Proceedings of IEEE IWADC, Perugia, Italy, 2003

Onboard Feature Indexing from Satellite Lidar Images

Yang Cai
Carnegie Mellon University
ycai@cmu.edu

Yongxiang Hu
NASA Langley Research Center
yongxiang.hu-1@nasa.gov

Mel Siegel
Carnegie Mellon University
mws@cmu.edu

Sriram Janakiram Gollapalli
Carnegie Mellon University
sriramg@andrew.cmu.edu

Ashish R Venugopal
Carnegie Mellon University
arv@andrew.cmu.edu

Ulas Bardak
Carnegie Mellon University
ubardak@andrew.cmu.edu

Abstract

The purpose of the onboard feature indexing system is to perform pattern recognition and data compression onboard. We use the unsupervised machine learning algorithm k-means to classify the lidar profile data and generate an index dictionary. Then we train the Radial Basis Function neural network with the index dictionary on ground computers. Finally, we use the same RBF model for the onboard feature recognition and indexing. We implemented a prototype of the onboard computer with ZISC (Zero Instruction Set Computing) chips and FPGA (Field Programmable Gate Array) so that it takes advantage of intrinsic parallel computing and reconfigurability. We tested a set of 44K profiles as the training set to learn prototypical profiles that make up the indexing dictionary. With 64 indices, we reach a high compression rate 99.17% with reasonable error range. We found the required neurons are equal to the indices. We also compared our method to wavelet algorithm and found that it significantly outperforms the wavelet compression technique.

1. Introduction

The multispectral and imaging systems on a spacecraft can produce more data than can be analyzed by humans on Earth. With the growing number of satellites and improving spatial and spectral resolutions, Earth Science researchers have to cope with massive data. For example the down-link rate will be 1 GB per second in 2010 and up to 10 GB per second by 2020. Much of the data are redundant and irrelevant to specific purposes. Scientists spend up to 70% of their time on preprocessing the imperfect and redundant data. Delays during the analysis hinder the discovery of disastrous situations or significant atmospheric features.

The concept of space-borne data processing systems has emerged since early 90s. For example, Jet Propulsion Lab proposed strategies, such as the technologies needed for dramatic data reduction, autonomous event recognition and response, hyper-spectral and radar data onboard processing. [1] A space-borne sensory system should be able to not only acquire these data, but also to evaluate them, act upon their scientific content, summarize their scientific significance and disastrous situations.

Our approach is inspired by human perception in daily life. Humans have natural ADC systems that efficiently organize information in our memory. Humans are capable of describing complex things with simple *feature indexing,* which can dramatically reduce information. For example, we often describe a traffic intersection with a letter ‘T’, or ‘X’. Thus we compress an image (e.g. 1 megabyte) to a letter (e.g. 1 byte). On the other hand, we often retrieve information with feature indexing in our memory. Cognition scientist Herbert Simon and his colleagues studied this phenomenon and developed a computer model EPAM (Elementary Perceiver and Memorizer) [2] to simulate how people learn and recognize features in words and images.

The feature indexing method has been used for video digesting and compression. [3] As the Media Lab Director Nicholas Negroponte predicted that the new technology might have enormous impact on our life in future. [4]

The objective of this study is to develop a prototype of the image feature indexing system that can be applied to onboard physical discovery. While we implement the prototype, we also consider the constraints of onboard environment, e.g. computing power, weight (bit per gram) and reconfigurability.

In this paper, we focus on satellite lidar data only because the data stream is relatively small to begin with and the data directly contribute to disastrous aviation weather detection. Lidar (which stands for *light detection and ranging*), like radar, is an active remote sensing technique. It involves the use of pulses of laser light directed from space toward the ground, measuring the time of pulse return. The return time for each pulse back to the sensor is processed to calculate the variable distances between the sensor and the various objects present above (or on) the ground. The algorithm developed can be transformed to other earth observatory systems, such as hyperspectral image, SAR, etc. Fig.1 shows an example of the lidar profile image. In this image, each vertical line is a *profile*. Each pixel represents the reflectance value. Similar to a fingerprint, the spectral data and lidar/radar data of each imager pixel or lidar/radar profile from earth observations can be divided into hundreds to millions of classes, depending on the complexity of the data. Each class of target has its unique spectral and spatial characteristics. Each class can be labeled with a simple integer index. Instead of sending down detailed spectral data and vertical profiles, each of which might be in the order of megabytes and may appear thousands of times during a satellite mission, we only down-link the integer index. The detailed spectral data and vertical profiles, if needed, can be easily looked up in a standard library with the down-link index. We call this approach "*Feature Indexing*".

