Integrating Spatial Data: A User's Perspective

Joseph M. Piwowar and *Ellsworth F. LeDrew*
Earth Observations Laboratory, Institute for Space and Terrestrial Science, Department of Geography, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada

ABSTRACT: An emerging trend in geographic information systems (GIS) applications is the use of multiple systems and diverse data sets in a single study. In practice, sharing spatial data among several different systems is difficult because of incompatibilities in spatial data formats and limitations within existing GIS. This leaves users unduly burdened with the task of trying to fit their data into the system they want to use.

New GIS must remove the burden of data formats from the user. This can be accomplished by (1) providing transparent access to a diverse range of spatial data sets, and (2) processing these data as if they were native to the system. We propose a new GIS design – a *multi-format* GIS – which has the potential to transparently integrate spatial data from diverse sources into a common operation. In this design, users identify the files they wish to include in their analyses without concern for data formats.

INTRODUCTION

RECENTLY, MUCH EMPHASIS has been placed on the integration of remotely sensed imagery from Image Analysis Systems (IAS) with other digital spatial data from Geographic Information Systems (GIS). It is widely recognized that this type of synergism has the potential to open new avenues for each (Estes, 1985; Jackson and Mason, 1986; Goodenough, 1988). In fact, many scientists already use a number of different IAS and GIS in a single project to take advantage of processing capabilities that some systems have and others do not (Abel, 1989). We have also seen how researchers initiating new studies first look into spatial data archives to see if an existing map can be used, or modified to suit their goals, before attempting to collect the information themselves. Thus, the emerging trend in GIS/IAS applications is towards multi-system use of multi-format spatial databases.

In practice, sharing spatial data among several different systems is difficult because of the unique way in which each system stores and processes its data. Users of spatial information systems, however, should be free to devote all of their time to analyzing their data: they should not be burdened with manipulating their data sets to get them into some prescribed format. This premise includes the ability to make transparent use of as many different data sets as required to solve a particular problem and is not restrictive to using a single IAS/GIS.

One solution to this difficulty is to standardize the methods of data manipulation. There has been a lot of effort recently to develop standards for spatial data exchange (as attested by the number of standards-related papers found in recent GIS and AutoCarto symposia), but these have so far resulted in yet more data formats and little standardization. Due to the non-structured nature of spatial phenomena, and the diverse agenda and jurisdictions of standards delegations, however, it seems unlikely that we will see a truly universal standard emerge from current efforts – at least not in the short term.

Developers of new spatial information systems must be encouraged to remove the burden of data formats from the user. This can be accomplished by (1) providing transparent access to a diverse range of spatial data sets, and (2) processing these data as if they were native to the system. In this paper, we suggest how these issues can be overcome by examining the data integration issue from a user's perspective. We review some of the fundamental issues of combining diverse data sets and categorize existing analysis systems by their methods of addressing these problems. Based on this review, we propose a conceptual design for a new type of information system which provides transparent access to many spatial data sets. We describe a prototype system which we have developed to transparently access foreign data (i.e., data stored in an external format) from within a second spatial analysis system. We also suggest a set of Integration Guidelines to guide the system during spatial analyses by determining the best method of resolving a query, given the formats of the data sets involved and the efficiency and accuracy of the available analysis tools.

The intent of this paper is to present a framework of ideas to stimulate thought among IAS/GIS users and developers on an alternative system design. We do not present a completed solution to the data integration problem. Instead, we provide a basis upon which such a solution can be built.

DATA INTEGRATION ISSUES: A USERS' PERSPECTIVE

The problems stemming from attempts to integrate spatial data sets can be grouped into (1) fundamental differences in the ways we model reality, (2) difficulties in format conversions, and (3) the limitations of many existing analysis systems.

SPATIAL MODELS OF REALITY

A spatial database represents a model of reality. As such, no one model is inherently better, or more accurate, than another (Holder, 1988). Spatial data are typically collected, stored, and manipulated in either a line (*vector*) or grid (*raster*) format. Vector formatted data are more representative of the way we think spatially, hence they are used most frequently in manual methods of data collection and presentation. Automated data collection, however, frequently produces data in a raster form because the values are gathered in a structure which is optimized to computer architecture – a regular grid. Both the vector and raster structures have advantages and difficulties which are well described in the literature (see Peuquet, 1984; Burrough, 1987; or Aronoff, 1989), yet their fundamental differences can make data conversion between them a complicated task (Piwowar *et al.*, 1990).

