Applied Technology Institute (ATIcourses.com)

Stay Current In Your Field • Broaden Your Knowledge • Increase Productivity

349 Berkshire Drive • Riva, Maryland 21140
888-501-2100 • 410-956-8805
Website: www.ATIcourses.com • Email: ATI@ATIcourses.com

Boost Your Skills With ATIcourses.com!

ATI Provides Training In:

- **Acoustic, Noise & Sonar Engineering**
- **Communications and Networking**
- **Engineering & Data Analysis**
- **Information Technology**
- **Radar, Missiles & Combat Systems**
- **Remote Sensing**
- **Signal Processing**
- **Space, Satellite & Aerospace Engineering**
- **Systems Engineering & Professional Development**

Check Our Schedule & Register Today!

The Applied Technology Institute (ATIcourses.com) specializes in training programs for technical professionals. Our courses keep you current in state-of-the-art technology that is essential to keep your company on the cutting edge in today's highly competitive marketplace. Since 1984, ATI has earned the trust of training departments nationwide, and has presented On-site training at the major Navy, Air Force and NASA centers, and for a large number of contractors. Our training increases effectiveness and productivity.

Learn From The Proven Best!

ON-BOARD PROCESSING FOR SPECTRAL REMOTE SENSING

Richard B. Gomez and Ambrose J. Lewis
Center for Earth Observing and Space Research
George Mason University
Fairfax, Va 22030, USA
rgomez@gmu.edu

ISPRS Special Session, FIEOS/ISPRS

KEY WORDS: On-Board Processing, Hyperspectral, Spectral Library, Remote Sensing

ABSTRACT:

This paper presents an information technology (IT) data processing approach that will provide cost-effective methods to operationally process and exploit spectral information on board a remote sensing satellite in real time. This IT data processing approach focuses on the development of a method to process and exploit, on board, huge amounts of spectral data without the need to have an expert spectral analyst in the loop. Special attention is paid to spectral libraries and novel uses of exemplar spectra. Exemplar spectra are single spectra, derived from a set of individual spectral signature measurements that capture the spectral essence of a class or subclass of materials. These involve techniques that are beyond the forefront of current hyperspectral research. Open Database Connectivity (ODBC) and JDBC (not an acronym) technology are used in the ground segment to provide cross-DataBase Management System (DBMS) connectivity to a wide range of SQL databases and other tabular data sources such as spreadsheets. In cooperation with other organizations that have spectral libraries, an open standard for the development of a prototype Internet-based, spectral information management architecture (SIMA) will be proposed. The prototype to be built will contain "canned" statistical and visualization routines. The final version of the database will be searchable by key words, etc., using hypertext-type software to extract desired information. It is envision that these tools will be developed with a client-server architecture knowing that these algorithms may be computationally intensive. We expect to use a COTS computational software language such as Matlab or IDL to code the statistical and visualization routines. The exact algorithm adopted will be decided upon during definition of a User Model that will assist in defining how the database will be designed. We will build upon the prototype data search and analysis engine developed at GMU/CEOSR. This WWW-enabled, Java front-end user-friendly engine will take full advantage of freely available or widely available commercial software and low-cost hardware architecture. For the ground segment, we will allow for the development of novel human-interfaced search implementations with efficient data mining, hierarchical data models making full use of extensible markup language (XML) technology. CEOSR has demonstrated this IT approach through the successful relationship between GMU and NASA's Global Change Data center that houses both the EOSDIS Goddard Distributed Active Archive Center (DAAC) and the TRMM Science Data Information System (TSDIS).

