of phantom points, usually equal to the number of initial real tracks; these are required to ensure that no final track has a local smoothness greater than the specified local smoothness constraint. Although the total number of tracks, including those with phantom points only, remains constant following the initialization, the actual number of tracks made up of real feature points will vary during the optimization procedure.

The cost function Ξ is constructed from the deviations of the tracks, calculated using the positions of feature points in three consecutive frames. This is to be minimized subject to the constraints so as to produce the smoothest possible trajectories. For each track the deviation can be calculated using a measure of the direction, speed, or magnitude of acceleration, or any combination of these. For this paper, the function of Salari and Sethi (1990) has been employed, which uses direction and speed only. The local deviation $\mathscr{D}_i^k$ at time level k for track i is defined as

$$\mathscr{D}_i^k(P_i^{k-1}, P_i^k, P_i^{k+1})$$

$$= \begin{cases} 0, & \text{if } P_i^{k-1} \text{ is a phantom feature point, and} \\ & \quad P_i^k \text{ and } P_i^{k+1} \text{ are real or phantom.} \\ \phi(P_i^{k-1}, P_i^k, P_i^{k+1}), & \text{if } P_i^{k-1}, P_i^k \text{ and } P_i^{k+1} \\ & \quad \text{are real feature points.} \\ \phi_{max}, & \text{otherwise,} \end{cases}$$

(3)

where ϕ_{max} is an upper bound on the local smoothness, which acts as a penalty for missing feature points, and P_i^k represents the position of a feature point. The nondimensional function ϕ is defined in terms of the displacements and measures the local smoothness. This is given by

$$\phi(P_i^{k-1}, P_i^k, P_i^{k+1}) = w_1\left(1 - \frac{P_i^{k-1}P_i^k \cdot P_i^k P_i^{k+1}}{\|P_i^{k-1}P_i^k\| \|P_i^k P_i^{k+1}\|}\right)$$
$$+ w_2\left[1 - \frac{2(\|P_i^{k-1}P_i^k\| \|P_i^k P_i^{k+1}\|)^{1/2}}{\|P_i^{k-1}P_i^k\| + \|P_i^k P_i^{k+1}\|}\right], \quad (4)$$

where $P_i^k P_i^{k+1}$ is the displacement vector from point P_i^k to point P_i^{k+1}, and w_1 and w_2 are weights. Notice that the first term of (4) is zero, if the three points are collinear, while the second term is zero, if the displacement between the first and second points is equal to the displacement between the second and third points. The cost to be minimized is then defined as

$$\Xi = \sum_{i=1}^{m} \sum_{k=2}^{n-1} \mathcal{D}_i^k(P_i^{k-1}, P_i^k, P_i^{k+1}), \quad (5)$$

where m is now the total number of tracks obtained during the initialization step and n is the total number of frames. As previously stated, for a phantom feature point, no coordinates or strengths need be specified; a fixed penalty is associated with the phantom points according to the definition of the local track deviation. Also notice that the tracks with only phantom points in them contribute nothing to the cost function due to the first condition in (3).

For the current applications, the distance norm is defined as

$$\|P_i^k P_i^{k+1}\| = \begin{cases} \|P_i^k P_i^{k+1}\|_e, & \text{if } P_i^k \text{ and } P_i^{k+1} \\ & \text{both true feature points.} \quad (6) \\ d_{max}, & \text{otherwise,} \end{cases}$$

where $\| \quad \|_e$ is the Euclidean norm and d_{max} is a parameter that must be chosen based on the expected maximum possible displacement of objects in the projected space.

