Geographic Information Systems and Remote Sensing Future Computing Environment

Nickolas L. Faust
Georgia Tech Research Institute, Georgia Institute of Technology
William H. Anderson
Center for Mapping, The Ohio State University, Columbus, OH 43210
Jeffrey L. Star
Remote Sensing Research Unit, University of California, Santa Barbara, CA 93106

ABSTRACT: When considering the research needs related to the integration of remote sensing information into a geographic information system (GIS), one of the factors that must be considered is the state-of-the-art in computer technology. Operations that once took hours or days on mainframes can now be performed in real time or near real time on microcomputers and workstations. In addition, many operations that were once considered impossible because of computing limits can now be revisited and, perhaps, implemented. This paper attempts to describe the essential functionality required for remote sensing and GIS integration, evaluate the state of the art in computing technology, articulate current impediments to remote sensing and GIS integration, and define research issues that should be attacked to further this integration.

INTRODUCTION

THE INTEGRATION OF IMAGE DATA INTO geographic information systems (GIS) is one of those great ideas whose time has come. GIS systems have become accepted as a standard way of handling geocoded data sets and performing analyses on those data for a multitude of applications. Historically, the major costs associated with the implementation of a GIS system are neither the necessary hardware or software, nor the personnel costs associated with the management of the system. The major costs of GIS have normally been in the gathering of the geocoded data. Recently, remote sensing (RS) images have been shown to be a natural and cost effective means for update of GIS data sets (Gernazian, 1989; Tilley, 1989). As we enter into the 1990s, the technologies for data entry, data storage, data access, and data analysis will progress such that a new version of GIS will be created with radically different constraints. The concept of real time modeling and interactive three-dimensional query will change the way in which we deal with spatial data. These new insights will in part be due to major advances in computer technology and in part be due to the institutionalizing of the GIS/RS process into everyday decision making (Lang, 1989).

NCGIA Initiative 12, Integration of Remote Sensing and GIS, is focusing on the impediments to rapid acceptance of GIS technology into the research, applications, and operational environments that exist worldwide. This overall topic is broken down into several discrete directions that will be reported on in this same volume (Estes *et al.*, 1991). Lauer *et al.* (1991) report on the impediments to the institutionalization of GIS and Remote Sensing Integration Technology. Data Integrity and Error Sources are investigated by another panel (Lunetta *et al.*, 1991). Data Structures and Data Access and their effects on acceptance of GIS/Remote Sensing integration are studied by Ehlers *et al.* (1991). Processing Flow limitations and impediments are discussed by another panel (Davis *et al.*, 1991).

OBJECTIVE

We are currently in the midst of massive changes in the computing environment that will affect all phases of GIS/RS integration. In this article, we will attempt to

- functionally identify the processes used in GIS/RS analysis,
- identify current impediments in the computing environment to rapid GIS/RS acceptance,
- identify trends in computing technology that will directly affect the GIS/RS systems of the future, and
- suggest research thrusts that might be pursued to overcome these obstacles.

Techniques for handling the interactions between related spatial variables were demonstrated by Ian McHarg (McHarg, 1960) in the 1960s using plastic overlays to show coincidence of geographic location for map based conditions. Individual conditions would be coded by placing a clear plastic overlay over a base map and coloring the plastic for areas for which the desired condition was satisfied. For example, proximity to roads would be shown by a buffer region about all major roads that were visible on the map. If land-use information were available on the map, another condition might be to show all areas for which the existing land use is non urban. A vegetation map might be used to show the location of environmentally sensitive areas. Multiple types of maps could be used for an analysis if they were drawn at the same scale and covered the same area. Each desired condition would have its own overlay with the colored areas representing the selected criteria. Once all the potential criteria were selected and colored in on the plastic overlays, an analysis may be performed by simply placing all the clear overlays in a stack on a light table. The areas that remain clear satisfy none of the desired conditions, and the darkest areas identify the areas that meet all or most of the selected criteria. The geographic areas so identified may then be transposed onto the base map for final presentation. Many of the techniques for GIS analysis operate in the same manner as this manual method. The only difference is that computers have made this type of analysis cheaper and easier to use than ever before. Planners have used similar techniques for years for development of master plans and evaluation of environmental impacts of proposed human activities. Master plans and environmental impact studies often involved years of study, resulting in the vendor's or consultant's best estimate as to the appropriate location for facilities. If the plan was not acceptable, or if other criteria needed to be considered, another long study might have to be commissioned. The use of GIS in the 1970s has provided a method for evaluation of numerous alternatives within a planning process and for actually involving the all concerned parties in the development of an appropriate strategy for implementation.

Modern GIS computer analysis began in the middle 1970s at the Harvard School of Landscape Design with Carl Steinitz. A

PHOTOGRAMMETRIC ENGINEERING & REMOTE SENSING,
Vol. 57, No. 6, June 1991, pp. 655–668.

0099-1112/91/5706–655$03.00/0
©1991 American Society for Photogrammetry
and Remote Sensing

cell based model for growth management was developed under NSF funding for the Boston metropolitan area. The model, called GRID, was a complex model of spatial parameters, including population distribution, that was run in a batch mode on an IBM mainframe. Given certain spatial variables such as the implementation of zoning laws, the model was used to allocate projected population growth with a gravity distribution within Boston and its neighboring areas. The model was used in teaching planning and resource allocation techniques to graduate students in the School of Design. Because of the batch nature of the model, and its dependence on complex interactions, GRID was difficult to use in a one- to two-year graduate program. A student revolt led to the design of a somewhat interactive geographic analysis set of tools called IMGRID by David Sinton which allowed the exercise of the geographic analysis functions without being tied to the complex GRID model. Prior experience with hand overlay models (Steinitz, 1974) led to definition of basic functions necessary for IMGRID. GRID and IMGRID form the basis for most cell based GIS analysis systems. Early students in the Graduate School for Design include Jack Dangermond and Chuck Kilpac of Environmental Systems Research Institute (ESRI), Dana Tomlin of Yale, and Bruce Rado, Lawrie Jordan, and Steve Sperry of Earth Resources Data Analysis Systems (ERDAS). ESRI developed a mainframe and minicomputer based grid analysis package and the first topologically based vector based GIS. COMARC, Inc. developed a competing polygon and vector overlay GIS system that was extensively used in the forestry industry. Oddessy, developed by the Harvard School for Computer Graphics was also developed as a vector based GIS. Dana Tomlin developed the Map Analysis Package (MAPS), and ERDAS developed the first microcomputer GIS and image processing system. ESRI and ERDAS developed the first integrated capability for image processing, grid GIS analysis, and topological vector processing. Other packages, such as the Map Overlay Statistical System (MOSS), were developed with a basis of software developed at the State University of New York.

All of these systems were initially developed on mainframes, minicomputers, or microcomputers with a Von Neuman, or serial, architecture. As computer power has increased, GIS systems have become oriented toward a workstation environment where single workstations have the power of full mainframes of the past, and networks of workstations allow resource sharing and multiprocessing. The implementation of GIS systems to take advantage of the vast resources of the multiprocessing environment has yet to be fully realized. Supercomputing systems are currently implemented that perform complex viscus flow modeling, finite element integration, and three-dimensional stress and heat flow modeling, but very little serious attention has been paid to the prospects for parallel implementation of GIS analyses.

Computer technology changes significantly from the 1960s to the late 1970s. Mainframes were being supplemented by minicomputers, and microcomputers were initially being introduced into the popular market. By the mid 1960s minicomputers (PDP 1, IBM 650) were being introduced by Digital Equipment Corporation (DEC) and International Business Machines (IBM) (IEEE Scientific Subcommittee, 1989). Scientific applications were becoming viewed as inherently different from business applications and new systems were being developed to meet this market. Personal computers became available using the Zilog Z-80 and the Motorola 6800 chips in the late 1970s, and IBM introduced its personal computer in the early 1980s. The introduction and wide acceptance of the personal computer led directly into the concepts of networks and shared access which represent the state-of-the-art in today's lower power computing arena.

Also during the 1970s, computer technology developed the first of the parallel processing (Illiac IV) and vector processing (CDC Star 100) systems. These systems signaled the emergence of supercomputers as a new class of computer. Techniques experimented with in the 1970s have led to commercial hardware and software implementation of parallel and vector processors. Mini-supercomputers, a lower cost and sometimes hardware specific implementation of supercomputer principles, have become available in the 1980s for intensive computational power at a workstation.

Spatial display of GIS and remote sensing information has been another area in which computer technology has advanced greatly in the last two decades. In the early days of GIS and remote sensing, the only display media for spatial information were line printer maps and plotters, with subsequent manual coloring. Advances in computer graphics for computer aided design and computer aided manufacturing (CAD/CAM), image processing (IP), and scientific visualization have provided a sophisticated method for interpretation and display of geographic information that is only recently being employed by GIS/RS.

IDENTIFICATION OF FUNCTIONS PROVIDED IN GIS/RS INTEGRATION

GIS systems in today's market may be conveniently broken down into two types that are indicative of their inherent data structure. Raster GIS systems have multiple variables which normally are coded into cell or raster grids. Attributes for variables are normally coded to a grid cell value with a lookup to extended attributes. While data may be captured in either a raster or vector manner, data analysis occurs in the raster domain. Examples of such raster systems include ERDAS by ERDAS, SPANS by TYDAC, GRASS by the U.S. Army Corps of Engineers, and MAPS by Dana Tomlin.

Vector systems, on the other hand, capture data in vector format, analyze data in vector format, and produce output in either vector or raster format. Generally, the data are either stored as whole polygons, lines, and points, or in a topologic structure of arcs and nodes. Attributes are coded to each basic representation and are generally stored in a relational database. Examples of current vector based GIS systems include ARC/INFO by ESRI, MOSS by Autometrics, and TIGRIS and MGE by Intergraph.