1. INTRODUCTION

1.1 Overview

Hyperspectral imaging (HSI) is proving to be one of the most powerful remote sensing techniques available. The cost of this approach is the generation of huge datasets that require fairly sophisticated computational techniques to exploit. The purpose of this paper is to show that these computational difficulties can be somewhat mitigated by using techniques from other areas in information technology. First, the paper introduces the concept of real-time systems to familiarize the reader with this area. Second is a discussion on the potential application of hardware clustering as a methodology to increase overall system performance and throughput. Included is a brief overview of the software libraries that can be used to perform parallel processing within such a hardware cluster. Third is an overview of spectral libraries and exemplar spectra. Effective utilization of these two system components is critical to the computationally intensive pixel unmixing and classification algorithms. Following this introduction is a discussion about the proposed Spectral Information Management Architecture (SIMA), which is proposed as a technique for organizing and structuring spectral information.

1.2 Hyperspectral Introduction

It is possible using hyperspectral sensors and appropriate data analysis techniques to remotely sense the composition of objects within a scene. There is a tremendous power in this simple statement. Let’s look at how this works: Imaging spectrometers collect a very large number of wavelength channels simultaneously. Hyperspectral imaging spectrometers collect on the order of hundred of wavelengths per each spatial pixel. For example, the CASI 2 sensor collects 288 bands of data for each pixel imaged. This differs significantly from multispectral systems, which collect on the order of ten bands per pixel. For example, the Landsat Thematic Mapper sensor collected three bands in the visible, one in the near infrared, two in the mid infrared, and one in the thermal infrared only.

Another way to distinguish hyperspectral from multispectral data is the spectral resolution or channel size. Hyperspectral sensors have very small wavebands when compared to their multispectral cousins. For example, the Landsat Thematic Mapper channels vary in size from 100 nm in the visible blue channel to 270 nm in the mid infrared channel. This varies considerably from a hyperspectral sensor, such as CASI 2, which has 288 channels with a spectral resolution of 2.2 nm full width at half maximum (FWHM) at 650 nm wavelength.

By having so many small wavebands over such a large portion of the electromagnetic spectrum, it is possible to use hyperspectral data to determine the unique spectral signature of remotely sensed objects. Hyperspectral data is typically shown in a three-dimensional cube. Two dimensions are for the spatial data and the third is for all the spectral band measurements. Data is captured in frames of pixels, the size of which is set by the sensors altitude and angular aperture.

We've already seen the major benefit of hyperspectral imaging, the determination of target composition. Next, we discuss some of the challenges associated with these techniques. The first challenge is the huge volume of data generated by these types of sensors. The complexity of dataset analysis and distribution is much greater here than in multispectral applications. This isn't so much of a constraint for airborne imaging spectrometers, where data can be hand carried off the platform. The second challenge is huge volume of data redundancy within these datasets. Much of the additional data doesn't add to the inherent information that can be extracted from the hyperspectral dataset. There are two forms of redundancy possible in these datasets: spatial and spectral. In spectral redundancy, information from one band may be fully or partly predicted from other bands within the dataset. Generation of correlation matrices based on the data from each band can help identify where the redundancy is lurking. The so-called Hughes effect comes into play here (Hughes, 1968). This effect states that the number of training pixels must grow to ensure reliable estimates from supervised data classifier algorithms. As the number of features increases, the classifier accuracy will also increase to a certain point. After this point, the performance of the classifications actually begins to decrease.

2. REAL-TIME SYSTEMS

2.1 Introduction

Real-time systems are utilized in applications that required a deterministic performance. Example applications include automated target recognition, navigation, networking, medical, and control systems. In all of these examples, the key is to have the system's hardware and software perform the required work within the defined timeframe. For example, a ship's navigation system that cannot keep up with positional updates from multiple concurrent sensor feeds is a hazard. Real-time operating systems and services provide system designers with the ability to predetermine minimal acceptable Quality-of-Service (QoS) metrics for the system. Based on these metrics, the hardware and software of the system must be designed to meet the stated mission performance requirements.