The aim is to find the set of tracks that maximizes the smoothness, that is, minimizes the cost function, subject to the maximum displacement constraint
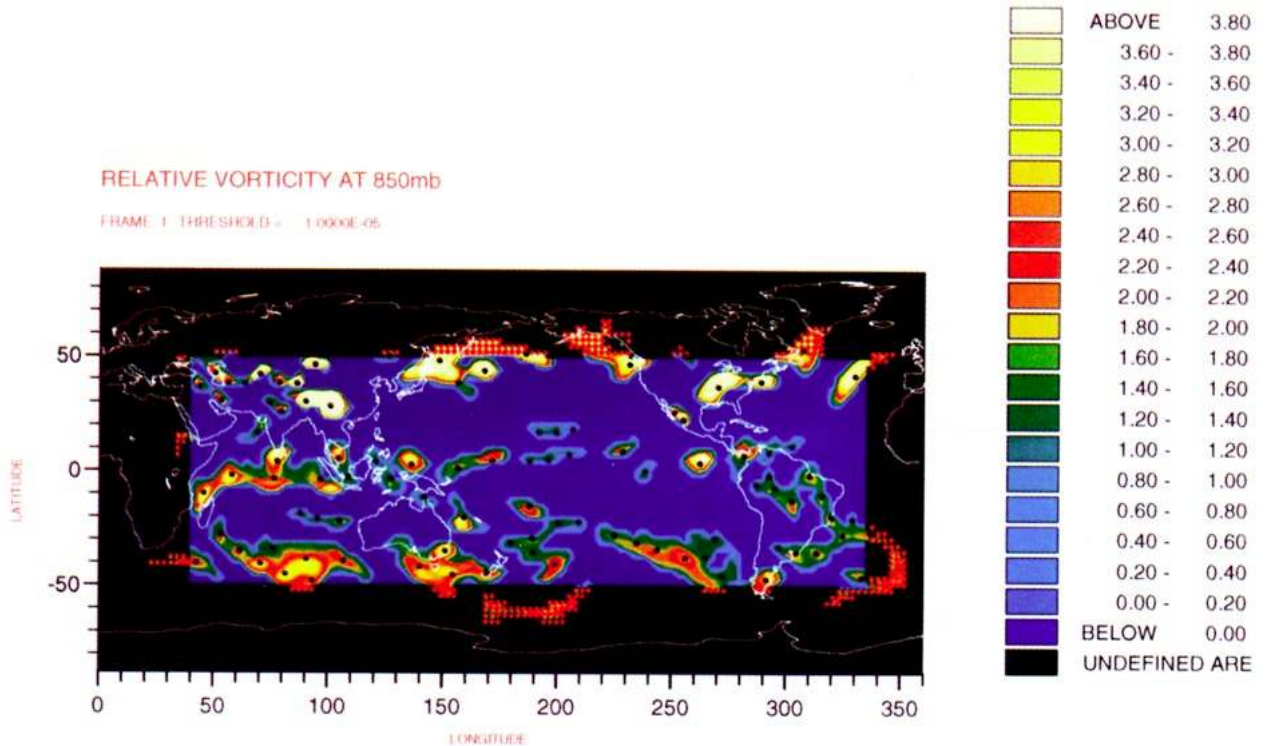

FIG. 6. Result of segmentation and feature point detection. The identified objects are color coded in units of 10^{-5} s^{-1} and superimposed on a blue background that identifies the region of interest chosen. The black dots are feature points, and red dots are object points found outside the region of interest up to a distance of six grid points from the user-defined region. The field is relative vorticity at 850 mb from a T42 UGAMP GCM. The segmentation threshold was set at 1.0×10^{-5} s^{-1}.

$\|P_i^k P_i^{k+1}\| \leq d_{\max}$ and the maximum smoothness constraint $\mathscr{D}_i^k \leq \phi_{\max}$.

The actual algorithm is described in Sethi and Jain (1987) and Salari and Sethi (1990), and so is not repeated here. The actual process of minimization consists of swapping points on tracks so as to provide the maximum gain in smoothness subject to the constraints. The gain for an exchange at time level k between tracks i and j is given by

$$g_{i,j}^k = \mathscr{D}_i^k(P_i^{k-1}, P_i^k, P_i^{k+1}) + \mathscr{D}_j^k(P_j^{k-1}, P_j^k, P_j^{k+1})$$

$$- \mathscr{D}_i^k(P_i^{k-1}, P_i^k, P_j^{k+1}) - \mathscr{D}_j^k(P_j^{k-1}, P_j^k, P_i^{k+1}),$$