The integration of remote sensing information into a GIS occurs naturally in a raster GIS because both data structures are approximately the same. Integration into a vector system requires somewhat more effort, but it has been recently achieved by several GIS and remote sensing vendors, at least to the extent of updating vector information by using an image as a backdrop for vector editing (Sperry, 1989; ERDAS, 1989).

Basic image processing and analysis tools such as geometric correction and pattern recognition are expected to be available to preprocess remote sensing information into a form in which attributes are assigned to raster data instead of raw data values from spectral bands of multispectral imagery.

Once the remote sensing information is in a form that can be analyzed as a part of a GIS, systems with both data representations apply a number of the same functions, even though they are implemented in a significantly different manner. We will consider each of the basic functions of a GIS and discuss the differences in the implementation for vector and raster systems (Davis, 1990).

INFORMATION DISPLAY

Display of GIS/RS information is, of course, the most visible part of a GIS/RS system. The information must be provided to a user in a concise and easy to understand manner so that the user may be able to comprehend the potential complex relationships that may exist between multiple spatial variables. The techniques for GIS/RS display will tend toward using all of the capability available in a graphic display system.

Display of raw and processed remote sensing data normally occurs in a "true color" mode in which raster images for three different variables (in this case, three spectral bands) may be assigned independently to the individual color guns of a display unit. Each variable normally can have up to 256 values (8 bits) in grey scale. Some RS imagery is captured with 10 bits (1024 levels) or 12 bits (4096 levels). A user defined, hardware-scaling look-up table (LUT) may be used to scale the input data into the desired 0 to 255 range. If the data occupy only a part of the potential 0 to 255 range, the LUT may also be used for dynamic enhancement of the image. Some of the current workstations will allow display of true color imagery but do not allow the passing of the data through LUTs. Because a large part of remote sensing analysis involves the dynamic enhancement of true color images, this architecture does not satisfy the GIS/RS users needs.

Raster GIS data variables may also be displayed in a true color mode, with the visual integration of the three primary colors of the display giving dynamic information showing a three-layer analysis.

Normally, however, GIS raster variables are displayed in a pseudo-color mode in which each GIS data value may be assigned a color value that is taken from either a set color palette or a user selectable combination of grey scales on the red, green, and blue primary color guns.

Because a modern graphics display system is a raster oriented device, vector data must be converted "on the fly" into the raster domain for display. This function may be provided in software or hardware. Hardware implementations of vector plotting, clipping, filling, and shading generally operate in a display list manner in which the software display program loads vector data from a database into the local memory of the display which can be operated on by dedicated hardware processors.

In some current graphics displays, an overlay plane is available as well as the three 8-bit planes of the true color image. In most cases, this overlay plane will have from 16 to 256 colors which will mask the underlying image or GIS file that has been placed into the true color memory. This overlay plane may be used to display vector information overlaid on true color remote sensing data or other GIS data that has been rasterized. The ability to simultaneously view image data and GIS vector information is crucial to the integration of GIS and remote sensing information.

When such an overlay plane is not available, extensive manipulation of images and display bits within a true color display system is necessary to provide a similar capability. Often this manipulation results in degraded display times. For GIS/RS, the overlay plane is clearly desirable.

Attribute Handling

The simplest case of handling identifiers for raster GIS variables is a code that identifies the value of a particular cell in the raster data set with a character string. Attributes for vector data are more complex in that single entities such as vectors may have multiple codes depending on directionality or its relationshp to other neighboring entities. This coding is generally developed at the time of data entry by means of a text manipulation program and is usually carried along with the data variable in the same or an auxiliary file.

When analysis is performed on a number of GIS variables, the resulting attributes of the analysis file should indicate which variables have been combined, and which functions were applied to perform this combination. The resulting attribute may then be a simple concatenation of the input attributes, or it may be a user defined text attribute that describes the process. For example, the combination of soil, vegetation, slope, and land ownership may result in an analysis called "suitability for siting of land fill areas" or it may contain a history of all processes that were applied without a meaningful description of the final result.

One of the basic GIS functions is the ability to display GIS data that satisfy a user selectable set of attributes. This capability assumes the ability to search through all GIS data and select only those GIS variables for display that satisfy all of the given criteria. Attributes may be handled in a flat file manner in which the application program must perform an exhaustive character string search through concatenated attributes. This function can be performed relatively easily for a small number of variables and simple analyses, but may become impossible, or too time consuming for complex analyses. Next, attributes may be handled in a linked list manner, in which a pointer is kept in the analysis file that points to all input data sets as well as the input analysis attributes. This method is reasonably efficient, but requires the GIS/RS system to keep track of a myriad of pointers, as well as have the capability for deleting, editing, and updating these pointers. A third method uses the tools available with a commercial relational database management system (RDBMS) to keep track of multiple attributes for GIS variables and resulting analysis variables (Sibershatz, 1990).

New techniques developed in machine vision and artificial intelligence applications are being investigated for storage and manipulation of GIS entities (points, lines, polygons, attributes, etc.) as objects in an object oriented database management system with an inheritance hierarchy (Ehlers, 1991). It is yet unclear how this procedure, which requires identification of features that are to be treated as objects, would interact with the raw data variables such as remote sensing imagery which provide continuous grey tone data values.

Vertical Analysis

In a raster based GIS/RS system, a vertical analysis is implemented easily. A vertical analysis allows the user to combine numerical attributes of a number of independent GIS/RS variables into a resulting analysis variable. Each input GIS/RS variable is preprocessed with user input to create a numeric value that represents each raster cell's relative importance to others within its own GIS layer for the desired analysis.

Once the numeric scaling has occurred for each input GIS/RS variable, a weighted sum of the variable's values for each raster cell is calculated and finally scaled by the user for final output. As mentioned above, the attributes are normally either lost with the only attribute being defined by the user for the analysis output, or they are concatenated to form a more complex textural description of the combination process. In some cases, both final attribute user definition and concatenation occurs.

The vertical analysis above may be described as a "point" process in which each raster cell within a GIS/RS database is considered without respect to adjacent values with the same variable or other variables. Image processing techniques utilize similar "point" functions. A "point" function allows the value of each raster cell in one variable to be processed with a mathematical function with respect to the same raster cell in another GIS/RS variable. The mathematical operations may include "+", "−", "/", "*" as well as any other legitimate function such as sin or cosine, logarithms, or exponentials. In the case discussed above, the operator was "+".

Ratios between GIS/RS variables can be handled by "point" functions with an operator of "/".

Masking of GIS/RS variables may be handled by preprocessing the values within one variable to have binary values of 0 and 1 and then performing a "point" operation with an operator of "*".

Logical operations may be handled in a "point" function mode with additional operators of "AND," "OR," "XOR," and "NOT."

Vertical raster functions are functionally identical to normal image processing functions used on remote sensing data sets. Ratios and linear combinations (weightings) of multispectral bands are often used functions for enhancement of multispectral imagery.

Vector GIS/RS operations for vertical analysis are substantially more complex than raster vertical operations. To perform vertical operations on vector data, a set of algorithms for point, line, and polygon overlay must be executed. Logical functions such as union and intersection of polygons, point inside polygon, vector intersection, and clipping must be performed on all data entities within each of the variables with respect to all entities in the second data variable. Clipping and windowing functions hopefully reduce the complexity of the analysis, but normally, substantial portions of each data variable must be processed. Most of these functions involve floating point logic, and are thus extremely time consuming. For analyses in which two simple GIS/RS variables are overlaid, the process may go fairly fast. However, after a number of combinations have been performed, the number of possible intersections for subsequent analyses may become prohibitive with the current implementation of technology. For simple variables, the storage of the GIS/RS variable in vector format provides a large compression in the amount of memory and disk space. For the storage of complex variables and the results of complex analyses, however, the vector file will often exceed the storage necessary for storage of the file in raster form.

Attributes in a vector system may be stored either in a linked list form, which again must be managed by the application program, or they may be stored in an RDBMS as discussed above. A typical inquiry of such a system might be: "find all areas with at least 60 percent pine forest, within Jasper County, containing a stream, with a size of 10 acres or more, and a cost of less than $2000 per acre." Polygonal areas may be defined with multiple attributes, such as the percentage pine cover, ownership, cost per acre, soil type, etc. These attributes may be searched along with multiple attributes from other GIS variables to find candidate areas. The solution to such a request requires several polygon overlays as well as a search into a relational database management system for attributes satisfying the request.

If data are stored in an arc/node representation, then directionality and right and left attributes must be dynamically interpreted to form polygonal areas, and to access their attributes. Access to the attributes may require a search command with the supporting RDBMS.

Proximity Analysis

Proximity analysis is one of the more powerful tools of GIS/RS systems. A proximity variable is one whose values represent the spatial distance of any point from a data value specified as a search criteria. For example, in a land-cover classification, a proximity analysis may be required to show the distance away from all water classes. The resulting GIS file will not have anything to do with the coincidence of a test pixel with any other land-cover class, but it will include the Euclidian distance of any point from the specified criteria, in this case, the existence of water in the land-cover classification.

This technique is extremely useful in assessing the relative spatial importance of a specified search class with respect to user specified criteria for an analysis. For example, in a siting problem, nearness to a water body may be an important factor. The best circumstances might be those areas directly adjacent to a water body, but the importance of a site might be only marginally less if it is less than 100 metres from water. It also might be acceptable if it is within 200 metres, because site modifications might be easily made to provide direct water access.

Proximity to certain criteria is often used as a weighting criteria in a subsequent vertical analysis.

Vector proximity analysis usually entails the definition of a buffer zone area defined by lines and/or polygon features showing areas a certain distance away from a specified criteria. The buffer zone may be used as a mask in a polygon overlay of vector data.

Neighborhood Operations

Neighborhood operations are those functions in raster analysis in which the resulting value or class for a cell at position (x,y) depends not only on cells at the same (x,y) location from different variables, but also on the cell values within a certain distance of the (x,y) location in all data variables. This neighborhood in raster analysis may be symmetric (a 3 by 3 window of cells surrounding the (x,y) cell, a 7 by 7 window, etc.) or it may be asymmetric (a 3 by 10-cell area). A neighborhood operation may be as simple as computing the sum of all values within the designated box, or it may be as complex as the dynamic computation of the mean, mode, median, maximum, and minimum of the box and zeroing values less than zero and greater than 10. The neighborhood function may also be performed on logical variables to indicate presence or absence of a specific criteria.

A neighborhood function may create more than one output variable that represent different functions applied to the window. For example, slope and aspect may be computed using the same window on digital elevation data. Some common neighborhood functions that are applied to GIS/RS data sets are

- average value
- minimum value
- maximum value
- mean value
- median value
- mode value
- AND, OR, NOT logical functions
- slope
- aspect
- coincidence
- absence
- diversity

Neighborhood operations are also common in image processing. While the functions are approximately the same, and the implementations are equivalent, the names of the functions often differ. For example, texture analysis of images uses neighborhood functions to define first-, second-, and third-order moments that can be thought of as higher derivatives of an image surface within the specified window.

Convolution filtering is another enhancement process used in image processing that uses a neighborhood window approach. A user specifies the size of his window (3 by 3, 7 by 7, etc.) as above; however, the user is then required to input a set of coefficients for each of the n by n elements of the neighborhood. The convolution algorithm then computes the resulting output cell's value as the sum of each cell's value within the window multiplied by the user specified coefficient for that (x,y) location within the window and then divided by the number of cells in the window. For example, if a user were to specify all coefficients to be equal to 1's, then the resulting value would be the average cell value within the window. This result would be called a low pass filter in image processing terminology. Low spatial frequency information within the image is saved and high frequency information is thrown away.

Low spatial frequency information may be demonstrated by a grey scale image in which the value in the image varies slowly from the left side of the image to the right. Thus, the grey scale only goes through one cycle of black to white in the entire image. In contrast, a high spatial frequency image is one with many black to white reversals across the image. The highest spatial frequency image possible is represented by a checkerboard image that changes from black to white in every adjacent cell. This highest frequency in an image is called the nyquist frequency, and the spatial frequency is $n/2$ where n is the total number of pixels across an image. Most images contain neither all low frequency nor all high frequency information within them. Most images contain many spatial frequencies of information, and the above convolutional filtering techniques may be used to enhance certain spatial frequencies and suppress others. Normally, high frequency enhancement is applied to remote sensing digital data to sharpen edges and make the image more interpretable.

By specifying different coefficients, a high pass filter may be generated. This filter saves only high spatial frequency information at the expense of low frequency information. By using other variations of coefficients, other filters such as band pass and directional filters may be implemented.

Because most polygon overlay operations are Boolian in nature, there does not seem to be an equivalent function to neighborhood processing in raster GIS and remote sensing images. As mentioned above, however, proximity masks may be used to extract regions within a specified distance of a selected value.

Time Based Operations

In many of the more sophisticated GIS/RS applications, time series events form a methodology for studying changes over a period of time. Because a GIS system is based on the same spatial coordinate system regardless of the acquisition date of the data set, procedures may be used to study spatial changes over a period of time.

The simplest application of time based analysis is the use of data from two different time periods to detect differences that are present from one date to another. This technique, called "change detection" in normal remote sensing analysis, can be used with imagery or GIS data entered from base maps of varying dates. Change detection is simply the identification of the fact that the map/image has changed spatially over time. A more detailed form of change detection allows the quantification of how much change has occurred over the period, and which GIS classes have changed into which new classes. For example, in remote sensing analysis, it is often advantageous to know what classes have had a change, and into what other classes are the changes placed. This change detection can also be used as a check on classification accuracy, because the probability of going from one class to another (i.e., forest to urban) may be high in one case but very low in another (i.e., urban to forest).

The more complex case of time series analysis has to do with trend analysis and prediction of a particular spatial distribution of GIS/RS classes given the prior spatial distributions of classes in a number of preceding data sets or years. In this case, it is insufficient to determine that a change has taken place. It is necessary that one develop a model for how these changes have occurred over time and to make the model so robust that it can predict future results based on specific assumptions in the model parameters.

Modeling can be thought of as a complex association of rules that allow the prediction of future events based on the past history of events. Spatial modeling may involve a high level decision making strategy which may rely on political, economic, and resources based criteria. Most spatial models employ the use of a GIS/RS system to manage the geographic information and the spatial interactions between information data sets. Current models may predict population growth and distribution, watershed runoff, or economic trends based on the availability of resources. The most complex model of them all, in today's use of spatial data, is the modeling of the global environment. The state of the environment has so many variables that operate in three dimensions as well as time, the fourth dimension, that it is unlikely that man will be able to accurately produce a successful predictive model in the near future. On the other hand, research into well defined three-dimensional time series analyses will continue to provide insights into the environment that could some day be the basis of a fully predictive model of the global environment.

Vector Raster Conversion

Vector to raster conversion is traditionally the easier of the two conversions between data representations. Normally, vector to raster conversion occurs by simply performing polygon fill logic on individual polygons, lines, and points within a database and writing the raster version of a polygon into a raster image buffer. If all polygons and all of the raster image buffer fit into memory, then the task is efficient. If, on the other hand, enough memory cannot be allocated, the procedure must operate on a polygon by polygon basis and an image block basis. Virtual memory normally will eliminate the need for the user to have to manage this partitioning, but disk space accesses will invariably cause longer run times.

One of the principal problems with vector to raster conversion is deciding what to do with the multiple attributes that may have been assigned to the polygons in the analysis or data entry phases.

Raster Vector Conversion

Raster to vector conversion normally requires a number of user intensive steps to define polygons that will fit into a topologically structured vector network. Initially, clustering or clumping GIS algorithms may be used to define contiguous regions with the same GIS/RS value. The boundaries of the regions may be formed by segmentation algorithms that normally operate on the detection and chaining of edges, or the growing of regions within the raster data set. The perimeter of the regions thus constitutes the initial "polygon" region. Algorithms for smoothing, to remove the jagged pixel edges, and thinning, to remove short vectors along a consistent path, must be run iteratively until the final vector file is acceptable. Next, the assignment of attributes to vectors, points, nodes, etc. must be assigned in at best a semi-automated manner. Operator experience in photointerpretation is often key to a successful raster to vector conversion. Spatial relationships that are often hard to define analytically can often be detected by manual interaction and then combined with automatically detected region definitions. The process from raster to vector representations is normally an extremely difficult process. No current totally automated process is known by the authors that can perform without exhaustive preprocessing of the raster input data sets.

COMPUTING TECHNOLOGY

Advances in computing technology have been so rapid in the last decade, that it is difficult to evaluate the many alternatives offered in today's and tomorrow's market for high speed interactive computing. The above functions describe the algorithms and techniques which are at work in the current generation of GIS/RS systems. The next decade of computing promises much more in the individual power of computers and the synergistic capability of distributed networks. Jack Dangermond, President of ESRI, predicted that, for geographic analysis in the next 10 years, CPUs capable of processing 1000 MIPS will be common in large organizations, and greater processing power will be packed into compact, less expensive systems (Dangermond, 1988). It is possible that, with the emerging technologies in the computing, user interface, and visualization areas, the concept of how GIS/RS analyses are performed will be totally redefined.

Single Processor Computation Power

One of the most dramatic areas of change in the overall computing environment that affects GIS/RS systems is in the raw computation power of computing systems. Initially we will look at changes that have occurred in single central processing unit (CPU) architectures, and later we will expand to consider multiprocessing and special purpose architectures.

The 1960s saw the development of a number of mainframe computer architectures with sophisticated supporting operating system and applications software. These machines were designed to respond to needs both from the scientific community (NASA, etc.), the oil exploration community (seismic analysis), and the Department of Defense as well as needs from a growing business comunity (IEEE Scientific Supercomputer Committee, 1989). The speed and power of these computers lay in the extensive instruction sets, and the ability to perform complex instructions in the lowest level clock cycle time for the computer hardware. The cycle time was also related to the speed of access of memory. A major advantage of a mainframe was that a single computer system could support a large number of users with little degradation in service. Multi-user support requires a sophisticated operating system and a management strategy for setting priorities between users. Large computational loads and high priority processing on such systems could cause the system to be ineffective as a multi-user system.

Differing physical implementations of memory were being developed, with the standard random access memory in the 1960s being core memory. In the 1970s metal oxide semiconductor (MOS) memory was introduced with faster access times but only volatile storage (the memory had to be refreshed by the computing system). Static random access memory (RAM) did not have to be refreshed, but was slower and more costly than the MOS implementation.

The density at which memory can be stored in computer chips has increased radically over the past few years; 64 kilobits, 256 kilobits, and now 1 megabit may be stored on a single chip. This storage density also translates to faster access times for memory, reducing one of the limitations on computing speeds.

Different vendors, throughout the 1960s and 1970s, produced general purpose computers with different word lengths and instructions sets. For example, the IBM 7000 series and the 360 and 370 series used 32 bits as its internal word length, the Harris series used 24-bit word lengths, the Control Data Cyber series used 60 bits, and the Univac 1100 series used 36 bits. The word length selection was made by each vendor in the design of computing specific hardware to optimize the number of instructions that could be executed per second, and the capability of each individual instruction. One standard by which the raw power of computer systems may be gauged is the number of million instructions per second (MIPS) that can be executed by that machine. Data access is also normally handled in terms of data words with the same number of bits. In some cases, the ability was created to execute an instruction and fetch data in the same cycle time. Each computer manufacturer had a different number of storage registers and accumulators that acted as temporary storage while a complex instruction was being executed. These strategies for hardware design and implementation and high power software environments led to an abundance of proprietary systems with little commonality.

Minicomputer systems also became available in the 1960s in response to the high cost required for mainframe processing. While the cost was less than mainframes, the performance also was less. Early minicomputer systems such as the DEC PDP 8 used an 8-bit CPU, with later generation minicomputers such as the DEC PDP 11, Data General Nova, and Hewlett-Packard 3000 using a 16-bit CPU. The shorter word lengths resulted in less complex instructions that could be executed in one clock cycle of the system. More complex instructions could be implemented in multiple cycles. In the mid to late 1970s minicomputers expanded their power, speed, and software and hardware sophistication by including 32-bit CPUs in minicomputer systems.

DEC's VAX and DG's eclipse/MV systems were developed in the early 1980s with a speed of approximately 1 million instructions per second, which was approximately equivalent to the larger mainframe computers being used for scientific tasks.

In the 1970s the market for a small computer, known as a microcomputer became viable, first for computer hobbyists, and later for business and scientific applications. Mainframe architectures and, later, minicomputer systems had been introduced with relatively sophisticated operating systems and high speed serial processing. The microcomputer, first an 8-bit CPU (Zilog Z80, Motorola 6800, and Intel 8086), was intended to have a simple operating system and a relatively slow processing rate (<0.1 mip). As time had progressed into the 1980s, however, microcomputers have extended their capabilities with 16- and 32-bit CPUs. The simple operating systems such as CPM have given way to more complex and general purpose ones. The integration of multiple functions into one computer chip, a microprocessor, caused an explosion in the system development. The MicroVAX system from DEC achieved approximately the same speed as the VAX 11/780 series machine, but at a fraction of the initial and ongoing maintenance costs.

During the 1980s we have seen a dramatic reduction in the differences in capabilities of mainframe, minicomputer, and microcomputer systems. Minicomputer systems have achieved mainframe speeds and most of their capabilities, and microcomputer systems are becoming the focus for a distributed processing strategy that is altering the whole concept of computing.

It should be pointed out that MIPS is only a rough measure of comparison of the speeds of different computer systems. The computer's ability to perform computations along with memory accesses and reading and writing to input/output (I/O) devices is what one likes to measure for a particular application code. Other standards currently exist which test the computer's speed on various mixtures of computation, memory access, and I/O processes. For problems needing intensive floating point calculations, a more realistic standard is MFLOPS, which is discussed below.

In 1977 Apple introduced its first entry into the microcomputer market with an easy to use, graphically oriented system which became popular because of its power and low cost, especially in education.

The major change in the acceptance of the microcomputer that propelled microcomputer systems into high visibility was the introduction in 1981 of the IBM Personal Computer based on an Intel 8086 computer chip. Almost all microcomputers became known as PCs, and the market penetration into all walks of life was initiated. IBM extended the PCs capability initially with the Intel 80286, a 16-bit processor, and then the Intel 80386, a 32-bit processor. Motorola products in this time period included the M68000, the M68010, M68020, and the M68030. These chips became the cornerstone of a number of third-party microcomputing systems that pointed toward a tailored application environment, or a sophisticated development environment.

In an about face from the trend of having longer word lengths to give more power and speed, the reduced instruction set computer (RISC) was introduced in 1986. The RISC computers achieve high speeds by optimizing the implementation of only a few instructions of the more complex systems. A 32-bit word length was used, then, to provide faster data access, and to combine multiple instruction executions in a clock cycle. The state-of-the-art in speed for single processor systems are the MIPS R2000, R3000, R4000, and R6000 RISC chips used in Silicon Graphics and STARDENT products as well as the SUN SPARC RISC chip. Depending on the clock speed for these chips, these RISC processors show a performance of between 8 and 30 MIPS. Most of the workstations in today's market use these two chips as their basis. IBM, on the other hand, has recently developed its own RISC chip with very attractive speed figures of between 30 and 50 MIPS. Other vendors are also in the process of releasing similar speed systems.

Thus, in the time span between the development of the VAX 11/780 minicomputer and the present, we have gone from a realizable speed of 1 MIP to approximately 50 on current generation single CPU workstations.

New generation serial processors include the Intel dedicated graphics processor called the 82786 and its follow-on general purpose i860. A graphics processor seeks to offload graphics processing from the host into the high speed special purpose processor. The i860 product boasts 33 MIPS and 66 MFLOPS and is today being imbedded into many imaging and graphics products (Keller, 1990; Wilson, 1990). Because of the speed of the new generation CPU chips, new memory technology was needed to prevent memory bottlenecks. The video RAM chips allow direct refresh of video screens from solid state memory, and take the load of memory management from the CPU.

Several companies have invested in the development of their own custom silicon processors to achieve greater speeds. Silicon Graphics developed special purpose geometry and shading custom processors to boost performance in scene rendering and graphics (Baum, 1988). Vitek developed its image computer in silicon to provide the processing power of 300 MIPS. The system is programmable and may be used for many applications, from image processing to scene rendering.

Vector Processing

Vector processing has been the basis for most "supercomputer" systems in the last decade (Cheng, 1989). Scientific computing has a need for fast computation using floating point or greater precision. A vector processor is generally a single controlling processor which sequences a long data vector through a number of "pipelined" stages. A pipeline operation, as the name suggests, is the process by which a complex operation is broken into a number of independent sub steps that can be implemented sequentially. Each step in a pipeline operation may be handled by a dedicated processing element (an adder, multiplier, etc.) and the results passed as input directly to the next processing element. Each step within an operation such as a vector multiply may also be broken down into simpler functions such as fetch, add, and store. Each operation in this sequential process has an inherent execution time, and the total time for processing the first element of a vector is the sum of the individual execution times. However, once the first vector element has exited the first sequential step and entered the second step, the second vector element is entered into the first step. After the sequential pipeline is full, the time for processing a vector element is equal only to the time taken by the longest of the individual sub steps. For long vectors, therefore, the pipeline processor gives significant speedups. For short vectors, however, the speedups may be much less and, in some cases, may not justify the use of a vector pipeline operation.

Vector pipelining assumes that the same operation is being applied to a large amount of data. The control and execution of the next instruction is sequential in nature, but the CPU must know whether the last batch of data has passed through the total pipeline. A timing interrupt or message passing strategy must provide this information.

Vector Supercomputing

Vector pipelining was introduced in the late 1960s and early 1970s on the Control Data STAR 100, the Texas Instruments Advanced Scientific Computer -TI-ASC, and the Cray-1 (Cragon, 1989; August, 1989). These systems were legitimately known as supercomputers (Rau, 1989; Jones, 1989). A performance measure for floating point operations based on a set of computer programs known as Linpack was developed by Dongarra (1987) to measure the effectiveness of such computers with the measure computed in millions of floating point operations per second (MFLOPS). The Cray 1S supercomputer was evaluated as having a performance of 12 MFLOPS. Current supercomputer vector architectures have greatly expanded the power of floating point computation, with the Cray X-MP having a performance measure of 235 MFLOPS per processor with up to four processors. Other competing supercomputer systems have the performance measures listed in Table 1 (IEEE Scientific Supercomputer Subcommittee, 1989).

New systems in design will have up to 5.5 and greater MFLOPS performance per processor; however, these supercomputer systems still fall into the category of "big ticket" items with price tags of 10 to 30 million dollars per system.

The currently available workstations, with 30 to 50 MIPS, will normally have a floating point performance of between 4 and 10 Linpack MFLOPS. These are the systems that will be applied most directly to GIS/RS problems in the near future.

Vector Array Processors

One method that can be used to add more floating point performance to a workstation or stand-alone CPU is the addition of an attached vector array processor. Common array processing systems include those by Floating Point Systems, Mercury Data Systems, CSPI, and SKY. Normally, these systems will be attached through direct plug into the bus of the workstation or through a parallel input/output channel. For maximum performance, a direct memory access (DMA) interface is necessary to minimize data transfer bottlenecks.

Array processors are normally implemented through intense vector pipelining, so only problems that can be approached in a way to guarantee long vectors will be efficiently implemented. If short vectors are used, there is a danger of spending more time transferring data than actually operating on the data. The efficiency of an array processer implementation of a particular problem is inversely proportional to the amount of time that the array processor spends idle.

Parallel Processing Power

When multiple CPUs or vector processors are linked together, the major differentiation between systems relates to the method of synchronization between the various processors and their memory. For a synchronous system, all operations are coordinated through a timing clock. The vector processing architectures shown above depend explicitly on timing to send the input data stream to multiple processors and various subparts of a pipeline. Multiple processors may operate through local memories or a global memory that is shared by all processors. If a processing algorithm needs only data that is not needed by any other processor, local memory may be used because communication across processors is minimized. If, however, an algorithm is implemented that requires that data be shared between processors, a complex addressing scheme must be used to avoid

Table 1. Supercomputer Performance Measures.

System	MFLOPS per processor	Max processors
(1) Cray-2	488	4
(2) Cray Y-MP	333	8
(3) CDC/ETA 10g	133	8
(4) IBM 3090S	1710	1
(5) Hitachi S-820-80	3000	1
(6) NEC SX 2	1300	1

collisions in memory access and update. If four processors need to access the same memory location, then care must be taken to lock out other processors during the instant that data are being read and to allow the next processor to read the memory as soon as it is available. If the algorithm is allowed to modify the contents of memory, the relative access order of the multiple processors could determine the final value. This result would be clearly undesirable. Dasgupta (1990) has developed a taxonomy for computing which represents serial and parallel processing alternatives.

SIMD

Another kind of synchronous parallel processing environment is the single instruction, multiple data (SIMD) system (Duncan, 1990). For this type system, multiple processors are required to execute the same instructions on multiple data streams. Image processing systems have been designed to take advantage of this architecture, and the same kinds of SIMD systems may be applied to most GIS/RS data sets. Synchronous timing is used to move data in and out of a SIMD system and to move data within the system.

Two-dimensional and three-dimensional arrays of processors can be assembled with mesh and crossbar methods of memory addressing. For a two-dimensional array, a single processor may be assigned to operate on each picture element (pixel) of an input image and to write the results into an output pixel array if there were no data dependencies on the output value from another processor. If such data dependencies did exist, the problem would not be applicable to SIMD synchronous processing.

SIMD processors may operate on complex or floating point values or they can operate on single bit values. One-bit operations such as masking may be applied by a huge array of one-bit processors such as the Loral massively parallel processor (MPP). In an image with x,y locations for individual pixels, an N by N array of processors may be used to perform logical and arithmetic functions on N by N image pixels. The CM-2 from Thinking Machines Corporation has a stated benchmark of 2500 MIPS or 2500 MFLOPS for a large matrix multiply in double precision (Meng, 1987) and 3500 MFLOPS for the same operations in floating point (Tucker, 1988).

Image array processors have been implemented in a SIMD mode in which muliple processors operate independently on a number of individual image pixels (Hogan, 1990). If the operation to be performed involves not only point operations (those which act on a single x,y image pixel with multiple layers of information) but also area operations (the output pixel's value is dependent not only on the x,y pixel's value, but also its neighbor's values), a memory access system must be employed to assure that no memory address conflicts arise. If the data to be processed are only available to individual processors through local memory, a mechanism must be employed to avoid boundary effects due to local memory boundaries. If the memory available to all processes is global, then only direct memory conflicts will potentially cause access problems. The experimental CLIP7A image processor uses SIMD elements with a certain level of autonomy (Fountain, 1988).

Associative memory may be used to store data words based on the word content rather than the location in an input array. The advantage of using associative memory rises when it is desired to search a large amount of data for a particular character or data pattern. An associative memory word usually has a large number of bits. Multiple processors then can search through associative memory in a bit-by-bit match mode to find the condition satisfying the specified requirements. SIMD systems can be used to implement associative processing.

"Systolic architectures (systolic arrays) are pipelined multiprocessors in which data is pulsed in rhythmic fashion from memory and through a network of processors before returning to memory," (Duncan, 1990). Systolic systems are SIMD systems that combine pipelining and parallel processing to optimize the processing that may happen to an individual element without significant degradations to performance by I/O. Many operations may be performed on the data without requiring intermediate storage of the temporary results.

Convex Computer Corp implements a SIMD vector and scalar processor based on RISC architecture.

MIMD

Multiple instruction, multiple data (MIMD) computer systems are the classic case that most of us think of when parallel computing is discussed. MIMD systems have multiple processors which may operate on different instructions and differing data. MIMD machines do not have synchronous timing with the same instruction being performed; therefore, a sophisticated intercommunication scheme is necessary to tell each processor when to execute its instruction and on which data set to operate. Asynchronous processing allows each processor to perform a number of different operations on its own local data without concern of the neighboring processors. Each processor acts alone, but when it finishes its process, it must notify the other processors.

Message passing between adjacent nodes on the architecture is normally the method by which one processor talks to its neighbors. This is normally considered as loose coupling between the processors and memory (Hornstein, 1986). (Because these systems employ local memory only, local memory to local memory transfers are necessary to update the state of the overall process). The topology of the MIMD architecture may employ a ring, mesh, tree, or hypercube structure. The hypercube structure may be represented by the Intel Personal Supercomputer or the NCUBE/10 system.

A global memory is another alternative to be considered when discussing MIMD systems. This alternative leads to a tightly coupled state between processor and memory. The constraints on dynamic update of the shared memory involve constant checking, especially when the memory is in the process of being updated by one process and, at the same time, being accessed by another processor. Bus contention can be handled by only having one bus to memory. When a processor is using the bus for memory access, the bus will not respond to other requests until the initial request is satisfied. Other memory access systems have a dedicated memory access path to a memory cache. Multiple switching logic is implemented in the BBN Butterfly processor as a method of memory access control. Other tightly coupled systems include the Alliant FX/8, the Sequent Balance 8000, and the Concurrent 3200 MPS (Duncan, 1990).

For an optimum implementation, an N processor architecture would represent a speedup factor of N over the serial implementation of the same algorithm. In practice, this is almost never the case because the complex message passing and testing logic will provide some inefficiency in the application of the N processors. The optimum case for a parallel system, whether synchronous or asynchronous, is to be able to use 100 percent of the capability of all processing elements.

MIMD processing is especially applicable to "coarse grained" parallelism in which an application may be broken down into functional sub units that then can be implemented on a number of processors. This is a high level parallel structure that may have one system performing totally different operations from another processor. For example, in image processing, one set of processors may perform edge enhancement and detection while another will perform edge chaining for polygons. The vector chaining procedure in this case cannot happen before the edge operations have been completed. A message must be passed from the processor performing the edge operations to all other processors saying that the results of the edge operation are available for further processing.

In some cases MIMD may be combined with SIMD processors to form a hybrid parallel processing scheme. One of the implementations involves a tree structure topology in which higher level MIMD processors send messages to SIMD subservient processors which then operate on multiple data in a synchronous mode.

In addition to the application of MIMD processing to numerical based algorithm evaluation, MIMD is being explored as an efficient means for implementing searches within a relational database (RDBMS) (Frieder, 1990).

Two other advanced MIMD architectures are data flow and reduction (Duncan, 1989). A data flow system continually scans to see if all of its operands are available (Gaudiot, 1990). Once the operands are complete from one or more processors, the instruction is executed. A token data flow system acquires tokens that are passed from completing processes and stores them until a check can be made as to whether the token matches the requirements as one of the operands for its own execution.

A reduction process, on the other hand, executes a process when it's results are necessary for the execution of another process. This "demand driven" strategy involves a clear idea of the data necessary for a particular operation, and where the sources of that data reside.

The costs of high speed computing closely coupled with a quality graphics system for visualization has dropped dramatically in the late 1980s and early 1990s with the advent of the mini-supercomputer class of machine which is expected to meet the computing and visualization needs of a project scientist with a deskside or desktop system (Diede, 1988).

Parallel Processing Software

One of the greatest challenges in parallel computing is the development of the software that will allow full use of the hardware capabilities of the new hardware systems (Prasanna-Kumar, 1989). Most of the new systems are trying to avoid the development of special languages for implementation of applications code and instead rely on optimization of FORTRAN and C code. SIMD systems have been developed with reasonably efficient parallelizing compilers because the synchronization between processors is a vital part of the architecture (Little, 1988). MIMD systems, on the other hand, often have to have special tailoring of the application program to achieve speedups. For example, in MIMD programs, the software developer must identify variables and sections of the code that must be kept in global memory with sophisticated access lockout protection. Other portions of the code may have variables in local memory that do not affect the other processors.

Transputers were designed along with the OCCAM programming language to allow efficient concurrent execution of loosely coupled processes with extensive message passing along one-way paths. A transputer may only address other transputers to which it has a direct connection. OCCAM is a special purpose parallel language which seeks to optimize the performance of loosely coupled parallel systems (Dinning, 1989).

A parallel system may have "coarse grained" or "fine grained" parallelism in its software implementation. Coarse grained parallelism involves the identification of whole code segments such as functions, expressions, or do loops that can be assigned to multiple processors. Fine grained parallelism, on the other hand requires the definition of individual variables that must be shared between processors. MIMD machines require a detailed understanding of the application code, and often require a rewrite of the applications to adequately take advantage of the parallel hardware. The implementation of an algorithm on parallel hardware may be divided into a number of processes. A heavyweight process such as the operating system may occupy resources and have high priority while "threads," or lightweight processes such as message passing, may also be implemented. There may be a number of threads within a process, each sharing memory and resources. Threads are one implementation of fine grained parallelism (Feitelson, 1990).

Xerox, in 1981, led the way into a graphical interface using bit mapped screens, mice, and icons which was dedicated to being able to respond to the casual user (Johnson, 1989). The What You See Is What You Get (WYSIWYG) philosophy was implemented in the top level document editor. Another major innovation of the Xerox Palo Alto Research Center was the development of smalltalk using the first vestiges of object oriented programming. New techniques were developed using bit mapped graphics to define the interaction between discrete code "objects," or self contained implementations of algorithms. An "object" in this sense knows everything about it's own environment, and has rules that allow it to interact with external "objects," (Kodosky, 1987). Carnegie Mellon University has developed GARNET, a support package for the development of highly interactive user interfaces (Myers, 1990). The Animation Programming Environment (APE) was developed at the Ohio Supercomputer Center to allow scientists to interact with their data without loading down the supercomputing resources (Upson, 1989).

Techniques have been developed that are attempting to define performance figures for the new parallel and serial processing machines. MIPS, MEGAFLOPS, and other newer measures such as the SPECMARK are not necessarily flexible enough to represent the true power of a particular machine for a specific application. In trying to establish a benchmark for the evaluation of image processing systems, Preston (1989) developed the Abingdon Cross Benchmark which exemplified most of the common image processing operations that are currently in use in the field. This technique would not necessarily represent the needs for GIS and remote sensing integration.

Custom Designed Silicon Chips

Another method of obtaining significant speedups in operation involves the design and implementation of special purpose chips that may be dedicated to a specific processing task. These chips, known as ASIC chips (application specific integrated circuits), have been designed for a number of applications including image processing, communications, and synthetic scene rendering. Silicon Graphics developed their proprietary "shading engine" and "transformation engine" for application in the simulation and visualization markets. We have seen above that the VITEK image processor has a dedicated, programmable, custom chip for performance of high speed computations on image data (Keller, 1990). The technology advances in recent years have allowed great expansion of the number of integrated circuits that may be placed on a chip. Very large scale integration (VLSI) and a new DoD mandated VHISC technology are being used for implementation of more and more complex processes.

DISPLAY TECHNOLOGY

The concept of value for any GIS/RS system lies in its ability to display geographic information in a meaningful way to a manager, technical specialist, or student. Display technology has changed greatly in the last decade, with new advances bringing on lower cost and more practical methods of information presentation. Hardcopy as well as softcopy displays are valuable as a media for GIS/RS information. In the past decade changes in display technology have followed several paths. First, for softcopy true color image display, the technology has moved from a 256 by 256 image, through a 512 by 512 image, to a standard 1024 by 1024. The 1024 displays now cost the same or less than the first 256 by 256 displays. A number of new workstation displays are greater than 1024 in resolution, but in many cases allow 512 and 1024 windows within the workstation display area for true or pseudocolor images. Pseudocolor or greyscale displays with eight or ten bits of color are commonplace on the least expensive personal computers. Integration of the capability of capturing images from a video source created the dilemma of saving and displaying images at video resolution (NTSC - 640 by 480) while having a greater display resolution. The advent of animation techniques on mini, micro, and workstation hardware also brought up the reverse problem of how to store information from 512 or 1024 images in video format. Current software and hardware on windowing workstation systems allow the creation of video windows within the workstation and the sending of image information out to an NTSC format device.

Image processing systems, in addition to having "true color," or 24-bit, displays, also normally have more image memory planes than are displayable by the red, green, and blue color guns of a cathode ray tube (CRT). These extra image memory planes may be used for computational scratch pad memory, or they can be used to hold other image spectral channels or other GIS layers. Because these memories are automatically refreshed, instantaneous display of a number of frames of information is possible.

New display technology has resulted in the creation of stereo image displays by use of alternating fields on a 60 Hz display or by using polarization (Johnson, 1989). Glasses are normally necessary to achieve the stereo perception necessary for detailed air photointerpretation and elevation extraction. Other new stereo viewing techniques are being investigated to achieve stereo viewing without glasses (Hodges, 1987). Holographic displays of GIS/RS information has interesting potential, but has not yet been proven practical. Volumetric displays have been implemented by Texas Instruments using the concept of writing to a spinning disk that is angled with respect to the shaft that drives it. Positioning occurs throughout the space volume, not on a two-dimensional projection (Williams, 1989).

Flat panel technology is now finally approaching its potential for large area/high density displays after years of promises. Displays such as 2k by 2k and 4k by 4k are now becoming practical using active matrix poly-si thin film transistor (TFT) technology. These displays are being developed aggressively for the high definition television (HDTV) market (Faughnan, 1989).

Medical imaging and other applications have also led to the development of 2k by 2k cathode ray tubes (Rosen, 1989). Displays with greater resolution using multiple steerable CRT beams are also being developed. Sony has developed a commercial 20-inch 2K by 2k display for air traffic control (Werner, 1989).

For animation purposes, it is necessary to be able to record a sequence of video frames on a frame by frame basis on a suitable device that will allow for real time playback of the sequence. It is not necessary that the individual frames be computed in real time for most applications. Technology for recording of video or digital information onto a media for real time playback has also advanced greatly in the last decade. High quality video editing systems such as those used for broadcast TV have the capability of single frame editing, and adding one frame at a time to one-inch video tape. Unfortunately, high quality video equipment is often prohibitively expensive. If animation frames can be generated on video tape, they may then be transferred to video disk by an expensive disk mastering process. The advantage of video disk is that, once the master video disk is made, many copies may be made at low cost.

Another method that may be used for animation is the optical memory disk recorder (OMDR) which is a direct read after write (DRAW) or "write once" technology. Information may only be recorded once in each position in the OMDR. The recorded image is then permanently on the OMDR and may not be recorded over. This type of system is often used for archiving images. In a DRAW system, a user may record frames from his rendering system in a frame-by-frame manner by converting the RS170 RGB signals from a reasonable quality color monitor to an NTSC signal through a converter and recording the video directly on the OMDR. A user then is able to play his video frames back from the OMDR in real time (Wilson, 1989). OMDR systems with DRAW technology are now within the ten thousand dollar range.

New technological advances now have resulted in the development of optical disks with limited read/write capability. These systems may also be recorded in a frame-by-frame manner and may be played back in real time.

SOFTWARE ENVIRONMENT ADVANCES

In addition to the advances in software for parallel processing, the workstation environment has evolved due to significant advances in software in the last 10 years.

GRAPHICAL USER INTERFACE (GUI)

Pioneered by the Xerox Corporation STAR and popularized by the Apple MacIntosh Computer, the graphical user interface allows users with relatively little experience to manipulate programs and data by merely "pointing" and "clicking" to appropriate places on a bit-mapped display system. The compute paradigm itself transformed from a fundamentally character-oriented, command-line model to "windows" of operation, either character or graphical in nature, operating independently (Johnson 1989). In addition, certain objects could be "cut," "dragged," and "dropped" into other windows, thereby providing an intuitive method of inter-application communication. Today, that paradigm is available in nearly all computers.

CLIENT-SERVER MODELS

By the time that the MacIntosh computer was unveiled in 1984, the banking industry, dominated by mainframe vendor IBM, had firmly established the concept of "clients" and "servers" which performed specialized functions in networked environments. Authenticated requests from remote "clients" were issued to specialized "servers" to perform a variety of functions. Centralized computer servers processed branch office accounts. Different banks electronically "wired" funds overseas. Monthly statements were sent to high-speed print "servers." Airline and hotels reservation systems adopted these methods as well.

In 1983, the Massachusetts Institute of Technology launched Project Athena, to provide a coherent computing environment for the tremendously diverse needs of its faculty and students. Among the project's many achievements has been the standardization of a new compute resource, the display server. Perhaps most significant about the display server standard has been its endorsement by nearly every manufacturer of workstation-class computers and their associated software vendors.

Consisting of a bitmapped display, a keyboard, and a mouse, the display server provides a powerful method for client application programs to obtain input and display output to users. A MacIntosh-style window paradigm was adopted, with the important difference that a client could open a window, dubbed an "X-window," on any or all machines accessible over the network. Multitasking machines, widely available in desktop configurations, can perform the tasks required of both client and server, while low cost dedicated display servers, commonly called "X-terminals," have appeared which run all clients remotely. Extensions are currently underway at MIT to standardize three-dimensional processing and image decompression in the standard display server functionality.

ICONIC PROCESSING

One example of the emerging class of GIS/RS applications is the iconic processing environment developed at the Lamont-

Doherty Geological Observatory in New York State (Sunexpert, 1990). Running on a UNIX-based network of file servers and workstations, the iconic processor allows a user to construct arbitrarily complex topologies of data files and processing steps using the full resources of the network. After beginning a remote execution service daemon, machines are eligible to participate in the networked imaging process. Icons representing disk files, processing algorithms, and output devices are selected and logically linked together, as by a user in a network configuration window.

Considerable management information systems (MIS) research is being done in the arena of active "agents," which, like their human counterparts, are dispatched, with sufficient authority, to support a high-level query or command, returning with appropriately formatted "reports." Such agents typically spawn a background process on a host machine, and may take hours or several days to run. For example, a planner may wish to look at the pattern of building. Because such high-level requests will increasingly interact with distributed databases under the control of several agencies and vendors, the model of processing the agent's request locally, and then "billing" the query process, has some merit. Of course, supporting such a model raises serious questions of security and privacy, as well as the technical issues of supporting a standardized set of remote procedure calls. Also, subtle problems concerning the consistency of rules applied by each agent in differing contexts may be difficult to detect. However, it is not unreasonable to expect computerized decision-support systems, fashioned after their traditional counterparts of industry and government, to enter mainstream computing in the near future (Wiggens, 1988).

VISUALIZATION

In the last several years, the computing power available in a workstation environment has made it possible for detailed analysis and incremental simulation to be used at a scientist's desk to help solve nonlinear differential equations or massively parallel problems that operate on huge amounts of geographic or image based data. As discussed in the section on Display Technology, systems are now becoming available that fall into a new class of personal supercomputing. Superior graphics on some of these systems allow for instantaneous viewing of time series or multi-dimensional data (Weigner, 1990). Software for these systems as well as the current generation of desktop workstations is now being developed to harness the power of the hardware for aids in interpretation of multi-dimensional data sets. The integration of remote sensing into GIS systems is an application that may be suitably shown in three-dimensional perspective (Long, 1989). Landscape planning is one application which can benefit from dynamic interaction of a three-dimensional GIS along with CAD models and the capability to instantaneously render a new view of a database showing the effects of potential changes in the GIS information (Robertson, 1990).

The huge amounts of data that will need to be analyzed to allow understanding of the spatial and temporal dynamics of the global environment make visualization techniques not only useful but absolutely necessary (Hibbard, 1989). New remote sensing instrumentation is being developed that will dwarf the data volumes of all previous remote sensing systems. These data sets must be integrated into multi-dimensional GIS systems which provide expanded capabilities for time and space modeling.

A number of vendors of single and multiprocessor computing systems have developed "visualization environments" that attempt to provide sophisticated display of scientific data without detailed programming knowledge by the potential user (Upson, 1989; Bishop, 1990; Myers, 1990).

IDENTIFICATION OF TECHNOLOGY LIMITATIONS

While the advances in computing technology have been spectacular in the last decade, the realization of its full potential for GIS/RS processing and analysis is yet to be realized. For full acceptance of GIS/RS systems into the everyday management process, the systems will have to provide data access and analysis solutions in "real time," not simply "near real time." The real time processing must apply to relational access to multiple attributes for a geographic database class, as well as complex overlay processing to answer "what if" questions. This real time analysis is becoming possible using the advances in raw speed of central processing units of serial, pipeline, and parallel systems as well as advances in the real time compression and decompression of image and map data (Clark, 1990).

A manager must be able to postulate questions to a system through a user interface that does not require sophisticated computer knowledge and to receive an answer in text, numeric, or graphical terms instantaneously. Expert and learning systems must be used to develop systems capable of learning a particular application and providing tailored default answers and processing streams. For example, the user should be able to input a Landsat image tape and have the system dynamically enhance the data, geometrically correct the data, classify the data by means of a layered classifier, and then load the data into a GIS, all by specification of only a few parameters at the beginning.

Another limitation lies in the difficulty of developing software that will fully utilize the computing power that is now emerging. The complex nature of spatial analysis requires the developers of remote sensing and GIS software to thoroughly understand applications of spatial analysis as well as the programming languages and operating systems of a myriad of workstations. There are only a handful of vendors of GIS and remote sensing systems whose development staff is well acquainted with the applications, algorithms, and interfaces that are necessary to provide a product to an increasingly diverse and less technically savvy user base. Standards for operating systems, programming languages, and user interfaces are beginning to be defined that will simplify the job of the system developer, and allow him to apply his efforts to efficient implementation of computer code for GIS and remote sensing analysis.

The most important limitation of all to widespread GIS/RS acceptance is the cost of initially entering geographic and attribute information into a database, and the cost of continuous updating of the data. Some steps have been taken toward a limited satisfaction of the availability of geographic data sets, with the service provided by several vendors and applications companies of assembling a number of the data variables necessary for a GIS/RS system for a particular geographic area.

A large impediment to GIS/RS integration has been the sheer volume of data a single image can represent, and the processing often required to remove any radiometric or geometric distortions necessary to make the image useful in any given context. Fortunately, several of these problems are shared by the electronic publishing and printing (EP&P) industry, and tremendous progress has been made toward providing solutions in low-cost configurations. In particular, image compression algorithms have been embraced to the point that low-cost, optimized integrated circuits are appearing which work in concert with popular personal computer software. The current standard was developed by the JPEG (Joint Photographic Expertise Group ANSI).

The JPEG standard will be incorporated into the next generation, "level 2," of the PostScript page description language specification, so that mainstream color imaging servers may transmit and receive compressed image files for printing and half toning purposes. Additionally, Kodak has announced that it will be supporting image compression in its Photo-CD family of products, whereby customers can have 35-mm rolls of film converted into CD-ROM format for digital manipulation, priced appropriately for a consumer market.

GIS/RS FUTURE COMPUTING RESEARCH ISSUES

If the advances in speed in computing through hardware development will continue to advance in a logarithmic manner during this decade with speed in MIPS for workstations roughly equivalent to "Joy's Law," (Gage, 1990), i.e.,

$$\text{speed} = 2^{**} (\text{Year} - 1986),$$

then in 1991 the speed of serial processing units will be 25 or 32 MIPS. We have already seen, in the last stages of 1990, workstations with at least this much power. Given a continuing growth of CPU speed, the potential capability for near real time analysis of geographic data seems achievable within a few years. Networking and the transfer of data across large distances for real time access seem to be the limitations of current technology.

The real issues in computing in the next decade involve innovations that will allow relatively unsophisticated users to access the power of the computer hardware, without having to become experts in programming and computer operating systems. The tools for GIS and remote sensing analysis should become easier to use for a novice, and at the same time be able to take advantage of the new advances in hardware and software technology.

The research issues associated with this problem involve the investigation of new and innovative techniques for the provision of powerful GIS and remote sensing analysis techniques to a relatively unsophisticated user. New techniques associated with artificial intelligence technology and the development of learning systems may approach a solution to the rapid acceptance of GIS technology in real world applications.

New research should be developed to take advantage of the new 1-, 2-, 3-, 4-, . . ., n-dimensional visualization techniques now being developed for the analysis and presentation of geographic data. These techniques will eventually allow for the coherent analysis of time series data sets as well as global data sets necessary for the effective characterization of parameters related to the global environment.

Research should be conducted into a standardization of the n-dimensional cataloging of spatial data sets which combine raw sensor information with either raw or interpreted GIS information. An expansion of the current GIS/remote sensing capabilities for dealing with data structures to allow acceptance of new sensor information as well as an archival and cataloging function is necessary for the ability to handle and analyze the massive data sets that will become available for global analysis during the 1990s. The ability to dynamically update and create new data structures for spatial data is critical to the integration of GIS/remote sensing technology into everyday decision making. The cataloging function must contain extensive history information which gives an explanation of what processing and analysis steps have been applied to each data set.

The integration of GIS and remote sensing analysis requires more than the simple combination of functions developed individually for either GIS or remote sensing analysis. Synergistic interactions exist between spatial data sets that require new techniques be investigated that actively combine the attributes of all data sets into a form that can be analyzed by a multidimensional data analysis algorithm. Research needs to be done into the definition of the interactions between GIS and Remote Sensing variables that could be implemented in more accurate formulations of analysis techniques.

The interactions between disparate data sources requires us to consider the scaling variations between the data ranges of the various data layers that would be used in a spatial analysis. While current remote sensing data layers or channels may only a have a basic data range of 8 bits, or 0 to 255, other spatial data sets may have a basic variation orders of magnitude higher than that for raw sensor data. For example, another spatial variable might be the distribution of population over a study area. This variable, depending on scale, might have a large value which is entirely inconsistent with the range of data for raw sensor information. A decision rule, for example, maximum-likelihood classification, might be overly biased by the data range associated with the population and underestimate the influence of the raw data set. A multi-dimensional scaling strategy must be developed to relate the data ranges of input GIS/remote sensing to the effective data analysis algorithms currently used in GIS and remote sensing analysis.

The new computing and network capabilities that are now becoming available through the explosion of the workstations in image processing and GIS analysis will need to be thoroughly tested through rapid prototyping with feedback provided to the hardware and software vendors. The concept of real time interactive spatial analysis with a number of scientific collaborators should be explored and defined further to define the limitations of immediate interactions and the concepts and techniques that would provide these desired capabilities in the long as well as the short term. High speed data networks, and real time data compression and decompression, will need to be combined with a modeling methodology for the immediate combination of insights and ideas of a number of scientists instantaneously. New techniques for multimedia communication using, voice, video, graphics, and digital imaging will need to be employed to provide the optimum working environment for effective collaboration.

Innovative techniques for data reduction and compression for raw sensor integration into GIS spatial analysis using on-board processing for specific applications should be considered as a method of achieving real time scientific interaction.

Finally, research needs to be implemented to define a set of standard GIS/remote sensing functions that may be implemented into a GIS/RS toolkit of C or FORTRAN callable programs. Using standards such as X and Motif for user graphical user interface (GUI) and display of data, a GIS/remote sensing algorithm toolkit would allow the vendor community to rapidly expand their offerings in the GIS/remote sensing market place, as well as provide a basis for the rapid acceptance of GIS/remote sensing functionality into an increasingly broad user community.

To satisfy the research needs addressed above, the following tasks are proposed:

- Investigation of discrete GIS/RS functions to determine speedup potential from pipeline and parallel computing.
- Determine advantages/disadvantages of SIMD versus MIMD for GIS/RS.
- Investigate potential of Expert Systems for simplification of the interface to non-scientific users.
- Determine speed versus cost issues in practical applications of GIS/RS (institutionalization cross issue).
- Investigate the advantages/disadvantages of the development of a low level GIS toolkit that may be called by superior planning/modeling programs.
- Investigate the practicality of performance of GIS/RS functions across a heterogeneous network.
- Investigate the practicality of dynamic compression/decompression of GIS/RS data on the fly.
- Investigate the implementation of vector GIS/RS functions (polygon overlay, masking, etc.) on parallel systems.
- Develop stadards and generic algorithms that would allow for automatic conversion of raster scanned data to polygons.
- Investigate the application of "Learning Systems" to GIS/RS analysis and user interface.

- Investigate methodologies and implementation for real time raster GIS/RS analysis using SIMD and vector hardware.
- Investigate the usefulness/application of stereo displays for three-dimensional update of raster/vector integrated data.

ACKNOWLEDGMENTS

The authors would like to thank Chris Nicholas of Sun Corp. for appropriate additions in definition of the current state of the art in software environments. Also, thanks are due to Jack Estes and Frank Davis of NCGIA for their advice in integration issues.

REFERENCES

August, M., G. Brost, C. Hsiung, and Schiffleger, 1989. Cray X-MP: The Birth of a Supercomputer, *Computer*, Vol. 22, No. 1, pp. 45-53.

Baum, D., and J. Winget, 1990. Parallel Graphics Applications, *UNIX Review*, Vol. 8, No. 8, pp. 50–61.

Bishop, G., M. Monger, and P. Ramsey, 1990. Multicomputing—A Visualization Programming Environment for Multicomputers, *IEEE Computer Graphics and Applications*, Vol. 10, No. 1, pp. 50–58.

Clark, D., 1990. C-Cube's JPEG Image Compression: Where's It Headed?, *Advanced Imaging*.

Cheng, H., 1989. Vector Pipelining, Chaining, and Speed on the IBM 3090 and Cray X-MP, *Computer*, Vol. 22, No. 9, pp. 31–45.

Cragon, H., and J. Watson, 1989. The TI Advanced Scientific Computer, *Computer*, Vol 22, No. 1, pp. 55–65.

Dangermond, J., 1988. GIS Hardware Trends, *Computer Graphics World*, Vol. 11, No. 12, pp. 104–104.

Dasgupta, S., 1990. A Hierarchical Taxonomic System for Computer Architectures, *Computer*, Vol. 23, No. 3, pp. 64–75.

Davis, F., D. Quattrochi, M. Ridd, and N. S-N Lam, 1991. Research Needs in Processing and Analysis of Remote Sensing Data, *NCGIA Initiative 12 Report*.

Diede, T., C. Hagenmaier, G. Miranker, J. Rubinstein, and W. Worley, 1988. The Titan Graphics Supercomputer Architecture, *Computer*, Vol. 21, No. 9, pp. 13–27.

Dinning, A., 1989. A Survey of Synchronization Methods for Parallel Computers, *Computer*, Vol. 22, No. 7, pp. 66–78.

Dongarra, J., and I. Duff, 1985. *Advanced Architecture Computers*, ANL/MCS-TM-57, Argonne National Lab.

Duncan, R., 1990. A Survey of Parallel Computer Architectures, *IEEE Computer*, Vol. 23, No. 2, pp. 5–17.

———, 1991. Choosing Parallel Architectures, *Military and Aerospace Electronics*, No. 8, p. 19.

Ehlers, M., D. Greenlee, T. Smith, and J. Star, 1991. Data Structures and Access, *NCGIA Initiative 12 Report*.

Estes, J., J. Star, F. Davis, and D. Maquire, 1991. Overview of Research Issues in the Integration of Remote Sensing and GIS, *NCGIA Initiative 12 Report*.

Faughnan, B., and R. Stewart, 1989. Polysilicon Active-matrix Liquid Crystal Displays, for Cockpit Applications, *Proc. of SPIE*, Vol. 1117, pp. 143–154.

Feitelson, D., L. Rudolph, 1990. Distributed Hierarchical Control for Parallel Processing, *Computer*, Vol. 23, No. 5, pp. 65–78.

Fountain, T., K. Matthews, and M. Duff, 1988. The CLIP7A Image Processor, *IEEE Transactions on Image Analysis and Machine Intelligence*, Vol 10, No. 3, pp. 310–320.

Frieder, O., 1990. Multiprocessor Algorithms for Relational-Database Operators on Hypercube Systems, *Computer*, Vol. 23, No. 11, pp. 13–28.

Gage, J., 1990., Personal communication, Presentation at I-12 Working Group Meeting, EROS Data Center, Dec 5, 1990.

Gaudiot, J., and A. Sohn, 1990. Data Driven Parallel Production Systems, *IEEE Transactions on Software Engineering*, Vol 16, No. 3, pp. 281–294.

Gernazian, A., and S. Sperry, 1989. Remote Sensing and GIS, *Advanced Imaging*, Vol. 4., No. 3, pp. 30–32.

Hibbard, W., and D. Santek, 1989. Visualizing Large Data Sets in the Earth Sciences, *Computer*, Vol. 22, No. 8, pp. 53–58.

Hodges, L., and D. McAllister, 1987. True Three-Dimensional CRT-based Displays, *Information Display*, Vol. 3, No. 5, pp. 18–22.

Hogan, B., 1990. High Performance Image Processing on a Massively Parallel Computer, *Advanced Imaging*, Vol. 5, No. 10, pp. 42–47.

Hornstein, V., 1986. Parallel Processing Attacks Real-Time World, *Mini-Micro System*, Vol. 19, No. 12, pp. 65–72.

IEEE Scientific Supercomuter Subcommittee, 1989a. The Computer Spectrum, *Computer*, Vol. 22, No. 11, pp. 57–64.

———, 1989b. Supercomputer Hardware, *Computer*, Vol. 22, No. 11, pp. 63–69.

———, 1989c. Software for Supercomputers, *Computer*, Vol. 21, No. 11, pp. 70–74.

Johnson, J., T. Roberts, W. Verplanck, D. Smith, C. Irby, and C. Upson, 19xx. Tools for Creating Visions, *UNIX Review*, Vol. 8, No. 8, pp. 38–48.

Johnson, J., T. Roberts, W. Verplank, D. Smith, M. Beard, and K. Mackey, 1989. The Xerox Star: A Retrospective, *Computer*, Vol. 22, No. 9, pp. 11–29.

Johnson, P., 1989. Hardware for Stereoscopic Computer Graphics and Imaging, *Information Display*, Vol. 3, No. 5, pp. 16–18.

Jones, T., 1989. Engineering Design of the Convex C2, *Computer*, Vol. 22, No. 11, p. 36–45.

Keller, J., 1990. The Rise of Image Processing, *Military and Aerospace Electronics*, p. 17.

Kodosky, J., and R. Dye, 1987. Graphical Programming, *Computer Graphics World*, Vol. 10, No. 12, pp. 77–80.

Lang, L., 1989. GIS Goes 3D, *Computer Graphics World*, Vol. 12, No. 3, pp. 38–44.

Lauer, D., J. Estis, J. Jenson, D. Greenlee, and T. Mace, 1991. Institutional Issues Affecting the Integration of Remotely Sensed Data and Geographic Information Systems, *NCGIA Initiative 12 Report*.

Little, J., G. Blelloch, and T. Cass, 1989. Algorithmic Techniques for Computer Vision on a Fine-Grained Parallel Machine, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 3, pp. 244–257.

Lunetta, R., R. Congalton, L. Fenstermaker, J. Jenson, K. McGwire, and L. Tinney, 1991. Remote Sensing and Geographic Information System Data Integration: Error Sources and Issues, *NCGIA Initiative 12 Report*.

McHarg, I., 1969. *Design with Nature*, The Natural History Press, Garden City, New York.

Meng, 1987. Parallel Processor Gets Data Intensive, *ESD*, Vol. 17, No. 6, pp. 17–17.

Myers, B., D. Giuse, R. Dannenberg, B. Vander Zanden, D. Kosbie, E. Pervin, A. Mickish, and P. Marchal, 1990. GARNET: Comprehensive Support for Graphical, Highly Interactive User Interfaces, *Computer*, Vol. 23, No. 11, pp. 71–86.

Prasanna-Kumar, V., and D. Reisis, 1989. Image Computations on Meshes with Multiple Broadcast, *IEEE Transactions on Pattern Anlaysis and Machine Intelligence*, Vol. 11, No. 11, pp. 1194–1202.

Preston, K., 1989. The Abingdon Cross Benchmark Survey, *Computer*, Vol. 22, No. 7, pp. 9–19.

Rau, R., D. Yen, and R. Towle, 1989. The Cydra 5 Departmental Supercomputer, *Computer*, Vol. 22, No. 1, pp. 12–34.

Robertson, B., 1990. Sculpting the Scenery, *Computer Graphics World*, Vol. 13, No. 6, pp. 48–52.

Rosen, B, and S. Kriz, 1989. Case study: Developing a 3000-line interactive CRT display, *Information Display*, Vol 4, No. 1, p. 12–15.

Siberschatz, A., M. Stonebraker, and J. Ullman, J. (editors), 1990. *Database Systems: Achievements and Opportunities*, TR-90-22, Department of Computer Sciences, The University of Texas at Austin.

Steinitz, C., P. Parker, and L. Jordan, 1976. Hand-Drawn Overlays: Their History and Prospective Uses, *Landscape Architecture*, Vol. 66, No. 9, pp. 444–455.

Sunexpert, 1990. Iconic Processing in a Scientific Environment, *Sunexpert*, Vol. 1, No. 8, p. 81.

Tilley, S., and S. Sperry, 1988. Raster and Vector Integration, *Computer Graphics World*, Vol. 11, No. 8, pp. 73–75.

Tucker, L., and G. Robertson, 1988. Architecture and Applications of the Connection Machine, *Computer*, Vol. 21, No. 8, pp. 26–39.

Werner, K., 1989. The flowering of liquid-crystal technology, *Information Display*, Vol. 4, No. 2, pp. 6–11.

Wiegner, K., 1990. Data and Vision, *Forbes Magazine*, Vol. 146, No. 10, pp. 193–196.

Wiggins, 1988. Expert Systems: *Applications to Planning*, Springer-Verlag.

Williams, R., and F. Garcia. 1989. Volume Visualization Displays, *Information Display*.

Wilson, R., 1990a. New Long-Word Architecture Threatens to Outshine RISC, *Computer Design*, Vol. 29, No. 9, pp. 26–28.

———, 1990b. Imaging and the Write Choice for Optical Mass Storage, *Advanced Imaging*, Vol. 29, No. 9, pp. 37–41.

43rd Photogrammetric Week
Stuttgart, 9-14 September 1991

This internationally-recognized "vacation course in photogrammetry" has been held at Stuttgart University since 1973. Because Professor Dr.-Ing. Friedrich Ackermann, one of those responsible for the scientific program, is to retire soon, this 43rd Photogrammetric Week will be his farewell seminar. Essential lines of his work have been chosen as the main topics for the meeting:

• GPS for Photogrammetry • Digital Photogrammetric Image Processing • Photogrammetry and Geo-Information Systems •

Lectures and discussions will be held in the mornings. Technical interpreters will be available for simultaneous translations into German or English. Demonstrations are scheduled for the afternoons. **For further information, contact:** Universitat Stuttgart, Institut fur Photogrammetrie, Keplerstrasse 11, D-7000 Stuttgart 1, FRG, telephone 0711/121-3386 or FAX 0711/121-3500.

Introduction to GIS for Water Resources Applications
Short Course
Louisiana State University, Baton Rouge
7-8 September 1991

Course Topics
• Value of GIS • GIS Theory • Case Study – Mapping Water Quality Parameters • System Components • Case Study – Urban Flash Flooding Prediction • Work Flows • Case Study-Assessment of Ground Water Vulnerability to Contamination • Implementation Issues • Evaluation and Review •

Contact: Michael C. Fink, Meeting Manager, American Water Resources Association, 5410 Grosvenor Lane, Suite 220, Bethesda, MD 20814, tel. 301-493-8600, fax 301-493-5844.

Air, Marine, and Land Radionavigation Systems Users

1990 Federal Radionavigation Plan
1991 Conferences

The U.S. Department of Transportation is conducting open meetings for all users of U.S. government-provided radionavigation systems. The purpose of the meetings is to obtain user perspectives on federal policits and future plans for these services. Federal radionavigation policies and plans are outslines in the 1990 DOD/DOT Federal Radionavigation Plan, single copies of which are available from the VOLPE National Transportation Systems Center. Users are encouraged to attend the meetings to provide inputs for the 1992 plan.

LORAN-C • OMEGA • TRANSIT • RADIOBEACONS • VOR/DME • MLS/ILS • GPS

Sponsors: Research & Special Programs Administration; Federal Aviation Administration; U.S. Coast Guard

Dates/Locations: 19-20 Nov. 1991, Alexandria, Virginia; 5 Dec. 1991, Seattle, Washington.

Information: *Federal Radionavigation Plan*: Elisabeth J. Carpenter, Volpe National Transportation Systems Ctr., Ctr. for Navigation (DTS-52), 55 Kendal Square, Cambridge, MA 02142-1093, tel. 617-494-2126.
Conferences: Conference Office (DTS-930), Attn: Radionavigation Users Conference, 55 Kendal Square, Cambridge, MA 02142-1093, tel. 617-494-2307.