As the examples above illustrate, real-time systems are typically utilized in mission critical environments. In these applications, failure of the system could lead to immediate and irreversible lose of property or even life. A mission critical remote sensing system must provide high availability performance. This constraint requires the system to perform consistently and reliably during its normal operations. A corollary to high availability is fault tolerance. This is the ability of the system to detect and react to internal and external errors in a controlled fashion. For example, a fault tolerant system may be able to select between two redundant input streams to utilize the "best" data for a signal-processing algorithm. This system may automatically switch from an input channel with too many communications errors to a more reliable channel.

2.2 Performance Monitoring and Fault Localization

High-availability, fault tolerant systems will typically employ redundant hardware to attempt to minimize any system-wide single points of failure. The system may be able to automatically detect and react to system failures utilizing the concepts of performance monitoring and fault localization.

Performance monitoring techniques analyse the inputs and outputs from the components within the system and track the progress of data through the system. The goal is to determine if the data produced at a given component of the system differs significantly from the expected result. A component is given a particular input, and the output of the processing is compared to the expected result, or signature. The performance monitoring software may determine that a component is working sub-optimally. In this case, the system would be directed to switch to another, redundant component. Performance monitoring is typically done by checking error rates and other metrics against a table of thresholds. When the threshold for a particular component is exceeded, the performance monitoring software will signal that a change is required.

Fault localization differs from performance monitoring. Here, known patterns, or signatures, are injected into the system and the results are compared to the expected. If the fault localization code isolates a problem component, the system will notify the operator and/or force a switch to another component.
Typically, a full system will be comprised of a variety of components with different performance requirements. Near real-time performance is often suitable for display systems. In this case, since there is an inherit delay as a human operator sees and interprets the display, some slight variations in processing performance can be tolerated. Another component in our hypothetical system could have no real-time constraints. For example, in some applications, the software that interprets user inputs need not have real-time performance constraints. In this case, the software would be "event driven" and react to user inputs when they occur. On the other hand, such system/operator interactions may need to be real-time, in the case of a system that must react to a user selection within a predetermined timeframe after the operator initiates an action.

3. HARDWARE CLUSTERING TECHNIQUES

3.1 Clusters

It may be feasible to increase the on-board processing power of remote sensing satellites by developing clusters of platform-based computers. A cluster is a collection of two to hundreds or even thousands of interconnected computers used as a unified computing resource. Clusters are used to achieve high performance, high availability, or horizontal scaling. Cluster technology can also be used for highly scalable storage or data management. These computing resources could be utilized to efficiently process the remotely sensed data before transmission to the ground. Alternatively, the various processors could look for different solutions within the same dataset. For example, a processor in the cluster could be dedicated to identify an important spectral in a hyperspectral dataset based on user tasking.

The clustering technique has been exploited with tremendous success on Earth-based problems. A generic class of clustered computers developed by NASA is the so-called Beowulf class (Ridge, et at., 1997). Computer systems in this class utilize low cost hardware and Linux base operating systems, combined with mature parallel software libraries such as MPI (Message Passing Interface) and PVM (Parallel Virtual Machine) to make some of the most powerful computing systems in the world. The classic example given to support this approach is "to pull a bigger wagon, it is easier to add more oxen than to grow a gigantic ox."

3.2 PC/104 Computers

A promising area for the space environment is work being done at by the Embedded Reasoning Institute (ERI) of Sandia National Laboratory in PC/104 Linux Mini-clusters (Williams and Armstrong, 2002). PC/104 is an IEEE standard (IEEE P996.1) that describes a single-board computer (SBC). These computers are compliant with industry standardized hardware and software of the PC architecture. Mechanically quite different from the PC form factor, PC/104 modules are approximately 3.6 X 3.8 inches in size. A self-stacking bus is implemented with pin-and-socket connectors composed of 64- and 40- contact male/female headers, which replace the card edge connectors used in standard PC hardware. Virtually anything that is available for a standard PC is available in the PC/104 form factor. PC/104 components are designed to be stacked together to create a complete embedded solution.

