Scenario-Driven Automatic Pattern Recognition in Nowcasting

ROBERT C. MCARTHUR AND JAMES R. DAVIS
Arthur D. Little, Inc., Cambridge, MA 02140

DAVID REYNOLDS
Cambridge Consultants Limited, Cambridge, England

(Manuscript received 18 March 1986, in final form 12 December 1986)

ABSTRACT

The purpose of this paper is to illustrate how the construction of a knowledge-based system (KBS) to support nowcasting, can be used to guide and facilitate the development of objective pattern recognition algorithms for use with meteorological data. We believe that a KBS based on the semantic interpretation of weather data, using the concept of weather scenarios, can assist the development and use of objective algorithms for pattern recognition in two ways:

1) it focuses the development of pattern recognition algorithms on only those phenomena which are most useful to operational forecasters;
2) its top-down logic constrains when, where, and how objective algorithms should be applied.

We first describe our understanding of nowcasting expertise and the use of pattern recognition ("manual") by human forecasters. We then briefly review the current use of automatic pattern recognition in nowcasting, present the elements within a scenario and discuss a KBS architecture for using scenarios. Finally, we close by discussing the practical benefits of merging a qualitative KBS with algorithmic pattern recognition techniques.

1. The nature of nowcasting expertise

The ideas in this paper stem from an ongoing project to develop a KBS to support nowcasting at NASA's Kennedy Space Center (KSC). Most of the knowledge engineering effort has centered on identifying and characterizing the nature of the expertise of two forecasters, each with over 15 years of experience in forecasting weather at KSC. Additional interviews have been conducted with a dozen forecasters whose experience at KSC ranged from 3 months to 3 years.

We believe that the modus operandi of expert nowcasters is to build or select one or more mental models of a given weather situation by matching past experience to current observed features. These models are used as guides in identifying and tracking individual patterns in available data sources and then both to extrapolate forward very short-period events, and to predict the likely evolution of the current mesoscale system on the basis of similarity to past situations (Schlatter, 1985).

When asked to describe these models, expert forecasters typically relate them as short stories, describing the development of a complex weather pattern. Typically, a number of experiences will have been summarized into an abstract story which the forecaster then uses as a guide in observing and interpreting new weather situations. We have termed these dynamic abstractions "scenarios" to distinguish them from the more static historical records of weather events and from "scripts" in the sense understood in the artificial intelligence (AI) community (Shank and Abelson, 1977).

Expert-level nowcasters appear to depend heavily on the use of weather scenarios which characterize generic classes of weather patterns and their development over time. Scenarios are used by expert forecasters to define the range of possibilities for how the weather will evolve over the next several hours. At any one time, forecasters may use several competing scenarios to guide them in developing a forecast.

Weather scenarios have a direct parallel in the "conceptual models" discussed by Brown (1985). According to Brown, conceptual models describe typical configurations of air flow, temperature and moisture. Similar to scenarios, the primary use of conceptual models lies in the interpretation of multiple data sources, and the assignment of specific weather phenomena to particular geographical regions. Scenarios, however, are explicitly concerned with the configuration and evolution of mesoscale patterns previously observed at KSC.

2. Pattern recognition and scenarios

A scenario can be thought of as a multidimensional hypothesis that describes the expected patterns of change in various weather systems across a varying number of meteorological data streams or sets.

Selecting a particular scenario as a good analogy of how today's weather might evolve requires a complex pattern identification and matching process. At least four separate levels of pattern recognition appear to be involved:

- Recognition of individual patterns or features *within a data set.*
- Recognition of clusters of related features *across data sets.*
- Fusion or differentiation of *features* within a data set *across time.*
- Recognition of patterns of change in clusters of related features *across time.*

The dynamics of meteorological phenomena add a level of complexity not normally found in pattern recognition tasks. In most high-level pattern recognition problems (such as scene identification), the properties of the objects are fixed. Even if the scene is a dynamic one, the moving "objects" do not change their intrinsic properties of shape, albedo and so forth at any significant rate. However, identifying a meteorological system is *not* just one of identifying a collection of objects but one of identifying a *process.* The individual entities in the scene (e.g., localized storm cells, clear areas, convergence zones) evolve and give only clues to the processes involved but do not themselves constitute that process. Two weather situations which are the same, in terms of the processes involved, can appear as being very different when viewed as collections of individual weather elements. These considerations become critical when the aim is to track the evolution of short-lived weather elements over most or all of their life cycle.

