

University of Redlands

InSPIRe @ Redlands

MS GIS Program Major Individual Projects

Theses, Dissertations, and Honors Projects

7-2009

A MODIS Imagery Toolkit for ArcGIS Explorer

Sean Maunder Pack
University of Redlands

Follow this and additional works at: https://inspire.redlands.edu/gis_gradproj



Part of the [Geographic Information Sciences Commons](#), and the [Remote Sensing Commons](#)

Recommended Citation

Pack, S. M. (2009). *A MODIS Imagery Toolkit for ArcGIS Explorer* (Master's thesis, University of Redlands). Retrieved from https://inspire.redlands.edu/gis_gradproj/134



This work is licensed under a [Creative Commons Attribution 4.0 License](#).

This material may be protected by copyright law (Title 17 U.S. Code).

This Thesis is brought to you for free and open access by the Theses, Dissertations, and Honors Projects at InSPIRe @ Redlands. It has been accepted for inclusion in MS GIS Program Major Individual Projects by an authorized administrator of InSPIRe @ Redlands. For more information, please contact inspire@redlands.edu.

University of Redlands

A MODIS Imagery Toolkit for ArcGIS Explorer

A Major Individual Project submitted in partial satisfaction of the requirements
for the degree of Master of Science in Geographic Information Systems

by

Sean Maunder Pack

Ruijin Ma, Ph.D., Committee Chair

Douglas Flewelling, Ph.D.

July 2009

A MODIS Imagery Toolkit for ArcGIS Explorer

Copyright © 2009

by

Sean Maunder Pack

The report of Sean Maunder Pack is approved.



Douglas M. Flewelling, Ph.D.



Ruijin Ma, Ph.D., Committee Chair

July 2009

Acknowledgements

First and foremost I would like to thank my wife Kirstin and our daughters Siena and Hallie for their willingness and eagerness to move to southern California for a year so I could attend the MSGIS program. Without their support and enthusiasm through many months of long days attending classes, finishing homework, and implementing this project, I could not have succeeded in completing the program.

Several individuals at ESRI provided useful suggestions and necessary technical expertise at critical junctures during the completion of this project. Ray Carnes and Bryan Baker provided much needed guidance during attempts to implement a client-server solution. In addition, Nathan Warmerdam provided Python code modifications for a custom geoprocessing script used in the same client-server solution. Without the ideas and efforts of these individuals, I would not have learned many of the intricacies involved in developing geoprocessing services for ArcGIS Server.

Edan Cain, who works on the development team for ArcGIS Explorer, deserves special acknowledgement for his and his family's friendship during our time in Redlands, as well as for the assistance he provided specific to ArcGIS Explorer and development of custom tasks for the software. His comments, suggestions, ideas, and technical expertise helped me to achieve much better project solutions than would have been possible on my own.

Nathan Strout and John Laska, both with the Redlands Institute at the University of Redlands, provided their time, suggestions, and technical expertise related to VB.NET and ArcGIS Server. It was always reassuring to know that I could count on their knowledge and willingness to help.

Michael Liebler, a fellow member of Cohort 14, provided significant assistance in the development of a prototype client-server solution for the project.

Finally, Dr. Ruijin Ma served as my primary advisor throughout the process of this project and provided valuable guidance, clarity, and technical expertise to its completion. His suggestions resulted in greater functionality, and his knowledge of remote sensing techniques and their implementation helped me to develop image classification tools that are computationally efficient and easy to understand.

Abstract

A MODIS Imagery Toolkit for ArcGIS Explorer

by

Sean Maunder Pack

NASA's medium spatial resolution MODIS sensor provides near-global, daily remote sensing coverage of the Earth in 36 spectral bands that are optimized for monitoring a wide variety of environmental parameters. MODIS data is provided by NASA at no cost and is easily accessible via the Internet. As such, MODIS provides a rich source of remotely sensed data that can provide timely environmental information to military operations, disaster monitoring, and relief efforts. However, current workflows for downloading MODIS and identifying environmental features of interest require the use of sophisticated software operated by experienced analysts. These software packages have the added limitations of being expensive and not readily available in combat and/or disaster relief environments. This paper discusses the development of a set of software tools using existing geographic information system technology. These tools can enable analysts with limited experience and operating in difficult environments to easily access MODIS data and develop environmental spatial data from it for further analysis. Two different system architectures were developed as solutions—one that exists as a set of standalone tools in a desktop environment using ArcGIS Explorer, and one that exists as a client-server framework using ArcGIS Server with ArcGIS Explorer as the client.

Table of Contents

Chapter 1 – Introduction.....	1
1.1 Client	1
1.2 Problem Statement	1
1.3 Proposed Solution	2
1.3.1 Goals and Objectives.....	2
1.3.2 Scope	2
1.3.3 Methods.....	3
1.4 Audience.....	3
1.5 Overview of the Rest of this Report.....	4
Chapter 2 – Background and Literature Review.....	5
2.1 Background	5
2.2 Review of Literature.....	6
2.2.1 Remote Sensing and GIS in Environmental Monitoring and Disaster Management	6
2.2.2 MODIS Data.....	7
2.2.3 MODIS Data Application to Environmental Monitoring and Disaster Response.....	7
2.2.4 Existing Tools for MODIS Exploitation	8
2.2.5 Classification of Remotely Sensed Data	8
2.2.6 Compression of Remotely Sensed Data	11
2.3 Summary	12
Chapter 3 – MODIS Data.....	13
3.1 MODIS Overview	13
3.2 MODIS Rapid Response Project Images and Image Subsets	14
3.2.1 MODIS RRP Near Real Time Images	15
3.2.2 MODIS RRP Image Subsets	15
Chapter 4 – System Design.....	17
4.1 Problem Statement	17
4.2 Requirements Analysis.....	17
4.3 System Architecture	18
4.3.1 Client-Server Architecture	19
4.3.2 Standalone Architecture	20
4.3.3 ArcGIS Explorer Custom Task Framework	20
4.4 Project Plan	22
4.5 Summary	25
Chapter 5 – Implementation: Standalone Solution	27
5.1 The MODIS Data Retriever Custom Task	27
5.1.1 MODIS Subset Query Points	28
5.1.2 MODIS Data Exploration and Retrieval	30

5.1.3	MODIS Data Export.....	33
5.1.4	Help Document.....	34
5.2	The MODIS Data Classifier Custom Task.....	35
5.2.1	Adding Training Points	36
5.2.2	Image Classification	38
5.2.3	Change Detection	42
5.3	Summary	44
Chapter 6 – Implementation: Client-Server Solution.....		45
6.1	ArcGIS Explorer and ArcGIS Server in a Client-Server Architecture	45
6.2	Maximum Likelihood Classification in ArcGIS	45
6.3	Development of a Maximum Likelihood Classification Geoprocessing Service	46
6.4	Summary	48
Chapter 7 – Results and Analysis		49
7.1	System Architecture	49
7.1.1	Analysis of the Client-Server Solution.....	49
7.1.2	Analysis of the Standalone ArcGIS Explorer Solution	50
7.2	Performance Assessment of the MODIS Data Classifier.....	50
7.2.1	Data	50
7.2.2	Image Classification Parameters in the MODIS Data Classifier Custom Task	51
7.2.3	Performance Analysis.....	53
7.2.4	Performance Analysis Summary	61
7.3	A Case Study: April/May 2009 Flooding of the Red River of the North	62
7.4	Summary	63
Chapter 8 – Conclusions and Future Work.....		65
8.1	Conclusions	65
8.2	Future Work	66
Works Cited.....		67
Programming References		73
Appendix A. Original Project Workplan		75
Appendix B. Code Examples		77

Table of Figures

Figure 2-1. General steps to appropriately classify remotely sensed data.	9
Figure 2-2. Parallelepiped classification envelopes of inclusion using two bands of image class data from a MODIS RRP subset and two standard deviations.	10
Figure 2-3. Equiprobability contours (illustrated, not actual) resulting from maximum likelihood classification of image data from a MODIS RRP subset.	11
Figure 3-1. Global MODIS data coverage for October 1, 2008 for the Terra (a) and Aqua (b) sensors.	13
Figure 3-2. MODIS RRP subset coverage.	16
Figure 4-1. Schematic view of the client-server solution.	19
Figure 4-2. Schematic view of the standalone solution.	20
Figure 4-3. Generalized AGX custom task framework with supporting classes (ESRI, 2009b).	21
Figure 4-4. Planned project workflow schematic.	23
Figure 4-5. Actual project workflow schematic.	24
Figure 5-1. Diagram of classes that compose the MODIS Data Retriever custom task.	27
Figure 5-2. ArcGIS Explorer user interface with the MODIS Data Retriever custom task interface added to the task panel (red rectangle). The red arrows indicate the extent of the MODIS subset availability feature class.	28
Figure 5-3. Detail of the MODIS Data Retriever user interface with query tools and interfaces highlighted.	29
Figure 5-4. Detail of a query point added to the AGX globe and the associated subset extent outlined in white.	30
Figure 5-5. Detail of the MODIS Data Retriever user interface with thumbnail tools and interfaces highlighted.	31

Figure 5-6. Detail of a subset thumbnail and retrieved subset image added to the AGX globe.	32
Figure 5-7. Example of metadata created automatically by the MODIS Data Retriever, as seen in ArcCatalog.	33
Figure 5-8. Detail of the MODIS Data Retriever user interface with the data export tool and interface highlighted.	34
Figure 5-9. Diagram of classes that compose the MODIS Data Classifier custom task.	35
Figure 5-10. ArcGIS Explorer user interface with the MODIS Data Classifier custom task interface added to the task panel (red rectangle).	36
Figure 5-11. Detail of the MODIS Data Classifier user interface with training point tools and interfaces highlighted.	37
Figure 5-12. Training points representing “snow” added to a false color MODIS subset on the AGX globe. Red arrows indicate the locations of some of the training points.	38
Figure 5-13. Logical flow used to conduct custom task image classification.	39
Figure 5-14. Results of image classification performed on a false color MODIS subset to identify pixels representing snow.	41
Figure 5-15. Detail of the MODIS Data Classifier user interface with the change detection tool and interface highlighted.	42
Figure 5-16. Change detection result from two image classifications of “snow” for the “AERONET_Fresno” RRP subset from the mornings of 3/30/2009 (top left) and 4/19/2009 (top right).	43
Figure 6-1. ArcGIS maximum likelihood classification workflow as viewed in ModelBuilder.	46
Figure 6-2. Custom geoprocessing workflow as viewed in ModelBuilder.	47
Figure 7-1. MODIS RRP false color subset selected for the performance analysis, with descriptions of the broad image classes present.	51
Figure 7-2. Results of classifying the “AERONET_Fresno.2009089.terra.250m” image subset using the ArcGIS “Maximum Likelihood Classification” tool to derive reference data.	52

Figure 7-3. Two dimensional spectral distribution of image classes present in the “AERONET_Fresno.2009089.terra.250m” subset.	53
Figure 7-4. Number of pixels classified as “water” by the MDC using varying standard deviations. The red dashed line indicates the number of “water” pixels in the reference data set, and the red arrow indicates an apparent plateau between 2 and 2.5 standard deviations.	55
Figure 7-5. Four different comparisons showing the number of pixels that were classified by the MDC custom task and the reference data set as either “water” or “not water”.	55
Figure 7-6. Overlap of “water” pixels identified by the MDC custom task with other image classes in the reference data set.	56
Figure 7-7. Visual comparison of results from the reference data set and the MDC custom task (using 2.5 standard deviations) for a region of cloud over water from the original image subset. Light blue indicates pixels classified as “water”.....	57
Figure 7-8. Spectral comparison of a subset of pixels from the area of interest in Figure 7-7. A clear continuum in spectral space exists between “water” and “cloud”.	58
Figure 7-9. Increasing accuracy and false positives with increasing standard deviations. Pixels classified as water are light blue. Red arrows indicate greater accuracy for some pixels (left column, transition from water to land) and greater error for others (right column, shadows).	60
Figure 7-10. Classification results using increasing numbers of non-selectively placed training points, with two sets of selectively placed training points for comparison.	61
Figure 7-11. Comparison of flood extents for the Red River of the North, spring 2009, using results from the MODIS Data Classifier custom task.	63

List of Tables

Table 1. MODIS spectral bands, associated wavelengths, and primary uses.	14
Table 2. Primary functional and technical requirements for the MODIS Toolkit.	18
Table 3. Standard deviation settings for the MODIS Data Classifier custom task and associated trackbar setting values.	52

List of Acronyms and Definitions

AGX	ArcGIS Explorer
API	application programming interface
AVHRR	Advanced Very High Resolution Radiometer
GEOINT	geospatial intelligence
GIS	geographic information system
MDC	MODIS Data Classifier
MLC	maximum likelihood classification
MODIS	Moderate Resolution Imaging Spectroradiometer
NASA	National Aeronautics and Space Administration
NDVI	Normalized Difference Vegetation Index
NGA	National Geospatial-Intelligence Agency
NOAA	National Oceanic and Atmospheric Administration
RRP	Rapid Response Project
SDK	software development kit
URI	Uniform Resource Identifier
USDA	U.S. Department of Agriculture

Chapter 1 – Introduction

The National Aeronautics and Space Administration's (NASA) Moderate Resolution Imaging Spectroradiometer (MODIS) sensors, onboard the Earth Observing System Terra (EOS AM) and Aqua (EOS PM) satellites, were designed to monitor changes in Earth processes (land, sea, atmosphere, and cryosphere). Observed changes in these processes are a means to understand human-induced and natural changes to the Earth's physical and biological systems, which can in turn help inform policy decisions related to environmental protection (NASA, 2008a).

Although originally intended to monitor long-term and broad scale processes, these sensors have also developed a niche in monitoring short-term, regional environmental effects and disasters, including wildfires, effects of hurricanes and typhoons, floods, possible droughts, and crop yields for famine monitoring. Additionally, MODIS data are very useful in informing global military missions due to MODIS's ability to monitor environmental parameters that may be factors in theaters of operation. These roles have developed primarily due to MODIS's high temporal resolution and large coverage area, and the fact that environmental parameters and the effects of many environmentally-related disasters can be detected at relatively low to medium spatial resolution (i.e., 250 m to 1 km).

The primary goal of this project is to increase user access to and exploitation of MODIS data through the development of custom GIS-based software tools, referred to collectively as the MODIS toolkit. A secondary goal is to make the resulting toolkit as broadly and easily available as possible through the use of existing technologies currently available at no cost via the Internet.

1.1 Client

This project is being undertaken specifically for the National Geospatial-Intelligence Agency's (NGA) Image Sciences Division. However, numerous entities within NGA stand to benefit from the MODIS toolkit, including the Environmental Issues Branch, the Office of Global Support, and analytical units that focus on a variety of regional analyses, including homeland security.

1.2 Problem Statement

NGA needs a tool to facilitate analyst access to and exploitation of NASA's MODIS imagery data. Currently the incorporation of MODIS data into projects ranging from direct military support to emergency response to long-term analysis is impeded by time consuming, inefficient, multi-step processing. This processing can also be intimidating to analysts not proficient in multiple software environments that can be complex to employ (e.g. ArcGIS, ERDAS Imagine, ENVI). These difficulties are exacerbated in deployed and disaster response settings where data and product turnaround have immediacy, access to software can be limited, and analytical expertise can be highly variable. As a result of these factors, NGA requires a set of MODIS analysis tools that:

- are easy to access in one software environment
- can be easily downloaded and utilized without the need for end-user licenses for complex software packages
- require little or no training prior to use
- are quickly assimilated by analysts with varied expertise and capabilities
- simplify as many tasks as possible, from data importation to feature identification

1.3 Proposed Solution

The solutions developed in this project rely on customizing existing GIS technologies—specifically ESRI’s ArcGIS Explorer and ArcGIS Server—to provide the requisite functionality. Furthermore, the solutions exist in a user environment (ArcGIS Explorer, build 500) that is both easy to use and familiar to users with minimal GIS experience or exposure to globe-based data viewers like Google Earth. In attempting to meet the client’s needs, two separate solutions were developed. A standalone toolkit was developed using only custom tasks in ArcGIS Explorer. The second solution was a prototype client-server solution that includes a custom geoprocessing service hosted by ArcGIS Server with ArcGIS Explorer as the client.

1.3.1 Goals and Objectives

The primary goal of this project is to increase NGA analyst access to and exploitation of MODIS data. Through applying existing GIS technologies, the process by which NGA analysts access MODIS data and derive useful spatial information will be streamlined and incorporated into a single user environment.

The intention is for the final user environment for MODIS access and exploitation to be a simple-to-use, freely available interface and toolkit that can be easily downloaded and used with access only to an Internet connection. Overall, the final MODIS toolkit’s intent was to improve the efficiency of the MODIS data processing stream and to make the toolkit widely available as a utility for developing environmental spatial data to aid in military and disaster relief support. As noted, these objectives are important to deployed NGA personnel because, while access to a computer and Internet connection is not generally a problem, access to end-user licenses for software may be. Furthermore, experience levels of deployed analysts are variable—the tools resulting from this project were simplified to a great extent in an effort to ensure success for analysts with wide ranging backgrounds and capabilities.

The tools resulting from this project have the further benefit of being of potential value to other governmental, non-governmental, and academic entities involved in environmental monitoring and disaster relief. The toolkit could also serve as a prototype for development of new functionality or additional tools for improving access to other dispersed remote sensing data repositories (e.g., NOAA AVHRR and USGS Landsat).

1.3.2 Scope

The final products of this project include two solutions based on different architectures: a standalone set of custom software tools in a desktop environment, and a client-server solution. Final functionality includes the ability to:

- identify and download MODIS imagery through geographic query
- classify environmental features of interest via supervised classification
- conduct change detection with classified datasets
- export resulting datasets in raster format
- display results with other datasets of interest (e.g., boundaries, terrain, roads, etc.) in a user-friendly interface
- create metadata automatically for downloaded data and any data derived through use of the toolkit

1.3.3 Methods

Solutions developed for this project relied exclusively on the use of ArcGIS Explorer (AGX) as the user interface for both the standalone and the client-server architectures. AGX was selected as the development platform for several reasons:

- its simple and friendly user interface
- its availability at no cost via Internet download
- its ability to be customized through a .NET programming interface
- its development and existence specifically as a client for ArcGIS Server
- its off-the-shelf, built-in data and functionality that provide useful context and a rich user experience

NASA's World Wind software was also explored as a possible environment in which to develop the MODIS toolkit but, in the author's judgment, World Wind is a less ideal platform for this project due to the following reasons:

- it has a more complicated user interface than AGX
- it is slow to load and refresh base imagery data
- its base imagery data has lower spatial resolution than that of AGX
- there is no available GIS server software with which it will work smoothly
- its presumed lack of technical support and/or ongoing development relative to AGX

With respect to the standalone desktop architecture, two custom tasks were developed for AGX that incorporate all of the desired access and exploitation functionality. For the client-server architecture, a prototype custom geoprocessing service was developed on ArcGIS Server that can be consumed by AGX. This geoprocessing service was developed specifically to conduct supervised image classification, and provides proof-of-concept that a similar service could be developed to enable users to download and view MODIS imagery.

1.4 Audience

This paper is intended for NGA analysts familiar with GIS technology who could benefit from using MODIS data in their analysis. Furthermore, the paper and associated solutions could be of interest to other governmental and non-governmental entities that have a need

for high temporal resolution environmental spatial data. Finally, prolific users of AGX and its associated custom tasks may find the functionality of the MODIS toolkit and/or its associated programming code to be useful.

1.5 Overview of the Rest of this Report

The next chapter provides more detailed background about the MODIS sensors, as well as a literature review detailing the application of remotely sensed data to environmental monitoring and disaster response. Chapter 3 includes more detailed information about MODIS data, specifically the MODIS Rapid Response Project image subsets that form the core data for this project. This is followed in Chapter 4 by an overview of the proposed solutions for the project. Chapters 5 and 6 then discuss the project implementation in detail, including the logic and programming code used to create the standalone and client-server solutions that were developed. Chapter 7 reviews the results of the project, including an analysis of the two architectures that were implemented, a performance analysis conducted on one of the custom software tools, and an example case study for how the MODIS toolkit could be applied. Chapter 8 concludes the report, highlighting project conclusions and areas for future work.

Chapter 2 – Background and Literature Review

2.1 Background

Two primary advantages of MODIS data are its high temporal resolution and its availability over the Internet at no cost. These factors make it a very rich source of remotely sensed environmental data, with a wide range of applications that include assessing the extent of flooding after severe storms or natural disasters, predicting drought conditions by analyzing snowpack for a given region, or determining mobility within a region as it pertains to current environmental conditions.

With access to an Internet connection, image analysis and GIS software, users can classify and extract image classes of interest (e.g., water, snow, vegetation) from MODIS imagery for monitoring change or for further spatial analysis. However, this type of analysis currently entails a relatively complex workflow that includes retrieving MODIS data from one of four networked distribution sites, performing any necessary pre-processing of the data, conducting feature identification and extraction in image analysis software (e.g., ArcGIS, ERDAS Imagine, ENVI). The resulting raster data or feature classes are then imported into a GIS software package for spatial analysis. This is an inefficient process accessible to only the most sophisticated users who have access to advanced software. As a result, despite the availability of MODIS imagery at no cost, a barrier exists for many individuals and groups who could otherwise benefit from MODIS's abundance of remotely sensed environmental data.

NGA, a Department of Defense combat support agency and a member of the U.S. national intelligence community, is an agency that could benefit from better access to MODIS data. NGA provides geospatial intelligence (GEOINT) to warfighters, but is also heavily involved in providing GEOINT in support of disaster relief and homeland security operations (NGA, 2008a). Due to these missions, their global nature, and their necessity for swift reaction, NGA has a direct need for data acquired by the MODIS sensors. Indeed, three central facets of GEOINT as it pertains to NGA support are the abilities to define environments of interest, describe influences those environments have on mission planning, and assess threats and hazards that may be identified through synthesizing information from the first two facets (NGA, 2008b). Despite NGA's need to apply information derived from MODIS imagery, the use of MODIS imagery in daily operations is limited by inefficient means to access the imagery, process it, and derive useful results that can be easily integrated into analysis and end-user products.

Modern GIS software packages provide a number of ways to improve NGA's access to, processing of, and integration of MODIS data. In addition to advanced geoprocessing capabilities that yield spatial data in a variety of formats, GIS software offers many alternative frameworks for development and implementation of solutions. These frameworks include customization of existing software and/or the development of client-server solutions that can be driven across networks.

2.2 Review of Literature

This project entails the application of GIS technology to improving access to and classification of remotely sensed data, specifically as it applies to disaster response. As such, the following literature review focuses on five main associated themes:

- the use of remotely sensed data in environmental monitoring and disaster management
- pertinent background information on MODIS imagery
- examples of MODIS applications to environmental monitoring and disaster management
- existing tools for exploiting MODIS data
- an overview of imagery classification as a means to derive spatial data and the effects that data compression has on remote sensing data quality

2.2.1 Remote Sensing and GIS in Environmental Monitoring and Disaster Management

Evidence of the important role that remotely sensed data and GIS are now fulfilling in the realm of environmental monitoring and disaster management is abundant (Gillespie, Chu, Frankenberg, & Thomas, 2007; Doescher, Ristyb, & Sunne, 2005; Kevany, 2003; Ostir, Veljanovski, Podobnikar, & Stancic, 2003). Remote sensing and GIS technology have also been demonstrated to be improving analysis and understanding of the longer term environmental effects of natural disasters across broad areas (Sirikulchayanon, Oyana, & Sun, 2008). In fact, it has been argued that the application of remote sensing to disaster response has not only substantiated its use in these scenarios but has also highlighted the “great need for remotely sensed data during disaster response operations” (Doescher et al., 2005, p.124). Cutter (2003) supports the conclusions of the empirical studies cited above in a discussion of the role that geographic information science plays in the disaster response cycle. She indicates that spatial data acquisition and integration is one of the challenges of geographic information science as it applies to disasters and emergency management, further stating that “there is a critical need for real-time data and information [in the disaster response cycle]” (p.443).

Due to its high spatial resolution, commercial satellite imagery has been an obvious tool applied to many disaster settings (Gillespie et al., 2007; Doescher et al., 2005). However, this high spatial resolution comes at the cost of lower temporal resolution, with revisit times of 1.7 days or more under ideal conditions (Digital Globe, 2008; GeoEye, 2008). Airborne remote sensing in support of the Hurricane Katrina response also proved to be a valuable source of data for mapping damage (Corbley, 2006), but the very high spatial resolution of airborne sensors comes at the cost of global access—airborne remote sensing platforms can be either non-existent or slow to deploy in many parts of the world. It is important to note that another important trade-off to these sources of high resolution (spatial and spectral) data is that they can be cost-prohibitive for those outside of government or well-funded non-governmental organizations.

As an alternative to the high resolution data sources indicated above, several authors have specifically demonstrated the value to environmental monitoring and disaster

response scenarios that exists in exchanging spatial resolution for high temporal resolution (Belward et al., 2007; Gillespie et al., 2007; Brakenridge & Anderson, 2006; Tralli, Blom, Zlotnicki, Donnellan, & Evans 2005; Domenikiotis, Loukas, & Dalezios, 2003). As noted, this tradeoff is inherent in sensors like NASA's MODIS (250-1000m spatial resolution) and the National Oceanic and Atmospheric Administration's Advanced Very High Resolution Radiometer (AVHRR) (1.1 km spatial resolution).

2.2.2 MODIS Data

In addition to a series of slow-to-produce standard data products, more time-dependent imagery products are also routinely developed from MODIS data. During the historic summer 2000 forest fire season in the western United States, the utility of MODIS imagery for detecting and monitoring fires became very apparent to the U.S. Department of Agriculture (USDA) Forest Service (National Aeronautics and Space Administration, Goddard Flight Center, 2003). However, standard MODIS fire products at that time were generated approximately two months after data acquisition; this was clearly not timely enough to be of use to first responders (MODIS Rapid Response Project, 2004).