provided $\|P_i^k P_j^{k+1}\| \leq d_{\max}$ and

$$\|P_j^k P_i^{k+1}\| \leq d_{\max}, \quad (7)$$

so the $k + 1$ points of the i, j pair are swapped if they correspond to the maximum gain ($g_{i,j}^k > 0$) at this time level. This process is performed for each value of k ($2 \leq k \leq n - 1$, for n frames). From (7) it is easy to deduce the role played by the phantom points. For example, suppose $\mathscr{D}_i^k(P_i^{k-1}, P_i^k, P_i^{k+1}) > \phi_{\max}$ (all three points are real feature points) and $\mathscr{D}_j^k(P_j^{k-1}, P_j^k, P_j^{k+1}) = 0$ (suppose all three points are phantom points). Then by swapping P_i^{k+1} and P_j^{k+1}, the local smoothness, $\mathscr{D}_i^k(P_i^{k-1}, P_i^k, P_i^{k+1}) = \phi_{\max}$ and $\mathscr{D}_j^k(P_j^{k-1}, P_j^k, P_j^{k+1})$ is unchanged. Thus, at the termination of the algorithm the local smoothness will be less than or equal to $\phi_{\max}$ for all tracks.

The process is an iterative stepwise process analogous to hill climbing techniques and, like hill climbing techniques, can encounter problems due to the local nature of the process. What is actually required is the global optimum, where as the algorithm used here produces an approximation to the global solution. To obtain a global solution, simulated annealing techniques or genetic algorithms might be employed; however, this would generally be more time consuming than a local technique and computationally prohibitive if the calculation must be repeated many times. Hence, for the purpose of building a statistical description of the climatology from the trajectories, the suboptimal local technique has been judged to be sufficient.

The algorithm must proceed alternatively forward and backward in time over the whole time sequence, unless some initial correspondence over a few frames is known. This is because if the initial correspondence is determined arbitrarily, then by stepping only forward in time, the result will be a function of the initialization. If some technique could be used to predetermine the correspondence in the first few frames, following the first appearance of an object, then only forward time stepping would be required. This has not been examined yet because of the complexity of objects appearing and disappearing throughout the frame sequence.

In practice, for a long time sequence the resultant tracks consist of locally smooth sections separated by

phantom points. This is a consequence of swapping real and phantom points due to violation of the smoothness constraint. These sections may or may not be related. If they are related, we can say there has been poor feature detection, possibly because of occlusion. In the application reported here, occlusion is impossible since the data are defined on a single pressure level that determines the vertical position. If occlusion, or some other reason for poor feature detection is known to occur, some analysis of continuity for the separate pieces may be needed. However, this is not required here. The set of tracks obtained from the tracking algorithm are scanned and segmented into separate tracks if a discontinuity is found. Finally, the phantom points are discarded (they are used only as a calculational aid) and the track data stored in records as previously described in this section.

Operationally, four parameters need to be chosen: the two weights w_1, w_2, and the two optimization parameters $d_{\max}$ and $\phi_{\max}$. The choice of these parameters will depend on the quality of the detected feature points (how smooth can the tracks be expected to be in terms of direction and speed when using these points) and the spatial and temporal time scales within the data. A sensitivity study could be performed as was done by Murray and Simmonds (1991). However, it has usually been found that an intuitive guess is sufficient. An animation of the data has been found to be of particular use in aiding the choice of the parameters $d_{\max}$ and $\phi_{\max}$ and in verifying the operation of the algorithm for short sequences. The technique has been found to be reasonably robust to the choice of the weights, provided $d_{\max}$ and $\phi_{\max}$ are well chosen.

The optimization process can become very inefficient if performed over a very long sequence of frames for data that admit a lot of short tracks, since a great many phantom points may be required. The most computationally efficient means of running the code is to ensure that tracks are as full as possible with true feature points. This is achieved by running the code for short overlapping sequences whose resulting tracks can be spliced together when, and if, track sections overlap. The length of sequence to choose for any set of data will depend on the temporal scales within the data.

6. Some results using model output

The initial motivation for the development of the method has been the analysis of data from the UGAMP GCM to explore the effects on the synoptic variability of using different convective parameterizations, and the degree of interaction between the tropics and the extratropics for these different parameterizations. The analysis is performed by examination of cyclonic (and possibly anticyclonic) activity. In this paper the results are restricted to a single example that illustrates the performance of the technique. Full results will appear in a future publication.