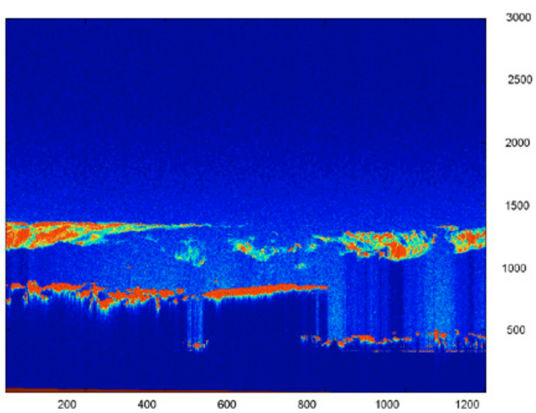

Fig.1 Sample of Lidar Profile Image

Feature indexing is different from "general-purposes" data compression methods, such as wavelet, or MP3 algorithms. Since it is based on scientific knowledge. For a satellite instrument, the number of types of distinguishable atmospheric features is limited. The spectral and spatial information of each type of feature can be characterized. Earth Science sensors repeatedly detect each feature many times during a satellite mission. The feature indexing technique effectively eliminates the redundancy of same spectral and spatial information.

We assume that each lidar image has highly repeatable and *limited number of classes*. Taking two days worth of space shuttle lidar (LITE) data for example, we found limited amount of physical property characterizations. It contains around 120,000 different lidar shots, 65% of lidar profiles have either no objects at all or only have one layer of cloud or aerosol object. 97% of lidar profiles have three layers of objects or less. In light of this property, we can use clustering methods to group atmospheric features into limited number of classes on the ground.

In this paper, we present the system architecture, the machine learning algorithms, hardware realization, and benchmarking for the performance of our approach against wavelet algorithm.

2. System Architecture

Feature indexing requires an index generator and index recognizer. The index generator classifies features into limited number of indices. The index recognizer retrieves indices according to features in the image. In our case, we assume that the onboard computer has very limited computing power so that we do feature classification on the ground computer which has significant computing power. We only use the onboard computer to do the indexing or feature recognition, which needs less computing power.

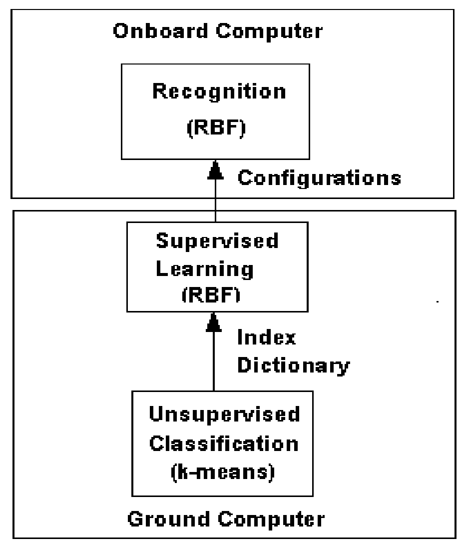

Fig. 2 The Machine Learning Scheme

Unsupervised learning algorithm is used because it is hard to train the pattern classification manually with enormous lidar data samples. The output of the classification process, k-means, is an '*index dictionary*,' which is used to train the recognition model on a simulator with a ground computer. Then the trained recognition model is loaded to an onboard computer. The advantage of this approach is that the two can share the same configuration file so that the onboard model can be updated from the Earth by transferring the configuration file. Once the onboard system recognizes a lidar profile, it will output a feature index for it and pass it to the Earth.