Given the fairly small size and low power consumption of PC/104 computers, they may be suitable for the satellite environment. Other small form factor and power consumption devices could be explored as alternatives. These include VME (Versa Module Europa) and Compact PCI based systems. A significant benefit of this approach is the availability of robust software and operating systems. These components have been significantly utilized on a variety of platforms by thousands of users around the world. Because of this huge diverse population, many "lurking" bugs have been exercised from the system.

Much of this software is standards based, which helps ensure interoperability. In a clustering environment, for example, the MPI libraries are used to share information between the various processes of the system. The official MPI standard defines the names, calling sequences, and results of subroutines that can be utilized by typically compilers, such as Fortran, C, and C++. These subroutines are used to send and receive messages designed to push the data out to available processors and to share the computational results. Remote sensing applications are well suited for this implementation since they have a high degree of data parallelism.

Another benefit is the utilization of open source software, such as the Linux operating system. One of the best methodologies found so far for developing consistently high quality software is the utilization of code reviews. In a review, other developers review and provide comments on the author's code. Open source software can be thought of as being continually reviewed. The source code of a computer program is made available free of charge to the general public. The rationale for this movement is that a larger group of programmers not concerned with proprietary ownership or financial gain will produce a more useful and bug -free product for everyone to use. Programmers have the ability to read, redistribute and modify the source code, forcing an expedient evolution of the product. The process of eliminating bugs and improving the software happens at a much quicker rate than through the traditional development channels of commercial software.

4. SPECTRAL LIBRARIES & EXEMPLAR SPECTRA

4.1 Spectral Libraries

In order to effectively exploit hyperspectral data, it is essential to determine what materials are being sensed within each pixel. Depending on the sensor and the mode of operation, the spatial coverage of a given pixel can vary greatly: from an area of less than a square meter to a square kilometer or more. Since each material has its own spectral fingerprint, spectral libraries must be developed to house the unique spectral signatures of all materials that may be in the scene of interest. Each pixel scene generally contains more than one material in the scene. Hence, the pixel composite spectral signature must be unmixed to determine the spectra of all the individual material constituants in the scene. Algorithms exist to perform pixel "unmixing" and classification operations, which strive to break up a composite pixel into its component parts and to define the individual materials contained within the composite pixel. It is possible for the spectral signature of a single material within a pixel to dominate the spectral characteristics of that pixel. Generally, however, we need to perform unmixing analysis. Hyperspectral data "unmixing or demixing" generally means detecting and identifying the presence and concentration of one or more specific materials in a pixel by the recognition of spectral signatures of these materials in the hyperspectral data. This is accomplished by determining which of the features of the materials are not shared by the other objects in the background scene. The analysis entails the determination of the spectral signatures describing the individual constituents or "endmembers" of the scene. The specrum from each pixel can then be represented as a weighted sum of endmember signatures whose positive coefficients add to one. The Naval Research Laboratory (NRL) has demonstrated a very fast and efficient method called the "Filter Vector Algorithm" for demixing when endmembers are known (Palmadesso, et al., 1995 and Bowles, et al., 1995). They provide an excellent overview of sevaral spectral demixing methods. This filter vector method and related techniques are collectively referred to as NRL's Optical Real-Time Adaptive Spectral Identification System (ORASIS). It also serves as an approximate method of constructing endmembers directly from the data when there is adequate diversity in the data set. Their method exploits the requirement of positive mixing coefficients, which implies that all the data points lie inside a simplex (convex hull). The vertices of a hypertriangle that contains the convex hull in n-dimensional space and is oriented in a way determined by the shape of the hull goves an indication of which endmembers are involved. These notions have been used (Boardman, 1995) to find approximate endmembers when no a priori knowledge of the material constituents of the spectral mixture is available. To be effective, these algorithms depend on access to a spectral library. It is critical that the spectral library contain sufficient and appropriate samples to assist the unmixing and classification algorithms.