3. The problem for automatic pattern recognition

In a modern forecasting environment, such as that provided by a PROFS or McIdas workstation (Mandics and Brown, 1985; Schlatter, 1985), the forecaster is deluged by the sheer quantity of data which is available. It is physically impossible for a forecaster to examine all of the data, yet somehow expert forecasters cope quite well. Inexperienced forecasters, on the other hand, are frequently at a loss to know precisely where to focus their attention.

We believe that one of the primary differences between expert and novice forecasters in such a data-intensive situation is that expert forecasters are *selective* in the data they choose to examine. Experts appear to ignore the majority of data and concentrate only on that which is *currently* most meaningful. In other words, they effectively use their knowledge and experience to guide them in deciding what subset of data to analyze. This can be thought of as a top-down approach in which top-level scenarios guide the expert in selecting data and features to examine.

In contrast, the typical approach to developing and using automatic pattern-recognition algorithms has been bottom-up. A set of processes is developed which identify any and all features that might be of interest in a data set, and aggregate them into any and all objects of potential interest.

Given the number of sensor systems available in a modern forecast facility and the significant computational requirements for most pattern recognition algorithms, it is nearly impossible to automatically identify every feature of potential interest in all data sets or streams. Even if it were possible, it is not clear that the forecaster could use all of the output because of the overwhelmingly large number of features available.

An expert human being, on the other hand, determines what patterns and features are of interest *before* doing any detailed analysis of hard data. We believe this selection is governed by which scenarios the expert has selected as potential analogies for today's evolving weather patterns.

Our thesis is that scenario descriptions can guide in the selection of which data should be used, which algorithms should be applied, and when the analyses should be conducted. Such an approach should allow automatic pattern recognition to replicate the expert human's selectivity in the face of too much data. We are using the scenario structure to describe the meteorological process. Scenario recognition is peformed by matching the scenarios to today's situation using global scene descriptors (such as overall synoptic flow, presence of local cells in different regions relative to the synoptic flow and existence of local convergence zones).

4. Status of pattern recognition in nowcasting

There are currently three areas of automatic pattern recognition in support of nowcasting operations:

1) derivation of inferred meteorological fields;
2) identification of individual meteorological features; and
3) tracking of individual meteorological features.

Inferring significant meteorological fields from observed data is perhaps the most advanced of current applications. In some cases there is a direct physical connection between the desired field and the observable data (for example, derivation of rainfall rate from radar reflectivity by use of a raindrop spectrum) in which case the procedure can be carried out purely algorithmically with recourse to recognition techniques. However, many times there is no simple physical relationship which can be used as the basis for a purely analytic solution, and statistical procedures must be used which deduce the desired field from weakly related observables.

This simple use of pattern recognition is equivalent to the image processing task of labeling pixels on the basis of purely local information. It provides the user with field information such as "there is a high rainfall signature is this local region" but does not specifically identify localized objects of meteorological significance. This local identification of signal patterns can be the first step in more sophisticated pattern recognition, and can also be a meteorological product for direct use by forecasters.

The second major role for automatic pattern recognition is the more complex one of actually identifying individual, meteorologically significant events. This corresponds to the segmentation and object-labeling phases of image processing systems. Typical entities of interest might be:

- region identification (sea, land, type of cloud cover) from satellite data
- wind shift boundaries (from doppler radar)
- storm cells

See Roska (1985) for an example of this type of pattern recognition.

The final role for automatic pattern recognition in current practice is the tracking of the features identified above, either for purposes of extrapolation forecasting (e.g., tracking storm cells), for prediction (e.g., looking for future intersections of wind shift boundaries which could lead to initiation of new cells) or for inferring indirect information on the bulk fluid flow.

The foregoing processes may be considered as a bottom-up flow from raw data streams to semantic-level interpretations. A simplified picture of this situation is shown in Fig. 1. Here the raw signal is first preprocessed to validate the data and peform basic noise reduction. Geometric transformations are also applied at this stage to map the data into a standard coordinate frame.

Three of the layers in Fig. 1 correspond to the three pattern recognition roles noted previously:

1) local transformation and labeling;
2) delineation of significant "objects" or regions within the data;
3) tracking of regions (possibly involving recourse back to unsegmented data).

Two final stages are required: semantic labeling of the individual regions and semantic interpretation of the complete weather situation. It is the encoding and application of this semantic knowledge that is addressed by the use of the scenario structures.