Even within the same generic spatial model there can be a multitude of format variations ranging from structural differences (e.g., row-order raster versus run-length encoding versus quadtrees), to software variations (e.g., the *spatial* data may be stored identically between two systems but the *ancillary* information has a different format), or hardware variations (e.g., byte-ordering changes on different computers). Each of these variations may be trivial to a system programmer yet can be real barriers at a user's level.

CONVERSION ISSUES

Integrating data stored in different formats can be tedious and error-prone because any conversion of data invariably leads

PHOTOGRAMMETRIC ENGINEERING & REMOTE SENSING,
Vol. 56, No. 11, November 1990, pp. 1497–1502.

0099-1112/90/5611–1497$03.00/0

to some generalization and loss of accuracy. This may be critical depending on the application. For example, when converting vector polygons to raster regions, there can be a significant change in the area of a polygon after conversion, depending on the relative sizes of the polygon and fineness of the grid. If we were conducting a study comparing the relative distribution of classes from two sources, this change could unjustifiably bias the results. In an evaluation of different types of vector to raster conversion algorithms, we found that, while holding all other processing parameters constant, different algorithms produced polygons of significantly different sizes (Piwowar *et al.*, 1990). Although users are beginning to routinely use stock conversion algorithms found in many IAS/GIS, they are rarely made aware of the nature and limitations of the algorithm which is employed.

If the conversion procedure and its processing options can be judiciously selected to suit the particular problem, it is possible to convert data between the two formats to minimize any errors (Saalfeld and O'Reagan, 1985; Piwowar *et al.*, 1990). We have found (using our previous example) that a polygon could be converted to its raster equivalent with only a 1 percent change in area, yet using the same algorithm its perimeter distance changes by over 4 percent.

The problem with many existing GIS/IAS, therefore, is not that the data must be converted to be usable, but that they are usually converted *a priori* before all possible applications are known. In order to minimize any loss of quality, the data should be left in their native format and converted only when required and in the manner best suited to their desired use.

GIS Limitations

Data sharing has also been hindered by incompatibilities imposed by the analysis systems we use. Every system requires that its data be stored in a specific, frequently unique, format. Consequently, spatial databases are usually incompatible from one system to another. Unfortunately, many system developers view their program as the producer of an end-product; the final destination of spatial data from other sources. As a consequence, facilities for importing data from other formats are far more common than facilities for exporting data.

TRADITIONAL GIS DESIGNS

Textbooks have traditionally identified four main components of a GIS: input, management, analysis, and output (Figure 1a) (Marble *et al.*, 1984; Burrough, 1987; Aronoff, 1989). This is the system developer's view.

We suggest that users' typically see a GIS as a set of analysis tools (which we call the *Function Module*) built on top of a structured database (*Data Module*) and equipped with an appropriate user interface (*Query Module*) (Figure 1b).

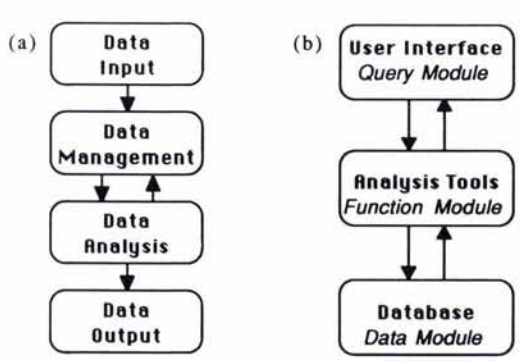

FIG. 1. Components of a geographic information system: (a) developer's perspective. (b) users' perspective (with headings used in the text).

In the Data Module, the GIS interacts with the computer's operating system to store and retrieve the data as required. In most analysis systems, the Data Module is linked to only one or two databases. (We use the term *database* in this context to refer to a collection of files, all structured according to the same format.)

Resting on the Data Module, the Function Module contains the generic spatial data manipulation functions required to satisfy any analysis operation. These functions are the toolbox for the system. Most operations would require the application of a series of these functions to complete their request. Table 1 lists fundamental GIS functions for which operators are defined in the Function Module.

The Query Module is generally the only portion of the database structure with which the user directly interacts. It is here that a complex spatial problem is presented to the analysis system for resolution. The Query Module breaks up the operation into fundamental spatial functions, feeds them to the Function Module, and reassembles the results returned from the Function Module into a complete solution.

This framework of GIS design can be used to group existing GIS into uni-format or dual-format categories, based on the structure of their data modules.

Uni-Format GIS

Many of the popular GIS now in use are uni-format systems. These geographic information systems are designed around a single spatial data model (Figure 2a). All data storage and analysis within the system operate on data in the format in which they reside in the database. Therefore, uni-format systems are typically categorized as either vector-based or raster-based. The singular nature of the data model tends to limit the applications to which a particular uni-format system is put. For example, we generally use vector-based GIS for cartographic analyses and apply raster-based systems to overlay analysis problems.

Importing data from external sources requires conversion from the foreign format to the local data structure. This is true if the data files are organized in different forms (i.e., one in vector, the other in raster), or even if the data are already in the same generic form (i.e., both vector or both raster). This conversion

TABLE 1. PRIMITIVE GIS SPATIAL ANALYSIS FUNCTIONS. THE FUNCTION CLASS IS LISTED IN BOLDFACE TYPE, FOLLOWED BY AN EXAMPLE. THE TERM *REGION* in the Examples Can Be Interpreted Equally as *PIXEL REGION* OR *POLYGON* (AFTER ESTES, 1981; MARBLE AND PEUQUET, 1983; AND WHITE, 1985).

1. **Location:** Given an attribute and a region file, find all of the regions containing that attribute.
2. **Inclusion:** Given a point and a region, does the point lie within the region?
3. **Containment:** Given a point and a region file, which region contains the point?
4. **Proximity:** Given a particular region in a region file, what are its neighbors?
5. **Dimension:** What are the areal coverages for each region in a region file?
6. **Intersection:** Given two overlapping region files, what are the proportions of each region from the first in each region in the second?
7. **Distance:** Given a point, *p*, and a point file, what is the point in the point file nearest to *p*? What is the distance from that point to *p*?
8. **Interior:** Given a region, what is the centermost point (i.e., an inside point which is farthest from the boundary)?

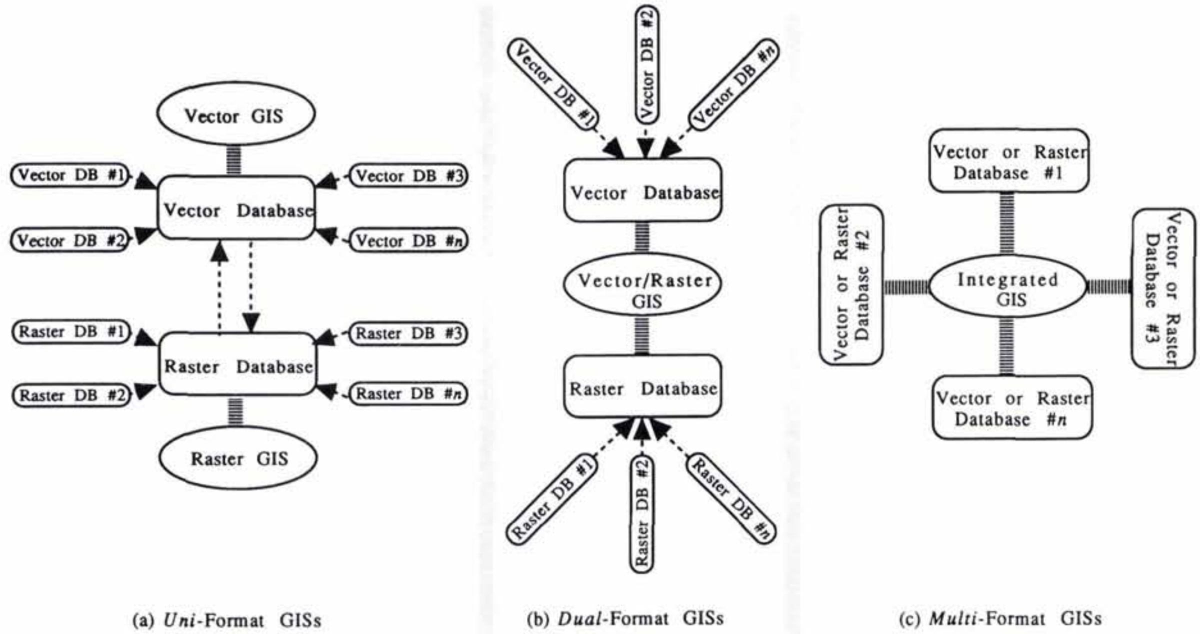

FIG. 2. Types of geographic information systems.

places an added burden on the user who must hope that the GIS in use is equipped with the appropriate programs to be able to exchange data with the desired foreign formats and convert each file to be imported.

DUAL-FORMAT GIS

An emerging trend in GIS is to design systems which can accommodate both vector and raster formatted data. Such dual-format geographic information systems have two principal databases, one for each data model (Figure 2b), and are equipped with conversion procedures to move data between their databases. Importing new information into either database of a dual-format system is limited by many of the same constraints as described for uni-format systems except that the user now has a choice of two native formats instead of one.

The processing and analysis of data within dual-format systems generally takes place in *either* vector *or* raster mode; rarely do these systems provide the functionality to perform a procedure in *both* vector *and* raster mode (e.g., polygon overlay *and* grid overlay). Consequently, users may find themselves burdened with converting their files several times throughout the course of their project, decreasing processing efficiency and data accuracies. In addition, the system may end up with two versions of the data files: one in the vector database and the other on the raster side. Not only is this an inappropriate use of disk space, but it can also lead to errors if updates to one copy of the file are made in isolation from the other version.

As an example, consider a simple map overlay problem. Because grid overlays are considerably more efficient than their polygon equivalents, overlay analyses are frequently executed in raster mode. Thus, if a vector file is to be overlayed with a raster file, the vector data would first be rasterized. Unfortunately, if both of the files involved were stored in vector format, they both would undergo a raster conversion before the overlay could be initiated.

A PROPOSED APPROACH: MULTI-FORMAT GIS

Clearly, while dual-format GIS add to our processing capabilities, they still leave room for improvement: these systems are still too closely tied to the organization of their data structures. An ideal system should be able to recognize and use data from a wide variety of sources without having to subject them to costly pre-conversions. We call such a system a *multi*-format GIS (Figure 2c). In this section we expand this theme by describing some of the conceptual differences between multi-format and more traditional modes of operation.

One way to provide this functionality is to make our GIS much more intelligent than they currently are. Multi-format systems must optimize their operations based on the relative size and complexity of the areas being considered, the formats in which the data are stored, the accuracies required, the capabilities of the system, and the nature and purpose of the operation. For example, a GIS may not need to convert two entire vector files to raster before performing a polygon overlay in one corner of the map. As a first step, the system should be able to extract only an appropriate window from each file and restrict the overlay operation to the region of interest. The system should also recognize that there is some point at which converting the entire file can be more efficient than working with only a portion of it, especially if the regions are large and convoluted. The system must determine the nature of the regions to be overlayed and take an appropriate action.

To achieve these goals, both the Function Module and the Data Module must be expanded, as illustrated in Figure 3.

DATA MODULE

Instead of having one or two dedicated databases, multi-format GIS could be linked to many different data sources, treating the information coming from each as though native to the system. Indeed, the files which are linked may originate in databases which are dedicated to other systems. An Integration Function placed in the Data Module would ensure that information is exchanged with the Function Module in a consistent manner. An Integration Function does not negate the need for some internal, *native*, file formats (which are used by the analyses functions), but it does remove these details from the user level (if desired).

The Integration Function would have two groups of operations:

Translation Operations and Conversion Operations. Translations are involved when two data sets have different formats but are structured according to the same generic spatial data model (i.e., both vector or both raster). When the files come from different data models, a Conversion Operation is performed. To provide maximum flexibility, the Integration Function would have to be designed to be able to read all of the necessary foreign data formats.

Although the Integration Function performs the same role as many of the conversion programs found as part of existing GIS, it is internal to the processing system, not part of a separate data exchange module. This means that its operation would not be controlled by the user, but by the GIS itself as it assembled the necessary data to satisfy an analysis request. The Integration Function would determine the format of each requested file (by interpreting its filename extension and/or by reading the first few lines of the file and/or by user specification) and perform any necessary conversions dictated by the nature of the operation involved (as directed by the Function Module). This totally removes any data pre-processing burden from the user: one needs only to specify which files are to be included and the system takes care of the rest. We demonstrate below that this does not significantly decrease processing efficiency. Of course, in a well-designed program, the user *could* be allowed to examine and modify the conversion parameters, if desired.

A side benefit to this approach is that the data do not need to be converted until their application is known. This helps to minimize any loss of data quality because the data would be left in their native format and converted only when required and in the manner best suited to their intended use. Thus, the Integration Function must not only be able to convert and translate data from one format to another but it should also be equipped with a variety of techniques to do so. As data then pass through the Data Module, they could be optimized for the spatial operators to be applied.

Prototype. To test the Integration Function concept, we have developed a prototype data translation program which allows us to access foreign format imagery from a second system. The program operates by interpreting the data format of the foreign format file by examining its filename extension and invoking the appropriate export routine to extract the spatial data and all available ancillary information. Our program differs from standard data import facilities by being designed to reside directly in the data stream of a GIS, not as an external module.

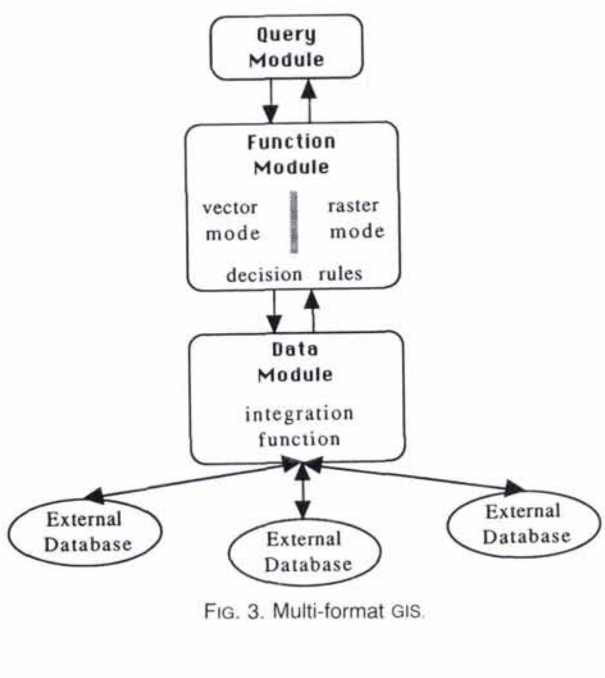

FIG. 3. Multi-format GIS.

The power of the integration software is best demonstrated by an example. Consider a map stored in a file of one format that we would like to display as a raster image using a second system's display program. In our prototype, we accomplish this with a command string similar to

export filename | display.

What is significant about this command string is that we are able to directly access files stored in one GIS's database from the analysis functions (in this case, simply display) of a completely different system. We do not pre-convert the data: they are only converted as required by an operation. We do not create any new files: the structure of the foreign format file is unchanged. We use a programming technique called *task-to-task communication* or *piping*, which allows us to pass the exported data directly to the analysis procedure, as if they originated in the system's own database. This reduces disk overhead and minimizes errors by eliminating data redundancy. If this command were built into the data access stream of a GIS, then this type of data sharing could become totally transparent to the user.

To get an indication of how system performance could be influenced by such an "in-line" integration, we completed several time trials using a 512 line by 512 pixel by 3 band image file. We compared the time used to process the file directly from the native format of our analysis function (as in uni- and dual-format GIS) with the time required for an in-line integration and analysis (as in multi-format GIS). We observed that the in-line conversion technique only required between one and four additional seconds to complete, on average (Table 2). This is well within the time tolerance of most analysts and is certainly offset by reduced data redundancy and disk storage.

Presently, our prototype is of limited scope: it can only translate data between several different raster file formats. It does, however, demonstrate that this type of data sharing is possible. We are in the process of adding several conversion algorithms to it so that we can access both vector and raster data.

FUNCTION MODULE

The Function Module must also be expanded to include a duality of GIS operations and a set of Integration Guidelines on how to apply them.

As a first step, the primitive GIS operations found in the Function Module must be twinned: they must operate with and between all data types presented to them (both vector and raster). Therefore, building on the examples in Table 1, *Distance* should also be able to answer the query: "given a pixel and a line segment file, which line segment passes closest to the pixel?"; or *Intersection* should solve: "given a raster image and a polygon file, which pixels are contained within each polygon?" This

TABLE 2. TIME TRIALS OF THE PROTOTYPE TRANSLATOR (TESTS CONDUCTED ON A SUN 3/260 USING A 512 LINE BY 512 PIXEL BY 3 BAND IMAGE)

Action	Elapsed Time (seconds)
display native file	0:02
display foreign file	0:03
resample native file	0:11
resample foreign file	0:14
3 by 3 median filter native file	0:28
3 by 3 median filter foreign file	0:32

functionality should be resolvable without file conversions. Because we already have some systems which perform the basic GIS operations in vector mode and other systems which process data in raster mode, the twinning of GIS primitives in a single system should not be problematic: the real challenge is in the specification of a set of guidelines for directing the data into either of the two processing modes.

Integration Guidelines

We have described the concepts of bringing together data from different formats in an Integration Function and the need for a duality of GIS operations. These techniques would enable us to access any number of "foreign" data sets and analyze them in either a vector or raster mode. We now must give the multi-format system the ability to choose the optimal mode of operation based on the data formats presented to it and the analysis task it is to complete. For example, at what combination of file size/complexity/accuracy requirements/program efficiency is it better to convert two vector files to raster form for an overlay operation?

We propose a set of Integration Guidelines, such as those in Table 3, to fill this gap. This is far from a complete list (it is doubtful whether a complete set of guidelines can be created before a multi-format GIS is actually constructed), but they are a base upon which we can build. The application of these guidelines could be in a set of derived decision rules, such as

if [format(file1) = format(file2)] &
[function_efficiency(format(file1)) = high]
then mode_of_operation_weight(format(file1)) = high

if [processing_window_size = small] &
[#regions_to_be_processed = small] &
[average(region_complexity) = small]
then mode_of_operation_weight(format(file1)) = high

if [planimetric_accuracy_required = high]
then mode_of_operation_weight(vector) = high

During processing, each term in the rules would be resolved to a functional equivalent. For example, *format(file1)* would reduce to "vector" or "raster." Values for operations, such as *function efficiency*, could be read directly from look-up tables derived by repeated testing of each GIS primitive function under a variety of conditions. Other constants, such as *processing window size* would resolve to values, such as 100,000 pixels, which could be derived by extensive testing. Once all of the rules for a particular operation have been resolved, the *mode of operation* weights are compared to determine the optimal solution. The Data Module would then be instructed to present data to the Function Module in the selected format.

In a well-designed system, the visibility of the implementation of such rules could be entirely selectable by the users. They could choose to have the system follow the rules without intervention, or elect to approve or modify each rule before it is executed.

Table 3. Considerations for Integrating Vector and Raster Files

- What is the nature of the data? How large an area is it? How convoluted are its boundaries? How many individual elements are in the area?
- What is the format of the data? If there is more than one data set, do they have the same format?
- What are the capabilities of the system? Can it do the desired operation? Can it do the desired operation in more than one mode?
- Is the system capable of converting between vector and raster format in several ways? What are the advantages/limitations of each capability?
- What is the nature of the operation? Can precision be sacrificed for speed? Is it more important to have an accurate area measure than a correct perimeter length?

CONCLUSIONS

One current trend in spatial information system design is towards the integration of data sets from diverse sources and in different formats. Most systems available today provide a "band-aid" solution to this problem by supplying the user with a few external conversion programs to accommodate these data. While these conversion routines are able to bring together some disparate forms of data, they tend to be an extra barrier to be crossed prior to performing any analytical operations.

Through this research, we want to examine ways of simplifying the use of GIS and IAS. We have focused our attention on data integration because it is the first, and often insurmountable, hurdle to be overcome when beginning to use a system: how to get data into a system-usable form. The prospective user, no matter how well prepared, may concede defeat before the battle has begun if he/she is faced with trying to decipher several spatial data formats.

Developments in data integration are key factors towards our goal of simplification. We have suggested how integration procedures can be automated, especially with the help of context-sensitive Integration Guidelines. To date, this has been demonstrated using raster files from several image file formats. Our next step will be to build in vector-based modules along with conversion algorithms which will transpose data between vector and raster formats, as required.

ACKNOWLEDGMENTS

This research was supported by a Centre of Excellence grant from the Province of Ontario to the Institute for Space and Terrestrial Science, and an NSERC Operating Grant to E. LeDrew. The comments made by the reviewers were quite constructive: their time and effort is much appreciated.

REFERENCES

Abel, David J., 1989. A Model for Data Set Management in Large Spatial Information Systems, *International Journal of Geographical Information Systems*, Vol. 3, No. 4, pp. 291–301.

Aronoff, Stanley, 1989. *Geographic Information Systems: A Management Perspective*. WDL Publications, Ottawa, 294 p.

Burrough, P. A., 1987. *Principles of Geographical Information Systems for Land Resources Assessment*. Clarendon Press, Oxford, 193 p.

Ehlers, Manfred, Geoffrey Edwards, and Yvan Bédard, 1989. Integration of Remote Sensing with Geographic Information Systems: A Necessary Evolution, *Photogrammetric Engineering & Remote Sensing*, Vol. 55, No. 11, pp. 1619–1627.

Estes, John E., 1981. Remote Sensing and Geographic Information Systems Coming of Age in the Eighties, *Proceedings, PECORA VII Symposium*, pp. 23–40.

———, 1985. The Need for Improved Information Systems, *Canadian Journal of Remote Sensing*, Vol. 11, No. 2, pp. 124–131.

Goodenough, David G., 1988. Thematic Mapper and SPOT Integration with a Geographic Information System, *Photogrammetric Engineering & Remote Sensing*, Vol. 54, No. 2, pp. 167–176.

Holder, Glenn H., 1988. *Selected Issues in GIS Research*, Earth Observations Laboratory, University of Waterloo, Technical Report ISTS–EOL–TR88–002.

Jackson, M. J., and D. C. Mason, 1986. The Development of Integrated Geo-Information Systems, *International Journal of Remote Sensing*, Vol. 7, No. 6, pp. 723–740.

Marble, Duane F., and Donna J. Peuquet, 1983. Geographic Information Systems and Remote Sensing, Chapter 22 in *Manual of Remote Sens-*

ing, Second Edition. (Robert N. Colwell, ed.) American Society of Photogrammetry, Falls Church, Virginia, pp. 923–958.

Marble, D. F., H. W. Calkins, and D. J. Peuquet, 1984. *Basic Readings in Geographic Information Systems.* SPAD Systems Ltd., Williamsville, N. Y.

Peuquet, Donna J., 1984. A Conceptual Framework and Comparison of Spatial Data Models, *Cartographica,* Vol. 21, No. 4, pp. 66–113.

Piwowar, J. M., E. F. LeDrew, and D. J. Dudycha, 1990. Integration of Spatial Data in Vector and Raster Formats in a Geographic Information System Environment, *International Journal of Geographical Information Systems,* in press.

Saalfeld, A. J., and R. T. O'Reagan, 1985. Streamlining Curve-Fitting Algorithms that Involve Vector-to-Raster Conversion, *Proceedings, AUTOCARTO 7,* Washington, D.C., pp. 491–497.

White, Denis, 1985. A Generalized Cartographic Data Processing Program, *Proceedings, AUTOCARTO 7,* Washington, D.C., pp. 560–569.

Inter-Society Color Council Call for Papers

The Inter-Society Color Council and the Technical Association of the Graphic Arts will cosponsor a conference on the difficulties encountered when comparing images presented in different media that are intended to simulate each other or another image. It will address such topics as color space transformations, ambient conditions, viewing geometry, surface properties, and adaptation. The conference will be technical in nature and will consist of invited and contributed papers emphasizing the exchange of information and discussion. Papers of a commercial nature will not be accepted.

Contributed papers will consist of a 30-minute presentation. Those wishing to contribute should submit a title and abstract (not exceeding 750 words) by March 1, 1991. Authors will be notified of acceptance by June 1, 1991. Send title and abstract, including name, affiliation, address, and daytime telephone number to:

Milton Pearson, RIT Research Corp., 75 Highpower Rd., Rochester, NY 14623
Telephone 716-475-5290; FAX 716-475-2361