As a result, the MODIS Science Team and the University of Maryland worked closely together to manually process MODIS data for provision to the USDA Forest Service. This customer-oriented initiative laid the framework for the MODIS Rapid Response Project (RRP), an alternative MODIS data processing and distribution system able to meet the quick turn-around demands of forest fire responders. Upon completion in 2001, the MODIS Rapid Response system was able to process data, identify probable fires, and display images with fire detections within two to four hours of data collection. Initial data products included true-color images only, but have since expanded to include false-color images, as well. These new products were developed as a result of diverse applications and new partnerships that now include the USDA Forest Service, USDA Foreign Agriculture Service, U.S. Environmental Protection Agency, U.S. Air Force, and U.S. Navy (MODIS Rapid Response Project, 2004).

2.2.3 MODIS Data Application to Environmental Monitoring and Disaster Response

The success of the MODIS Rapid Response System in providing near-real-time data (2-4 hours after acquisition, National Aeronautics and Space Administration, Goddard Flight Center, 2003) to the USDA Forest Service resulted in expansion of fire monitoring and mapping to the entire globe. The Fire Information for Resource Management System (FIRMS) provides web mapping of hot spots and active fires, as well as downloadable spatial datasets, for a subset of high risk regions across the globe (Fire Information for Resource Management System, 2008).

In addition to fires, the USDA's success with MODIS data led to a similar partnership with the Rapid Response Project for monitoring droughts and crop conditions as part of USDA's Foreign Agricultural Service (USDA Foreign Agricultural Service, 2007). The Dartmouth Flood Observatory has also demonstrated the effectiveness of using MODIS as an "almost ideal floodsat" in monitoring and mapping floods globally (Dartmouth Flood Observatory, 2002), and as a disaster response and damage assessment tool (Brakenridge & Anderson, 2006). Its application to disaster relief was very

effectively realized in May, 2008, after Cyclone Nargis crossed the Irrawaddy Delta in Myanmar, killing tens of thousands of people. MODIS Rapid Response imagery was used to determine damage extent and to help target relief supply shipments (J. Schmaltz, personal communication, October 2, 2008).

Notwithstanding its relatively low spatial resolution, MODIS imagery has become a useful tool for monitoring natural disasters and environmental problems, as well as for relief efforts necessitated by these conditions. This is due to MODIS's high temporal resolution, as well as its availability at no cost.

2.2.4 Existing Tools for MODIS Exploitation

Currently the only available tool specifically designed for MODIS imagery exploitation is the 4-D Viewer application within the MICRODEM GIS software utility developed by the U.S. Naval Academy (2007). This freeware utility allows for interactive selection and download of MODIS RRP image subsets that can then be combined with digital elevation data to create 3-D imagery scenes. Additional functionality includes the ability to rotate 3-D scenes, scroll through other downloaded MODIS scenes for one's area of interest, and create oblique block views. While MICRODEM is very useful for downloading and viewing MODIS imagery, it does not include the ability to classify image data and extract information for further spatial analysis.

2.2.5 Classification of Remotely Sensed Data

The procedures for transforming remotely sensed data into information are known collectively as classification (Jensen, 2005, p. 337). Classification techniques are well-established within the discipline of remote sensing and provide the means to transform raw pixel values into thematic environmental information for GIS analysis. Several sources provide thorough reviews of the various methodologies that have been developed for multispectral image classification (Jensen, 2005; Narumalani, Hlady, & Jensen, 2002; Duda, Hart, & Stork, 2001; Tso & Mather, 2001). These methodologies typically require users to determine an appropriate image classification logic (parametric, non-parametric, non-metric), followed by an appropriate classification algorithm (minimum distance, parallelepiped, maximum likelihood). Figure 2-1 summarizes some of the general steps one would follow to appropriately classify remotely sensed data (Jensen, 2005, p. 339).

Although a comprehensive discussion of all classification schemes is beyond the scope of this paper, it is pertinent to provide background for two established classification methods: parallelepiped and maximum likelihood classification. This is due to the fact that the parallelepiped algorithm is the basis for image classification conducted using the AGX custom task toolkit developed for this project. Furthermore, maximum likelihood classification is one of two primary image classification tools available in ArcGIS, and it is the basis for the prototype client-server solution developed for this project.

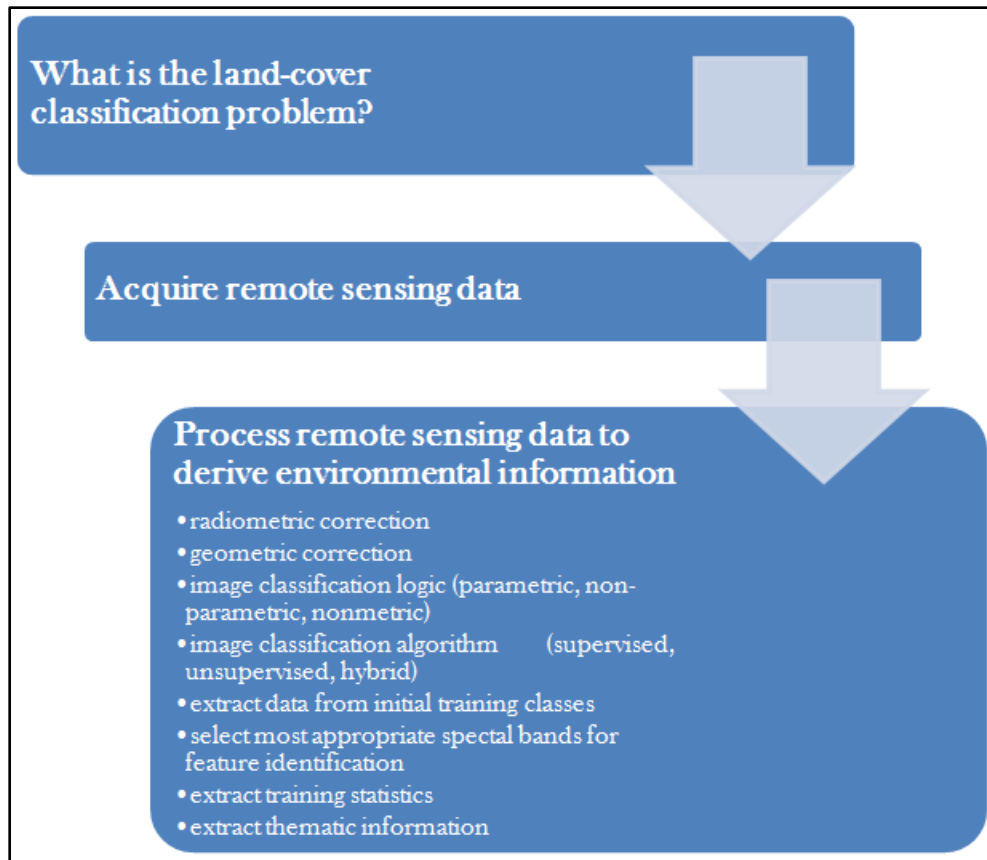


Figure 2-1. General steps to appropriately classify remotely sensed data.

2.2.5.1 Parallelepiped Image Classification

The parallelepiped classification scheme is a non-parametric, supervised classification technique. This indicates that the pixel data for an image are grouped into classes using some prior knowledge or information about the image classes present (supervised), but the pixel data are not assumed to be normal, or Gaussian, in distribution (non-parametric). Furthermore, the parallelepiped classification scheme is a per-pixel method that uses hard classification logic. These terms indicate that each pixel is evaluated for class membership individually (per-pixel) and each pixel must belong to one class only (hard logic); no pixel can be given partial membership in more than one image class (fuzzy logic) (Jensen, 2005).

In practice, the parallelepiped method is applied through developing sets of training pixels for each class present in an image. These training pixels can be used to calculate a mean digital number and associated standard deviation for each class in each image band. The mean and standard deviation are then used to define a minimum-maximum envelope of inclusion for a given class, per the following expression:

$$\mu_{ck} - \sigma_{ck} \leq DN \leq \mu_{ck} + \sigma_{ck}$$

where DN represents the digital number or value of a pixel for class c in band k (Jensen, 2005). The sensitivity for inclusion of pixels can therefore be increased or decreased by

simply increasing or decreasing the size of the envelope defined by the standard deviations. A pixel can only be included in a given class if it falls within the minimum-maximum envelope for *every* band in that class. Figure 2-2 illustrates this logic applied to two bands of image data from a MODIS RRP subset using double standard deviations to define the envelope for class inclusion. In two dimensions, these envelopes are clearly defined as rectangles, and will therefore be defined as rectangular or parallelogram prisms in three dimensions. When applied to n dimensions, or image bands, the envelopes take on forms that are known mathematically as *parallelepipeds* (Aronoff, 2005), the term also used to name the method.

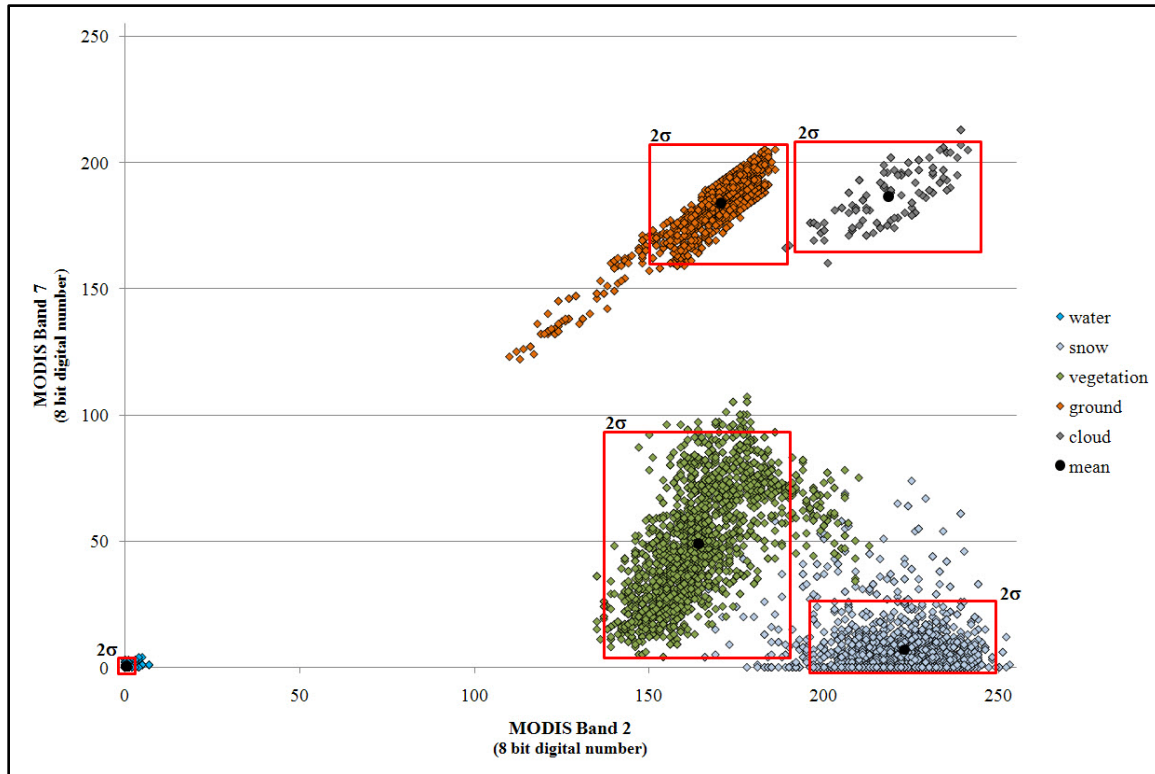


Figure 2-2. Parallelepiped classification envelopes of inclusion using two bands of image class data from a MODIS RRP subset and two standard deviations.

The primary benefit of the parallelepiped classification technique is that it is very simple to calculate. However, as is evident from Figure 2-2, it may not always be the best technique to use because of the possible erroneous inclusion of pixels in an image class. This can be seen at double standard deviations for the envelope of inclusion for the vegetation class which will erroneously include pixels representing snow. This problem clearly increases proportionately with the number of standard deviations used.

2.2.5.2 Maximum Likelihood Image Classification

Maximum likelihood classification is similar to the parallelepiped approach in that it is a supervised method that operates per-pixel using hard classification logic. In contrast, maximum likelihood classification is parametric, which indicates that it assumes pixel

values are normally distributed. The most important difference between the two methods is that the maximum likelihood strategy is based on probability theory, assigning a pixel to the class for which it has the highest probability of belonging. This concept is illustrated in Figure 2-3 for the same set of image data used in Figure 2-2. In two dimensions, the likelihood or probability of belonging to an image class is defined by equiprobability contours (Aronoff, 2005). In three dimensions, these contours become three dimensional Gaussian curves. This concept can be extended to n dimensions through calculation and application of covariance matrices. The advantages of maximum likelihood classification are quite evident in Figure 2-3 since the directional alignment and shape of the regions of inclusion (equiprobability contours) results in less overlap between training classes. Furthermore, where overlap does occur, a pixel is assigned to the class for which it has the greatest likelihood of being a part. The primary disadvantage of the method is that it is very intensive from a computational standpoint.

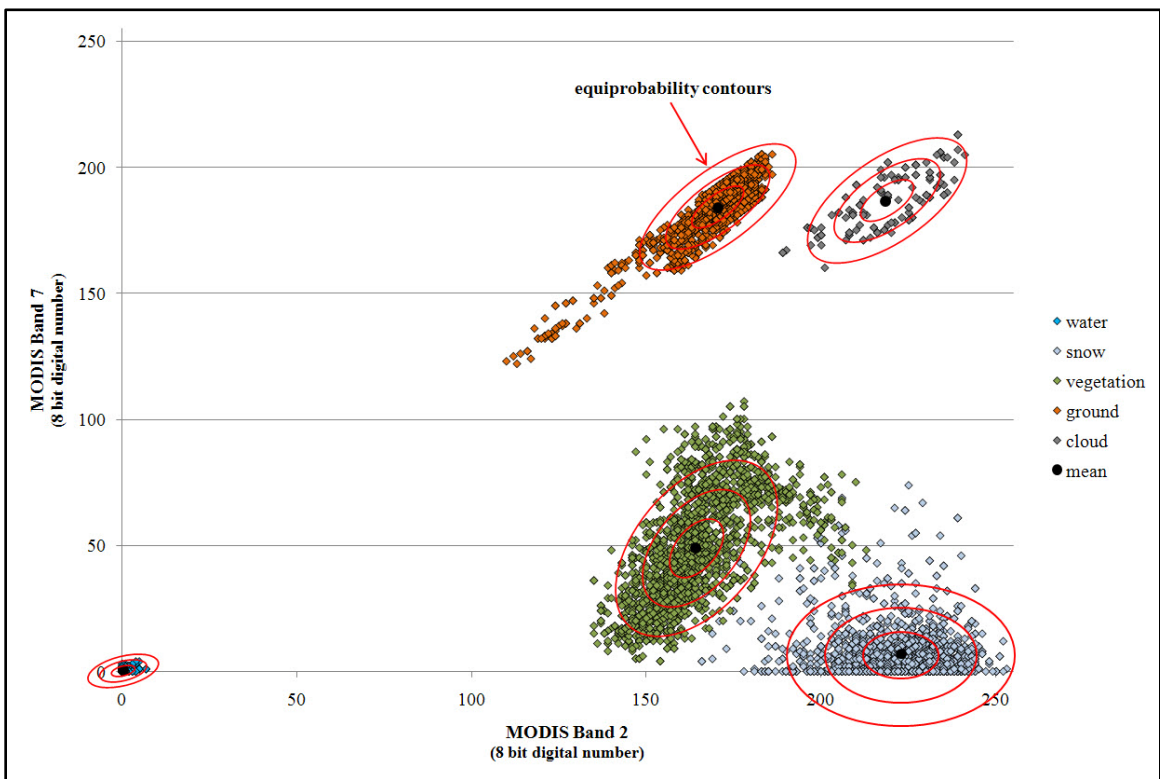


Figure 2-3. Equiprobability contours (illustrated, not actual) resulting from maximum likelihood classification of image data from a MODIS RRP subset.

2.2.6 Compression of Remotely Sensed Data

As a result of the vast amounts of data now being transmitted and downloaded by image scientists, file compression has also become an important consideration in remote sensing. Lam, Lau, & Li (1999) noted that compression techniques (e.g., JPEG, TIFF) “have been widely accepted in the remote sensing discipline to optimize data storage and to reduce data transmission time.” However, the employment of compression has given

rise to questions about the loss of information inherent in many compression techniques and their effects on the accuracy of image classification.

In an investigation of the effects of JPEG compression on classification of SPOT (Satellite Pour l'Observation de la Terre) multispectral imagery, Lau, Li, & Lam (2003) found that: compression ratios of less than 10:1 had minimal effects; with ratios between 10:1 and 30:1, classification accuracy decreased steadily, but was not more than 20-30% in error depending on the nature of the terrain being classified; but ratios beyond about 35:1 resulted in significant reduction of overall classification accuracy. On the contrary, Webb, Robinson, & Evangelista (2004, p. 660) concluded “that JPEG compression of up to 46:1 was not detrimental to the classification results [of astronaut-acquired orbital photographs]”. Although Lau et al. (2003) investigated both supervised and unsupervised classification, the results from both studies cited above were for supervised classification, using the maximum likelihood classifier.

2.3 Summary

The literature abounds with examples of successful application of remote sensing data to environmental monitoring and disaster management. In fact, some authors suggest that this is a niche which could benefit from even greater use of remotely sensed data. The successful application of remote sensing for monitoring and management has, in particular, been associated with MODIS data due in large part to the development of the MODIS RRP and the project's willingness to provide support and data to a broad group of diverse users. The following chapter will provide more detailed information about MODIS data, with specific focus on the image subsets produced by the RRP since they are the principal source of MODIS data for this project.

Chapter 3 – MODIS Data

This chapter provides broad information about the MODIS sensors, the data they collect, and the standard MODIS data products. In addition, the concluding sections of the chapter present more detailed information about MODIS RRP image subsets which are the sole source of data accessed and exploited by the MODIS toolkit developed for this project.

3.1 MODIS Overview

With a viewing swath of 2,330 km, the MODIS instruments carried on the Terra and Aqua satellite platforms have the ability to measure specific spectral reflectance values for the entire Earth surface every 24 to 48 hours (NASA, 2008b). The two satellites operate on 705 km circular, near-polar, sun-synchronous orbits, with Terra on the descending node (morning pass) and Aqua on the ascending node (afternoon pass). Figure 3-1 illustrates the global coverage for both platforms on a typical day (NASA, 2008c). Voids in coverage change daily based on satellite orbital characteristics.

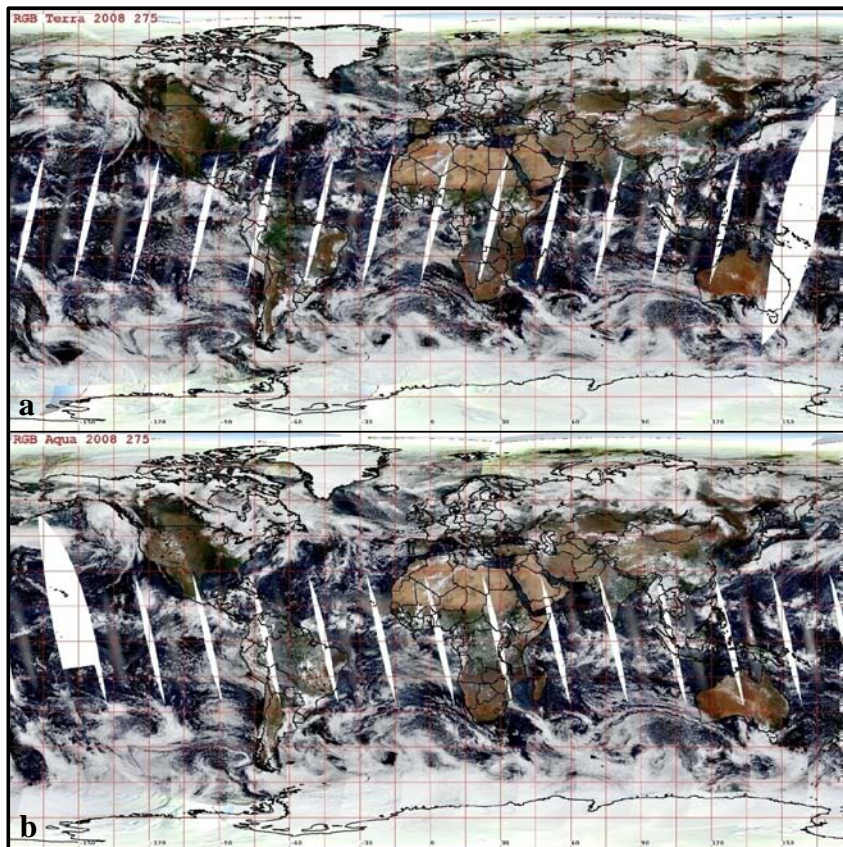


Figure 3-1. Global MODIS data coverage for October 1, 2008 for the Terra (a) and Aqua (b) sensors.

Spectral reflectance values are measured by the MODIS sensors in 36 different bands, each with a unique range of wavelengths and purpose in monitoring global change (Table 1) (NASA, 2008b). Subsequent to data collection and download via satellite link, numerous calibrated data products are developed and disseminated to the scientific community. The raw radiance data sets (referred to as Level 1A) are calibrated and geolocated to produce what are known as Level 1B data sets, which are generally available eight or more hours after initial data collection. The Level 1B data sets then serve as the base data from which higher level atmosphere, land, ocean, and cryosphere products are created (National Aeronautics and Space Administration, 2008d & 2008e). These products are distributed via four different disciplinary host sites. NASA uses the HDF-EOS (Hierarchical Data Format-Earth Observing System) file format for disseminating all standard MODIS data products.

Table 1. MODIS spectral bands, associated wavelengths, and primary uses.

Primary Use	Band	Bandwidth (µm)	Primary Use	Band	Bandwidth (µm)
Land/Cloud/Aerosols Boundaries	1	0.620 - 0.670	Surface/Cloud Temperature	20	3.660 - 3.840
	2	0.841 - 0.876		21	3.929 - 3.989
Land/Cloud/Aerosols Properties	3	0.459 - 0.479		22	3.929 - 3.989
	4	0.545 - 0.565		23	4.020 - 4.080
	5	1.230 - 1.250	Atmospheric Temperature	24	4.433 - 4.498
	6	1.628 - 1.652		25	4.482 - 4.549
	Ocean Color/Phytoplankton/Biogeochemistry	7	2.105 - 2.155	Cirrus Clouds Water Vapor	26
8		0.405 - 0.420	27		6.535 - 6.895
9		0.438 - 0.448	28		7.175 - 7.475
10		0.483 - 0.493	Cloud Properties	29	8.400 - 8.700
11		0.526 - 0.536	Ozone	30	9.580 - 9.880
12		0.546 - 0.556	Surface/Cloud Temperature	31	10.780 - 11.280
13		0.662 - 0.672		32	11.770 - 12.270
Atmospheric Water Vapor		14	0.673 - 0.683	Cloud Top Altitude	33
	15	0.743 - 0.753	34		13.485 - 13.785
	16	0.862 - 0.877	35		13.785 - 14.085
	17	0.890 - 0.920	36		14.085 - 14.385
	18	0.931 - 0.941			
	19	0.915 - 0.965			

3.2 MODIS Rapid Response Project Images and Image Subsets

In addition to the standard, calibrated MODIS data sets, a series of near real time images and image subsets are also produced from raw MODIS data by the MODIS RRP. The background and origin of the RRP and these images and subsets are discussed in more detail in Chapter 2, but their data characteristics are summarized in the following sections.

3.2.1 MODIS RRP Near Real Time Images

MODIS RRP near real time images are unique due to their availability within a short amount of time after initial data collection by the sensors. The RRP images are available as true-color (MODIS bands 1-4-3) and false-color (MODIS bands 7-2-1 and 3-6-7) images, with an additional land surface temperature (LST) image as well. All of the images are available via the MODIS Rapid Response website and downloadable in HDF-EOS format at spatial resolutions of 250m, 500m, 1km, 2km, and 4km (MODIS Rapid Response System, 2008a). Despite their availability shortly after collection, the near real time images are provided without geometric correction for distortion. Users must geometrically correct the images prior to use through one of several second party software utilities.

3.2.2 MODIS RRP Image Subsets

In addition to the near real time images, the RRP also provides a series of true-color, false color (MODIS bands 7-2-1), and Normalized Difference Vegetation Index (NDVI) image subsets in geometrically corrected and geopositioned JPEG, TIFF, and KML format. A select few subsets are also available in false-color composed of MODIS bands 3-6-7. The term “subset” is used for these files because they are available as subsets of MODIS’s daily global coverage. This fact and the subset file formats and their geometric correction and positioning are what really distinguish the image subsets from the near real time images. While the near real time images maintain scientifically calibrated spectral measurements, the JPEG, TIFF, and KML image subsets suffer from at least some loss of original information due to file compression and should therefore not be used for any application that requires accurate spectral measurements.

All of the image subsets are available at spatial resolutions of 250m, 500m, 1km, and 2km. Although the MODIS RRP near real time images are available for the Earth’s entire surface, RRP image subsets are more limited, focusing on the Earth’s land masses (Figure 3-2). Areal calculations in ArcMap indicate, however, that despite their limited extent, the image subsets still provide coverage of approximately 81% of the Earth’s land surface, not including Antarctica.

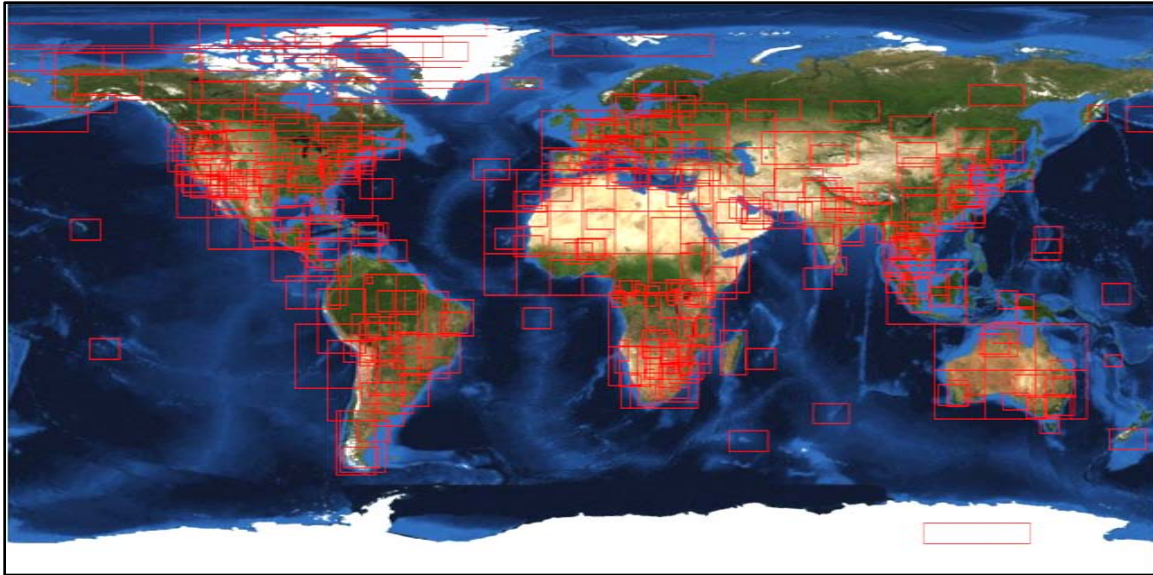


Figure 3-2. MODIS RRP subset coverage.

The MODIS toolkit was developed specifically to work with the JPEG format subsets. Although there is some inherent loss of information due to JPEG compression, these image subsets have many advantages over the HDF-EOS near real time images, particularly for users in remote locations or those with limited experience with image data. Specifically, the JPEG files are relatively small (less than 9MB versus 28MB or more for the HDF-EOS near real time images), already orthorectified and geopositioned, archived in an easy-to-query manner, and easily imported by most GIS and imagery software. Their disadvantages compared to the HDF-EOS near real time images are their inherent loss of information due to compression, greater time lag for initial production (approximately 2-4 hours after Level-1B products), and their lack of complete global coverage. However, the image subsets provide good coverage for most of the major populated areas of the Earth.

Chapter 4 – System Design

As indicated in previous chapters, two different architectures were developed for this project—a set of custom tasks for ArcGIS Explorer in a standalone framework, and custom geoprocessing services in a client-server framework. This chapter summarizes the requirements analysis conducted during the project’s initial phases, followed by a discussion of the broad nature of the two different architectures developed as final solutions. Details about AGX’s role in both architectures are also discussed since AGX is the primary user interface used in both solutions. Finally, the chapter concludes with an analysis of the initial project plan and proposed solution, as well as a discussion of how closely reality followed the initial plan.

4.1 Problem Statement

NGA needs a tool to facilitate analyst access to and exploitation of NASA’s MODIS imagery data. Currently the incorporation of MODIS data into projects ranging from direct military support to emergency response to long-term analysis is impeded by time consuming, inefficient, multi-step processing. This processing can also be intimidating to analysts not proficient in multiple software environments that can be complex to employ (e.g. ArcGIS, ERDAS Imagine, ENVI). These difficulties are exacerbated in deployed and disaster response settings where data and product turnaround have immediacy, access to software can be limited, and analytical expertise can be highly variable. As a result of these factors, NGA requires a set of MODIS analysis tools that:

- are easy to access in one software environment
- can be easily downloaded and utilized without the need for end-user licenses for complex software packages
- require little or no training prior to use
- are quickly incorporated into analysis by analysts with varied expertise and capabilities
- simplify tasks, from data importation to feature identification

4.2 Requirements Analysis

The primary functional and technical requirements for this project are outlined in Table 2. The functional requirements include all the operations necessary to enable analysts to easily access MODIS data, derive useable spatial data from it, and view it in context. These functions include the ability to query for data using either a point tool, or by manually entering coordinates of interest; the ability to classify a MODIS image to identify image classes of interest (e.g., snow, water, vegetation, etc); and the ability to view classified images with other associated data (e.g., boundaries, place names, satellite imagery). In addition, due to the prospect that many MODIS image subsets and classified datasets may be downloaded and derived by users in a short amount of time, automated metadata creation is a necessary function that will save users time. Automated metadata

Table 2. Primary functional and technical requirements for the MODIS Toolkit.

Functional Requirements	Technical Requirements
MODIS subset query via point query or geo-coordinates	one software environment (AGX)
automated subset retrieval	easily downloaded and used without the need to pay for software licenses
image classification—to enable the derivation of spatial data	limited or no training required for use
automated metadata creation	Windows workstation with minimal requirements needed to use AGX
data export in vector and/or raster format	broadband Internet connection
built-in base data	Microsoft .NET 2.0 or later

creation will also ensure that all data stemming from use of the MODIS toolkit will have the necessary associated information to ensure the data's longevity and accurate use. Finally, users will be able to export data from the custom software environment to the local storage of their choice. This also serves to maximize the data's longevity and use.

The technical requirements are driven by a combination of the client's needs and the decision to use AGX as the core software environment for the project solution. The decision to use AGX as the final development platform fulfills the basic technical requirements that the core software for the solution be easily downloaded without the need to purchase end-user licenses. As a result of this decision, AGX has a number of minimum system requirements for use, including the following basic hardware and software requirements (ESRI, 2009a):

- a Windows workstation with Microsoft Windows 2000 SP1 or later
- Intel Core Duo, Pentium 4 or Xeon Processors (1.6 GHz or higher)
- a minimum of 1 GB of RAM
- 64 MB of free disk space for loading the software
- a broadband Internet connection of 380 kbps or greater
- Microsoft .NET 2.0 or later

4.3 System Architecture

In order to fulfill the basic project requirements—availability without purchasing end-user licenses; access to built-in base data; and the broadest availability possible—it was decided early on to use AGX (build 500) as the core software environment in which to develop customizations for MODIS access and exploitation. This decision opened up two possible architectures for the final solution—a set of custom tasks in AGX that could be used in a standalone desktop environment, or the development of custom geoprocessing services on ArcGIS Server that could then be accessed using AGX as the client.

During the initial project planning phases, a client-server solution was the architecture of choice due to the vast number of tools in ArcGIS Server that are available for use in geoprocessing services. However, as the author's skill increased with AGX and its custom task framework, the architecture of choice quickly transitioned to the

standalone framework due to its characteristic of independence from any additional software.

4.3.1 Client-Server Architecture

Figure 4-1 provides a schematic view of the client-server architecture. As illustrated, AGX would only serve as a simple client through which parameters for query and other operations would be passed to ArcGIS Server. Query and download of MODIS data would then be conducted using a custom geoprocessing script on ArcGIS Server that would communicate with the MODIS RRP subset web server. ArcGIS server would then pass the resulting MODIS image back to AGX for viewing. Users could then establish the necessary image classification parameters (e.g., supervised or unsupervised classification, training points/polygons, etc.) in AGX which would then be passed back to the server. ArcGIS Server would then conduct the classification using another custom geoprocessing service developed using the built-in image classification functions that already exist in ArcGIS (maximum likelihood classification, principal components). Finally, the resulting classified dataset would be passed back to AGX for viewing or further analysis. Clearly, AGX's only role is as a user interface in which to create parameters and view results.

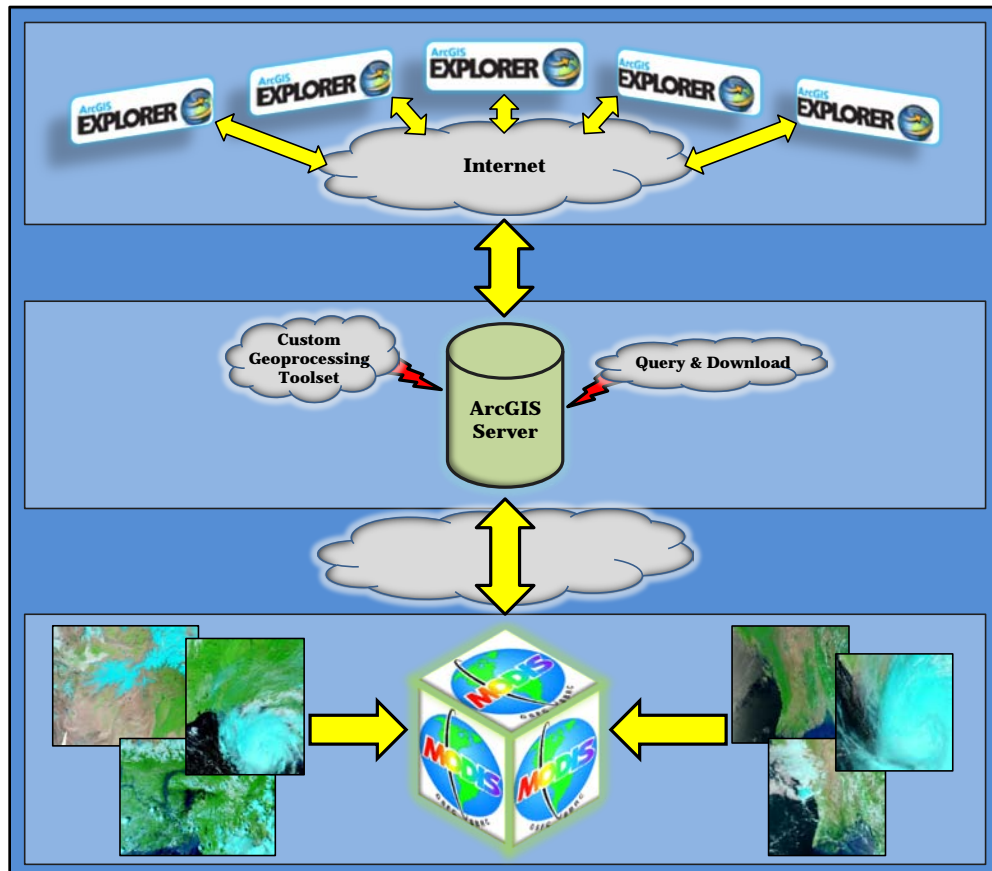


Figure 4-1. Schematic view of the client-server solution.

4.3.2 Standalone Architecture

The standalone solution has a schematic architecture that is very similar to the client-server version with one notable and obvious exception: the absence of any need for processing on ArcGIS Server (Figure 4-2). In the standalone framework, AGX would be used as the primary user interface for creating query parameters and viewing results. Meanwhile, custom tasks developed in AGX would also provide all the necessary functionality for retrieving data and conducting image classification. These custom tasks would communicate directly with the MODIS RRP web server to find and retrieve subset data. The obvious benefit of this architecture is its greater independence from additional software like ArcGIS Server. The clear disadvantage is the need for all functionality to be custom coded—pre-existing geoprocessing tools would not be available for use.

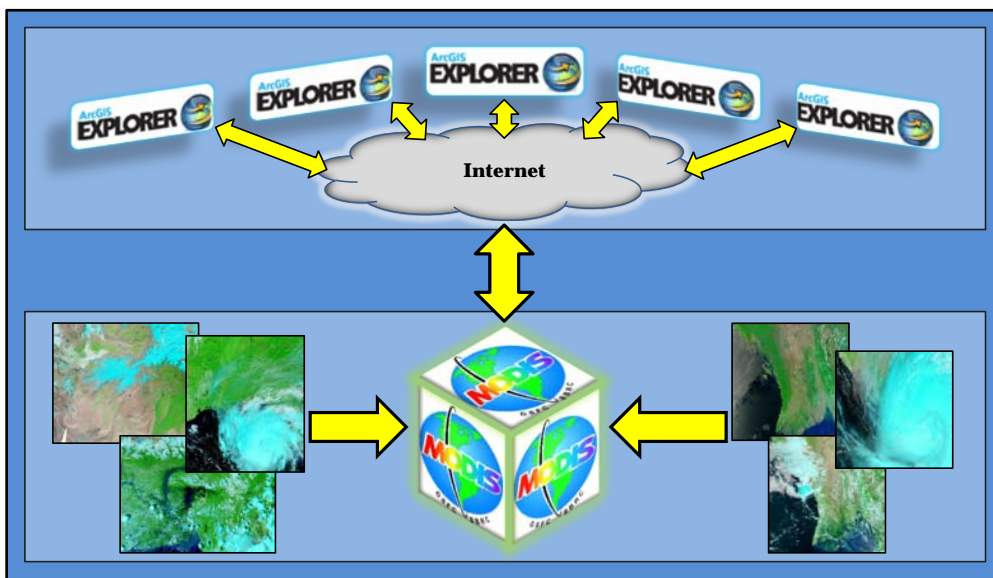


Figure 4-2. Schematic view of the standalone solution.

The ability to directly customize AGX is a powerful feature of the software and one of the reasons it was selected as the core technology for this project. Examples of pre-existing tasks for AGX include Identify, Measure, Find Place, Find Address, and Get Driving Directions—all of which are included as default tasks with installation of AGX.

The application programming interface (API) for AGX is built on Microsoft's .NET technology, which creates even more customization power for developers due to the vast library of programming classes that are available within .NET. There is also a very large .NET developer community active on the Internet, providing code snippets, suggestions, and ideas to other developers. These facts make the .NET programming languages very accessible to even the uninitiated.

4.3.3 ArcGIS Explorer Custom Task Framework

The AGX custom task framework provides the core functionality for the standalone architecture and can also be a factor in the client-server architecture. Custom tasks are developed using either VB.NET or C#.NET through a software development kit (SDK)

provided by ESRI. In addition to all the classes available via .NET, the AGX SDK enables access to an API which provides a number of programming classes specific to GIS that are very useful in developing custom spatial tools. For example, the API includes classes for working with different geometries (i.e., point, polyline, polygon), geodatabases, and raster and vector layers.

A custom task in AGX is created through overriding abstract classes provided with the SDK in conjunction with creation of a custom user interface using Windows forms classes (ESRI, 2008d) (Figure 4-3). There are two primary classes that need to be overridden and/or included to create a custom task: a “TaskUI” class, which is used to instantiate a custom task and provides the necessary information to AGX in order to do so; and a “CustomTaskControl” class (also referred to as a task window class) that is derived from a Windows forms class and supplies the user interface and supporting code that actually drives the task. It is also possible to develop custom tasks that run asynchronously for operations that may take a long time to execute, such as tasks that use connections to remote web servers. Asynchronous tasks require the addition of a third, overridable abstract class, a “Task” class, which accepts a set of parameters from the main CustomTaskControl class and runs operations on a separate thread. This class eventually creates results and passes its status back to the task window class upon completion. Asynchronous tasks add versatility to AGX because they allow users to continue actively working with the software while operations run in the background.

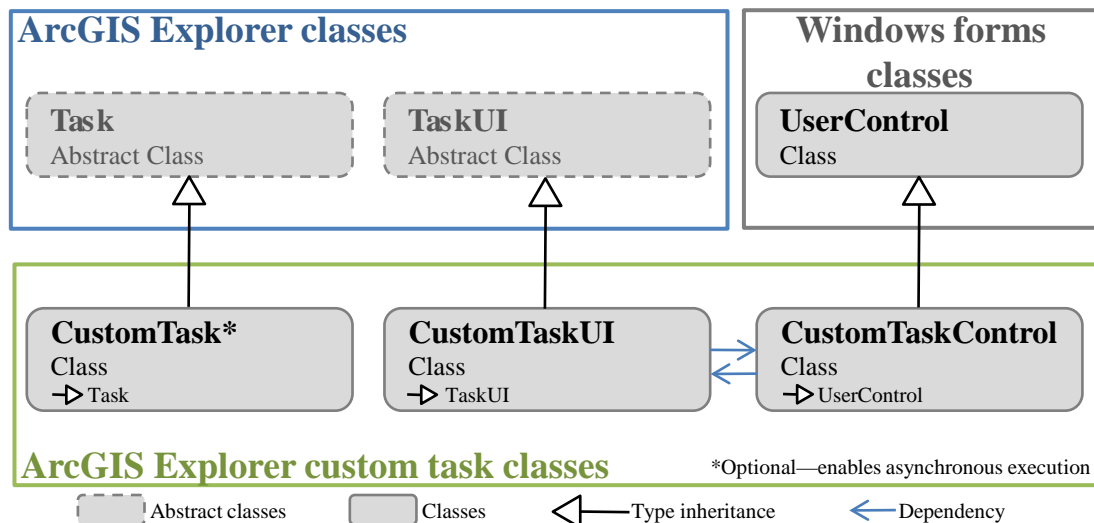


Figure 4-3. Generalized AGX custom task framework with supporting classes (ESRI, 2009b).

Once completed and compiled, custom tasks are saved as NMF files that can be shared and loaded directly into AGX by any number of users. This design, coupled with the relatively simple framework of the tasks, their development in .NET, and their ability to operate asynchronously enables a wide range of task functionality. This functionality can include everything from simple feature identification tasks to sophisticated data retrieval and database tasks that can incorporate data from remote web servers.

4.4 Project Plan

Initially, a solution based on a single architecture was envisioned that would be a hybrid of the two frameworks discussed in previous sections. The original plan was to develop a set of custom tasks that would provide some functionality within the AGX interface (e.g., point and coordinate query), while also relying on a connection to ArcGIS Server for more sophisticated functionality that might be difficult to code (e.g., image classification). During the initial development stages for the custom tasks, the author realized that all of the requisite functionality could be coded directly into the custom tasks, making them standalone tasks. This functionality included the more complicated image classification routines. Despite transitioning from a client-server framework to a standalone framework, the original project plan still provided valuable milestones. Rather than splitting the development of the required functionality between AGX and ArcGIS Server, all the same functionality was simply refocused to AGX as a standalone toolkit. The original timelines thus did not need modification. Figure 4-4 provides the originally planned workflow for the project, and Appendix A includes the project workplan that was devised to outline major tasks and guide completion. Although the required functionality was transitioned to a standalone framework, the original project workplan was still valid since none of the necessary functionality was altered. The original workflow, however, was minimally altered and will be discussed subsequently.

A number of other minor modifications were made to the original plans and goals during the process of completing the project, but these also had no major impact on the original project workflow and workplan. One of the largest discrepancies between stated goals and reality, however, was the timing of completion of the project. Initially, a fairly aggressive timetable for completion was established (see Appendix A) with the goal of completing all project tasks by the end of March. In reality, all goals were on track until early February when scheduling changes resulted in delayed completion of some tasks until early May. Despite this, enough extra time was budgeted at the end of the project to enable successful completion. This extra time was budgeted specifically for unforeseeable changes that might occur during project execution.

Another project design that turned out differently in reality was the anticipation that the three custom tasks that were originally envisioned could be developed in parallel, as illustrated in Figure 4-4. In reality, it was easier and more efficient to develop all the custom tasks in series, as illustrated in Figure 4-5. In other words, it was simpler to complete each custom task, from user interface design through documentation, prior to starting the next custom task. Despite this, the overall amount of time it took to complete the tasks was similar to what was originally planned.

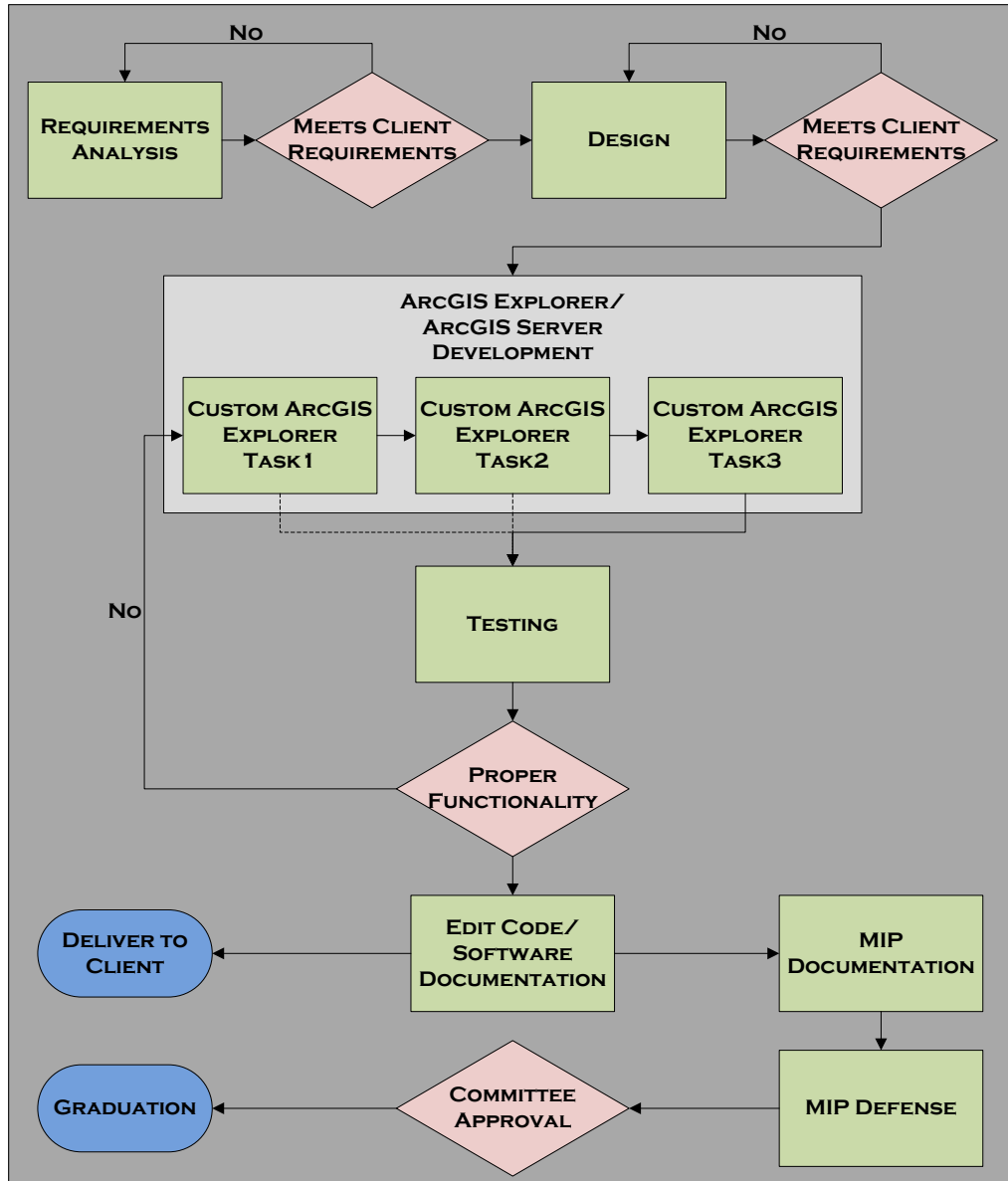


Figure 4-4. Planned project workflow schematic.

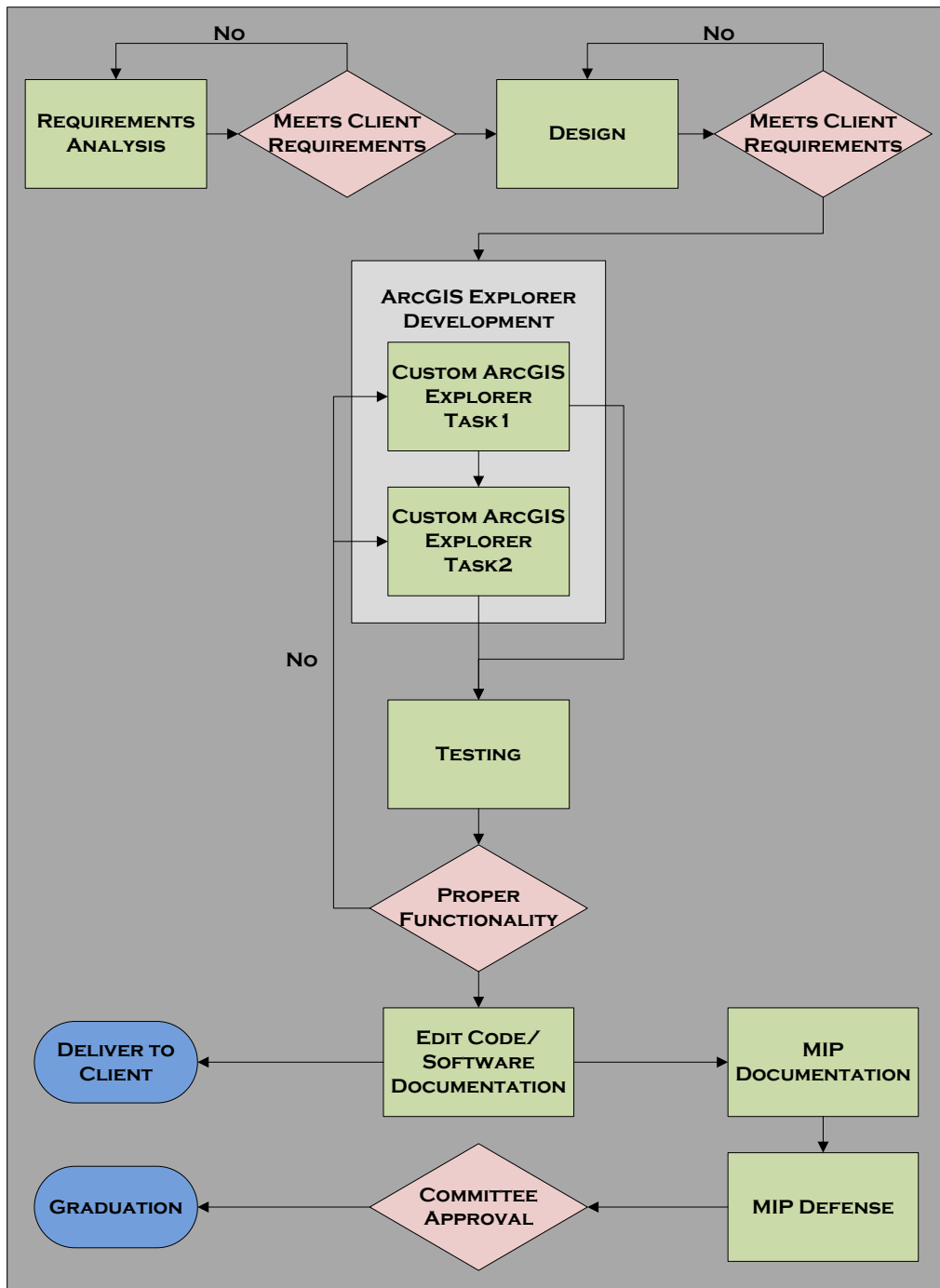


Figure 4-5. Actual project workflow schematic.

The custom tasks also changed somewhat from what was originally planned. At the outset, tasks were planned that could be used for querying and importing MODIS data, for classifying MODIS data, and for exporting classified results in either vector or raster format. While the first two tasks were completed as planned, the functionality to allow users to export data in vector *or* raster format was not fully implemented. As will be discussed in subsequent chapters, all data can be exported, but only in raster format. This

was largely due to altering the solution's architecture to a standalone framework. The custom tasks could no longer rely on built-in data management tools in ArcGIS (e.g., raster to vector conversion), so everything had to be independently coded. A tenable solution for raster data creation and export was discovered and used, but the means to enable vector export, including any existing code, was not found. Despite this, time budgeted for the vector functionality was reallocated to developing additional, unplanned functionality that added a lot to the final solution. This additional functionality included a tool for conducting change detection of classified datasets, as well as a thumbnail query tool that made exploration of MODIS more efficient.

Finally, once it was decided to alter the final architecture to a completely standalone framework, there were no plans to attempt a client-server solution. However, an opportunity arose and time was therefore allotted to trying to develop a client-server solution. Although a working client-server solution was never realized for reasons that will be discussed in Chapter 6, it was possible to explore the custom geoprocessing service framework and create a prototype solution that can be developed further in the future.

Overall, the project was completed as planned. The project benefited from the fact that AGX provided all the expected functionality needed to complete the project, so it was not necessary to change software platforms part way through the project—an event that would have dramatically affected the project's schedule. Similarly, the amount of days needed to complete the custom tasks was similar to what was planned for.

4.5 Summary

The work plan developed for this project proved to be an effective outline for completion of major tasks. Some changes were made to the proposed solution established at the outset of the project, but these modifications had only a minor impact on the final functionality and the timing of delivery.

Although both of the AGX-based final solutions outlined in the opening sections of this chapter could provide the requisite functionality that the client desired, development of a standalone framework was more congruent with the goal for the final solution to be as independent as possible. As a result, most effort throughout the course of the project was directed towards developing a standalone solution. However, in an effort to provide proof-of-concept for the client-server architecture, a prototype custom geoprocessing service was developed for ArcGIS Server. The final solutions, their advantages, and disadvantages will be discussed in chapters 5 and 6.

Chapter 5 – Implementation: Standalone Solution

In its final implementation, the standalone solution for this project, dubbed the MODIS Toolkit for ArcGIS Explorer, consists of two custom tasks developed for ArcGIS Explorer and designed to work in tandem. The tasks contain all the functionality requested by the client, but also include some additional features which increase the toolkit's scope and utility. The first custom task is referred to as the MODIS Data Retriever and provides all the necessary query and retrieval functions to enable users to search for and download MODIS RRP image subsets. The second custom task is referred to as the MODIS Data Classifier and provides image classification capabilities that enable users to derive environmental spatial data from retrieved MODIS RRP image subsets. The MODIS Data Classifier also includes a tool for conducting change detection on previously derived classified data. This chapter discusses each of the custom tasks, their functions, and their user interfaces in more detail.

5.1 The MODIS Data Retriever Custom Task

The MODIS Data Retriever was developed in VB.NET and includes two essential classes required for any AGX custom task: a *TaskUI* class and a *CustomTaskControl* class (the AGX custom task framework was discussed in more detail in Chapter 4) (Figure 5-1). Since its primary function is to query and retrieve data from a remote web server, the custom task was developed to operate asynchronously and therefore has an overridden *Task* class which enables operation on a background thread. Finally, the MODIS Data Retriever has an additional task window class that provides the user interface and functionality for data export.

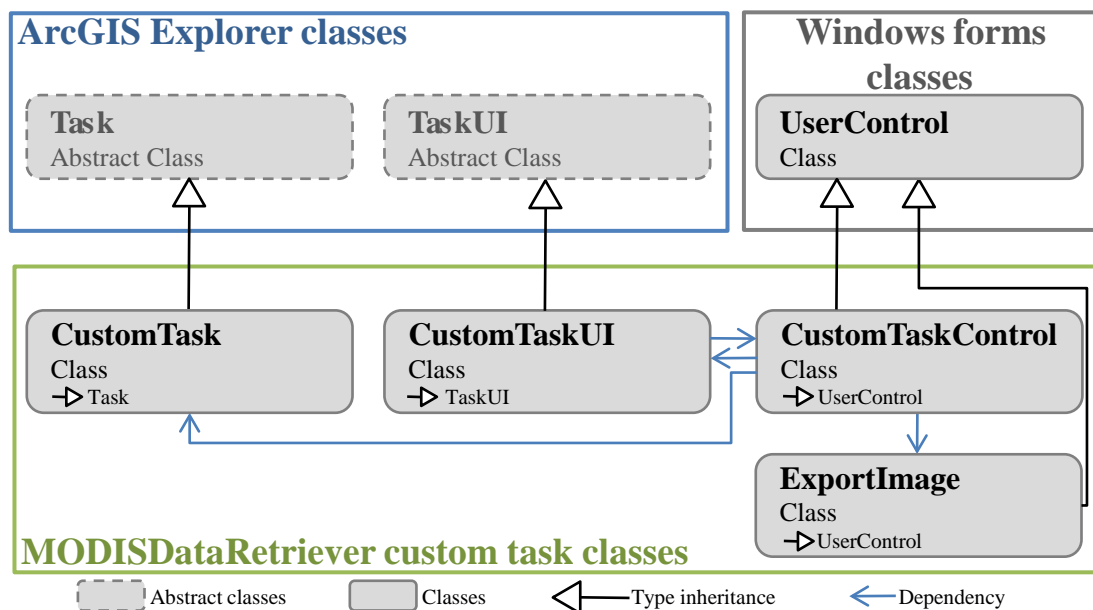


Figure 5-1. Diagram of classes that compose the MODIS Data Retriever custom task.

In addition to the main VB.NET classes that compose the custom task, the MODIS Data Retriever includes several support files that are installed to a user's local computer upon execution of the setup file for the task. These files include a geodatabase with feature classes of MODIS subset availability and of individual subset extents. These feature classes are used for display and query purposes. The setup executable also installs a help document in PDF format to a user's local computer. This document can be accessed from the main task interface.

The custom task user interface is shown added to the main AGX user interface in Figure 5-2. Upon initiating the custom task, a feature class from the supporting task geodatabase is added to the globe and given a red fill with 80% transparency. This feature class illustrates the availability of MODIS RRP image subsets. The goal of this feature class is to reduce the futility of querying for data where it does not exist, but the feature class can be turned off or removed at the user's discretion.

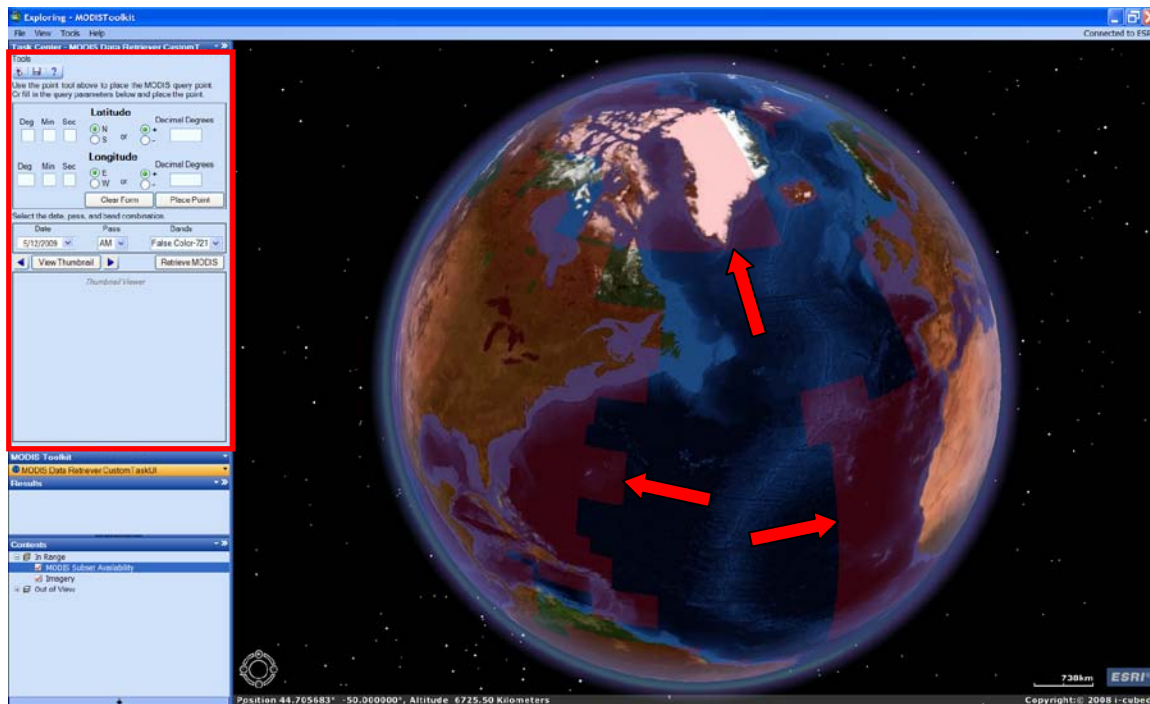


Figure 5-2. ArcGIS Explorer user interface with the MODIS Data Retriever custom task interface added to the task panel (red rectangle). The red arrows indicate the extent of the MODIS subset availability feature class.

5.1.1 MODIS Subset Query Points

Figure 5-3 details the primary user interface for the MODIS Data Retriever, with query tools and interfaces highlighted. Query points can be added to the AGX globe either by using a query point tool or by manually entering coordinates. When the query point tool button is selected (Figure 5-3, number 1), it activates a cursor that allows the user to manually add a point of interest to the globe. The coordinates of this point then form the basis of a subsequent MODIS data query. Alternatively, users can manually enter text-based coordinates into the text query form (Figure 5-3, number 2) and add query points to

the globe. The text query form will accept coordinates in either degrees-minutes-seconds format or decimal degrees. A significant amount of built-in validation was coded to support the text query form so that no invalid coordinates can be entered.

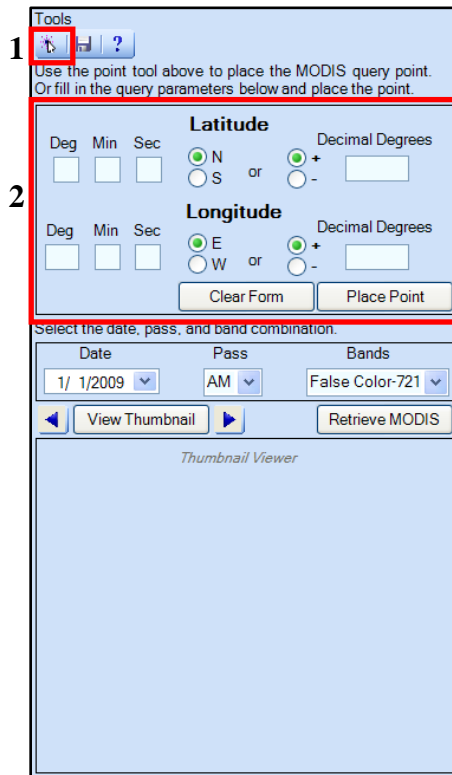


Figure 5-3. Detail of the MODIS Data Retriever user interface with query tools and interfaces highlighted.

Upon placement, query points appear as a small globe icon added to the AGX interface, and the extent of the most appropriate RRP subset appears as a white outline, indicating the actual subset on which the query will be based (Figure 5-4). This subset is determined through coded logic that:

- determines all the subsets that contain the query point (since there can be significant overlap amongst the subsets)
- iteratively calculates the great circle distance between the query point and the centroid of each intersecting subset
- iteratively updates a variable with the name of the subset whose centroid is closest to the query point

(see Appendix B, Code Example 1)

The purpose of this logic is to determine the subset with the greatest coverage around a given query point. Additional built-in validation ensures that query points beyond the extent of subset availability cannot be added to the globe.

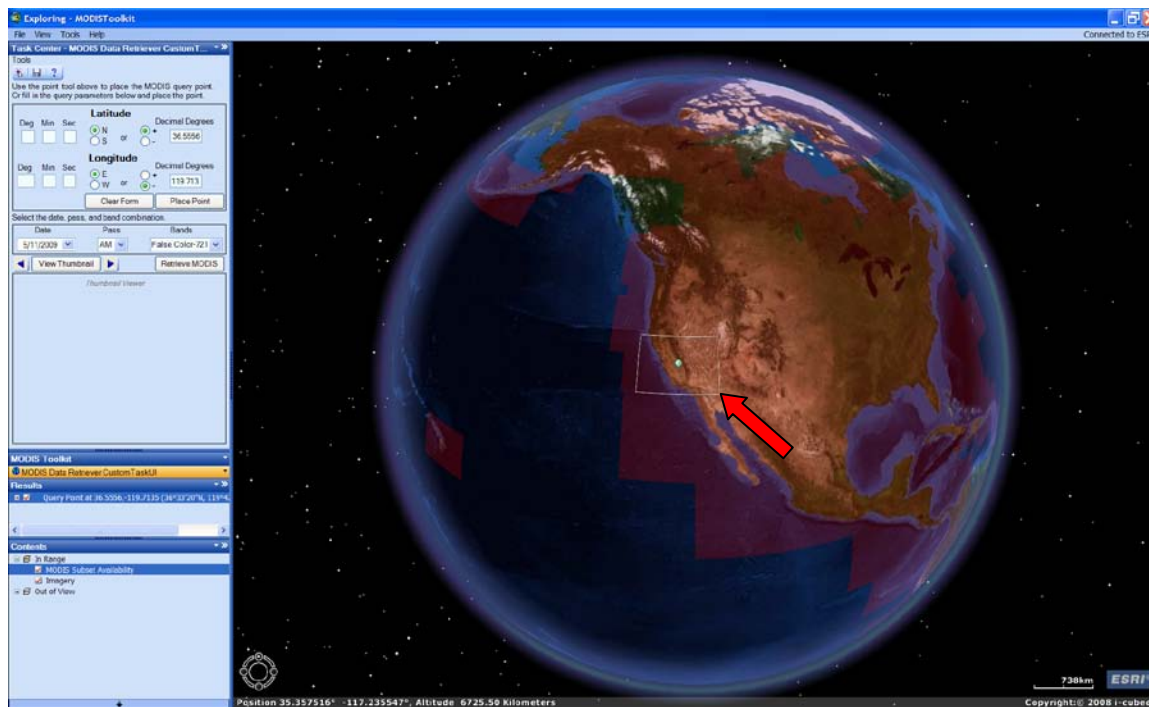


Figure 5-4. Detail of a query point added to the AGX globe and the associated subset extent outlined in white.

5.1.2 MODIS Data Exploration and Retrieval

Once a valid query point has been added to the AGX globe, three additional MODIS image subset parameters must be selected prior to retrieval of thumbnails or image subsets:

- the date of the subset
- the subset’s satellite platform (Terra or Aqua), which is also referred to as the pass (AM or PM)
- the subset band combination of interest (true color, false color, or NDVI)

These parameters are set by the user in the parameter section of the form (Figure 5-5, number 1). Note that these parameters are also necessary for final subset retrieval. With a valid query point and the three subset parameters set, a thumbnail image of the subset can be retrieved using the “View Thumbnail” button (Figure 5-5, number 2), or the highest resolution version of the subset can be retrieved using the “Retrieve MODIS” button (Figure 5-5, number 3).

Thumbnail queries allow users to view a lower resolution thumbnail of an image subset. These lower resolution images have much smaller file sizes than their high resolution counterparts and therefore can be downloaded relatively quickly. If a thumbnail appears to satisfy a user’s needs, then the user can retrieve the high resolution version of the subset. Once retrieved, thumbnail images are displayed in the thumbnail viewer embedded in the user interface (Figure 5-5, number 4; Figure 5-6). If a subset of interest is not available, a message box to that effect is displayed to the user.

To aid in subset exploration, forward and backward buttons are available next to the “View Thumbnail” button (Figure 5-5, number 2). These retrieve the next or last subset relative temporally to the one that is currently displayed. For example, if the currently displayed thumbnail is the true color subset from the morning of January 1, 2009, using the forward button will display the true color subset from the afternoon of January 1. The backward button will display the true color subset from the afternoon of December 31, 2008. This functionality was added so that users would not have to continually open the date/time and satellite pass selection tools if they are simply looking for the most appropriate subset around a specific date (e.g., the least cloud covered).

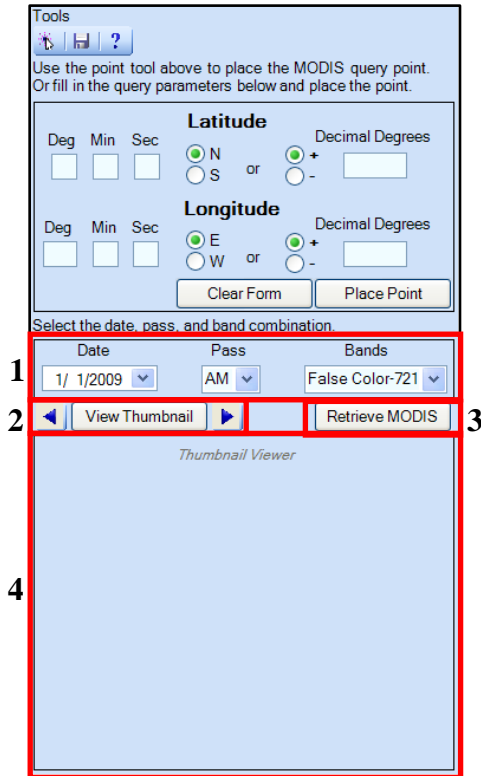


Figure 5-5. Detail of the MODIS Data Retriever user interface with thumbnail tools and interfaces highlighted.

Retrieval of the high resolution subset images uses the same parameters established for thumbnail retrieval and is achieved using the “Retrieve MODIS” button. This button will initiate download of the subset and its associated world file using a uniform resource identifier (URI) created from input parameters and the subset name associated with the query point (Appendix B, Code Example 2). The following text reveals how the URI string is constructed (Appendix B, Code Example 3):

```
"http://rapidfire.sci.gsfc.nasa.gov/subsets/?subset=" +
  [MODIS Subset Name] + "." + [Year] + [Day] + "." + [Pass] + [Bands] + ".250m.jpg"
```

Similar variables and logic are used for thumbnail download, the primary difference being the resolution of the image retrieved. Thumbnail URIs and filenames replace the

“.250m” term with a “.2km” term. The high resolution images and thumbnails are retrieved asynchronously using the threading functionality enabled by inclusion of a Task class in the final custom task framework (Figure 5-1).

Downloaded images and world files are saved to a temporary directory that is programmatically created on the user’s local computer when the task is used. Note that similar code and another temporary directory are also used for retrieving the smaller thumbnail files during thumbnail queries. The temporary directories for both thumbnails and final images are automatically cleaned up every time the custom task is used. Any files older than one day are automatically removed from the directories through coded logic to ensure that the directories do not become filled with unneeded data. As a result, however, users must export any data of interest to another local directory of choice. Tools to conduct data export are discussed in the next section.

Depending upon subset file size, bandwidth of a user’s Internet connection, and traffic on the web server, subset download can take anywhere from 15 seconds to several minutes. Once download is complete, the image is automatically added to the AGX globe (Figure 5-6).

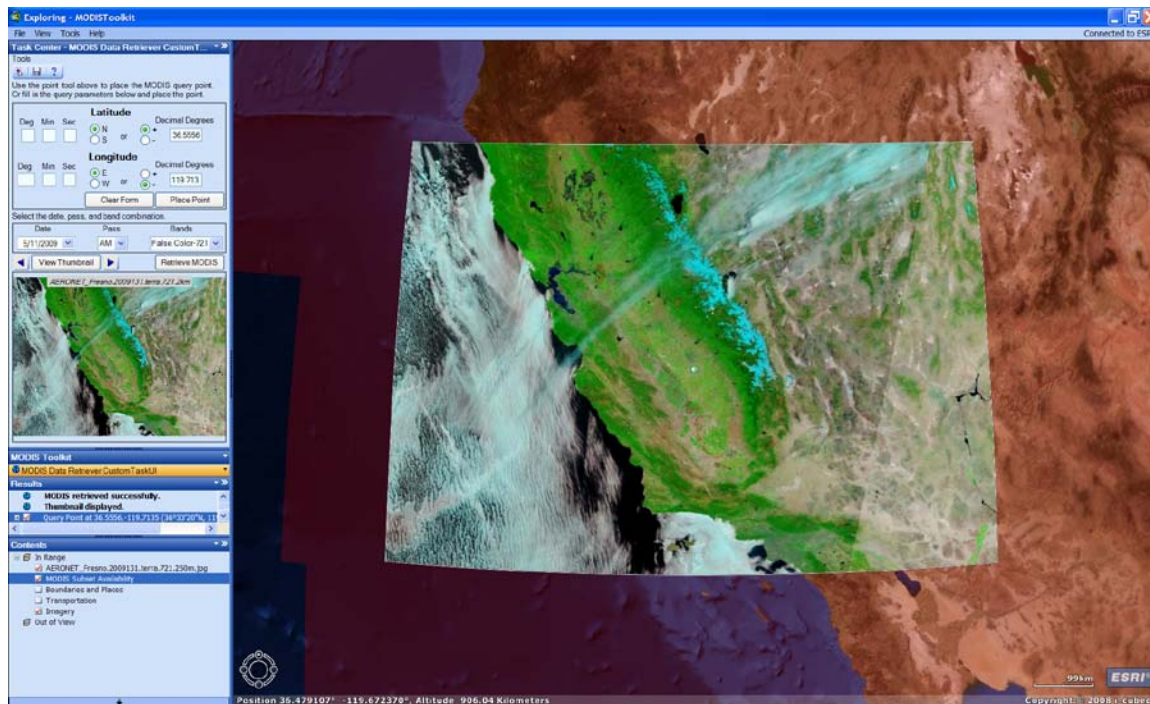


Figure 5-6. Detail of a subset thumbnail and retrieved subset image added to the AGX globe.

Addition of subset images to the AGX globe is dependent upon two necessary files: a world file for accurate geopositioning of the data (e.g., coordinates of the subset’s origin, pixel dimensions, rotation terms), and a spatial reference. Although some subsets with unusual spatial references do exist, particularly in polar regions, the vast majority of the RRP subsets, and all of those accessed by the MODIS Toolkit, use WGS-84 as their spatial reference. The world file is downloaded from the RRP web server, but the spatial

reference files are created programmatically as XML files for every downloaded image and thumbnail (Appendix B, Code Example 4).

Given the large number of images that individual users may retrieve and save locally using the MODIS Data Retriever, manual metadata creation by users could be very time consuming and would likely lack standardization. As a result, the custom task also programmatically creates metadata in a standard ESRI XML format for every image that is retrieved (Appendix B, Code Example 5) (note that metadata is not created for thumbnails since they are only used for data exploration). This metadata file is consistent with the requirements of ArcGIS and includes information about the data's source, its original filename (on the RRP web server), acquisition date, and a number of keywords related to the subset. Figure 5-7 illustrates an example of this metadata as it appears in ArcCatalog.

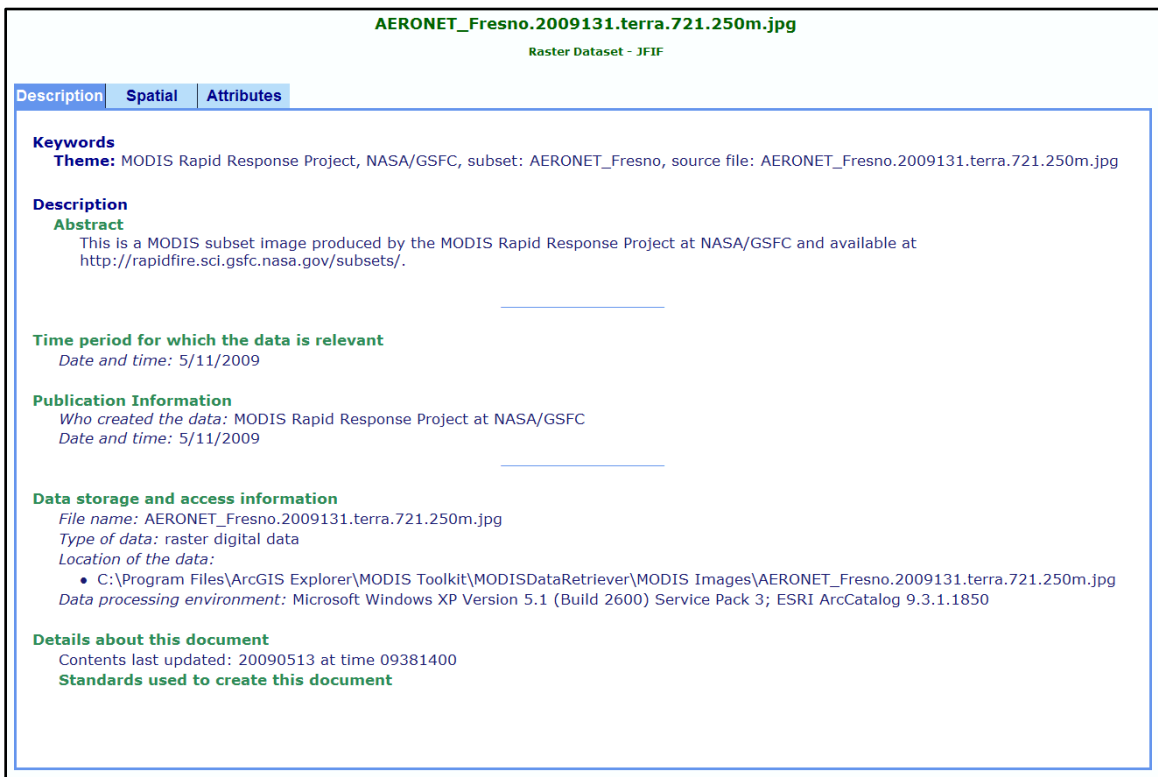


Figure 5-7. Example of metadata created automatically by the MODIS Data Retriever, as seen in ArcCatalog.

5.1.3 MODIS Data Export

Retrieved image subsets are stored only in a temporary directory that is established on the user's local computer. This directory is cleaned up every time the custom task is used by removing files that are older than one day. Therefore, if interested in saving a retrieved image subset, a user must export it to a local directory using the export tool (Figure 5-8). Reasons for developing this extra tool are twofold: it was assumed that users would not want to archive every image subset that they retrieved. It is therefore easier to programmatically maintain these images and obligate the user to archive only those that are of interest; and AGX does not provide the necessary functionality to save datasets in

their native formats—they can only be exported as AGX layer files (.lyr)—so the export tool provides this capability for raster data (vector data will be discussed in subsequent sections).

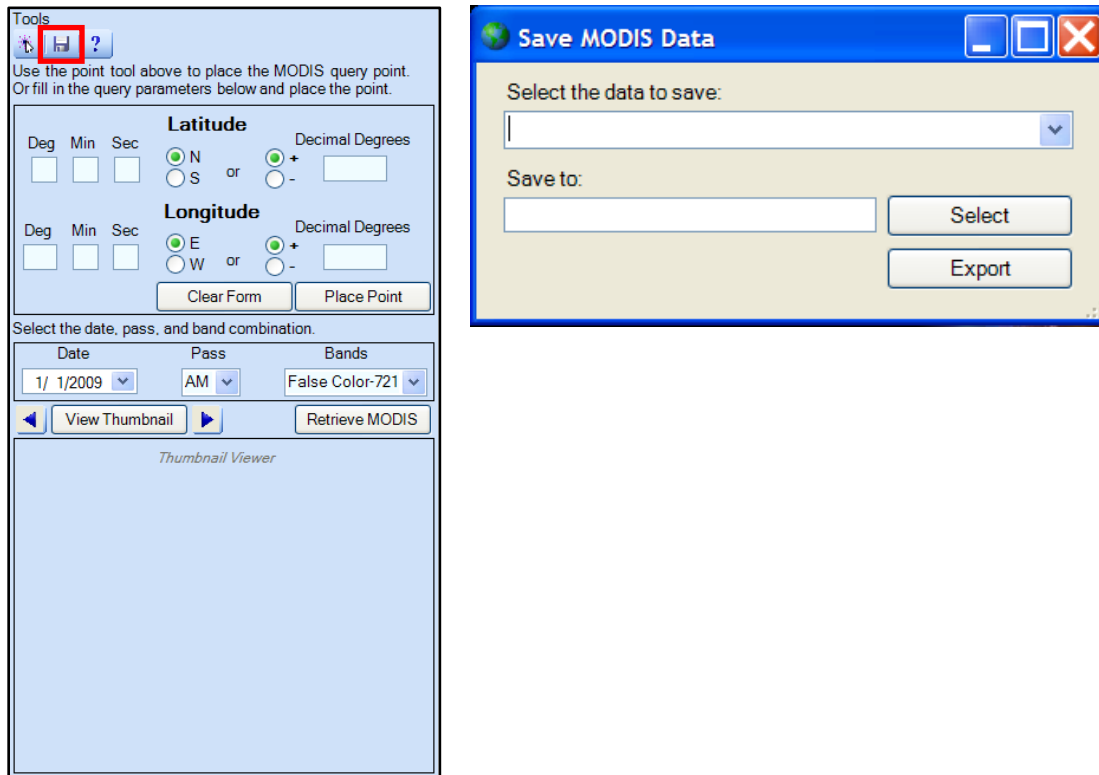


Figure 5-8. Detail of the MODIS Data Retriever user interface with the data export tool and interface highlighted.

When the file export user interface is launched, the user must select a file from the upper combo box dropdown menu. Since the tool was specifically designed for MODIS subset archiving, this combo box will display only raster datasets that are currently added to the AGX globe view. The lower text box is associated with a “Select” button that opens a file system explorer window, enabling the user to select the destination directory. When the “Export” button is implemented, the image subset is saved to the local directory specified by the user. All supporting files (i.e., world file, spatial reference, metadata) are also automatically saved to the new directory. Once completed, the AGX path associated with the file is updated and the export user interface displays a message indicating whether or not the file was exported successfully.

5.1.4 Help Document

Another tool available to users within the MODIS Data Retriever interface is the help button, which appears as a question mark on the main toolbar at the top of the interface. This button opens a help document in PDF format which contains information about the background and functionality of the custom task as an aid in using the task.

5.2 The MODIS Data Classifier Custom Task

The MODIS Data Classifier was also developed in VB.NET and includes the two essential custom task classes: a *TaskUI* class and *CustomTaskControl* class (Figure 5-9). This task was also developed to operate asynchronously and therefore includes a *Task* class. The primary reason that asynchronous execution was selected for the MODIS Data Classifier is that, depending on the image subset being classified, a second subset may need to be retrieved from the MODIS RRP web server prior to classification. This will be discussed in more detail in subsequent sections. Like the MODIS Data Retriever, this custom task also has a task window class to enable data export, but it has an additional task window class and supporting code that enable change detection.

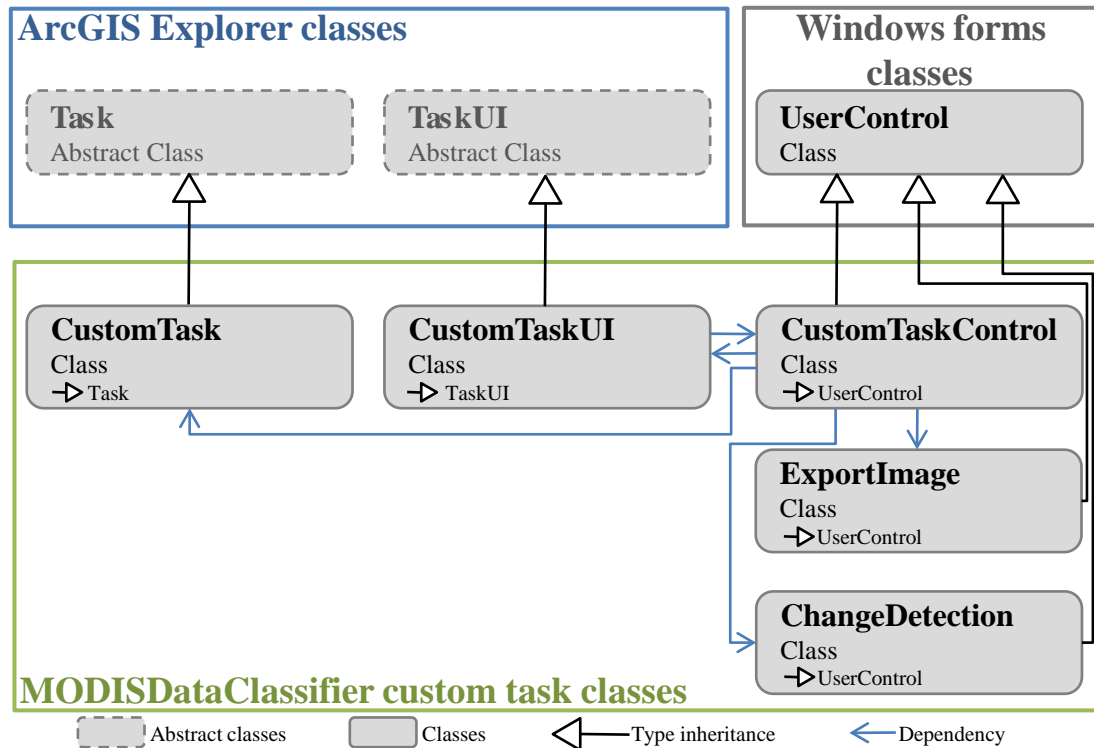


Figure 5-9. Diagram of classes that compose the MODIS Data Classifier custom task.

The MODIS Data Classifier also includes a supporting PDF help document that is installed to the user's local computer upon executing the setup file for this task. Figure 5-10 shows the MODIS Data Classifier user interface added to the main AGX user interface.

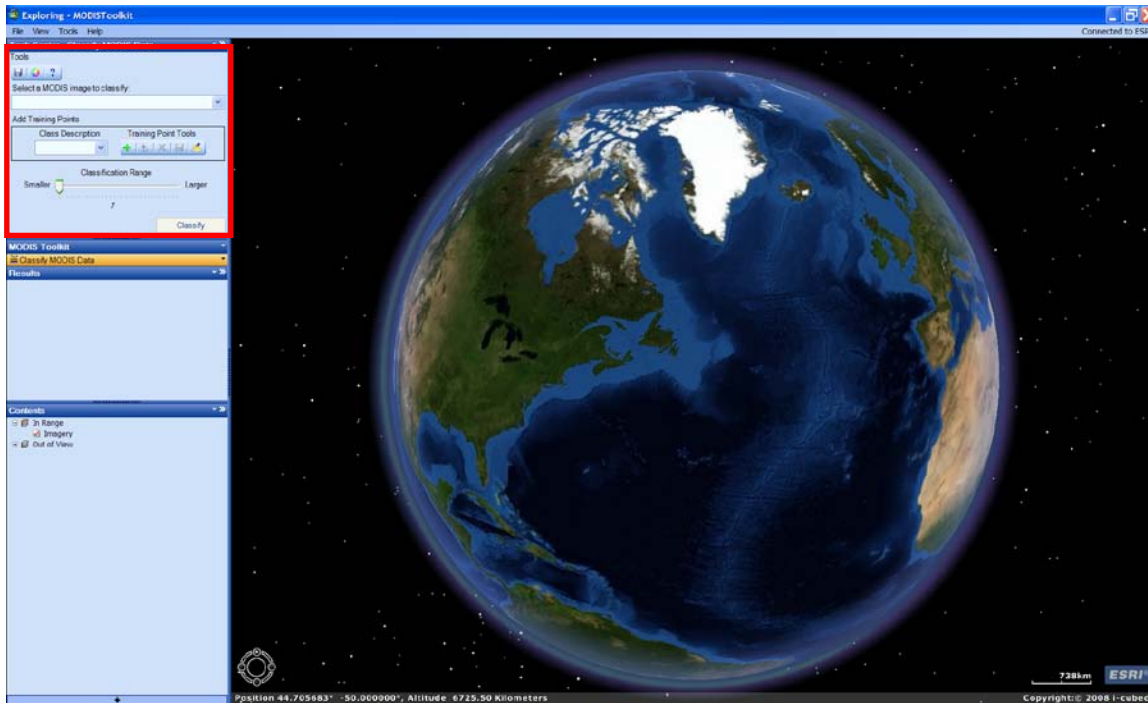


Figure 5-10. ArcGIS Explorer user interface with the MODIS Data Classifier custom task interface added to the task panel (red rectangle).

Unlike other image classification tools available in software like ArcGIS and ERDAS Imagine, the MODIS Data Classifier was designed to classify only one image class at a time. The reason for this stems from the fact that the toolkit was developed for analysts deployed in military support and disaster response settings. These analysts will be primarily interested in developing data for singular environmental parameters, such as snow cover, flooded areas, wildfire extent, etc. Despite this design, multiple image classes can be derived from the same MODIS subset by simply reapplying tools in the MODIS Data Classifier.

5.2.1 Adding Training Points

Training points are pixels that are selected from an image to represent an image class of interest (i.e., water, snow, vegetation, etc.). The values, or digital numbers, associated with these pixels are used to define feature class characteristics in spectral space. Regarding the MODIS Data Classifier, image classes are defined statistically using the mean and standard deviation derived from the training points.

Figure 5-11 highlights the tools and interfaces used for defining training points. Prior to adding any training points the user must select a MODIS image from the combo box dropdown menu (Figure 5-11, number 1). Built-in validation ensures that only MODIS subset images retrieved using the MODIS Data Retriever are allowed to be classified with the custom task. This is due to the fact that the custom task was developed to work specifically with MODIS RRP image subsets and no other image data. The MODIS Data Classifier ensures this by scanning the metadata file (if it exists) for any image that is currently added to the AGX globe view. The metadata must include the term “MODIS Rapid Response Project,” but must not include the terms “classified” or “change.” The

former ensures that the file is associated with the RRP and the latter ensures that the file is not a previously created classified or change detection dataset developed using the MODIS Data Classifier. The terms “classified” and “change” are used in the metadata of classified and change detection datasets, respectively (Appendix B, Code Example 6). This logic is necessary because classified and change detection datasets could also be added to the globe concurrent with image subsets. The result of this logic is a dropdown list of image subsets.

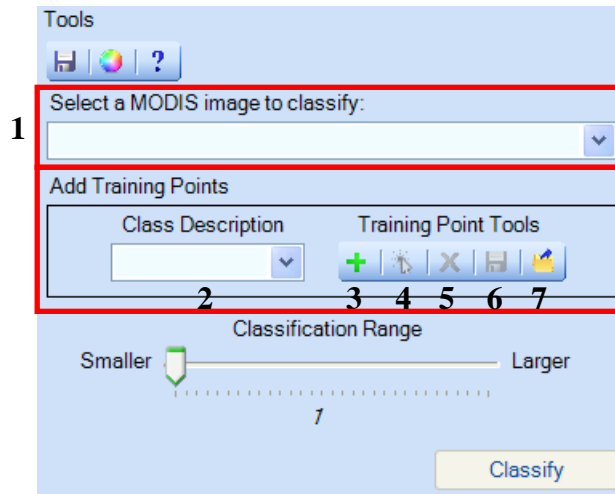


Figure 5-11. Detail of the MODIS Data Classifier user interface with training point tools and interfaces highlighted.

Another reason for the prerequisite that an image be selected is that the task also includes built-in validation that allows training points to be added to the globe only within the envelope of an image. Selection of an image by the user enables the boundaries of valid training point addition to be programmatically determined and enforced. This ensures that no training points can be added close to, but outside of an image, and also ensures that all calculated classification statistics are for one image only. Overall, this functionality ensures that training points are associated with the correct image and thereby reduces errors in calculating image class statistics for a given image.

Once an image is selected, an image class description must also be selected before classification of an image can occur (Figure 5-11, number 2). This description is used in metadata creation and helps ensure accurate description of all resulting datasets. A number of pre-existing descriptions exist within the dropdown menu of this combo box, but users can also manually enter a description that is most suitable for the class they are interested in deriving.

With both an image and class description selected, the “Add Training Point Set” button (Figure 5-11, number 3) will activate the point selection tool (Figure 5-11, number 4). The point tool enables users to begin adding training points to the globe. The necessity of having to first select the “Add Training Point Set” button before adding points ensures that all points added to the globe are part of the same training set. Once the first training point is added, the “Remove Training Point” and “Export Training Point” buttons will be enabled (Figure 5-11, numbers 5 and 6). The “Export Training Point” button allows users to save a set of training points as a CSV file to their local computer.

This CSV file contains spatial locations of training points, but not any spectral information associated with the current image.

The final tool is the “Import Training Point” tool that then allows users to re-add training points from a previously saved CSV file (Figure 5-11, number 7). Allowing users to save and re-use training point sets from a local directory is an important time-saving feature given the effort and time required to create a quality training point set that accurately captures an image class’s statistics. It is important to note that reusing a training point set for an image other than the one for which it was created could be problematic, since the extent of image class will change through time. Figure 5-12 shows a false color image with a series of training points for “snow” added to the globe.

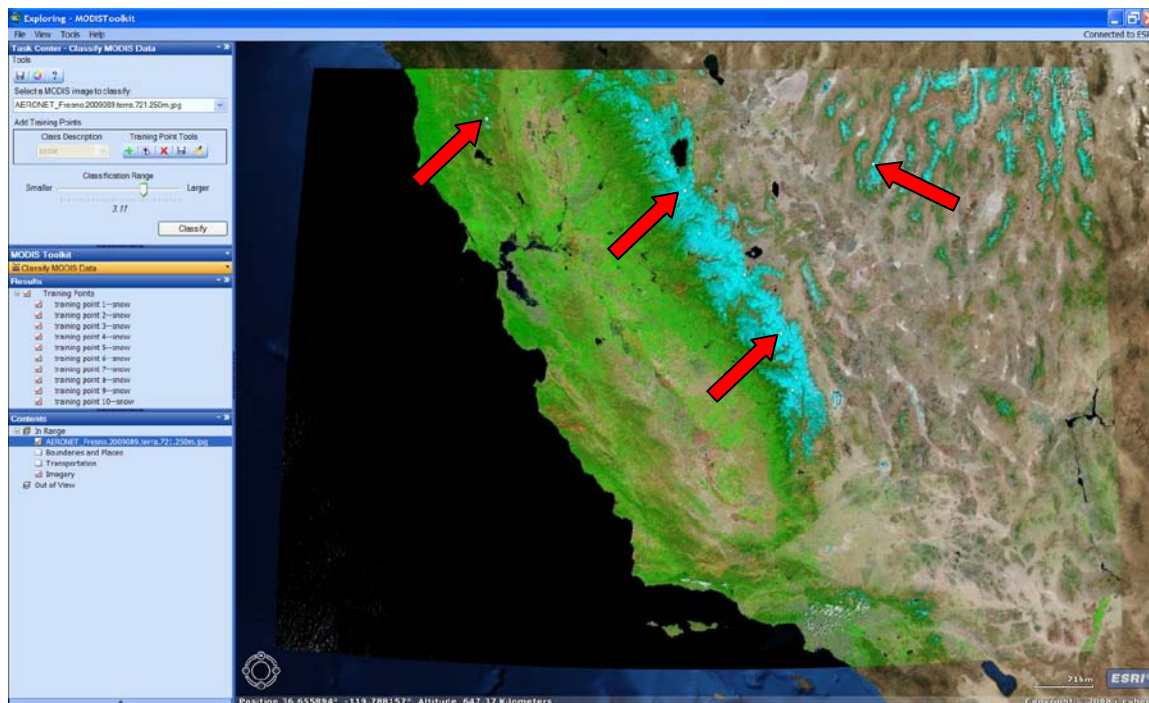


Figure 5-12. Training points representing “snow” added to a false color MODIS subset on the AGX globe. Red arrows indicate the locations of some of the training points.

5.2.2 Image Classification

With a training point set added to an image, the only other necessary parameter to set prior to processing an image classification with the MODIS Data Classifier is the size of the classification envelope. This is set via the “Classification Range” trackbar which allows settings between 1 and 4. Classification range is a more user-friendly term used to define the number of standard deviations that will be used by the task in calculating an envelope of inclusion, or parallelepiped, for the image class of interest. The custom task will calculate the training point mean and standard deviation for each MODIS band in the selected subset. The custom task will then increase or decrease the range for pixel inclusion into a class by increasing or decreasing the number of standard deviations via the range trackbar. Therefore, the two parameters available to a user in conducting a

MODIS subset classification are the number and variability of training points and the range setting.

With a training point set added to the globe and the range set, classification is conducted using the “Classify” button. This button initiates a series of logic that produces the classification results (Figure 5-13). The first step in this logic is to create a two-dimensional array of training point locations, based on row-column coordinates, that are determined using the decimal degree coordinates of each training point and parameters from the subset world files, including the coordinates of the subset’s origin and the pixel size in decimal degrees (Appendix B, Code Example 7). This array of row-column coordinates is used later in the logical flow to develop training point statistics.

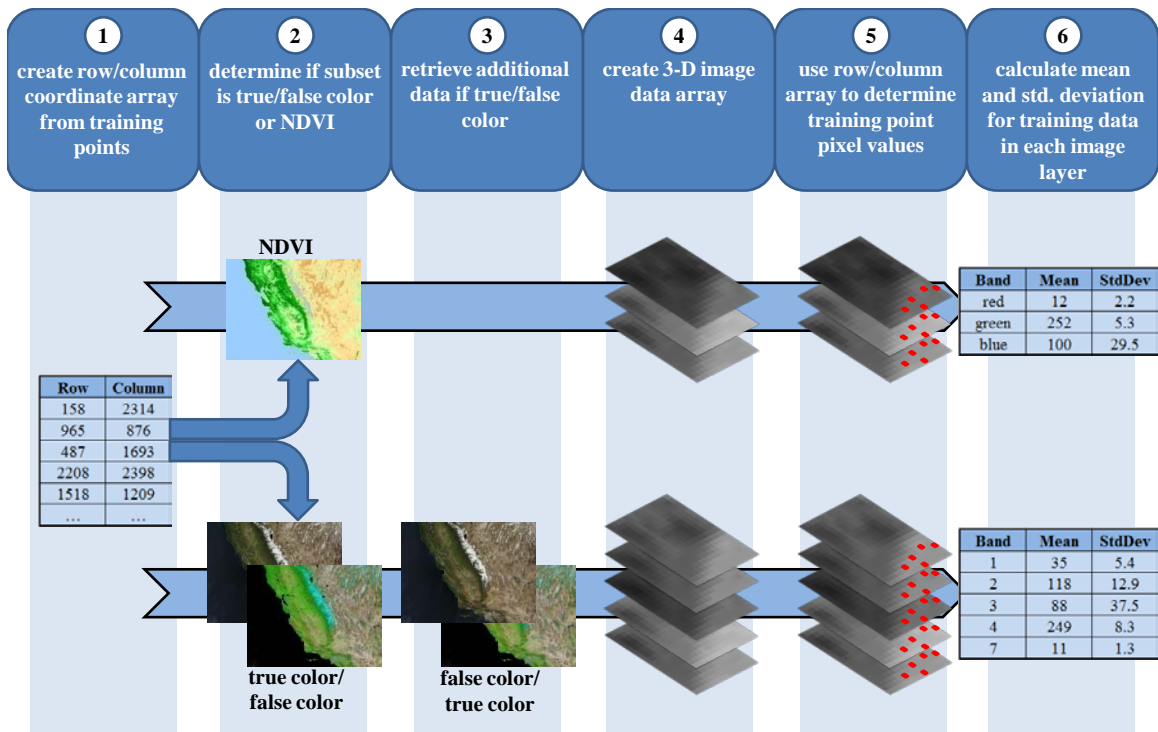


Figure 5-13. Logical flow used to conduct custom task image classification.

The second logical step is the determination of which subset image was selected for classification—true color/false color, or NDVI. If the image is either true color or false color, then the procedure used to classify the image is the same; the custom task retrieves the alternate image subset, hence the asynchronous execution. In other words, if the image selected for classification is the true color subset, then the task will retrieve the false color subset and vice-versa. Note that the extra image will be saved to the same directory as that of the image being classified—either the temporary or an archive directory. The goal of this is to use all available MODIS subset bands to improve the results of the classification. Using both the true color and false color subsets yields a total of five bands—MODIS bands one, two, three, four, and seven. Both of these subsets have three bands, but they overlap coverage of MODIS band one, resulting in five total bands.

If the image selected for classification is the NDVI version of the subset, then it is processed by itself since there are no other associated subsets. It is important to clarify that the NDVI subsets represent calculated NDVI values that are color-coded. As such,

the three bands that compose the NDVI JPEGs are not bands of MODIS data. Rather, they are bands representing the red, green, and blue color values that create the color-coded NDVI image. Therefore, using the custom task classification tools on NDVI subsets is not true image classification, but it allows users to select specific NDVI values and separate them for analysis. This functionality was added specifically for users interested in applying the MODIS Toolkit to drought and/or crop monitoring. The following discussions pertain to classification of the five true MODIS image bands from the true and false color subsets, but the logic supporting NDVI “classification” is similar, using only three bands.

Prior to retrieving any additional data, the custom task will first scan the directory containing the image being classified to determine if the additional image has already been retrieved. If it has been retrieved, then the task will use this image rather than try to retrieve it again. This logic clearly saves time and reduces redundant data. In general, it also makes conducting additional classifications of the same image subset much quicker to execute since the data is already available locally.

With all image data available, the next step in the classification algorithm is the creation of an image data array. For five bands of data, this array is a three-dimensional array that can essentially be visualized as a stack of five two-dimensional arrays (one for each band) with dimensions matching the height and width (in pixels, or rows and columns) of the source images. Each layer in the array stack contains single band digital numbers for each pixel in the source images. Image data (digital numbers) for the five bands are then copied into the image data array (Appendix B, Code Example 8).

The image data array is then used in association with the previously created training point row-column coordinate array to calculate individual band statistics for the training points. This is done by iterating through the training point row-column coordinate array and using the row-column coordinates to access values in the image data array that are then used to calculate means and standard deviations for each MODIS band (Appendix B, Code Example 9).

Finally, the means and standard deviations calculated for each band of the training data are used in conjunction with the range value set on the range trackbar to define minimum-maximum value ranges for each band, per the following equations:

$$\begin{aligned} \text{minimum value} &= \text{mean} - (\text{range setting} * \text{standard deviation}) \\ \text{maximum value} &= \text{mean} + (\text{range setting} * \text{standard deviation}) \end{aligned}$$

where the range setting is converted back to standard deviations prior to the above calculations. These minimum-maximum ranges therefore delimit the envelope of inclusion, or parallelepiped, for the image class being classified.

The entire subset of interest is then classified by iterating completely through the image data array, testing each pixel to see if all its band values fall within the envelope of inclusion (Appendix B, Code Example 10). Parallelepiped classification is based on the logic that if any values fall outside the envelope of inclusion, then they are not considered to be part of the image class. An entirely new PNG format bitmap is then used to record the results. This new PNG is identical in dimensions (height and width, or rows and columns) to the original image subset, and its world file and spatial reference are derived by copying the same files from the original subsets. Pixels whose values fall within the

envelope of inclusion are set in the new bitmap to be red, while all others are set to be transparent. The 32-bit PNG format was selected to record final classification results due to the extra 8-bit channel it includes for transparency (i.e., alpha channel). This allows the classification results to be displayed in AGX without obscuring regions of the globe that do not meet classification criteria (Figure 5-14).



Figure 5-14. Results of image classification performed on a false color MODIS subset to identify pixels representing snow.

Upon completion of a classification, the text of the “Classify” button changes to “Reclassify”. The purpose of this is to indicate to a user that modifications to classification parameters can be made to produce better results, and is useful given that many attempts are usually necessary in order to obtain the desired results. Modifications can include the addition of more training points to improve training statistics, or further adjustment of the range trackbar. Subsequent classifications are given the same filename, but are numbered sequentially starting with the number one.

Like the MODIS Data Retriever, all classification results from the MODIS Data Classifier have an XML metadata file automatically created upon execution of a subset classification. The metadata for classified results includes additional information to indicate how the results were derived, the source MODIS subsets used, and the image class represented by the data.

Like the MODIS Data Retriever, a tool for exporting classified image results to an archive directory of choice is located on the main toolbar. In addition, a PDF help file for the MODIS Data Classifier can be accessed through a help button on the main toolbar.

5.2.3 Change Detection

The final tool available to users in the MODIS Data Classifier custom task interface is the change detection tool that can be accessed via the change detection button on the main toolbar. Similar to the export data tool, a new user interface opens that requires input (Figure 5-15). This interface only requires the selection from the combo box dropdown menus of two previously classified image results that must adhere to the following criteria:

- datasets must be derived from the same MODIS RRP subset (i.e., same geographic extent)
- datasets must be from different dates/times; they can be from the same date, but one must be derived from the morning satellite pass (i.e., terra) and the other from the afternoon pass (i.e., aqua)
- datasets must be derived for the same image class (i.e., snow, water, vegetation, etc.)
- datasets must already be added to the current AGX globe view

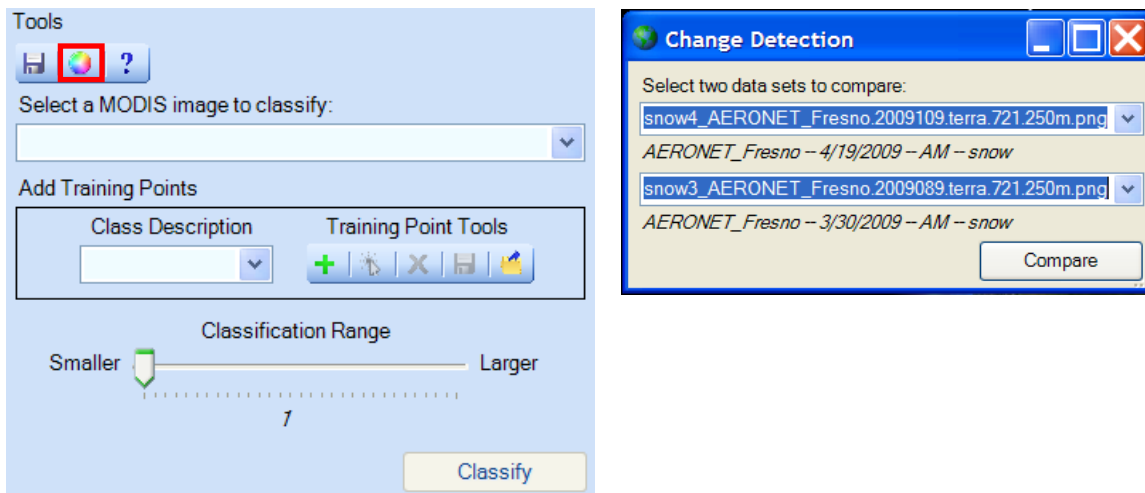


Figure 5-15. Detail of the MODIS Data Classifier user interface with the change detection tool and interface highlighted.

Once a dataset has been selected, pertinent information (i.e., subset name, image class, date/time) is displayed below the combo box to aid the user in selection of appropriate datasets. Built-in validation exists such that the “Compare” button will only be enabled when two datasets that meet the above criteria are selected (Appendix B, Code Example 11). Datasets do not need to be selected in chronological order. The custom task will determine this independently.

Areas of change are identified by comparing the two classified datasets on a per-pixel basis. The results are written to a new PNG format bitmap with pixels colored as follows based on results:

- green pixels indicate no change in the image class between the two images

- red pixels indicate areas representing the image class in the first dataset (chronologically) that were no longer present in the second dataset; in other words, the image class is no longer represented by the pixel values
- blue pixels indicate areas representing the image class in the second dataset (chronologically) that were not present in the first image; that is, new areas where the image class is present.

The PNG format was used once again to allow pixels absent in both classifications to be made transparent for data visualization purposes. Figure 5-16 illustrates two original image classifications from two separate dates and the results of change detection conducted on these datasets.

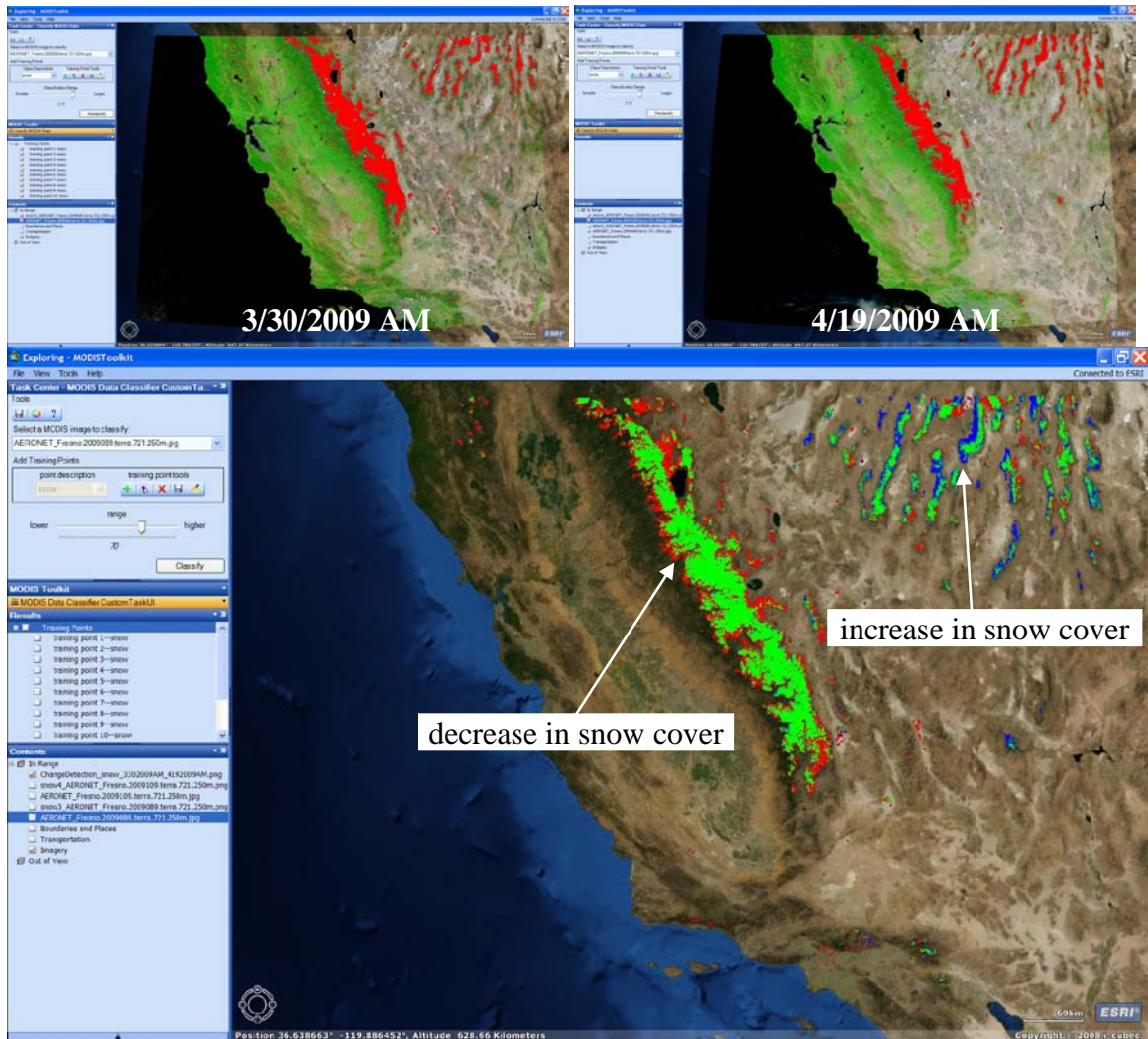


Figure 5-16. Change detection result from two image classifications of “snow” for the “AERONET_Fresno” RRP subset from the mornings of 3/30/2009 (top left) and 4/19/2009 (top right).

5.3 Summary

The MODIS Toolkit for ArcGIS Explorer, consisting of the MODIS Data Retriever and MODIS Data Classifier custom tasks, provides a standalone solution for the needs of this project. As a whole, the toolkit provides users with the ability to query for, explore, and retrieve MODIS RRP image subsets. The toolkit also provides the ability to classify the subsets to derive environmental spatial data, then compare classified datasets to determine change. Automated data handling and metadata creation aid users in managing all resulting data from the toolkit, including all image subsets and classified and change detection datasets.

Chapter 6 – Implementation: Client-Server Solution

As previously noted, most of the effort put forth towards developing a final solution for this project was aimed at the standalone solution. This is due to the assessed superiority of a standalone solution because of its independence from any additional software. Despite this fact, a client-server approach has some advantages over a standalone solution, the most important of which is access to powerful pre-existing spatial analysis tools available within ArcGIS Server. The following sections outline a prototype client-server solution developed to provide users with the ability to access the maximum likelihood classification (MLC) tools present in ArcGIS for application to MODIS RRP subset images.

6.1 ArcGIS Explorer and ArcGIS Server in a Client-Server Architecture

ArcGIS Server is a server-based GIS software utility developed and marketed by ESRI. It provides a number of advertised key features, including data management, GIS Web services, Web mapping, some spatial editing, mobile GIS support, and server-based analysis and geoprocessing (ESRI, 2009c). With the addition of server-based geoprocessing, ArcGIS Server provides much of the same functionality available to users in ArcMap, but without the need to host the software locally.

Users can access data and services hosted by ArcGIS Server using a variety of clients, including ArcMap, web browsers, and mobile devices. In addition, ArcGIS Explorer was developed specifically as a client for ArcGIS Server, so it provides a lot of built-in functionality specifically for its use as a client interface.

GIS content and functionality, including maps, data, and geoprocessing tools, are created and published as services on ArcGIS Server. Clients then connect to these services and use them remotely. As such, a spatial analysis tool like the “Maximum Likelihood Classification” tool that is readily available in ArcGIS can be published as a geoprocessing service to ArcGIS Server. This is true of most existing ArcGIS spatial analysis tools, but there are some limitations that will be discussed below.

6.2 Maximum Likelihood Classification in ArcGIS

Prior to explaining how an ArcGIS Server geoprocessing service was created for use in classifying MODIS subsets, an explanation of the workflow used in conducting MLC in ArcGIS is necessary. As with any supervised classification scheme, MLC in ArcGIS requires the user to define image classes of interest by identifying pixels that represent the individual classes. Through empirical observations, the MLC tool in ArcGIS performs best when hundreds of pixels are used to define each class. This is different from the approach used in the development of the MODIS Data Classifier custom task which can perform well with only tens of pixels identified.

From a user perspective, identifying hundreds of pixels is most easily achieved by defining training polygons rather than points. These training polygons are then used in conjunction with the original image data, through application of the “Create Signatures”

tool, to calculate the statistics and covariance matrices required by MLC. The result of this tool is an ASCII file that contains all the necessary information about the user-defined image classes to accurately classify an image. This reveals another difference between the ArcGIS MLC tool and the MODIS Data Classifier—all classes present in an image must be identified by the user when applying the MLC tool, rather than just a singular image class.

With a signature file created, users then classify images with the MLC tool using the original image and the signature file as inputs. It is therefore clear that MLC in ArcGIS requires the application of a multi-step process to achieve classification results. Figure 6-1 illustrates the ArcGIS maximum likelihood classification workflow as viewed in the ArcGIS ModelBuilder interface.

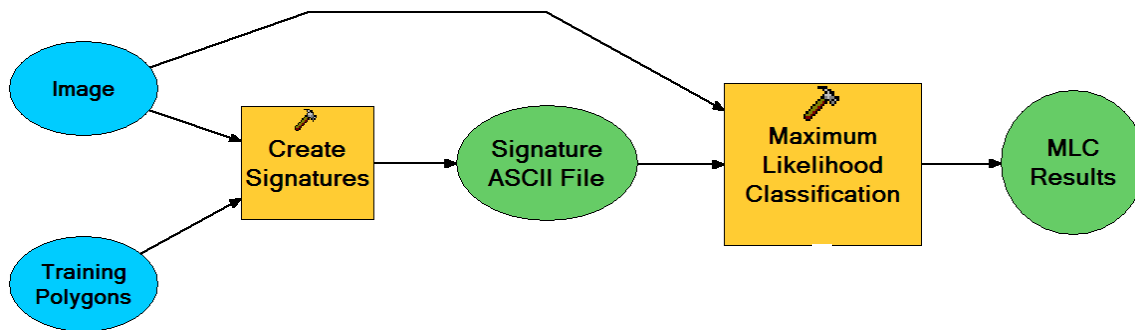


Figure 6-1. ArcGIS maximum likelihood classification workflow as viewed in ModelBuilder.

6.3 Development of a Maximum Likelihood Classification Geoprocessing Service

Despite the pre-existing spatial analysis tools available within the ArcGIS Server software, creating geoprocessing services that can be consumed by remote clients is not a trivial task. This is a fact that only becomes more obvious as the complexity of the desired service increases. Much of the difficulty stems from the need to pass user parameters and data from client to server, and subsequent server results back to the client. The difficulty is exacerbated by limitations on data formats permissible for input and output, and these formats vary depending on the client being used (ESRI, 2009d).

In addition to input/output difficulties, on the server side there is the additional complexity of creating a geoprocessing service that operates as expected. These services are usually developed from a model created in Model Builder. Publishing models that operate as desired can be a difficult and frustrating process.

With respect to the current project, training polygons and a MODIS subset image need to be provided from the client (AGX) to ArcGIS Server and, after execution, a raster output of the classification results need to be returned from server to client. Therefore, one of the first limitations encountered in attempting to apply a client-server architecture is the fact that AGX cannot input a feature class to a geoprocessing service. Input features such as training polygons are generally created as feature sets and passed to the server, but feature sets have several limitations relative to this project. Principally, feature sets cannot be saved and they must be manually given a text attribute (e.g., training class

description). Both of these characteristics make them less than ideal for image classification because, although not mandatory, the ability to save training polygons is a time-saving feature, and having to manually define training polygons would be cumbersome.

To negotiate this input roadblock, a new custom task was created for AGX that allows users to select an image class from a combo box dropdown menu and then draw training polygons on the AGX globe. Each image class text description is then programmatically assigned to a drawn polygon, removing the tedium of multiple manual entries. The task then allows the user to save the polygons as a .txt file, which is an allowable data format for input from AGX to ArcGIS Server per ESRI documentation (ESRI, 2009c). The custom task then programmatically formats the training polygon coordinates and image class descriptions in a manner that allows their re-creation on the server side using the pre-existing “Create Features from Text File” tool.

The second roadblock that was encountered is the fact that raster data (i.e., MODIS image subsets) cannot be used as a direct input from AGX to the server. This is therefore handled by passing the server the exact MODIS subset ID as a string that includes the subset name, date, band information, and passes information. A custom Python script on the server side then uses this text to programmatically create a URI and retrieve the image to the server’s scratch workspace for use in the service. Although retrieving the data on the server side is redundant, this is transparent to the user and is the only solution that could be determined given the input data limitations.

With input training polygons and the MODIS subset image available to the server, a model could be developed in ModelBuilder to execute the multi-step MLC process and subsequently published as the geoprocessing service (Figure 6-2). However, a multitude of attempts to create a model that executed properly when published as a service all failed. As well as can be determined, much of the failure is attributable to parameters that must be set for some of the tools in the ArcGIS MLC process, and attempts to hardcode these parameters into the model were ineffective. In the end, the best solution was to develop a custom Python script that includes hardcoded parameters for these tools, as well as code to create a training polygon feature class from the .txt file passed from AGX, retrieve the MODIS subset to be classified, create a signature file, and execute the maximum likelihood classification (Appendix B, code example 12).

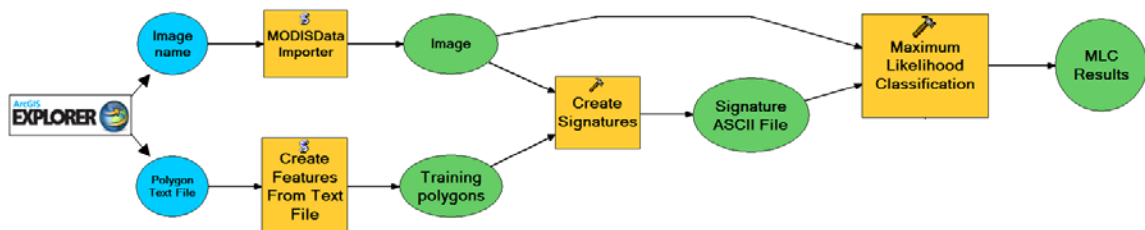


Figure 6-2. Custom geoprocessing workflow as viewed in ModelBuilder.

With inputs and the custom geoprocessing script working properly, the next roadblock encountered was an issue with permissible outputs that can be returned from ArcGIS Server to AGX. Although raster datasets are not allowed to be output directly to AGX, a workaround exists so that raster data can be added to a map service and then the updated map service returned to AGX. A map service is essentially a TIFF image of data

that can be updated and served by the server. As a result, in addition to the custom geoprocessing service, an associated map service must be created to display the MLC results. The AGX client therefore has to connect to both services in order to view any resulting classified datasets. It is important to note that although this allows the client to view the results, one major disadvantage of this workaround is that the client never has access to the resulting data, only the map service. Although not implemented in the current prototype, one potential workaround to enable client access to actual data would be to convert the MLC raster results to a feature class and return this data, since feature classes are an allowable output format for AGX. This raster to vector conversion could be added as a final step in the aforementioned custom Python script.

In their final form, the custom geoprocessing service and associated map service worked flawlessly when using ArcMap as a client. However, the services did not function at all in AGX, as was also the case after numerous and varied modifications were made to ArcGIS Server settings and the custom Python script. Finally, after consultation with the AGX development team at ESRI, it was revealed that there is a bug in the current version of AGX (build 500) such that text files cannot be used as input from AGX in geoprocessing services. As a result, the training polygon text file cannot be used as input and results in errors when the service is executed. This is contrary to ESRI's documentation (ESRI, 2009d), but in hindsight makes sense because the service produces errors immediately upon use since the required input files are never uploaded to ArcGIS Server. This bug is slated to be fixed prior to the release of the 900 build of AGX in summer, 2009.

6.4 Summary

In its current form, the framework under which geoprocessing services can be published and consumed from ArcGIS Server provides a fair amount of functionality for spatial analyses related to vector data. This is supported by the currently permissible data types allowed as input and output for these services. Although feature sets have to be used for input, feature classes are a valid output data format. On the other hand, raster data is supported as neither input nor output unless it is added to and returned in a result map service. This indicates that most raster analysis results from tools available through the Spatial Analyst extension would require conversion to vector format if the goal was to return actual data to the user. In addition to input and output data types, the experience gained in development of this client-server prototype suggests that any advanced spatial analysis, whether with raster or vector data, may require significant workarounds (i.e., additional custom tasks in AGX, custom Python scripting) to be developed in order to create the full functionality one may desire. With respect to the current project, until greater input and output support are added to ArcGIS Server, it appears that the standalone architecture is the most efficient and user-friendly framework for achieving the functionality desired by the client.

Chapter 7 – Results and Analysis

Analysis of the solutions for this project has been divided into three sections. The first section focuses on an analysis of the two different architectures developed for the project—standalone and client-server. Since the standalone “MODIS Toolkit for ArcGIS Explorer” prevailed as the superior solution, the second section of this chapter focuses on a performance analysis of the toolkit’s MODIS Data Classifier custom task. The analysis is concerned specifically with how well the custom task functions in deriving environmental spatial data, and what the optimal settings and best practices are in applying the custom task. To conclude the analysis of the MODIS Toolkit, the final section of the chapter provides a simple case study illustrating how the toolkit could be applied to a real-world environmental disaster.

7.1 System Architecture

The development of a toolkit to increase access to and exploitation of MODIS imagery proved successful at the conclusion of the project and yielded two solutions—one based on a standalone architecture and one on a client-server architecture. Each of these architectures has strengths and weaknesses. These solutions also revealed that ArcGIS Explorer is a valuable software utility in which to develop custom GIS solutions with sophisticated functionality.

7.1.1 Analysis of the Client-Server Solution

The originally proposed project solution included the use of ArcGIS Explorer as a client to access custom geoprocessing services developed for ArcGIS Server. This solution provides a number of advantages, the most important of which is access to pre-existing spatial analysis tools within the ArcGIS framework that can be leveraged in creating custom tools. In addition, this architecture allows for transparency to the user of software upgrades. It is possible to design this architecture such that all software changes can be made on the server side. Despite these benefits, this architecture proved not to be an ideal solution for this project, primarily due to existing limitations in the custom geoprocessing service framework. These limitations require significant workarounds to be designed and developed in order to create more sophisticated spatial tools, particularly those designed for raster data analysis. Future versions of ArcGIS Server may reduce this disadvantage, but for now it is significant enough to warrant serious consideration prior to developing a client-server solution.

In addition to issues with the geoprocessing framework, from the perspective of this project the client-server solution is also limiting because it requires Internet connectivity at all times for functionality. The standalone solution needs connectivity to initially download the AGX software and the custom tasks, as well as for initial data access. But once data has been retrieved locally, connectivity is non-essential for the standalone framework. This can be an important characteristic in deployed and disaster-response environments where continued connectivity may be an issue.

7.1.2 Analysis of the Standalone ArcGIS Explorer Solution

For the purposes of this project, the development of a set of custom tasks in AGX as a standalone toolkit for MODIS imagery access and exploitation proved to be the more valuable of the two possible paths to a solution. This is primarily due to AGX's availability at no cost via the Internet, the complete independence of AGX and its custom tasks, AGX's well-developed API and SDK supported by .NET, the ability to share custom tasks amongst many users, and AGX's simple, user-friendly interface and built-in access to base data. Over the course of the project, it became clear that development of sophisticated custom GIS functionality in AGX is really only limited by a developer's ability to leverage the AGX API and .NET, which together provide a wealth of programming classes for creation of custom GIS functionality. The combination of these factors enabled the final solution to meet and exceed all client-requested functionality.

The standalone architecture's only real limitation with regard to this project is its inability to be updated in a manner that is transparent to users. Any changes or upgrades to the custom tasks will require users to download new versions of the tasks.

7.2 Performance Assessment of the MODIS Data Classifier

As discussed in Chapter 5, the MODIS Data Classifier (MDC) custom task was developed specifically to enable classification of image subsets produced by the MODIS RRP. As such, the custom task provides users with the ability to develop environmental spatial data from MODIS RRP subsets that are otherwise simply images of the Earth. Despite the implicit utility of the MDC, data resulting from its use are only beneficial if they actually represent the classes of data captured by the original MODIS images (e.g., water, snow, vegetation, etc.). The following sections summarize a performance assessment conducted on data derived from the MDC in an effort to determine its overall utility in deriving environmental spatial data. As part of this assessment, the two primary parameters available in the MDC that affect results—the number and variability of training points and the range setting—were analyzed to try to determine their optimization.

7.2.1 Data

To initiate the study, one RRP subset was selected to meet the following criteria:

- The subset needed to contain a number of unique image classes that included as many of the broad image classes as possible that are regularly seen in MODIS images (e.g., water, snow, vegetation, bare ground).
- The subset needed to cover a large geographic area to be representative of most of the image subsets produced by the RRP.
- The subset needed to have limited cloud cover since clouds can be difficult to classify and can obscure the ground features that most users are interested in classifying.

The “AERONET_Fresno.2009089.terra.250m” subset, collected March 30, 2009, over southern California, met these criteria and the subset’s true color and false color images were used as the basis for this study. The false color image is illustrated in Figure 7-1.

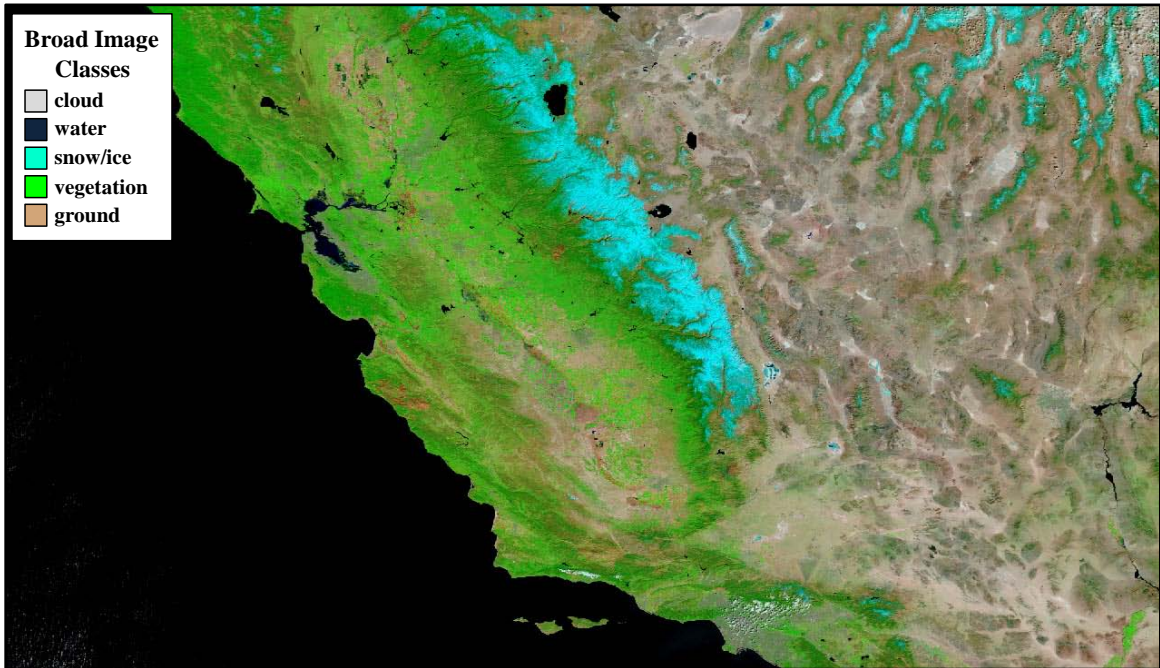


Figure 7-1. MODIS RRP false color subset selected for the performance analysis, with descriptions of the broad image classes present.

Both images are composed of three bands, but have one band in common—MODIS band 1. As a result, together both images represent five unique MODIS bands that include bands 1, 2, 3, 4, and 7, which provide data for the visible portion of the electromagnetic spectrum, as well as some coverage within the shortwave infrared (SWIR) portion of the spectrum.

With the primary experimental image data selected, the next step was to develop a reference classified data set against which to compare experimental results. This reference served to represent the “true” classes present in the image and provided a yardstick for assessing results from the MDC. For the present study, image classification results derived from commercially available software (e.g., ESRI ArcInfo or ERDAS Imagine) were deemed to be an appropriate reference. The reference classified data set was created using the maximum likelihood classification algorithm because this technique is one of the most widely used supervised classification methods (Jensen, 2005). The reference data were derived using the “Maximum Likelihood Classification” analysis tool available in ArcGIS (Figure 7-2).

7.2.2 Image Classification Parameters in the MODIS Data Classifier Custom Task

With regard to the parallelepiped classification scheme use in the MDC, there are two primary ways that the envelope of pixel inclusion can be altered: adjusting the mean and standard deviation for that class, and adjusting the number of standard deviations used to create the envelope. While the mean and standard deviation for a class may be adjusted

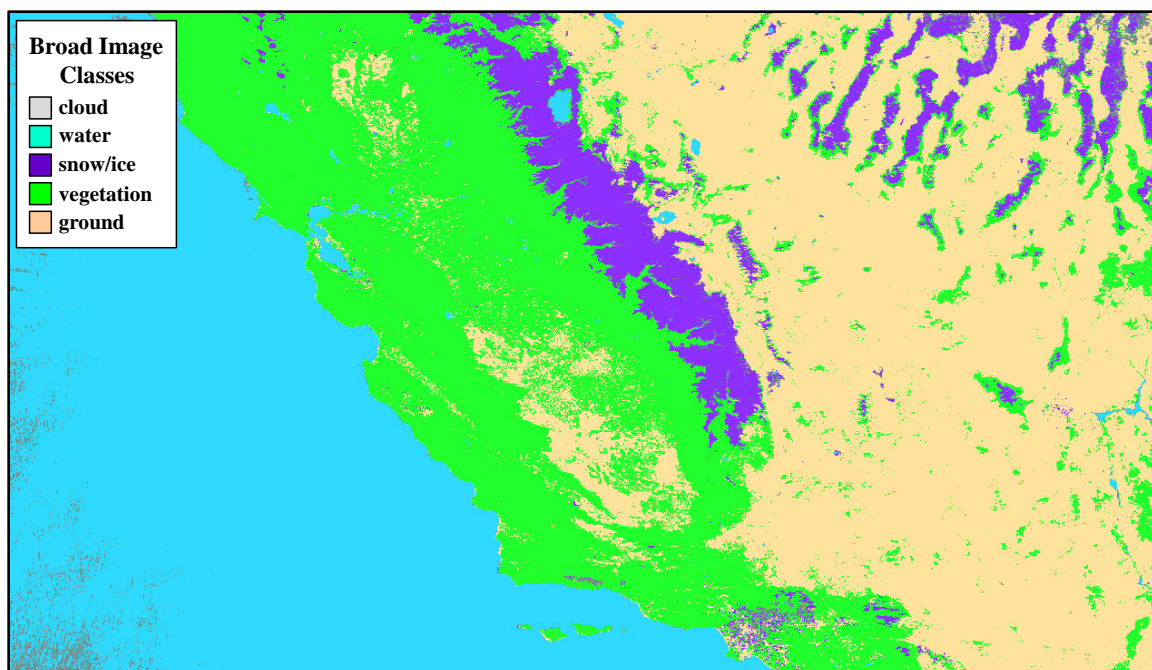


Figure 7-2. Results of classifying the “AERONET_Fresno.2009089.terra.250m” image subset using the ArcGIS “Maximum Likelihood Classification” tool to derive reference data.

indirectly through varying the number and variability of training points used to define the class (i.e., training pixels), the number of standard deviations used in the classification can be modified directly in the MDC using the adjustable “range” trackbar control in the custom task user interface (see Figure 5-11). This trackbar allows the user to adjust the number of standard deviations, from one to four, used to define the envelope of inclusion. The readout below the trackbar provides a unit-less measure from 0 to 100. Table 3 summarizes the basic allowable standard deviations and their associated trackbar settings; fractional standard deviations can be applied using intermediate settings.

Table 3. Standard deviation settings for the MODIS Data Classifier custom task and associated trackbar setting values.

Number of Standard Deviations	Trackbar Setting
1	0
2	33
3	66
4	100

The performance analyses that follow are specifically focused on how classification results are influenced by adjusting the range and the number and variability of training points, as well as how results of the MDC custom task compare to results from the ArcGIS maximum likelihood classification tool.

To be clear, the intent of this study was not to determine the accuracy of the MDC results versus reality, but rather to compare the MDC results to those derived using commercially available tools. The purpose of this was simply to ensure that the MDC

functions as designed and that there are no anomalous results which might be the product of an error programmed into the logic of the tool.

7.2.3 Performance Analysis

7.2.3.1 Reference Image Class

The MDC custom task was developed to classify only one class in a given MODIS RRP subset. The primary reason for this design is that the client for this project is interested in using the resulting custom task toolkit for identifying and monitoring environmental conditions related to natural disasters or conditions that may affect mobility (e.g., flooding, vegetation/ground cover, snow/ice, dust). Therefore, generally only one image class would be necessary to extract at any given time. In addition, designing the MDC custom task to operate on only one class at a time made development of the task much more straightforward and also makes executing the task simple and fast from a computational standpoint.

Due to this focus on only one image class, it was necessary to select an appropriate class for the present study. Since the goal of the study was to analyze the performance of the MDC custom task, “water” was selected as the baseline image class to use for comparison because it appears to have the most class separation of all the broad image classes present, and also has the least variability with regard to spectral values (Figure 7-3).

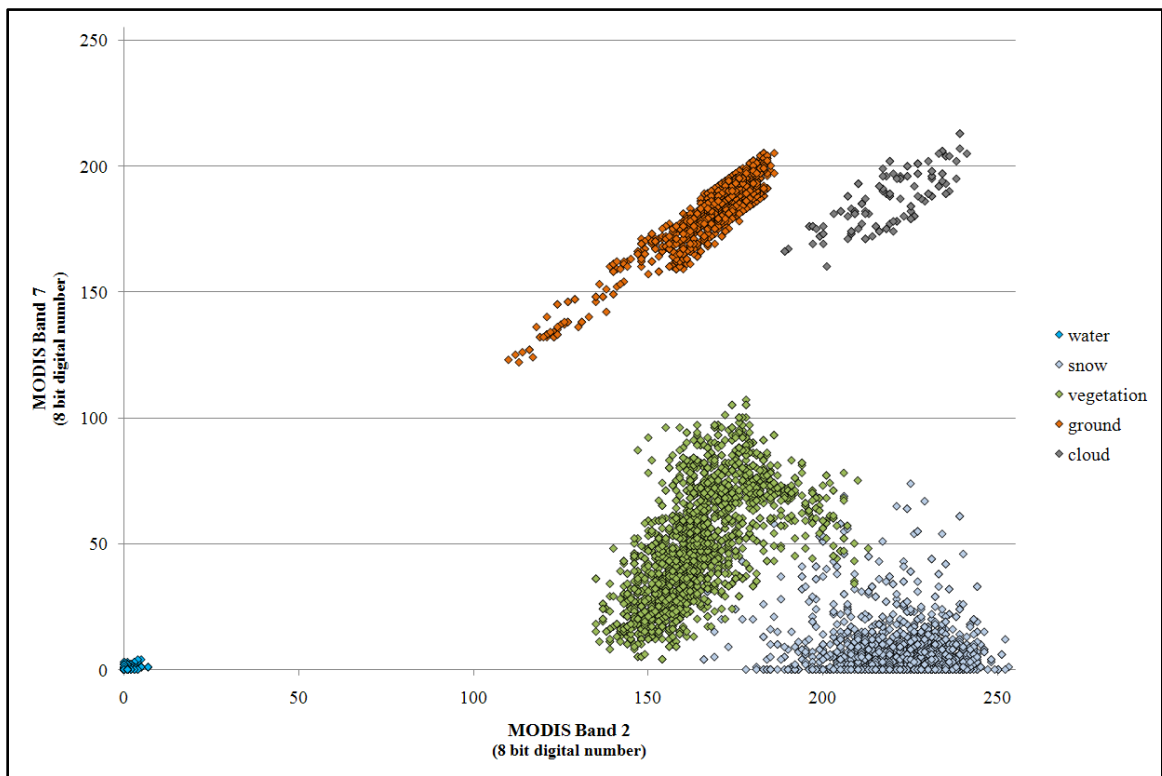


Figure 7-3. Two dimensional spectral distribution of image classes present in the “AERONET_Fresno.2009089.terra.250m” subset.

These characteristics provide the ability to assess the MDC's performance while limiting any variation or error that could result from using an image class that is in proximity to or overlapping the spectral space of another image class. All of the results that follow focus on the performance of the MDC custom task in classifying pixels that represent "water" versus pixels classified as "water" by the ArcGIS maximum likelihood classification tool.

7.2.3.2 Adjusting the Number of Standard Deviations Used

Methods

In an effort to assess the impact of changing the number of standard deviations used with the MDC custom task, 30 training points were selected that represent as much of the variability as possible for "water" pixels in the "AERONET_Fresno.2009089.terra.250m" subset. The subset was then classified using the MDC custom task to identify all "water" pixels using trackbar settings equivalent to 1, 1.5, 2, 2.5, 3, 3.5, and 4 standard deviations. The results of each of these classifications were then compared to "water" pixels from the reference classified data set.

Results

Figures 7-4 and 7-5 illustrate the overall results of the analysis. In Figure 7-4, the overall number of "water" pixels that were identified is plotted along with a line indicating the overall number of "water" pixels from the reference classified data set. Since results from the two different methods will not produce exactly the same results, Figure 7-5 illustrates the differences using graphs that plot the number of pixels for four different comparisons.

These graphs are intended to show differences in the numbers of pixels classified either as "water" or "not water" for both methods. The most noticeable characteristics of these results are that:

- the number of "water" pixels from the MDC custom task quickly exceed the number of "water" pixels in the reference data set
- while the number of "water" pixels increases rapidly between 1 and 2 standard deviations then more slowly from 2 to 4 standard deviations, the number of pixels reached a plateau of sorts between 2 and 2.5 standard deviations
- this same plateau is visible in the four comparisons of "water" and "not water" pixels
- there are a large number of pixels classified as "water" by the MDC custom task that were classified as "not water" in the reference data set
- there are also a large number of pixels classified as "water" in the reference data set that were classified as "not water" by the MDC task.

Although the results of the MDC custom task are similar to those from ArcGIS, increasing the number of standard deviations used beyond 1 resulted in an apparent over-prediction of "water" by the MDC task of anywhere from 11,450 to 44,746 pixels, or as much as 1.3% of those identified in the reference. It is clear from Figure 7-5 that nearly

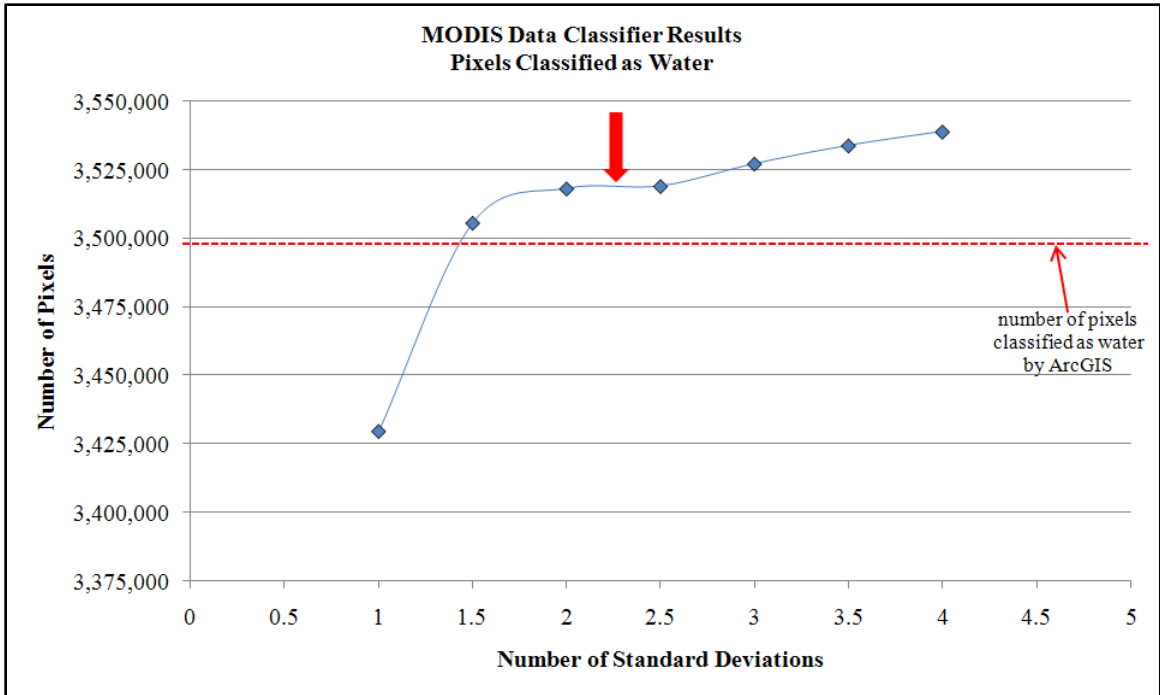


Figure 7-4. Number of pixels classified as “water” by the MDC using varying standard deviations. The red dashed line indicates the number of “water” pixels in the reference data set, and the red arrow indicates an apparent plateau between 2 and 2.5 standard deviations.

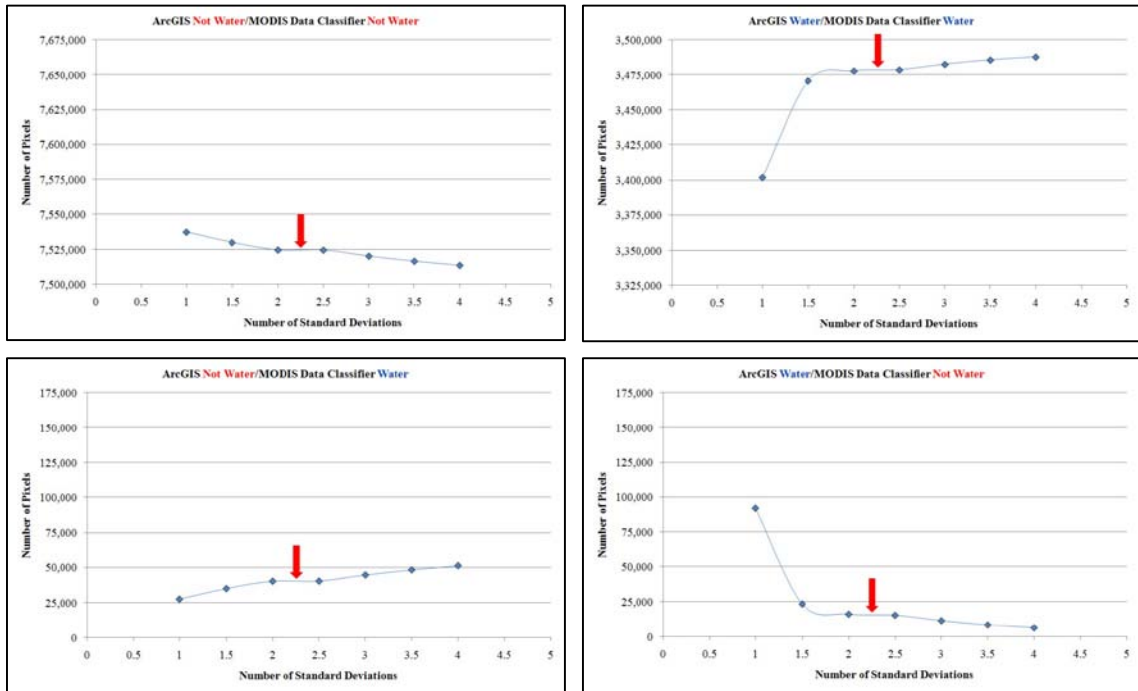


Figure 7-5. Four different comparisons showing the number of pixels that were classified by the MDC custom task and the reference data set as either “water” or “not water”.

all of this difference is explained by pixels classified by the MDC task as “water” that are “not water” in the reference.

To determine what other class or classes in the reference data set could explain the large differences between the two methods, pixels classified as “water” by the MDC task were compared with the four other broad image classes from the reference (i.e., cloud, snow/ice, vegetation, and ground). Figure 7-6 illustrates the results of these comparisons and reveals that virtually all of the difference is explained by pixels classified in the reference as “cloud” that the MDC task classified as “water”.

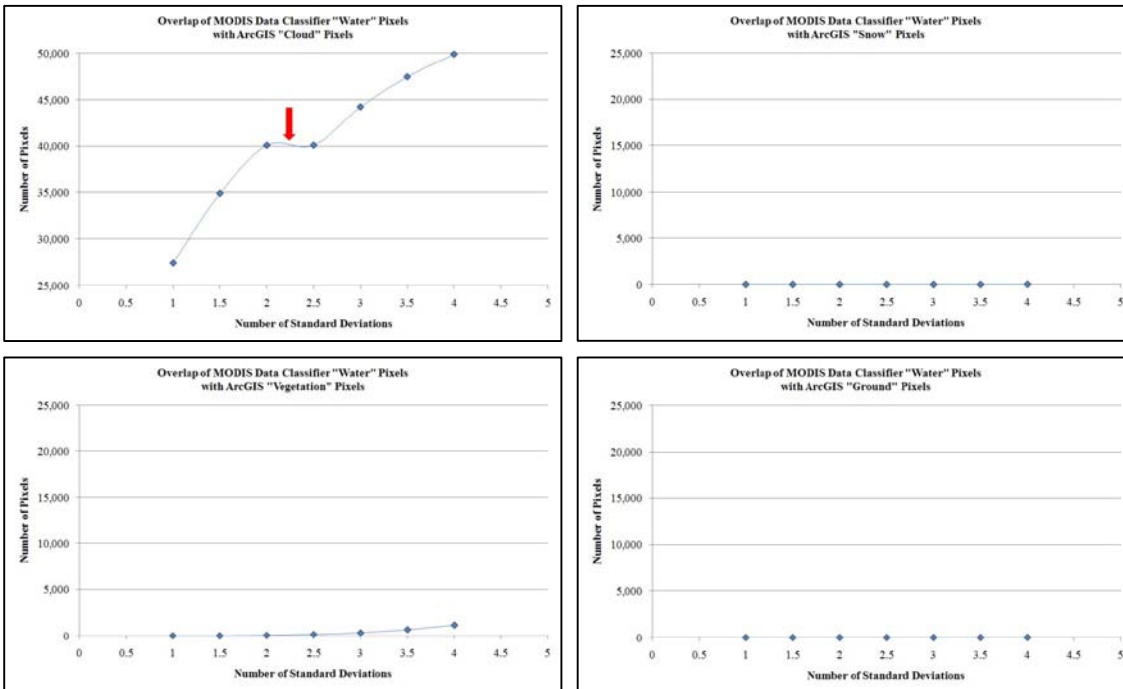


Figure 7-6. Overlap of “water” pixels identified by the MDC custom task with other image classes in the reference data set.

To evaluate this apparent misclassification, results of both methods were analyzed visually and in spectral space. One region of the original image containing cloud over water reveals that the MDC custom task did, in fact, classify many more pixels as “water” that are “not water” (i.e., cloud) in the reference data set (Figure 7-7). However, close visual inspection of these pixels indicates that classifying them as “water” is probably more accurate than classifying them as “cloud”. Analysis of a subset of these pixels in spectral space (MODIS bands 2 and 7) supports this conclusion and indicates that there is likely a significant amount of overlap between the “water” and “cloud” image classes (Figure 7-8). In fact, there appears to be a continuum of “cloud” values between the “cloud” and “water” end-members (see Figure 7-3 again for end-member locations in the same spectral space). This makes sense since pixels with decreasing density of clouds will take on more and more of the spectral characteristics of the ground features over which they exist. Therefore, a similar continuum is likely to exist between the “cloud” end-member and all other image class end-members.

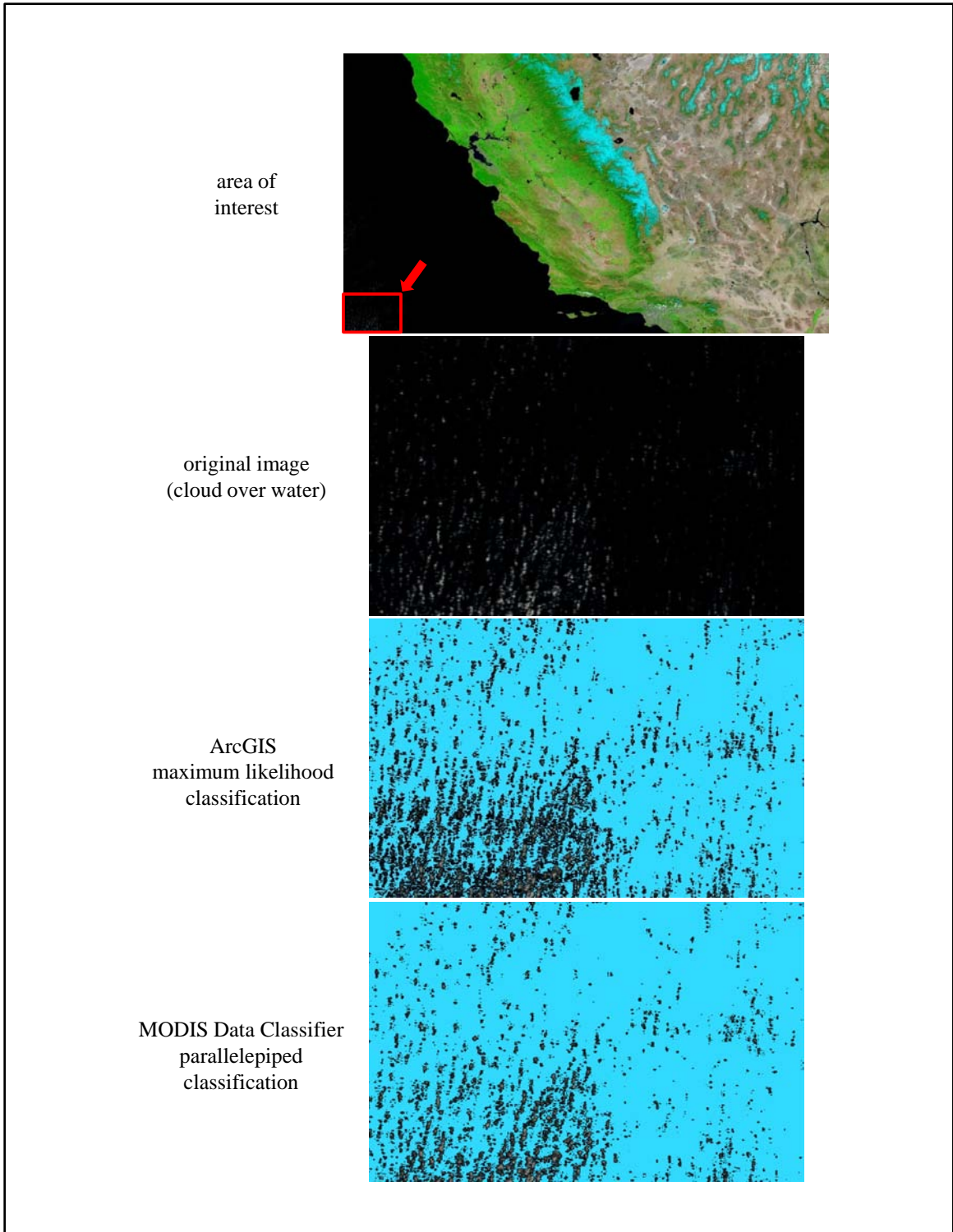


Figure 7-7. Visual comparison of results from the reference data set and the MDC custom task (using 2.5 standard deviations) for a region of cloud over water from the original image subset. Light blue indicates pixels classified as “water”.

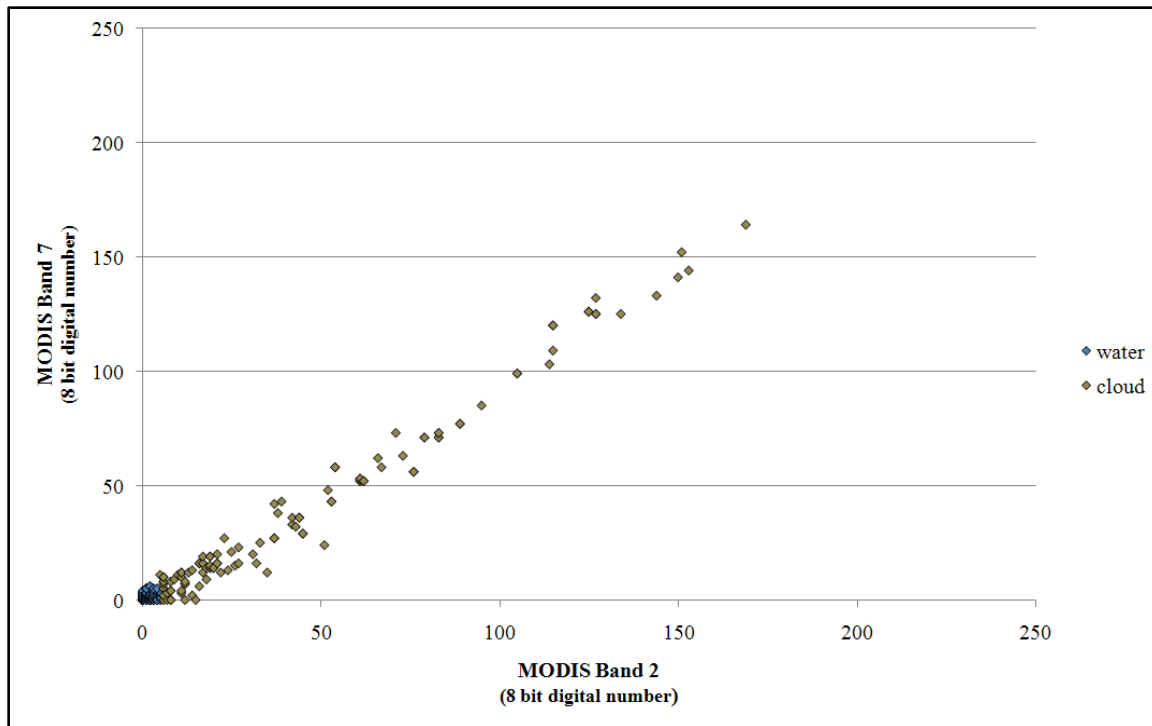


Figure 7-8. Spectral comparison of a subset of pixels from the area of interest in Figure 7-7. A clear continuum in spectral space exists between “water” and “cloud”.

Conclusions

The analysis of standard deviations has provided some useful conclusions that are very relevant to the use of the MDC custom task. Perhaps most evident is the suggestion that the task’s design to function on only one specific image class may actually provide better results than what can be derived from classification methods that yield all classes present in an image. It is not that the ArcGIS maximum likelihood classification tool is any less accurate, but rather that there is a competitive relationship that is in effect when pixels are being grouped into more than one class. To state this more clearly, when classifying an image using multiple training classes there is essentially a competition between classes for pixels. The MDC task, however, simply determines whether a pixel is inside a user-defined class envelope. It is not competing with any other class envelopes for pixels. This may give the user more flexibility in fine-tuning results for a given class.

The rapid rise in the number of classified pixels with the number of standard deviations to a plateau between 2 and 2.5 standard deviations suggests that this may be the optimal setting for classifying “water” pixels. However, per Figure 7-3 this setting will change depending upon the image class being classified and the location of all classes in spectral space. With respect to the current study, between 2 and 2.5 standard deviations, the number of pixels remained constant but steadily increased again between 2.5 and 4 standard deviations. The assumption is that increasing the number of standard deviations beyond 2.5 may increase accuracy but may also increase the number of false positives, or erroneous image class pixel assignments, due to the image class envelope overlapping other image classes, as illustrated in Figure 7-3. This is supported by visual

inspection of difficult to classify regions (e.g., transition between water and land) and regions of false positives in the original subset with increasing standard deviations (Figure 7-9). It is evident that more and more pixels in the difficult to classify transition zones are classified with increasing standard deviations, but it comes at the cost of including more and more erroneously classified pixels. The conclusion is that more pixels will be classified correctly using higher standard deviations, but there is also a higher likelihood of errors of commission (false positives), so more care must be taken in interpreting the results.

7.2.3.3 Varying the Number and Selectivity of Training Points

Methods

To assess the effect that increasing the number and selectivity of training points has on MDC classification results, an initial set of five training points was created and used to classify the image. These points were not selectively placed, rather just quickly added to areas of the image that contained water. Additional non-selective training points were added in increments of five, up to 50 total training points, and the image was reclassified after adding each new increment. The resulting classification results were then compared for overlap with the reference data set of pixels classified as “water.” Since the experimental classifications were compared for overlap with the reference, the greatest number of pixels that could be classified was that which was contained within the reference. This provided a baseline against which to compare experimental results.

For comparison, following the classifications using non-selective training points, two classifications were conducted using sets of 10 and 25 selectively placed points that were added with the purpose of trying to characterize as much variation as possible in the “water” image class (i.e., selective placement).

Results

Results of increasing the number and variability of training points are summarized in Figure 7-10. It is clear and probably not too surprising that increases in the number of training points are, in general, positively correlated with the number of pixels that are classified. The small variation in pixel count evident between increments is also to be expected, since the addition of new training points will modify the effective class mean and standard deviation used in the parallelepiped classification.

It is also clear from Figure 7-10 that selectively placed training points yield more accurate classification results, and good initial results were achieved with as few as 10 training points. In fact, 10 selectively placed points yielded better results than 40 non-selectively placed points, and 25 selectively placed points were able to account for almost the same number of classified pixels as 50 non-selectively placed points. Again, this is not surprising, given that selectively placed points have a greater likelihood of capturing more of the overall variability of the class, creating a more accurate envelope of inclusion for an image class.

The fact that the MDC custom task did not yield results that overlapped all or nearly all of the same pixels as those derived for the reference data set is expected; application of different classification methods and training points will result in different class

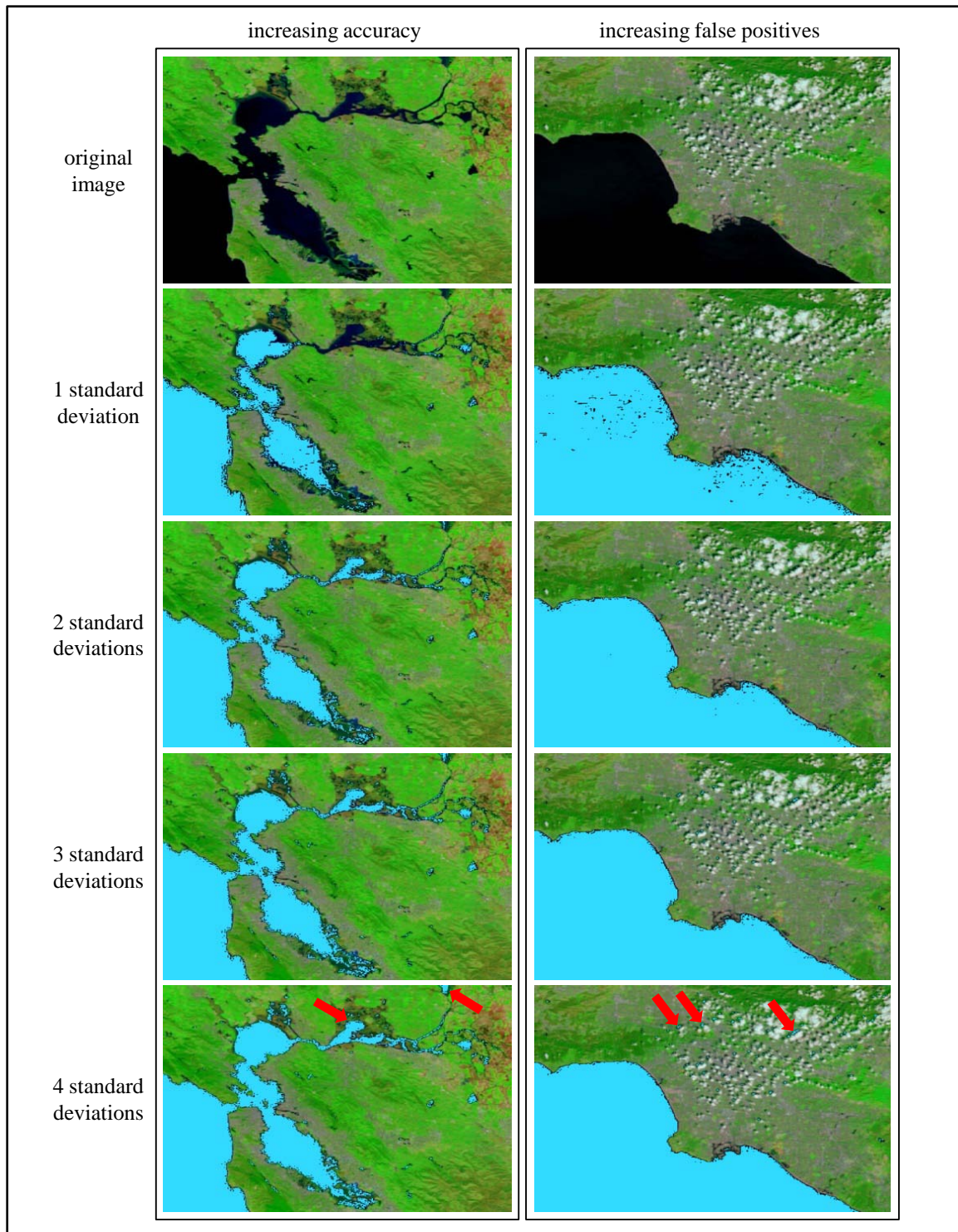


Figure 7-9. Increasing accuracy and false positives with increasing standard deviations. Pixels classified as water are light blue. Red arrows indicate greater accuracy for some pixels (left column, transition from water to land) and greater error for others (right column, shadows).

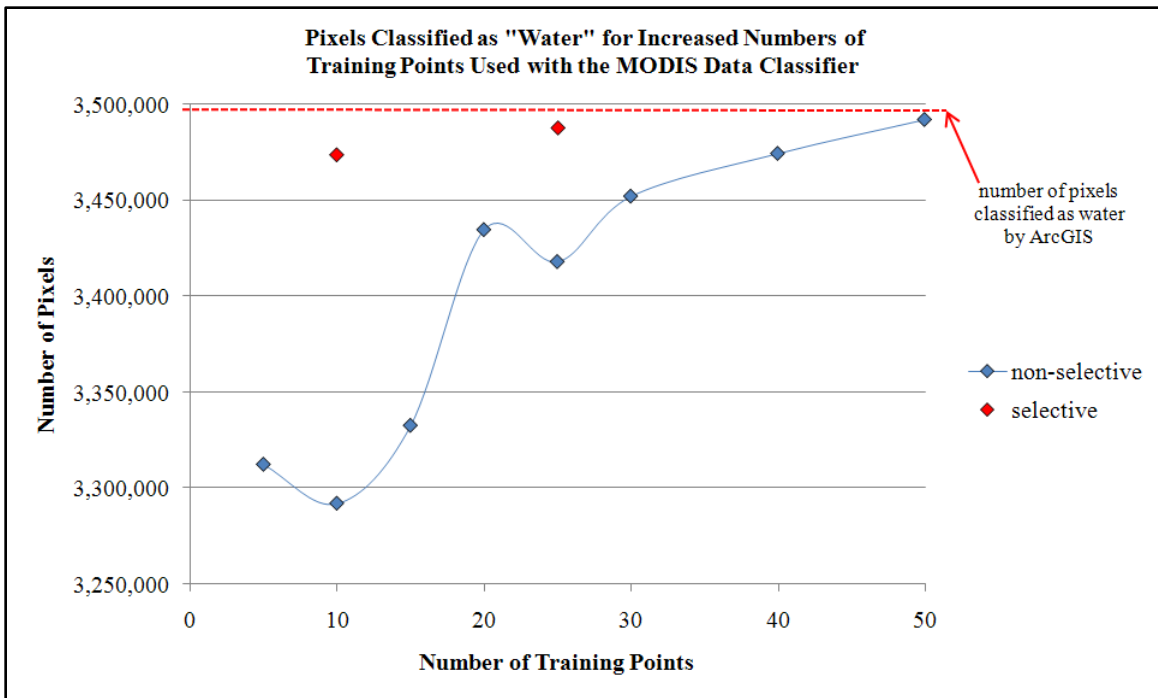


Figure 7-10. Classification results using increasing numbers of non-selectively placed training points, with two sets of selectively placed training points for comparison.

statistics. Close inspection of the various classification results reveals that all of the non-overlapping pixels resulted from small bodies of water with a unique spectral signature. The addition of one or two more selectively placed training points to these bodies of water would remove this discrepancy.

Conclusions

The training point analysis yielded results that provide important considerations for applying the MDC custom task. It is clear, albeit not surprising, that adding training points that capture most or all of the variations in an image class renders the most faithful classification results. What is surprising, however, is that very accurate results can be accomplished with relatively few (i.e., 10 to 25) training points. This is an informative result since the MDC custom task only provides users with the ability to add points and not polygonal training areas.

7.2.4 Performance Analysis Summary

Performance analyses conducted on the MODIS Data Classifier custom task for ArcGIS Explorer provided valuable information for application of the task in deriving environmental spatial data. Although the study is preliminary, based on a single class, the analyses reveal that the custom task can yield classification results comparable to results obtained through commercial software packages like ESRI's ArcGIS and ERDAS Imagine. However, the analyses also reveal that even better classifications may result through careful placement of image class training points, using at least 10 training points.

Further accuracy in classification results may be attainable through carefully increasing the applied standard deviation to four standard deviations, as long as one is cognizant that larger standard deviations can also yield more erroneous pixel classifications. In hindsight, it also appears that the design of the custom task to operate on only one image class at a time may also operate in the task's favor from the standpoint of overall classification accuracy using the parallelepiped method.

7.3 A Case Study: April/May 2009 Flooding of the Red River of the North

The Red River of the North originates along the southern border of North Dakota with Minnesota and forms much of the border between the two states. The river flows from south to north through the cities of Fargo and Grand Forks, North Dakota, eventually crossing the border into Canada, flowing through Winnipeg, Manitoba. At its terminus, the river empties into Lake Manitoba (Wikipedia, 2009a).

The river has a history of major flooding, with record floods occurring in the 1880s, 1950, 1997, and most recently in March and April, 2009 (Wikipedia, 2009b). As previously indicated, due to its high temporal resolution and large coverage area, the MODIS RRP provides a valuable source of data for monitoring large-scale environmental events like floods.

To provide an example of how the MODIS Toolkit for ArcGIS Explorer can aid in analyzing an event like the 2009 flood of the Red River of the North, three MODIS subset images covering the river were classified and compared for change. The first subset was from May, 2008, and was selected to serve as a reference for normal river levels. Two more subsets from 2009, one from April 9 and the other from May 3, were then analyzed and compared to reveal the 2009 flood extents and how the flood progressed from North Dakota northward into Manitoba. Figure 7-11 illustrates the results, with the left-hand image showing normal conditions; the center image showing change between the normal conditions of May, 2008, and the flood conditions of April 9, 2009; and the right-hand image revealing changes in flood extent between April 9 and May 3, 2009.

The left-hand image shows normal conditions as of May, 2008, with red pixels indicating the presence of water. Note that the Red River of the North is normally so narrow that it is not even detected by the MODIS sensor. The center image shows the extent of floodwaters on April 9, with blue pixels indicating water that is not normally present. The right-hand image reveals how floodwaters moved northward between April 9 and May 3, from the Fargo and Grand Forks, North Dakota area into Manitoba. Red pixels indicate areas flooded on April 9 that were no longer flooded on May 3, blue pixels indicate areas not flooded on April 9 that were on May 3, and green pixels indicate flooded areas that did not change between April 9 and May 3.

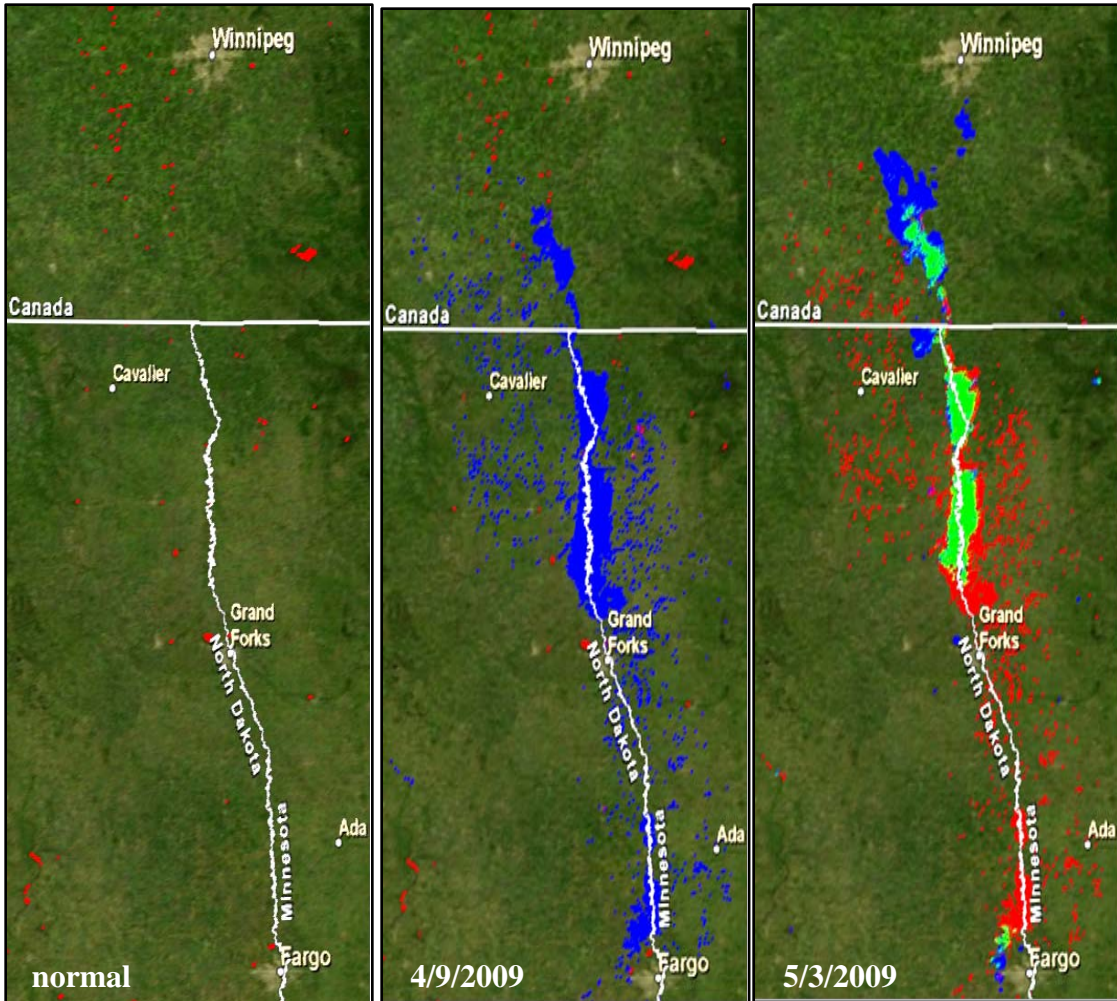


Figure 7-11. Comparison of flood extents for the Red River of the North, spring 2009, using results from the MODIS Data Classifier custom task.

7.4 Summary

The standalone “MODIS Toolkit for ArcGIS Explorer” is a solution that meets and exceeds all client-requested functionality for exploiting MODIS data. Furthermore, its availability at no cost via the Internet makes it an ideal, truly independent solution. The performance analysis results, coupled with the example real-world case study, reveal that the toolkit functions as designed and yields useful environmental spatial data that can be applied to military support and disaster response scenarios. Initial client feedback has been positive and, while not yet widely applied by the client, the toolkit was used to help complete a project for which MODIS RRP subsets were an ideal data source.

Chapter 8 – Conclusions and Future Work

8.1 Conclusions

At the outset of this project, the primary goals as defined by the client included the development of a set of tools in one user environment that would enable and improve user access to MODIS imagery, and the development of environmental spatial data from it. ArcGIS Explorer proved to be a very suitable environment in which to develop these tools and in many ways exceeded initial expectations. This is particularly true of the standalone capability of AGX, due to its custom task framework and well-developed API which allowed for the development of the MODIS Toolkit for ArcGIS Explorer. This toolkit includes all the functionality requested by the client, including:

- MODIS data query and retrieval
- image classification capability
- automated metadata creation
- data export capability

The toolkit also includes additional functionality not requested by the client, but which yields a more functional and user-friendly solution. This added functionality includes:

- thumbnail image queries to enhance data exploration
- training point import/export
- change detection capability

The only characteristic of this final solution that was not realized is the ability to upgrade software in a manner which is transparent to the user. However, this characteristic can really only be implemented in a client-server framework where all necessary software modifications are made on the server side, and was therefore exchanged for the characteristic of greater software independence provided by the standalone framework.

The development of a prototype client-server solution also yielded some important results related to the development of distributed GIS functionality for end-users without access to software licenses for commercial GIS software like ArcGIS. Specifically, the current framework for development of custom geoprocessing services is relatively limited, especially for more sophisticated GIS functionality and for certain data types, especially raster data. However, future improvements to ArcGIS Server may make this architecture a viable and more user-friendly solution that would provide access to an abundance of pre-existing spatial tools. Access to these tools is an important consideration. The standalone framework is ideal due to its independence, but this also requires custom development of any functionality that may be desired. This can be problematic in that development of custom tasks, whether simple or complicated, can be time consuming and requires the involvement of a programmer skilled in .NET and its use with the AGX API.

8.2 Future Work

There are a number of possible avenues for future work related to this project. As discussed in Chapter 3, the core data source for the MODIS toolkit is the JPEG image subsets created by the MODIS RRP. However, the RRP also creates HDF-EOS format images for the entire Earth whose radiometric accuracy remains intact. Extending the present tools to enable access to and exploitation of these image files would provide users access to MODIS data that could be used for more scientific endeavors.

In addition to MODIS data, another important data source that would increase the value of the toolkit is the archive of Landsat data now available at no cost through the USGS. Adding access to this archive would provide users with another source of environmental data—one with lower temporal resolution, but much greater spatial resolution (i.e., 30-90 meters). While MODIS data is a valuable source of environmental data, certain applications could benefit from an alternative, higher spatial resolution imagery source.

As should be apparent from preceding discussions, the MODIS toolkit was also developed to classify only MODIS JPEG images that were retrieved using the MODIS Data Retriever custom task. Another extension to this project that could add value would be to reprogram the tools to allow users to classify any images that were loaded into AGX, not just RRP JPEG subsets. This would give users greater flexibility in the data they could use and create a toolkit that might be of use to more groups.

In addition to developing tools to access different data sources, improving the existing parallelepiped classification algorithm could increase the toolkit's usefulness. More specifically, the existing algorithm could be modified to allow for mathematical rotation of the calculated parallelepipeds. This rotation would enable the envelopes defined by the parallelepipeds to adhere more closely to the data, resulting in greater accuracy. Furthermore, developing tools to enable different image classification algorithms could also be beneficial to users. These tools could include both supervised and unsupervised schemes to maximize user friendliness and usability.

Finally, the custom tasks discussed in this paper were all developed specifically for the 500 build of AGX. A completely revamped version of the software (build 900), complete with a totally new API, is currently scheduled for public release in summer 2009. Since the new API includes a set of completely new programming classes, there is no overlap between the AGX build 500 API and the AGX build 900 API. As a result, tasks developed in build 500 will have to be re-developed using the build 900 API in order to continue working in the new software. Upgrading this project's tasks using the new API and SDK would ensure their longevity and broader use.

Works Cited

- Aronoff, S. (2005). *Remote Sensing for GIS Managers*. Redlands, CA: ESRI Press.
- Belward, A.S., Stibig, H.J., Eva, H., Rembold, F., Bucha, T., Hartley, A., Beuchle, R., Khudhairi, D., Michielon, M., & Mollicone, D. (2007). Mapping severe damage to land cover following the 2004 Indian Ocean tsunami using moderate spatial resolution satellite imagery. *International Journal of Remote Sensing*, 28, 2977-2994. DOI: 10.1080/01431160601091803
- Brakenridge, R., & Anderson, E. (2006). MODIS-based flood detection, mapping and measurement: the potential for operational hydrological applications. In Marsalek, J., Stancalie, G., & Balint, G. (Eds.), *Transboundary Floods: Reducing Risks through Flood Management*. Amsterdam: Springer.
- Corbley, K.P. (2006). Rapid post-disaster mapping: Airborne remote sensing and Hurricane Katrina. *GIM International*, 20, 19-21.
- Cutter, S.L. (2003). GI Science, disasters, and emergency management. *Transactions in GIS*, 7, 439-445.
- Dartmouth Flood Observatory. (2002). *Potential Uses of Space-based Flood Mapping for Disaster Management in Asia and the Pacific* [PowerPoint slides]. Retrieved October 5, 2008 from http://www.unescap.org/icstd/SPACE/documents/RWDM_Bangkok/Acrobat/USA_DFO_Caquard.pdf
- DigitalGlobe. (2008). *Content Collection Systems*. Retrieved October 8, 2008, from <http://www.digitalglobe.com/index.php/82/Content+Collection+Systems>
- Doescher, S.W., Ristyb, R., & Sunne, R.H. Use of commercial remote sensing satellite data in support of emergency response. Proceedings. [International Society of Photogrammetry and Remote Sensing]. October 2005. Retrieved October 6, 2008, from http://www.commission4.isprs.org/workshop_hangzhou/papers/121-124%20S_Doescher-A079.pdf
- Domenikiotis, C., Loukas, A., & Dalezios, N.R. (2003). The use of NOAA/AVHRR satellite data for monitoring and assessment of forest fires and floods. *Natural Hazards and Earth System Science*, 3, 115-128.
- Duda, R.O., Hart, P.E., & Stork, D.G. (2001). *Pattern Classification*. New York: John Wiley & Sons.
- ESRI. (2008a). *ArcGIS Server: A Complete and Integrated Server GIS* [Brochure]. Retrieved October 10, 2008, from <http://www.esri.com/library/brochures/pdfs/arcgis-server.pdf>

- ESRI. (2008b). *ArcGIS Image Server: Fast, Dynamic Image Distribution and Processing* [Brochure]. Retrieved October 10, 2008, from <http://www.esri.com/library/brochures/pdfs/esri-image-server.pdf>
- ESRI. (2008c). *Explore Your World with ArcGIS Explorer* [Brochure]. Retrieved October 10, 2008, from <http://www.esri.com/library/fliers/pdfs/arcgis-explorer.pdf>
- ESRI. (2008d). *Architecture of a Custom Task*. Retrieved November 1, 2008, from <http://edndoc.esri.com/arcobjects/9.2/Explorer/GettingStartedArchitecture.html>
- ESRI. (2009a). *ArcGIS Explorer 500 System Requirements*. Retrieved January 29, 2009, from <http://wikis.esri.com/wiki/display/ag93bsr/ArcGIS+Explorer?>
- ESRI. (2009b). *Class Diagram of a custom task*. Retrieved June 1, 2009, from <http://edndoc.esri.com/arcobjects/9.2/Explorer/CustomTaskClassDiagram.pdf>
- ESRI. (2009c). *ArcGIS Server: Key Features*. Retrieved May 14, 2009, from http://www.esri.com/software/arcgis/arcgisserver/key_features.html
- ESRI. (2009d). *Geoprocessing with ArcGIS Server: Input and Output Data Types*. Retrieved May 14, 2009, from http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=Input_and_output_data_types
- Fire Information for Resource Management System. (2008). *Home*. Retrieved October 5, 2008, from <http://maps.geog.umd.edu/firms/default.asp>
- GeoEye. (2008). *Imagery Sources*. Retrieved October 8, 2008, from <http://www.geoeye.com/CorpSite/products/imagery-sources/Default.aspx>
- Gillespie, T., Chu, J., Frankenberg, E., & Thomas, D. (2007). *Assessment and prediction of natural hazards from satellite imagery* (Report No. CCPR-030-07). Retrieved October 8, 2008 from California Center for Population Research: http://www.ccpr.ucla.edu/ccprwpseries/ccpr_030_07.pdf.
- Jensen, J.R. (2005). *Introductory Digital Image Processing*. