Classification of Spectral Imagery Using Genetic Programming

Patrick J. Rauss
U.S. Army Research Laboratory
AMSRL-SE-SE
2800 Powder Mill RD
Adelphi, MD 20783-1197
prauss@arl.mil

Jason M. Daida
Atmospheric, Oceanic and Space Sciences Dept.
University of Michigan
2455 Hayward Ave.
Ann Arbor, MI 48109
daida@umich.edu
734-647-4581

Shahbaz Chaudhary
Atmospheric, Oceanic and Space Sciences Dept.
University of Michigan
2455 Hayward Ave.
Ann Arbor, MI 48109
chaudhar@umich.edu

ABSTRACT

This paper describes an initial use of genetic programming as a discovery engine that derives two sets of information from hyper-spectral imagery. The first consists of a set of classification algorithms learned from the data. The second consists of reduced subsets of the most germane bands for use in a given classification, since not all spectral bands are of use in deriving a particular classification algorithm. Currently, there are only a few techniques to discover which bands would be the most useful for a specific classification task. We describe the design of a prototype system and discuss its efficacy on a novel data set from an imaging system that uses an acoustically tuned optical filter. The data preprocessing, training data extraction, training data formatter, GP implementation, and classification image generation tasks are detailed.

1 INTRODUCTION

Genetic Programming (GP) offers a unique way to process hyper-spectral imagery. Being an adaptive learning technique, GP requires few, if any, assumptions about the data or features of interest. Since there is no requirement that the user be an expert in spectroscopy, GP offers non-specialists a tool to aid in development of hyper-spectral applications.

An additional advantage is that GP generates explicit equations. With a classification equation in hand, a user can see what bands are being used and how. This allows GP to be used as a discovery engine, thereby giving the user insight into what spectral bands are important for an application, as well as showing how these bands are being used to obtain the desired classification.

1.1 PROBLEM STATEMENT

Our goal is to construct a computer-assisted-design system for the discovery of classification algorithms and germane band information, using a small number of examples of sensor data from user-defined classes.

The potential value of hyper-spectral imaging is high, but the number of existing applications is small for several reasons. First, hyper-spectral imagers are relatively new. Not surprisingly, applications unique to hyper-spectral data have yet to be identified. Many current applications are based on work with multi-spectral imagery and often do not exploit the strengths of hyper-spectral data, resulting in only incremental improvements over multi-spectral applications. Second, most existing tools for exploiting hyper-spectral data require extensive knowledge of spectroscopic techniques, which is uncommon for many of the potential users. Third, existing remote sensing techniques often do not exploit the fine spectral resolution of hyper-spectral data. Fourth, tools to investigate new uses are not common or require expert knowledge. Finally, the data files themselves can be very intimidating, since even small collections can represent gigabytes of data.

Using GP on hyper-spectral data has the potential to address these difficulties by supplying a relatively simple tool to allow one to conduct application experiments. This would hopefully encourage the development of new applications for hyper-spectral imagery. GP would be supplied with all the band information during training, thereby allowing the exploitation of the finer spectral resolution of the data. Spectroscopic techniques are not required to design classification experiments. If a poor choice of classes were made, GP would likely fail to evolve a solution. GP tools can be relatively easy to use, and would not require expert spectroscopic knowledge to develop useful applications. Finally, GP results in a specific equation. This equation can be evaluated to attempt to understand the underlying physics of the classification and to reduce the storage capacity needed for application data. Reduced storage can be accomplished by determining the spectral bands necessary for the particular classifications and allowing the use of smaller subsets of the data to reside in active storage.

1.2 HYPER-SPECTRAL IMAGERY

Multi-spectral imagers for remote sensing have been in operation since the mid-1970s. Since that time, they have proven their value in many applications, such as crop yield estimation, forest harvest monitoring, evaluation of soil conditions, crop and timber type identification, regional planning, and cartographic updates. However, since multi-spectral data inherently have low spectral resolutions, their usefulness is limited to discriminating between objects with significantly different and broad spectral characteristics. Objects that differ only at finer spectral resolutions cannot be distinguished easily, if at all.

Although current hyper-spectral imagers are mostly experimental research systems, they are becoming more common. A number of airborne systems exist, and several satellite systems are scheduled for launch in the near future.[1] The term *hyper-spectral* refers to the large number of spectral bands and the narrow width of each band. Table 1 highlights some of the key differences between multi- and hyper-spectral systems.

Table 1: Comparison of multi and hyper-spectral systems.

System features	Multi-spectral	Hyper-spectral
Number of bands	7 (LandSat TM)	224 (AVIRIS) 112 (AOTF)
Spectral Resolution (nm)	450 to 1250 (LandSat)	10 (AVIRIS) 10–20 (AOTF)
First deployment	1972 (LandSat 1)	1987 (AVIRIS) 1997 (AOTF)
Common Deployment	Satellite	Airborne (AVIRIS) Ground (AOTF)
File size (Mb) or 640 by 480 image	2.1 (Landsat TM)	68 (AVIRIS) 34 (AOTF)

Visualization of hyper-spectral data is also a difficulty. One common visualization technique is a spectral image cube, a three-dimensional data structure. *X* and *Y* coordinates represent the standard spatial representation of an image; the *Z* coordinate represents the spectral dimension, as shown in figure 1.[2] Each slice in the *X*-*Y* plane is an image of the target scene at a particular wavelength. A slice parallel to the *Z*-axis represents the spectral signature along that slice. The vector of values taken along a line parallel to the *Z*-axis represents the spectra at that particular point.

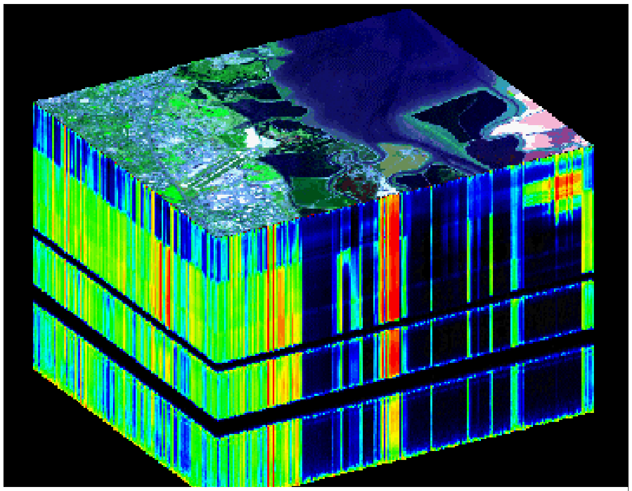


Figure 1:Hyper-spectral image cube of AVIRIS data from NASA.

1.3 PREVIOUS WORK IN CLASSIFYING SPECTRAL IMAGERY

There are a number of well-established techniques for exploiting the spectral content of multi-spectral imagery (Tassel Cap [Kauth 1976] [Crist and Cicone 1984], Atmospherically Resistant Vegetation Index (ARVI) [Kaufman and Tanre 1992], Normalized Difference Vegetation Index (NDVI) [Goward et al. 1991], and Principle Component Analysis (PCA) [Jiaju 1988]). However, when these techniques are applied to hyper-spectral data, there are several difficulties. Many of these techniques are forms of band ratioing. NDVI, for instance, is simply a normalized ratioing of a near-infrared band and a red band, which highlights healthy vegetation imagery by exploiting broad spectral features of chlorophyll. Determining useful band ratios in multi-spectral imagery is relatively simple. There are *n(n-1)* possible ratios (*n* is the number of spectral bands), and half that if we ignore the inverse ratios (*band 1/ band 3* is the inverse of *band 3/ band 1*). For LandSat data (ignoring the thermal band) there are 6(6-1)/2 = 15 possible ratios. It is a simple matter to visually examine all of these ratios to determine which ones highlight the desired classifications. It is even relatively simple to use these ratioed images as inputs for false color images, and visually investigate them for the desired highlighting. However, for hyper-spectral imagery the number of

[1] The Moderate Resolution Imaging Spectroradiometer Proto-flight model was launched in Dec. 1999. First light data was collected in February 2000. With 36 bands of bandwidths from 20 to 300 nm, it is considered a hyper-spectral system by some.

[2] Hyper-spectral cube from http://rst.gsfc.nasa.com/Sect13/Sect13_9.html

combinations rapidly becomes unmanageable. The data used for this project consists of 28 spectral bands, yielding 378 ratio images to examine. For the AVIRIS system of 224 bands there are 24,976 unique ratio images. Clearly some method of reducing the spectral search space is necessary if one is to pursue this approach for hyper-spectral data.

One method for reducing the dimensionality of a search space is principle component analysis (PCA). PCA is a common tool used in classifying multi-spectral imagery. The technique works best when knowledge of atmospheric conditions is available (for radiometric corrections to the data). This knowledge represents a nontrivial effort in data correction that at best requires corroborative measurements. Such measurements are usually obtained in fieldwork using *in-situ* instrumentation. Furthermore, the relevance of the corroborative measurements usually persists only for the time of the data collection; the measurements do not generally apply to any other time or location. Since corroborative data sets are difficult to obtain, most multi- and hyper-spectral image data do not contain atmospheric measurements that would allow accurate atmospheric corrections to support PCA.

We note that for low-spectral-resolution data — from well-studied multi-spectral instruments such as the NASA LandSat thematic mapper—atmospheric modeling programs exist (e.g. LowTran and ModTran). Such models can be used to obtain the required atmospheric corrections for a given image. However, because of the much higher resolution of hyper-spectral systems, these models are insufficient for accurate correction. A higher spectral resolution model does exist (HighTran), but with higher spectral resolution, the number of potential contaminants increases significantly and cannot be accurately modeled, which reduces the usefulness of any PCA done with data corrected with this atmospheric model.