As long as no hypothesis is being used to guide the process (as is the case in all current nowcasting workstations) then the data flow has to proceed purely bottom-up as previously described. However, once a complete path exists up to a semantic interpretation it then becomes possible to run the interpretation graph top-down and use the established interpretation to

1) guide in the selection and application of lower-level processing and to concentrate the available computing resources on the important regions and phenomena;
2) provide a context in which to validate low probability identification and reduce false alarm rates (at the risk of detecting only what the high level interpretation expects to detect!).

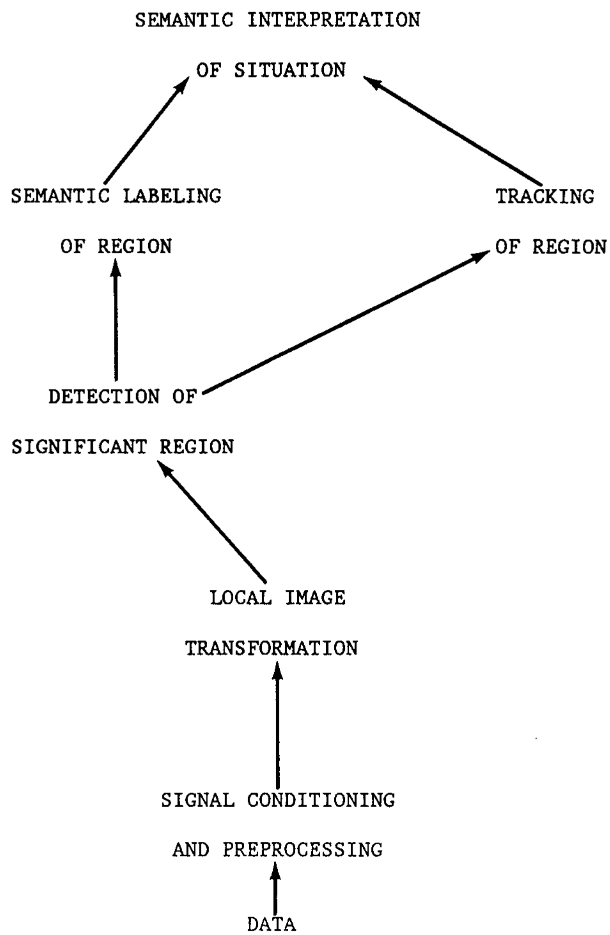

FIG. 1. Simplified sequence of processes involved in performing a pattern recognition task in nowcasting.

While the major data sets in use are from satellite and radar imagery, our aim is to interpret the complete meteorological situation, including all the other standard data sources within the interpretation: mesoscale networks, vertical soundings, hourly surface observations, field mill and lightning detection data.

5. Scenario structure

The current formulation of scenarios consists of a four-tiered hierarchy of data types, shown schematically in Table 1. The scenario level represents the highest level of abstraction. Lower levels of abstraction occur with lower rows. The order of procedural control when processing these data structures in a KBS proceeds from scenarios down the hierarchy. The four basic data

TABLE 1. Principle attributes for the four data types used to model the various levels of abstraction required to perform the pattern recognition task in Fig. 1.

Data types	Attribute type: Static	Attribute type: Dynamic
Scenario	Events: ordered list dependency constraints Assumptions: necessary sufficient Rules of thumb	Status History
Event	Predicates	Scenario Status Associated features Dependency constraints Monitoring window Time stamp
Feature	Associated variables Pattern recognition algorithms Observation interval Climatology	Events History: status location size
Variable	Measurement function	Value history Magnitude Direction

structures are represented as schemas or frames within an inheritance mechanism which allows a taxonomic description of the attributes (i.e., slots) that define the characteristics of each data structure. For more information on frames and schemas see Minsky (1975), and Fox (1979) and Winston (1984).

At the bottom of the hierarchy lies the *variable,* which usually consists of time-value pairs. The *feature,* in turn, is an identifiable weather entity, ranging in complexity from a steering level wind to an individual thunderstorm cell, described by certain variables. An *event* is a qualitative change in some feature, and occurs at a specific point in time. Finally, a *scenario* is composed of multiple events, as well as the dependencies between those events.