The cyclonic activity in the model output may be explored by examining fields such as the relative vorticity or pressure. However, the pressure field is not a good indicator of cyclonic activity in the tropics, because the pressure anomaly associated with a tropical disturbance is generally weak, although it has been used to good effect in the extratropics (cf. Murray and Simmonds 1991). The low-level (850 mb) relative vorticity field is a much better indicator of tropical variability associated with convective activity—for example, equatorial synoptic waves (Reed et al. 1986). Both the cyclonic and anticyclonic activity can be explored with this field, although typically only the cyclonic activity is examined.

The UGAMP GCM used to produce data for this study is a spectral model derived from the cycle 27 version of the ECMWF model (released in 1987). Horizontal components of variables are represented by truncated series of spherical harmonics, while in the vertical there are 19 levels represented by a hybrid coordinate. A more detailed discussion of the spectral technique and the parameterizations that have been incorporated into the model can be found in Simmons et al. (1988) and Tiedtke et al. (1988). Several changes to the physical parameterizations have been incorporated to the UGAMP GCM since its acquisition; these include changes to the vertical advection and the radiation parameterization (Slingo et al. 1993).

The data used for this paper were obtained by running the model for 360 days at horizontal resolution T42 in perpetual January mode, so January climatological sea surface temperature (SST) forcing has been used. The convective scheme that has been used is that of Betts (1986) and Betts and Miller (1993), which models both deep and shallow convection by the simultaneous relaxation of temperature and moisture fields toward observed thermodynamic structures.

The results shown in this paper have been obtained by applying the algorithm to the low-level relative vorticity field (850 mb). The parameters used for the image processing and tracking are as follows. The threshold used for the image processing part of the process is 1.0×10^{-5} s^{-1}, while the tracking parameterization used was 1) $w_1 = 0.6$; $w_2 = 0.4$; 2) $d_{max} = 6°$; and 3) $\phi_{max} = 0.25$.

More weight is given to the directional part of the cost function to compensate for the fact that the centroids are not the best feature points to use. In other words, there is greater error in the displacements than would be the case if the local maxima or minima could be used.

The tracking code was run on overlapping 15-day sections of the time sequences (6-h time steps), and the resultant tracks were spliced together. The region covered is the zone 50°S–50°N latitude and 0°–360° longitude. Two passes of the data are performed for each set of data because of problems experienced at the equator, as will now be explained. The initial interest is in the cyclonic activity in the tropics and how it can affect the midlatitude storm track. However, the cyclonic activity in the relative vorticity field, by definition, has a different sign in the Southern Hemisphere than in the Northern Hemisphere, so there is a continuity problem in dealing with the cyclonic part of the field in the equatorial region. One way to tackle this difficulty is to multiply the Southern Hemisphere part of the field by −1, so that it can be treated on the same basis as the Northern Hemisphere part in the same run of the code. However, performing this inversion can lead to an observed lack of continuity across the equator with a hole in the track distribution that is clearly artificial. This approach also takes no account of those systems that have sufficient strength to cross the equator. Thus, two passes of the data are performed for the two signs of the field. This has the beneficial result that continuity across the equator is maintained and that cyclonic and anticyclonic activity may be explored at the same time. Thus, for the positive part of the field we have cyclonic activity in the Northern Hemisphere and anticyclonic activity in the Southern Hemisphere, while for the negative part of the field this is reversed. However, we will show only the result of the process for the positive part of the field.

The results are shown in Fig. 7 for a 90-day period. The colored dots are at the positions of the detected feature points (at positions 6 h apart for each track), the color coding of the dots indicates the scaled strength of the system, and the arrows (colored dark blue; the arrow head may be difficult to see on pictures of this size covering the geographical area under consideration) indicate the correspondence and therefore the direction and speed of the system. The dot separation also gives an indication of velocity where there is difficulty discerning the arrows.