Since 1990, the U.S. Geological Service has been developing a system called Tetracorder (formerly known as Tricorder) [Clark 1995] for classifying hyper-spectral data. Tetracorder replicates the process an expert in spectroscopic analysis would follow, using over 120 high-quality laboratory spectra of environmental materials for comparison. It uses a least-squares curve fitting method to compare the pixel spectra with the library spectra to find the best match. However, Tetracorder requires expert understanding and experience in spectroscopic analysis for proper use on any but the simplest of classification tasks. It is very sensitive to the accuracy of the atmospheric correction, as comparisons of materials with similar spectral curves would be greatly affected by inaccurate correction. To match a particular imaging system, Tetracorder must be modified by an expert. Neither atmospheric data nor details on the imager are readily available with existing hyper-spectral data sets. It also appears that a significant effort is required to add new materials to the system library, thereby limiting the method's usefulness in developing new or radical applications.

Adaptive learning techniques, like neural nets and GP, are attractive because such techniques can adaptively compensate for these radiometric corrections, without the user having to explicitly state (or even be aware of) what those corrections are. However with neural techniques, it is difficult to extract the actual classification equation for analysis. On the other hand, GP supplies both the algorithm and the subset of useful bands. Both have geophysical meaning: that is, simply having the appropriate bands available says something of the physics involved with the phenomena to be studied. Having the equation shows why and how the bands are significant.

1.4 THIS EFFORT

To develop and use new hyper-spectral applications requires some level of automated processing. A challenge now is to develop automated analysis techniques to take advantage of hyper-spectral systems, to assist users in using this new capability, and to process the vast amount of data that are and will soon be available. In addition, as hyper-spectral applications are relatively new, it would benefit the remote sensing community to have available development tools to aid in determining what spectral features are useful for particular applications. GP has the potential to be a key tool in image processing and feature extraction [Tackett 1993] [Brumby et al. 1999] [Howard and Roberts 1999].

Encouraging results have been obtained from GP techniques used on other types of remote sensing data, such as: synthetic aperture radar [Daida et al. 1996a, 1996b] and infrared line scanner data [Roberts and Howard 1999]. We report on our initial attempt to use GP to generate classification equations suitable for processing an entire spectral image. The goal being, to use only a small number of training pixels from that image to generate a classification equation that generalizes well.

In the next section, we present the system design and briefly discuss each segment. Section 3 discusses the data used for development. Section 4 presents results from the first complete run on a single image. We present our conclusions in section 5.

2 SYSTEM OVERVIEW

The goal for the first phase of this project was to demonstrate two things: first, that it is possible to use GP techniques to classify pixels in a spectral image, and second, that it is potentially useful to do so. Therefore, we set out to show that this could be done using as many existing software tools as possible. We felt that there was little point in designing a streamlined, user-friendly, optimized system from scratch if it was not going to be useful.