Each data type is defined by referencing the next-simplest data type. Events, for example, are defined as changes in features, while features reflect patterns in particular variables. Thus a scenario, which directly refers only to a set of events, indirectly depends on the definitions of features and variables which underlie those events. Table 1 also gives examples of the two types of attributes required to define each data type. Static attributes are those which are a part of a data type's primitive definition, and remain constant over the lifetime of any specific object. Dynamic attributes, will change over time as conditions change and more becomes known about a specific object. The distinction between static and dynamic attributes is a useful one, since static attributes are all that is required when defining objects residing in a permanent knowledge base, while dynamic attributes are those of interest to the forecaster when performing a problem-solving task.

Variables are relatively simple. In addition to a value history, variables typically also have indicators for their current magnitude, direction and rate of change. Methods for obtaining measurements are associated with each variable; such a method might reference a particular type of accessing function into a data base, for example, or might simply "ask the user."

Features represent weather entities with identifiable behavior. A feature corresponds approximately to something which an operational forecaster might refer to as "it" in common speech: a "cell", "sea breeze" and "frontal zone" are all examples. In addition to a list of associated variables, a feature requires a history of its status and location, and an observation interval, the length of time between observations. A feature may also have a climatology which describes statistical aspects of its historical behavior.

Events are qualitative changes in a specific feature, or by extension, a group of features. To simplify the process, an event is assumed to occur at an instant in time rather than over a finite duration; this allows an event to have a time stamp indicating exactly when it occurred, or was observed. An event also has a predicate, which is a statement that can be tested (i.e., quer-

TABLE 2. Definitions and explanations of various modules in knowledge-based scenario system in Fig. 2.

Module	Description
Detect	Using future expectations as a guide, gather data and update the current set of features being tracked.
Anticipate	Based upon the current state of all active scenarios, generate a set of expectations about the future.
Monitor	Compare current conditions with expectations and modify the list of active scenarios accordingly.
Edit	Editing and composition facility for building the knowledge base.
Data	Description
Data base	Conventional data base of meteorological observations.
Expectations	List of events (with specific monitor windows) being actively watched.
Today	List of features identified during the recent past, each of which has its own history, etc.
Activated scenarios	List of scenarios being actively monitored.
Knowledge base	Static definitions for all scenarios, events, features and variables used by the system; also a body of productions rules for testing predicates.

ied) as to whether or not it has already occurred (Kowalski, 1979; Clocksin and Mellish, 1980). Such a predicate may be considered a top-level hypothesis which, when discovered to be true, proves that the event has "happened." In addition to its predicate, an event has a monitoring window which contains the time period during which the event is expected to "happen." Various actions are typically taken depending on whether an event "happens" before, during or after its monitoring window.

A scenario is defined by an ordered list of events and dependencies between those events. Simple event dependencies take the form of

[event-1 (time-1 time-2) event-2]

which translates as "event-2 should happen between time-1 and time-2 after event-1" where time-1 and time-2 are time intervals (expressed, for example, in number of minutes). The list of event dependencies defines a highly constrained temporal sequence which is analogous to, but much more general than, the event chains and fault trees used in operations research (Taha, 1971); they are similar in many ways to the various types of event chains as discussed in the AI literature (McDermott, 1985).

In addition to the events and their dependencies, a scenario also requires a set of assumptions, both sufficient and necessary. Typically these assumptions are statements which may be queried in the same way as event predicates. Scenarios are tested to see if they should be activated by periodically querying their sufficient conditions; once activated, their necessary conditions are periodically queried to ensure that they are still worth monitoring.

6. Scenario processing

The architecture of the KBS is intended to model to a considerable degree the way we believe expert nowcasters use scenarios. Figure 2 shows the major modules and data flow within the KBS. Table 2 contains definitions of module functions. The top-level functions in Fig. 2 are "monitor," "anticipate" and "detect". The editing facility initially is an off-line facility for working on the knowledge base of scenarios. The monitor function is reponsible for monitoring the ongoing weather events of today and activating or deactivating scenarios as appropriate. Anticipate works off of the activated scenarios to generate predictions or expectations of future weather events. The detect function translates the expectations into requests for specific data or analyses and schedules the requests appropriate to the observational intervals and the monitoring windows. Each top level box can be expanded, but the details are beyond the scope of this paper.

We view this architecture as providing a means for capturing the reasoning and experience of forecasters at a high level of abstraction. It is qualitative and symbolic, and is exclusively concerned with the logic and structure of forecaster expertise. The architecture is also semantic, in that it describes and interprets weather conditions in much the same way as a human forecaster.

Pattern recognition algorithms, on the other hand, attempt to capture the detection and tracking of me-

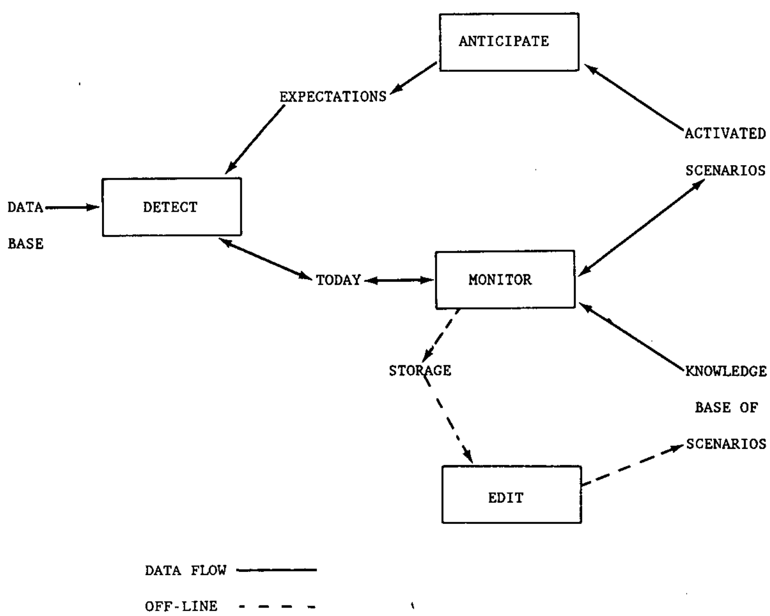

FIG. 2. Top-level description of important modules within knowledge-based system for use by nowcasters at Kennedy Space Center.

teorologically significant features in a quantitative manner. Typically, the thought processes and reasoning strategies of the human forecaster are of little interest, except as a general guideline to how pattern recognition might be completely automated. Moreover, an objective algorithm for pattern recognition almost never contains a mechanism for incremental improvement of the algorithm, other than parameter reestimation.

7. Scenario-driven automatic pattern recognition

It is our belief that practical considerations argue for the parallel development of qualitative nowcasting KBS and objective pattern recognition algorithms. We believe this for two reasons:

1) The modeling of forecaster expertise can guide the search for algorithms which are most useful to operational forecasters;
2) The top-down logic of a KBS, particularly the use of scenarios, greatly constrains when, where, and how objective algorithms should be applied.

Given the range of possible patterns which could be detected automatically in meteorological data, the first reason is fairly obvious. That is, there is an extremely large number of patterns which could be analyzed objectively, and a large number of data sets or streams in which to search for them.

By delving into the logic of forecaster reasoning, it is possible to uncover those types of features which are the most important to the forecaster and define what sorts of information should be provided about those features. The development of automatic algorithms for pattern recognition should also take into account the task-related needs of the forecaster by considering the worst-case combinations of weather, time pressures and data overload. There is little point, for example, in providing sophisticated algorithms for tracking isolated storm cells if the most pressing operational problems are caused by simultaneous monitoring of surface winds, precipitation rates and lightning frequency.

The second reason for parallel KBS and algorithm development has more to do with how algorithms are applied at run time. A KBS which uses top-down logic to find "interesting" scenarios and events to monitor can provide a powerful set of constraints concerning the use of objective analysis.

Once a symbolic model of the situation has been selected and instantiated by choosing a suitable scenario, the problem becomes one of monitoring the evolving situation both to check that the model is correct and to forecast the weather. This consists of looking for the appearance of individual events (onset of sea breeze, development of local convergence zone, development of storm cell complex) and the tracking of these events (tracking storm cell and extrapolation to predict future path and evolution).

The scenario also will indicate the geographical locations and times where the most important events are likely to occur. Thus the scheduling of computing resources to look for such events can be based on this expected likelihood and importance. A concrete example of this is the interpretation of synoptic flow to infer those regions where storm cell formation is important and to concentrate processing largely in just those regions.

The deployment of automatic pattern recognition resources will be determined by the particular set of dependency constraints between events, the set of monitoring windows across events, the observation interval for each feature within an event, the particular pattern recognition algorithms associated with each nested feature, the area in which the feature is expected to be found and the particular data-set or stream.

Given these constraints, improved efficiency can also result from selection of algorithms by compute-intensiveness. For example, a relatively inexpensive algorithm might be used early in the unfolding of a scenario when false positives might not be a problem. Then more expensive algorithms could be used to validate the occurrence of an event.

It would be possible to utilize algorithms which are more compute-intensive, on a pixel-by-pixel basis, because the algorithms will be applied in a highly selective manner. It is also possible to contemplate a much wider range of algorithms than would otherwise be possible, since only a small subset of all available algorithms will need to be applied at any one time.

A scenario-driven automatic pattern recognition scheme would have a number of implications for the pattern processing systems. A large library of algorithms would have to be available either in software or as programmably reconfigurable hardware. Both of these trends can be identified presently within the machine vision and pattern recognition industries.

The scenario structure would make the implementation of a parallel architecture relatively easy because each measurement function, algorithm call or event predicate can be handed off as an asynchronous task. Again, parallel systems architectures are beginning to appear in AI machines and in proposed machines combining AI and image interpretation.

This structure also means that the development of the KBS can proceed ahead of the development of automated pattern recognition systems, because each task can also be handed off to the forecaster for detection and/or measurement.

The advantages of a scenario-driven system are bought at the risk of detecting unexpected events somewhat later than they could have been under a purely bottom-up recognition scheme. This is, in any case, the price paid by humans with their expectation-driven view of the world. We can only strive for a compromise which allows a suitable trade-off in wasted computing power against the risks of failing to see the unexpected.

8. Future directions of nowcasting KBS

During the first phase of development, the nowcasting KBS domain has been restricted to summertime thunderstorms. The scenario structures are precompiled and not modifiable at run-time. They are either active or inactive.

Subsequent phases will extend the domain of the system to year-round weather phenomena and will incorporate a dynamic scenario modification system so that a variation on a given scenario can be constructed during run-time based upon partial matches to existing data. Such an extension could begin to incorporate some elements of causal modeling (Bobrow, 1985; Hayes, 1979).

Future extensions could incorporate a facility whereby new scenarios were automatically generated by the system based on symbolic descriptions of today's actual weather events. Such a system implies a well-understood vocabulary of symbolic descriptors and an extensive taxonomy of scenarios (among other things).

Additional types of knowledge could be included as part of the scenario selection process. For example, the launch of a shuttle orbiter is accompanied by a large number of weather constraints. Given those constraints, scenarios could be monitored which have events outside NASA guidelines which could occur within the launch window.

9. Summary

We urge the use of a KBS architecture which employs a semantic interpretation of current conditions to govern, in a top-down fashion, the execution of objective algorithms. We envision that such an architecture will imply the need for algorithms which are considerably simpler than those needed in the absence of semantic information. On the other hand, there will likely be a greater variety of algorithms required, since they largely will be specialized to handle the detection and monitoring of very particular types of meteorological features. We firmly believe that the combination of qualitative and quantitative knowledge in such a hybrid system will be far more powerful (and useful!) than the current generation of support tools for operational forecasters.

REFERENCES

Bobrow, D. G., 1985, Ed: *Qualitative Reasoning about Physical Systems,* MIT Press.

Brown, K. A., 1985: Conceptual Models of Precipitation Systems, *ESA Journal,* **9,** 526–538.

Clocksin, W. F., and C. S. Mellish, 1980: *Programming in Prolog,* Springer-Verlag.

Fox, M. S., 1979: On Inheritance in Knowledge Representation. *Proc. Sixth Int. Joint Conf. Artificial Intelligence,* Tokyo, 1.

Hayes, P. J., 1979: The Naive Physics Manifesto. *Expert Systems in The Micro-Electronic Age,* D. Mitchie, Ed., Edinburgh University Press.

Kowalski, R., 1979: *Logic for Problem Solving,* North-Holland.

Mandics, P. A., and Brown, R. C., 1985: When Will the Storm Get Here? *Research and Development,* Technical Publishing Group.

McDermott, D. V., 1985: A temporal logic for reasoning about processes and plans. *Cognitive Science,* **6,** 101–155.

Minsky, M. L., 1975: A framework for the representation of knowledge. *The Psychology of Computer Vision,* P. Winston, Ed., McGraw-Hill.

Roska, K. O., 1985: Large-area histograms from meteosat images. *ESA Bulletin,* **27,** 13–15.

Schlatter, T. W., 1985: A day in the life of a modern mesoscale forecaster. *ESA Journal,* **9,** 235–256.

Shank, R. C., and R. P. Abelson, 1977: *Scripts, Plans, Goals and Understanding,* Lawrence Erlbaum Associates.

Taha, H. A., 1971: *Operations Research,* The MacMillan Company.

Winston, P. H., 1984: *Artificial Intelligence,* Addison-Wesley.