The set of 44K profiles serves as the training set to learn a set of prototypical profiles that make up the indexing dictionary. We use the common k-means clustering technique, a generative method that assumes that the complete training set is generated from a set of k-distributions. Its goal is to find the means of these distributions (in the 3000 feature space, this corresponds to k prototypical profiles) to maximize inter distribution similarity (distance between elements said to have been generated from a given prototype) and minimize cross cluster similarity. The algorithm consists of a re-estimation procedure as follows. First, the data points (the profiles) are assigned at random to the K sets. Then the centroid (average profile) is computed for each set. These two steps are alternated until a stopping criterion is met, i.e., when there is no further change in the assignment of the data points, minimizing the sum of squares criterion below where K is the number of clusters, Sj defines a current profile cluster, and x is the 3000 feature profile.

$$J = \sum_{j=1}^{K} \sum_{n \in S_j} \|x_n - \mu_j\|^2$$

Creating the dictionary profiles constitutes the training step within our technique. Each of these profiles is used to initialize a Neural Network that can perform the recognition step of associating a new profile to one of these dictionary profiles. The Neural Network model uses Radial Basis Functions [5] that measure the similarity of a new profile to the set of existing prototype clusters. While the feature classification is processed on Earth, the feature recognition model is implemented onboard. We use Radial Basis Function (RBF), which is similar to the feature classification model, for feature recognition. The RBF neural network is a three layer network where each input node, corresponding to the component of a feature vector is connected to every node of the second layer (hidden layer), each node of the hidden layer is connected to one output node which corresponds to an index. Fig. 4 shows the network structure.

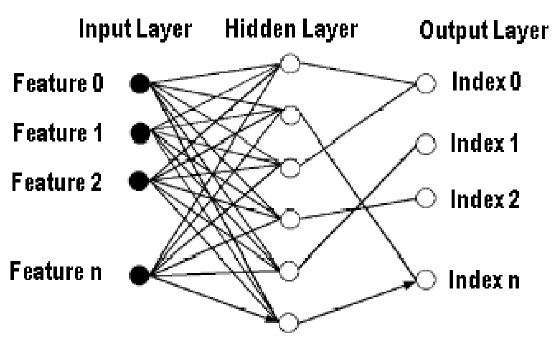

Fig.4 The RBF Network

3. Hardware Implementation

To solve the down-link bottleneck problems, we have to implement the feature indexing system on an onboard computer. ZISC78 (Zero Instruction Set Computing) chip from Silicon Recognition is used to implement the RBF (Radial Basis Function) algorithm. Each chip has 78 neurons. It is not limited in volume due to its parallel architecture. In this study, two ZISC78 chips are used for the rapid prototype.

To make the onboard computer reconfigurable, Field Programmable Gate Array (FPGA) systems are used in this design. The FPGA chip SPARTAN II from Xilinx contains 50K gates. The hardware scheme is illustrated as Fig.4.

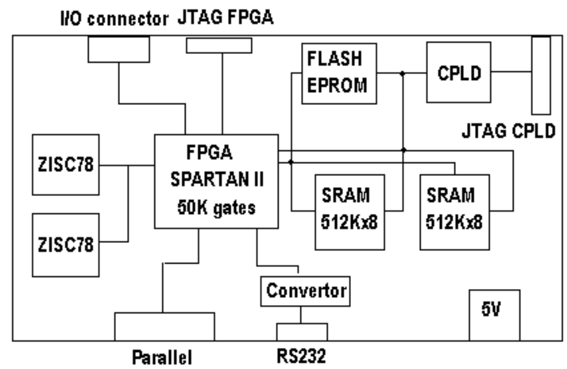

Fig. 5 Hardware for Onboard Recognition

The prototype system has limited capacity. Each data instance has up to 64 features, where each parameter can take values between 0 and 255. These limitations are inherent in the chip design.

4. Experiment Results

We tested a set of 44K profiles as the training set to learn prototypical profiles that make up the indexing dictionary. Considering each profile as a 3000 element feature vector, we cluster this data with k-means model to have 64 feature indices. Then we train the recognition model (RBF) on a simulator that has far larger capacity. Finally, we load the learned RBF configuration file to the onboard computer for recognition tests. On the ground computer, the profile set is reconstructed by selecting the dictionary profile that corresponds to the index received from the satellite. Since the prototype represents the closest profile to the unseen profile, an inherent loss occurs. Each reflectance value in the recovered image is compared to its original reflectance value. The difference between these values is squared and averaged over each profile for the measurement of loss due to the indexing operation. Also, the compression ratio is defined as the percentage of size reduction.

We use two ZICS78 chips for index recognition. After loading the base classes onboard, we measured that it took about 0.078 seconds to recognize each profile.

Table 1 Indexing Performance

Index	Error Rate	Comp. Ratio	Neuron
16	0.031082	99.95%	16
32	0.029709	99.68%	32
64	0.028133	99.17%	64

Table 1 shows the performance of the indexing test at different resolutions. In general, the data compression ratio, which is defined as the percentage of data reduction, is as high as 99.95%. Increasing number of indices reduces compression ratio in a very subtle level. On the other hand, as the number of indices increases, the error rate, which is defined as the distance of the reflectance values between original and recovered per point in average, decreases. In addition, we found that the neurons required for recognition is proportional to indices. In this case, it is in 1:1 ratio.

We compare our approach to wavelet compression, a popular lossy data compression algorithm. The wavelet compression is achieved through the use of Discrete Wavelet Transform (DWT). It is an orthogonal transform with a basis function that's localized both in time and space. Through it's application a wavelet coefficient matrix is produced. Depending on how much energy (roughly - the amount of variance in the data) we want to preserve, we can keep applying this matrix to the trimmed data matrix "smoothing" it every time. Note that the more we trim and reapply the DWT coefficients to the data, the smoother our data will be and more and more details will be lost. Table 2 shows that our approach has a better compression ratio at any given error rate.

Table 2 Wavelet Performance

Energy Retained	Error Rate	Comp. Ratio
91.66%	0.1262	94%
99.00%	0.0392	75%
99.86%	0.0166	50%

5. Conclusions

In this paper, a design prototype of the onboard feature indexing system is presented. The purpose of the system is to perform pattern recognition and data compression onboard. Instead of transmitting raw data from sensors to Earth, the onboard system recognizes the physical features from multiple sensors and then sends the feature indices or alerts to the Earth. On the ground, the profile data can be recovered for aviation alerts or further atmospheric studies. Although we focus on the lidar sensor in this project, the technology can be extended to other sensors.

We use the unsupervised machine learning algorithm, k-means, to classify the lidar profile data and generate an index dictionary. Then we train the Radial Basis Function neural network with the dictionary on ground computers. Finally, we use the same RBF model for the onboard feature recognition and indexing.

We implemented a prototype of the onboard computer with ZISC (Zero Instruction Set Computing) chips and FPGA (Field Programmable Gate Array) so that it takes advantage of intrinsic parallel computing and reconfigurability.

The NASA LITE data are used for experiments. We tested a set of 44K profiles as the training set to learn prototypical profiles that make up the indexing dictionary. With 64 indices, we reach a high compression rate 99.17% with reasonable error range. We found the required neurons are equal to the indices. We compared our method to a popular lossy data compression algorithm, wavelet and found that it significantly outperforms the wavelet compression technique.

However, the speed of the prototype board has not been fast enough: 0.078 seconds for indexing one profile. We haven't fully tested the reconfigurability of the FPGA unit yet.

We are encouraged from this preliminary experiment. It is a trend that hardware and software for ADC and image processing will merge into so called *configureware,* which can significantly help the onboard inverse physics and discovery.

Acknowledgement: This study is sponsored by Creativity and Innovation Initiative in NASA Langley Research Center, VA, USA and Silicon Recognition, CA, USA.

Reference

[1] Lansing, F., L. Lemmerman, A. Walton, G. Bothwell, K. Bhasin and G. Prescott, 2002: Needs for Communication and Onboard Processing in the Vision Era. 2002 IGARSS conference proceeding.
[2] Feigenbaum, E.A. and Simon, H.A. EPAM-like models of recognition and learning. In Cognitive Science, 8:305-36 (1984)
[3] Golshani, F., Dimitrova, N., Multimedia Tools and Applications, May 1998, vol.6, no.3, pp. 289-312(24)
[4] Brand, S. The Media Lab, Penguin Books, 1988
[5] Bishop, C. Neural Networks for Pattern Recognition, Oxford, 1995
[6] Graham Bothwell, Loren Lemmerman, Amy Walton, 2002, On-Board Processing, NASA Earth Science Enterprise Technology Planning Workshop, January 24-25, 2001
[7] Tessier, R. et al, Reconfigurable Computing for digital signal processing: A survey, The Journal of VLSI Signal Processing, May, 2001, vol. 28, no. 1 and 2
[8] Silicon Recognition Corporation: www.silirec.com