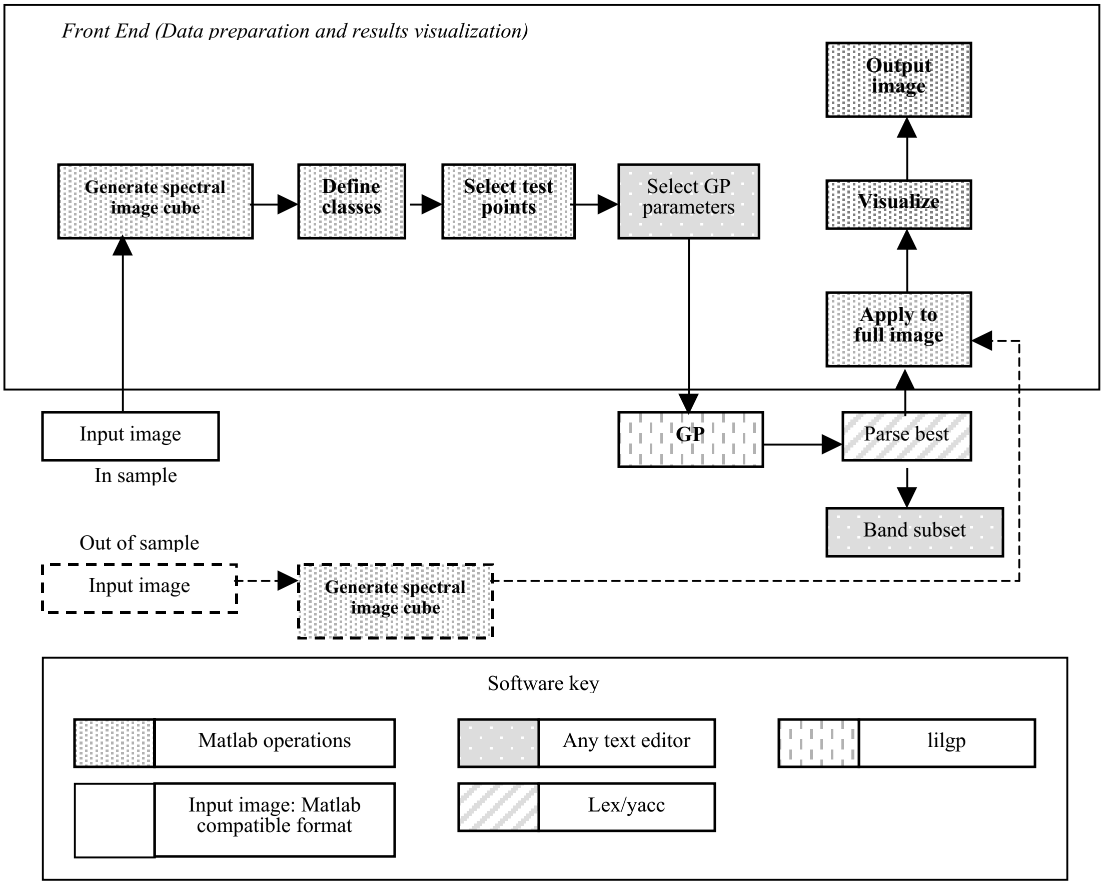


Figure 2:Data flow

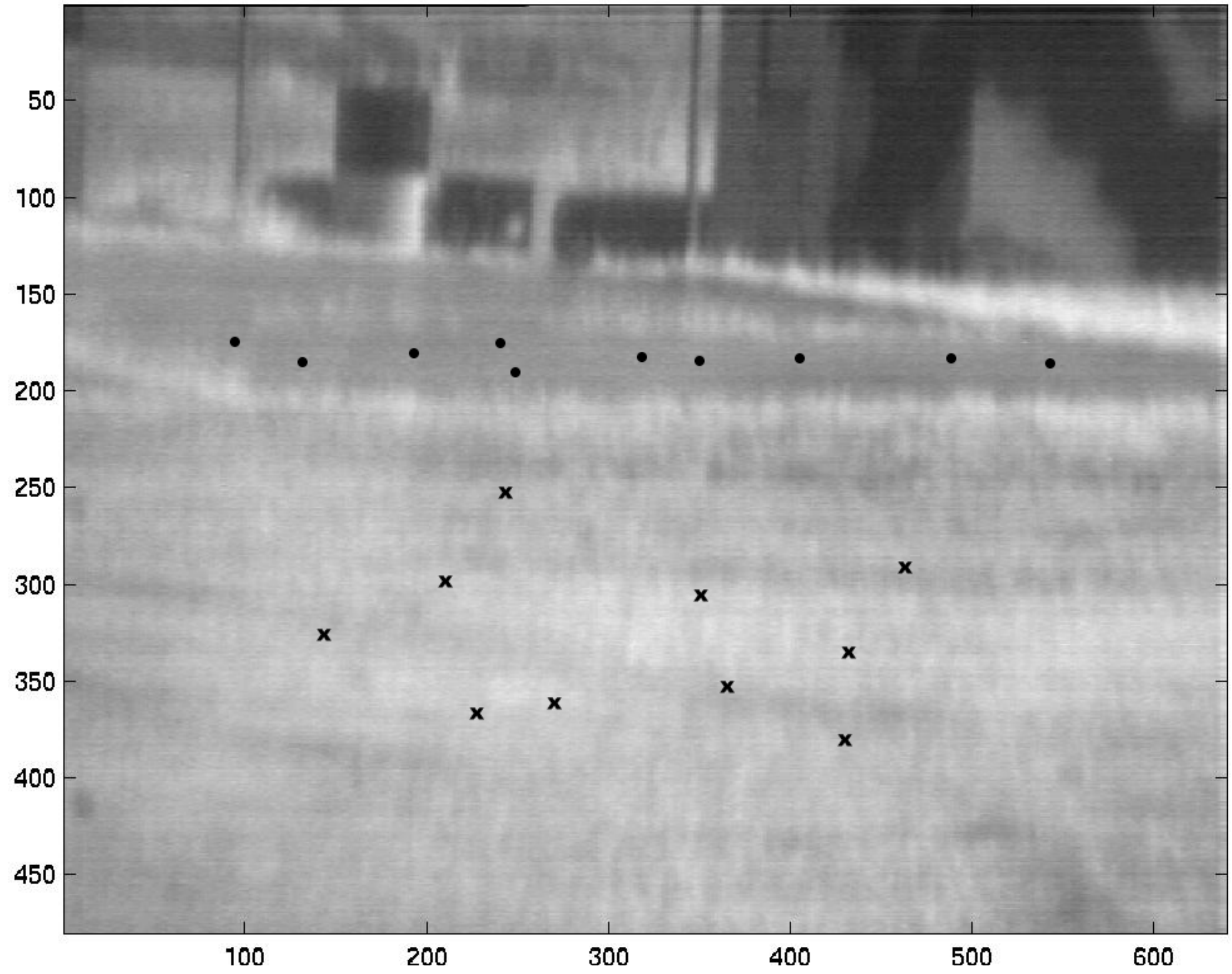
Figure 2 shows the data flow of the system we designed for evolving a classification equation. (For comparison, see also [Bersano-Begey et al. 1997].) The GP portion of the system was built around the lilgp software package [Zongker and Punch 1995]. We modified lilgp to recognize multiple, independent variables (the bands). To limit the amount of development required with lilgp at this stage, we used only the simplest of operators: +, -, *, and protected division (/). To avoid modifying the core code of lilgp, we added the capability to recognize multiple variables through modifications to the user functions.

Other tools were required to extract training data to present to lilgp, as well as tools to use lilgp's output to process a complete image. Figure 2 indicates that most of this is done using Matlab (by Mathworks). The first task for preparing the data is to import the spectral data into Matlab and to form the data into a spectral image cube for visualization.

Once the spectral image is organized into a spectral cube in Matlab, the data may be visualized in a number of ways to aid in determining the number and location of the different classes of objects to be discriminated. Optimally, a spectral image would have areas that can be "truthed." Truthed areas are locations on the image where field notes or spectral sampling would have been collected to obtain positive identification of these specific locations. Once the classes are defined, a Matlab tool is used to select training pixels representative of each class. The spectrum of each pixel is extracted, normalized, and saved to a file in the format that lilgp will accept.

We normalized individual pixel spectrum to a range of -1 to 1 to minimize against the possibility of GP using intensity as a discriminant (intensity varies greatly depending on changing illumination, which would not remain constant even in the same image, due to shadowing effects). Normalization should result in a more generalized classification equation, which can be used on a broader set of data. The normalized data are saved as a text file that is formatted for acceptance as a lilgp input file.

Next, a text editor is used to modify the parameter file of lilgp to define the number of generations, starting population size, various breeding parameters, and maximum tree size. Lilgp is then run with the extracted training data. For this initial test, the fitness function used was a simple zero threshold. A correct response was indicated if the equation resulted in a value greater than zero for the desired target class or less than zero for the undesired classes. Once lilgp has evolved an equation of suitable performance, this best equation can be used to generate a classification image for an entire image or data set.



Some reformatting of the resulting equation is required before our image-processing tool can be used to process an image. Lilgp outputs its best equation in a root-lhs-rhs style, resulting in equations that look like

```
(* (/ (+ (+ b1 b2) b3) (- (+ b4 b5)
          b6))(/ b7 b7)),
```

where b stands for band.

Matlab, however, requires a standard mathematical format (lhs-root-rhs). Fortunately, a parser was available elsewhere in our research group [Daida et al. 2000]. The parser consists of lex and yacc routines (lex and yacc are standard programs common in most UNIX C programming packages). Lex is used as a token generator that reads an equation, extracts tokens, and executes specific C instructions upon finding certain tokens. The lex code was designed to evaluate and simplify instances of *1, $/1$, $+0$, and -0, as well as checking for instances of *var/0*, which is replaced with 1, since protected division has been used. This generates a lexical analyzer for use with yacc. Yacc creates the program that performs the actual parsing and outputs the reordered (and possibly simplified) equation. This results in the above lilgp output being parsed and transformed as

$$((((b1)+(b2))+(b3))/(((b4)+(b5))-(b6))).$$

This equation is then copied into a Matlab script. The script loads the full image cube, applies the equation to every pixel in the image, builds a classification image and displays the resulting classification image, like that shown in figure 6.

3 SPECTRAL DATA

The spectral image used during this software development is from an experimental acousto-optical tunable filter (AOTF) imaging system developed by Carnegie Mellon Research Institute [Denes et al. 1997]. The proprietary

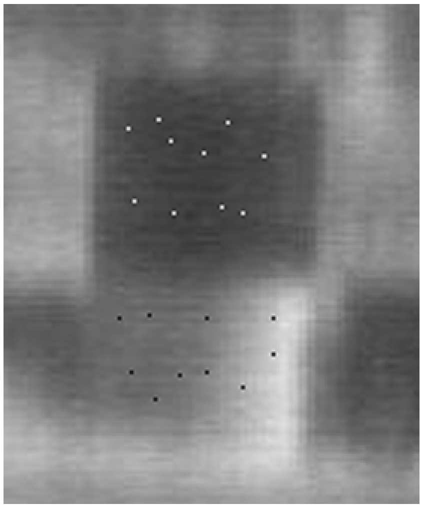


Figure 4: Enlargement of barrel with regions highlighted where classes *barrel-bright* and *barrel-dark* were extracted (approximate)

image was collected for the U.S. Army Research Laboratory (ARL) [Gupta et al. 1999] [3] in 1998.

The spectral imager is a visible to near-IR system with an overall spectral range from 450 to 1000 nm. An AOTF filter allows the instrument to be selectively tuned to collect multiple spectral images with bandwidths as narrow as 10 nm. The system also has an electronically controlled liquid crystal phase retarder, which allows collection of polarization information in each spectral band. For the work reported here, we used a single polarization image cube of 28 bands, where each band is approximately 20 nm wide. The sensor recorded each spectral band as a bit-mapped image and stored each image as .bmp files. These files were then read into Matlab and assembled into the spectral image cube.

Figure 3 shows band 19 of the spectral image cube. Several of the classes used are visible in this image. The circles and *X*'s depict the approximate location of the ten pixels extracted for training spectra for these two classes. The *X*'s represent the ten pixels chosen as the class *grass*. The circles represent the class labeled as the class *pad*. It appears to be rectangular pad of asphalt, concrete, or bare earth (perhaps a parking area) in front of the building.

Figure 4 shows an enlargement of the barrel. The upper dark portion of the barrel has been defined as the class *Barrel-dark*. The bright pixels show the approximate location of the ten pixels extracted for the training set for this class. The lower section of the barrel is defined as the class *Barrel-bright*. The black pixels in this area show the approximate location of the ten pixels chosen for this class.

Figure 5 shows an enlargement of the left side of the building. This area is where the ten pixels of the class *wall* were extracted. The crosses show the approximate location of these pixels.

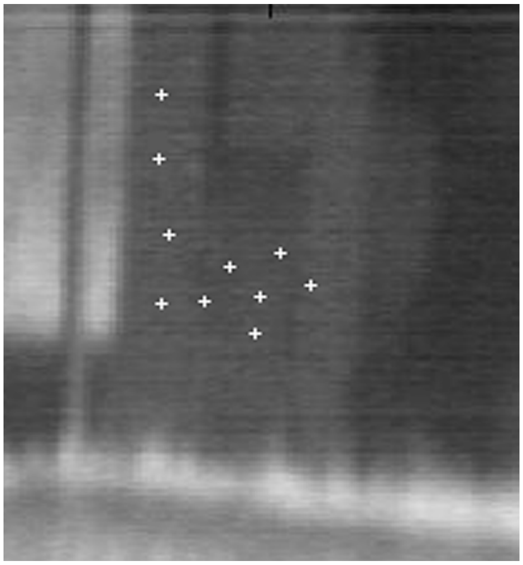


Figure 5: Enlargement showing wall area where pixels for the *wall* class were extracted. Crosses show approximate location of pixels used.

4 GP RESULTS

These results are from the first successful run of the complete system. The classification desired is what we assume would be the simplest separation (i.e. grass vs. everything else.) [4]

Using the equation from this run, we successfully processed the full spectral band image with the GP equation. Lilgp was instructed to accept class 1 (grass) as positive values (>0) and all other classes as negative (<0). Most of the GP parameters are similar to those mentioned in chapter 7 of *Genetic Programming: On the Programming of computer by Means of Natural Selection* [Koza 1992]: population size = 100; crossover rate = 0.9; replication rate = 0.1; population initialization with ramped half and half; initialization depth of 2-6 levels; and fitness-proportionate selection. Other parameter values were maximum generations = 100 and maximum tree depth = 20.

The following tree is a best-of-run individual. GP obtained a perfect score of 50 correct classifications on the 10 grass pixels within the desired class and the 40 other pixels (10 pixels from each of the remaining four classes) within the undesired class (again *b* stands for band).

```
(+(+(/(*(* b8 b3)
        (/ b8 b15))
      (/(/(- b23 b14) b22)
```

[3] This single image was made available to Mr. Rauss to support his masters research project at the University Of Michigan by his employer, the U.S. Army Research Laboratory.

[4] [NOTE: To effectively obtain the best solution, many more runs should be performed, as well as numerous runs to generate classification equations for the other classes.]

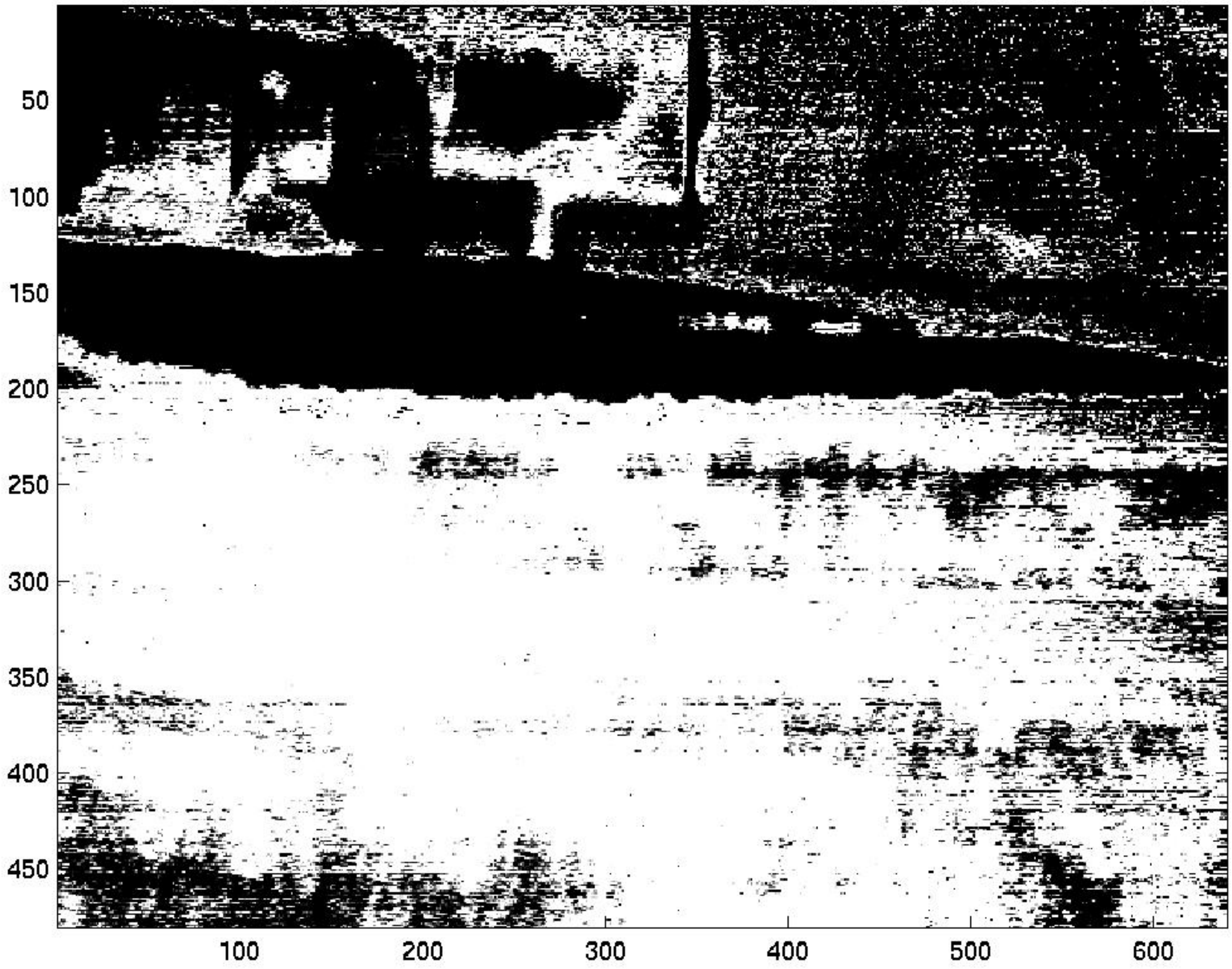


Figure 6: Classification image for class *grass*

```
   (/(*(* b8 b3)
       (/ b8 b15))
     (/(/ b21 b22)
       (+(-(+ b8 b26)
           (/ b5 b4))
         (+(/ b25 b6)
           (* b22 b2)))))))
 (-(+(- b27 b21)
   (+ b3 b18))
 (*(- b4 b28)
   (/ b5 b1))))
(/ b20 b28)).
```

After this file is parsed, the equation reads as follows:

$$((((((b8)*(b3))*((b8)/(b15)))/((((b23)(b14))/(b22))/((((b8)*(b3))*((b8)/(b15)))/(((b21)/(b22))/((((b8)+(b26))-((b5)/(b4)))+(((b25)/(b6))+((b22)*(b2))))))))+((((b27)-(b21))+((b3)+(b18)))-(((b4)(b28))*((b5)/(b1)))))+((b20)/(b28))).$$

Applying this equation to the entire image results in figure 6.

Note that the gray scale is set to saturate at values below -1 and above +1. White should indicate the class *grass* and *non-grass* areas should be dark.

5 CONCLUSIONS

We have taken the first steps in demonstrating that GP techniques can be useful for spectral image processing and classification. This single initial attempt has shown that classification images can be generated.

One useful aspect of a GP approach (as opposed to say a neural network approach) was the ability to obtain an actual equation. This equation could be analyzed and used for purposes of data mining. For example, for the image shown in this paper, the GP equation obviously did not use all 28 of the available bands. For the grass class GP is only using 18 of the 28 bands: 1, 2, 3, 4, 5, 6, 8, 14, 15, 18, 20, 21, 22, 23, 25, 26, 27, and 28. Using this approach with the other classes may similarly reveal that only particular bands are useful.

We have obtained a classification equation for separating grass from the other four classes defined in the image. Figure 6 shows grass in the foreground, as expected; though there were areas where *not-grass* has been incorrectly classified. Some of these pixels could be areas where the grass spectral signature was mixed with or overwhelmed by the spectrum of the soil. The areas around the barrel and crates in the left background were likely a mixture of correct results and errors. In figure 3, it does appear that there was tall grass around the crates and barrel, as well as a grassy area behind them. One area that did not respond well is at the base of the building. It appeared that there is grass along the base of the building in figure 3, but the classification does not highlight that particularly well. This could have been due to reflected light from the building reflecting off the grass on its way to the imager, which would subsequently contaminate the grass spectra with wall spectra. Since training pixels for grass were all chosen from the center part of the grass region, GP did not train on any examples containing both grass and wall spectra. To improve performance then, one might select a few pixels for each class (if possible) from areas where reflected light is included; this could allow GP to learn how to correctly classify these types of pixels as well.

Acknowledgments

Mr. Rauss would like to thank Neelam Gupta at the U.S. Army Research Laboratory for supplying him with the AOTF image used for developing the multi-variable capability of lilgp.

We also thank Paul Litvak for his invaluable help in modifying the parser.

References

Bersano-Begey, T.F., Daida, J.M., Vesecky, J.F., Ludwig, F.L. (1997). A Java Collaborative Interface for Genetic Programming Applications: Image Analysis for Scientific Inquiry, In *IEEE International Conference on Evolutionary Computation 1997*, Piscataway: IEEE Press, pp 477–482.

Brumby, S. P., Theiler, J., Perkins, S. J., Harvey, N. R., Szymanski, J. J., Bloch J. J., and Mitchell, M. (1999). Investigation of Image Feature Extraction by a Genetic Algorithm, In *Proceedings of SPIE 3812*, pp 24–31.

Clark, R. N. and Swayze, G. A. (1995). Mapping Minerals, Amorphous Materials, Environmental Materials, Vegetation, Water, Ice and Snow, and Other Materials: The USGS Tricorder Algorithm, In *Summaries of the Fifth Annual JPL Airborne Earth Science Workshop*, R.O. Green, Ed., JPL Publication 95–1, pp 39–40.

Crist, E. P. and Cicone R. C. (1984). Application of the Tasseled Cap Concept to Simulated Thematic Mapper Data, *Photogrammetric Engineering and Remote Sensing*, Vol. 50 no. 3, pp 343–352.

Daida, J, M., Hommes, J. D., Bersano-Begey, T. F., Ross, S. J., Vesecky, J. F. (1996a). Algorithm Discovery Using the Genetic Programming Paradigm: Extracting Low-Contrast curvilinear features from SAR Images of Arctic Ice, In P. Angeline and K. Kinnear Jr. (ed.) *Advances in Genetic Programming II.* Cambridge, MA: The MIT Press, pp 417–442.

Daida, J.M., Onstott R.G., Bersano-Begey T.F., Ross S.J., and Vesecky J.F. (1996b). Ice Roughness Classification and ERS SAR Imagery of Arctic Sea Ice: Evaluation of Feature-Extraction Algorithm by Genetic Programming. In *Proceedings of the 1996 IGARSS*, Washington: IEEE Press, pp 1520–1522.

Daida, J.M., Polito 2, J.A., Stanhope, S.A., Bertram, R.R., Khoo, J.C., Chaudhary, S.A., and O. Chaudhri (2000) What Makes a Problem GP-Hard? Analysis of a Tunably Difficult Problem in Genetic Programming, In *Journal of Genetic Programming and Evolvable Hardware*. Accepted

Denes, L. J., Gottlieb, M., Kaminsky, B., and Huber, D. F. (1997). Spectro-Polarimetric Imaging for Object Recognition, *SPIE Proceeding 3240*, Paper #02.

Goward, S.N. et al. (1991). Normalized Difference Vegetation Index Measurements from the Advanced Very High Resolution Radiometer*, Remote Sensing of the Environment*, Vol. 35, pp 257–277.

Gupta, N., Dahmani, R., Gottlieb, M., Denes, L., Kaminsky, B., and Metes, P. (1999). Hyperspectral Imaging using Acousto-Optic Tunable Filters*, SPIE Proceedings 3718*, pp 512–521.

Howard, D. and Roberts, S. (1999). A Staged Genetic Programming Strategy for Image Analysis, *In GECCO-99: Proceedings of the Genetic and Evolutionary computation Conference*, W. Banzhaf, et al. (eds.) Morgan Kaufmann, San Francisco: pp 1047–1052.

Jiaju, L. (1988). Development of Principle Component Analysis Applied to Multi-temporal Landsat TM Data, *International Journal of Remote Sensing*, Vol. 9, no. 12, pp 1895–1907.

Kaufman, Y. J. and Tanre, D. (1992). Atmospherically Resistant Vegetation Index (ARVI) for EOS-MODIS, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 30, no. 2, pp 261–270.

Kauth, R. and Thomas, G. (1976). The Tasseled Cap–A Graphic Description of Spectral-temporal development of Agricultural Crops as Seen by Landsat*, Proceedings 2nd International Symposium on Machine Processing of Remotely Sensed Data*, Purdue Univ.

Koza, J.R. (1992) *Genetic Programming: On the Programming of computer by Means of Natural Selection*, Cambridge, MA: MIT Press.

Roberts, S. and Howard, D. (1999). Evolution of Vehicle Detectors for Infrared Line Scan Imagery*, EvoIASP'99 and EuroEcTel'99 Joint Proceedings*, Springer LNCS, pp 111–125.

Tackett, W. (1993). Genetic Programming for Feature Discovery and Image Discrimination. *Proceedings of the Second International Conference on AI Planning Systems*, pp 669–673.

Zongker, D. and Punch, W. (1995). lilgp, Lansing, Michigan State University, GA Research and Applications Group.
http:// garage.cps.msu.edu/software/software-index.html