4.2 Exemplar Spectra

Exemplar spectra can be used to increase the performance of the unmixing and classification algorithms. An exemplar spectra is a single spectra which is a composite derived from a variety of individual spectra. The goal is to define a single spectra, which can be used to describe an entire class of materials. This approach is not unreasonable. Even under highly controlled laboratory environments, it is standard scientific procedure to make and average multiple measurements to reduce noise. Also, spectral libraries typically contain spectra made from a variety of instruments taken under diverse environmental conditions and following different protocols. Even under practically identical conditions, noise and instrument calibration may adversely effect the measurements. To yield a useful representation, the sample used to calculate the exemplar spectrum must be adequate in number and must be a representative sampling of the parent class.

Let's look at an example of exemplar spectra. Let's say the desired outcome is to develop a exemplar spectrum of corn, which could be used to successfully classify corn anywhere in the remotely sensed scene. The exemplar may be created based on a set of different types of corn, taken at a variety of times, under different lighting and weather conditions, and with different instruments.

There are two typical uses of exemplar spectra. First, they can be used to filter undesired pixels out of the scene. For instance, if the user is interested in looking at the health and distribution of evergreen trees, it would be beneficial to filter any pixels that are classified as corn using our hypothetical exemplar. The second typical use is to identify areas where further analysis is required. For instance, if the study was worried about moisture content in corn plants, the exemplar could be used to determine at quick glance if there is any corn plants present in a particular scene. Once the exemplar is used to identify any pixels of interest, individual spectra can focus the analysis exclusively on these regions.

In either case, the use of the exemplar can greatly enhance the performance of the classification algorithm by significantly reducing the amount of individual spectra that need to be compared. Since the exemplar is designed to emphasize the spectral features associated with the general class of object, the amount of number of potential matches the algorithm needs to consider is greatly reduced.

4.3 Spectral Information Management Architecture

The Spectral Information Management Architecture (SIMA) is proposed as a plan for overcoming the limits in existing spectral libraries. Existing spectral libraries vary in format significantly. Since there is no community-wide common standard for spectral library format definition, sharing data from a variety of sources can become quite difficult. Also, the community-wide standard requires appropriate metadata. Metadata is a major, hot-topic buzz word in information systems these days. A recent issue of Fortune magazine ran a column titled "Without Metadata, Content is Just Bits." This article (Alsop, 2000) describes the importance of metadata and defines it as "Metadata is information about information". It offers a few examples of what metadata is and points out some generic examples of where metadata would be helpful. The most concise definition of metadata is that it is the "data about the data." Metadata provides supporting information that describes some particular aspect of the target data. What does this mean?

Let's say, for example, we have some data from the result of a chemical experiment. Relevant metadata associated with this experiment could include information like who conducted the experiment, what equipment was used to measure these results, when was the equipment last calibrated and by what techniques, and which experiment's methodology was used. One of the keys to any experiment, we all learn in grade school, is the concept of repeatability. A result is considered worthless unless someone else following the prescribed methodology can produce identical results. Metadata comes into play as the repository for the information about the conditions surrounding the target-data: how was this data collected? When and where was the experiment run? How many times the experiment was ran? Were all the results consistent and reliable?

Also, the shelf life of data can now be quite long. We are sure folks doing "bucket" sea temperature reading in 1910 would be very surprised indeed if they knew that almost a hundred years later their measurements were being used for climate analysis. Also, this information was collected long before the tools and techniques being currently used to analyse it were even created! Metadata also comes into play with regard to scientific data quality. In fact, without the associated metadata, the scientific target-data can become worthless. For example, suppose analysis is conducted based on observations from a particular type of sensor on several satellites. Now, assume that after the fact, it is determined that one sensor on one particular satellite was out of calibration and its results are meaningless over a certain timeframe. Without the metadata to identify those measurements that came from the questionable sensor, all data from that timeframe is suspect.

4.4 XML Methodology

The recommended methodology for developing a standard spectral library exchange format is extensible mark-up language, a.k.a. XML. XML is a World Wide Web Consortium (W3C) standard for exchanging structured data. XML is a mark-up language for documents containing structured information. Structured information contains both content (words, pictures, etc.) and some indication of what role that content plays (for example, content in a section heading has a different meaning from content in a footnote, which means something different than content in a figure caption or content in a database table, etc.) The tags which are used to mark-up the data provide a definition of what role that data plays in the domain. Also, since XML is a W3C standard, it is not a proprietary technology. Many tools exist to assist in the generation and parsing of XML documents. The benefit here is that the SIMA isn't locked into a single vendor's closed format.

Finally, XML can be considered a meta-mark-up language. This means that it allows the user to define syntax to create other domain-specific, semantic, structured mark-up languages. There are many examples of this being done in practice. For example, the mathML is a mathematical mark-up language done in XML, which is designed to document describing mathematics as a basis for machine to machine communication. Creating such common languages within an XML framework is quite easy. The developers can create an XML Schema or Data Type Definition (DTD). Both of these techniques create a standard that other XML documents can be compared against. Schemas and DTDs can be used to define the type and format of required and optional data and metadata. Also, XML document itself can contain references to other XML documents. This is valuable as a way of creating linkages between related objects within the spectral library. For example, an exemplar spectra could contain references to all the associate base spectra and to all the related metadata. This combination would enable the user to not only utilize the exemplar, but to also break it into its component parts.

5. CONCLUSIONS

We have presented an information technology (IT) data processing approach that we feel provides cost-effective methods to operationally process and exploit spectral information on board a remote sensing satellite in real time. This IT data processing approach offers a method to process and exploit, on board, huge amounts of spectral data without the need to have an expert spectral analyst in the loop. Key element of this method is the proposed Spectral Information Management Architecture (SIMA), which emphasizes the standardization of spectral libraries and the novel useage of exemplar spectra in hyperspectral data analysis. Exemplar spectra are single spectra, derived from a set of individual spectral signature measurements that capture the spectral essence of a class or subclass of materials. We believe the use of exemplar spectra and XML methodology in hyperspectral sensing applications will result in real time spectral analysis techniques that extend the forefront of current hyperspectral research.

REFERENCES

Alsop, S., 2000. Without Metadata, Content is Just Bits. Fortune Magazine, 142(13).

Boardman, J., 1995. Analysis, understanding and visualization of hyperspectral data as convex sets in n-space. SPIE Proceedings, Vol. 2480, page 14.

Bowles, J., Palmadesso, P., Antoniades, J., Baumback, M., Grossmann, J., and Haas, D., 1996. Effect of spectral resolution and number of wavelength bands in analysis of a hyperspetcral data set using NRL's ORASIS algorithm. SPIE Proceedings, Vol. 2821, page 171.

Bowles, J., Palmadesso, P., Antoniades, J., Baumback, M., and Rickard, L. J., 1995. Use of filter vectors in hyperspectral data analysis. SPIE Proceedings, Vol. 2553, page 148.

Hughes, G.F., 1968. On the mean accuracy of statistical pattern recognizers. IEEE Transactions on Information Theory, Vol. IT-14, pp. 55-63.

Palmadesso, P. J., Antoniades, J. A., Bowles, J., and Baumback, M., 1995. The use of filter vectors and fast convex sets for the analysis of hyperspectral data. ISSSR Proceedings.

Ridge, D., Becker, D., Merkey, P., and Sterling, T., 1997. Beowulf: harnessing the power of parallelism in a pile-of-PCs. IEEE Aerospace Proceedings.

Williams, M. and Armstrong, R., 2002. Building a Linux Minicluster using commodity components. http://www.linuxdevices.com/articles/

ACKNOWLEDGEMENT

This work was supported by the George Mason University Virginia Access (VAccess) Project and the TRW Foundation.