We will make only qualitative remarks about the results here and leave a more detailed discussion concerning comparisons between convective schemes on intraseasonal and interannual time scales together with statistical analyses to future publications. In Fig. 7 the apparent strong activity over the Himalayas and other mountainous regions is spurious and is caused by the intersection of the 850-mb surface with the orography. We see that the algorithm has performed well. In the tropical Pacific two distinct groups of easterly waves can be seen in the region of the intertropical convergence zone (ITCZ) between 0° and 10°N. The first group is situated between 230° and 250° longitude with tracks originating due to cyclogenesis over southern Central America and propagating northwestward, with the majority terminating around 10°N. The second group originate around 225° and propagate westward more coherently than the first group. Tracks also increase in strength, as indicated by the change in color of the dots. The growth in the strength of these cyclonic
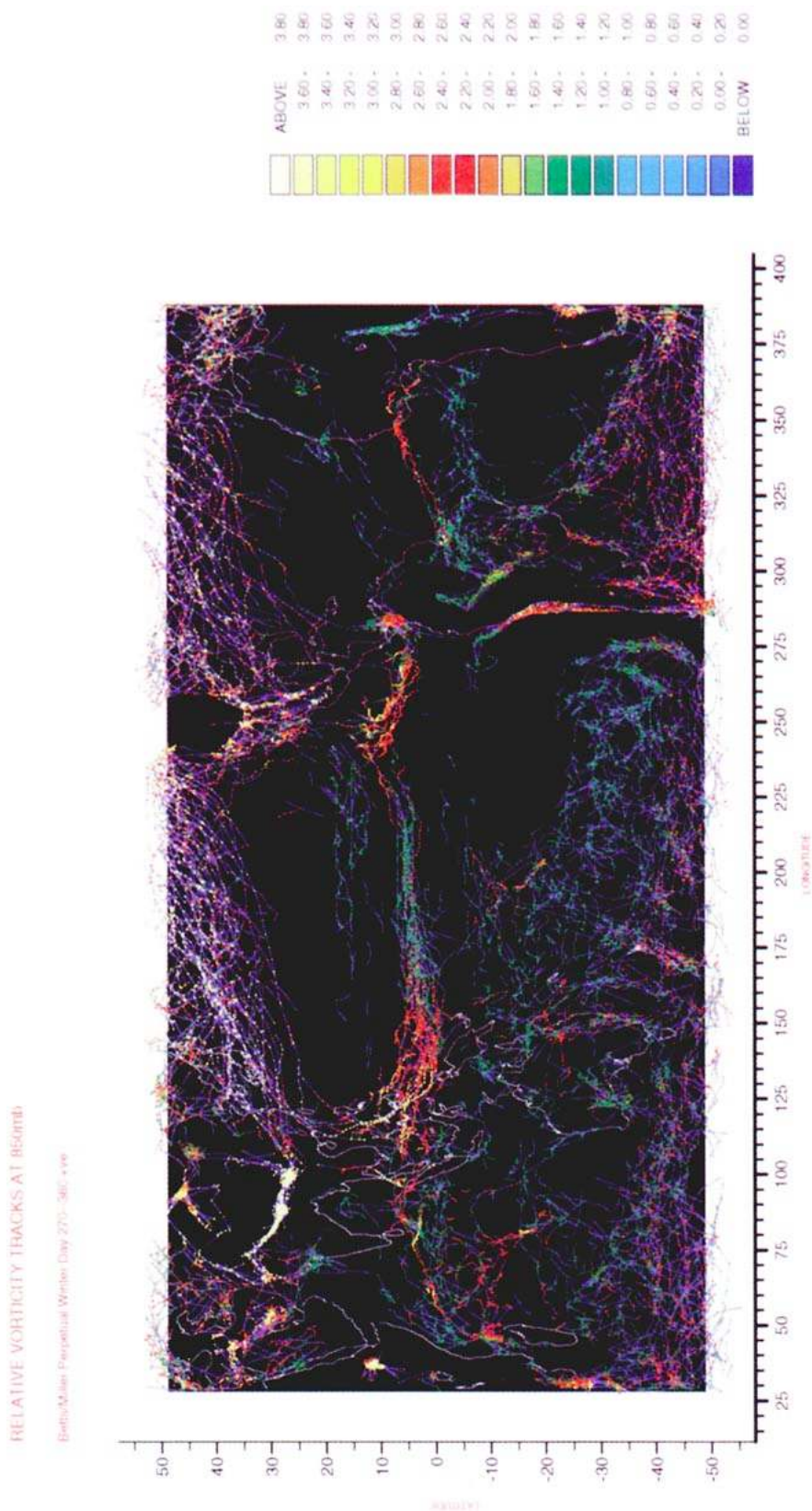
MONTHLY WEATHER REVIEW

FIG. 7. Tracks obtained using 850-mb vorticity data from the UGAMP GCM with the Betts–Miller convective parameterization, 90 days, 6-h intervals for the positive part of the field. Color scale is in units of 10^{-5} s^{-1}.

systems because of latent heat release in the ITCZ is most pronounced west of the date line. Also notice that several intense systems deviate northward away from the ITCZ at the western side of the Pacific to interact with systems in the midlatitude storm track. The tracks for this second group appear to be tightly confined within the ITCZ, apart from the western side of the Pacific. This together with the intensification and poleward migration may be overdone because of the use of the constant SST boundary condition.

In the tropical Atlantic there are a few southwestward propagating waves originating around the Gulf of Guinea. However, the January conditions used in the GCM run are not conducive to producing the easterly waves that are apparent during the northern summer months in this region.

We turn now to the northern storm tracks. In the Pacific the majority of the cyclonic activity originates over China. These systems move eastward and grow in strength to form part of the Pacific storm track. This accords well with the observed winter behavior (Boyle and Chen 1992). However, by examining other 90-day sections of the GCM run (not shown here), there is some variation in the region of cyclogenesis, with some 90-day periods showing the South China Sea to be a region of cyclogenesis. For the Atlantic midlatitude storm track, the continental United States is the major region of cyclogenesis but with some activity in the Gulf of Mexico.

For the Southern Hemisphere, the anticyclonic activity is generally weak and disorganized; however, at the latitudes shown, we are not yet into the southern storm track, which may show more organization for the anticyclonic activity.

A better indication of the variability in the data is obtained by looking at shorter time sequences—for example, on the intraseasonal time scale (14–30 days). Indeed, the results do show considerable variability. However, the best method of analysis is to construct a statistical representation of the results. This is currently being explored so that quantitative comparisons can be made.

7. Conclusions and further work

This paper outlines a general method of tracking objects through GCM output and remotely sensed datasets. Further work to improve the method is under way with the aim of implementing the surface-fitting algorithms, as already mentioned. This will enable different types of projections to be dealt with and, together with a local optimization algorithm, will allow local extrema to be found accurately. Other types of projections will be implemented, particularly for treating the high latitudes and for performing statistical analyses. Eventually, the aim is to work directly on the sphere instead of using projections; this is also being explored.

An example has been shown of how the method performs for data obtained from the UGAMP GCM. This has shown that coherent equatorial waves can be identified and tracked and that their poleward migration from the ITCZ can be followed into the midlatitude storm track.

The main thrust of future work will be to explore and implement methods of statistical analysis of the sets of tracks obtained from the GCM data and from remotely sensed data. Thus, it is hoped that quantitative comparisons between various datasets can be made and that the variability predicted by the models can be compared to that actually observed in the atmosphere.

REFERENCES

Aggarwal, J. K., and N. Nandhakumar, 1988: On the computation of motion from sequences of images—A review. *Proc. IEEE*, **76**, 917–935.

Akyildiz, V., 1984: Systematic errors in the behaviour of cyclones in the ECMWF operational models. *Tellus*, **37A**, 297–308.

Betts, A. K., 1986: A new convective adjustment scheme. Part I: Observational and theoretical basis. *Quart. J. Roy. Meteor. Soc.*, **112**, 677–691.

——, and M. J. Miller, 1993: The Betts–Miller scheme. *The Representation of Cumulus Convection in Numerical Models, Meteor. Monogr.*, No. 46, Amer. Meteor. Soc., 107–121.

Boyle, J. S., and T. J. Chen, 1992: Synoptic aspects of the winter East Asian monsoon. *Monsoon Meteorology: Oxford Monograph on Geology and Geophysics*, No. 7, Oxford University Press, 125–160.

Canters, F., and H. Decleir, 1989: *The World in Perspective: A Directory of World Map Projections*. John Wiley & Sons, 159–163.

Dierckx, P., 1980: An algorithm for surface fitting with spline functions, Report TW 50: Department of Computer Science, K. V. Leuven.

Endlich, R. M., D. E. Wolf, D. J. Hall, and A. E. Brain, 1971: Use of a pattern recognition technique for determining cloud motions from sequences of satellite photographs. *J. Appl. Meteor.*, **10**, 105–117.

Gates, W. L., 1992: The validation of atmospheric models. PCMDI Report No. 1, Lawrence Livermore UCRL-ID-109810.

Kernighan, B. W., and D. M. Ritchie, 1988: *The ANSI C Programming Language.* 2d ed., Prentice-Hall International Limited.

Knuth, D. E., 1973: *The Art of Computer Programming. Vol. 1: Fundamental Algorithms.* Addison-Wesley.

König, W., R. Sausen, and F. Sielmann, 1992: Objective identification of individual cyclones. CAS/JSC Working Group on Numerical Experimentation: Research Activities in Atmospheric and Oceanic Modelling, Report No. 17, WMO/TD No. 467, 7.40–7.41.

Lambert, S. J., 1988: A cyclone climatology of the Canadian Centre general circulation model. *J. Climate*, **1**, 109–115.

Leese, J. A., and C. S. Novak, 1971: An automated technique for obtaining cloud motion from geosynchronous satellite data using cross correlation, *J. Appl. Meteor.*, **10**, 118–132.

Le Treut, H., and E. Kalnay, 1990: Comparison of observed and simulated cyclone frequency distribution as determined by an objective method. *Atmósfera*, **3**, 57–71.

Murray, R. J., and I. Simmonds, 1991: A numerical scheme for tracking cyclone centres from digital data. Part I: Development and operation of the scheme. *Aust. Meteor. Mag.,* **39,** 155–166.

Rangarajan, K., and M. Shah, 1991: Establishing motion correspondence. *CVGIP: Image Understanding,* **54,** 56–73.

Reed, R. J., A. Hollingsworth, W. A. Heckley, and F. Delsol, 1986: An evaluation of the performance of the ECMWF operational forecasting system in analyzing and forecasting tropical easterly wave disturbances Part 1: Synoptic investigation, ECMWF Tech. Rep. No. 58, 1–74.

Rosenfeld, A., and A. C. Kak, 1976: *Digital Picture Processing.* Academic Press.

Salari, V., and I. K. Sethi, 1990: Feature point correspondence in the presence of occlusion. *IEEE Trans. PAMI,* **12,** 87–91.

Samet, H., 1989: *Applications of Spatial Data Structures: Computer Graphics, Image Processing and GIS.* Addison-Wesley Publishing Company.

Sethi, I. K., and R. Jain, 1987: Finding trajectories of feature points in a monocular image sequence. *IEEE Trans. PAMI,* **9,** 56–73.

Simmons, A. J., D. M. Burridge, M. Jarraud, C. Girard, and W. Wergen, 1988: The ECMWF medium-range prediction models. Development of the numerical formulations and the impact of increased resolution. *Meteor. Atmos. Phys.,* **40,** 28–60.

Slingo, J. M., K. R. Sperber, J. Morcrette, and G. L. Potter, 1992: Analysis of the temporal behavior of tropical convection in the ECMWF model. PCMDI Report No. 2, Lawrence Livermore UCRL-JC-109815.

——, M. Blackburn, A. K. Betts, R. Brugge, B. J. Hoskins, K. I. Hodges, L. Steenham-Clark, and J. Thuburn, 1993: Mean climate and transcience in the tropics of the UGAMP GCM. Part 1: Sensitivity to convective parameterization. *Quart. J. Roy. Meteor. Soc.,* in press.

Streten, N. A., and A. J. Troup, 1973: A synoptic climatology of satellite observed cloud vorticies over the Southern Hemisphere. *Quart. J. Roy. Meteor. Soc.,* **99,** 56–72.

Tiedtke, M., W. A. Heckley, and J. Slingo, 1988: Tropical forecasting at ECMWF: The influence of physical parameterizations on the mean structure of forecasts and analyses. *Quart. J. Roy. Meteor. Soc.,* **114,** 639–664.

Williamson, D. L., 1981: Storm track representation and verification. *Tellus,* **33,** 513–530.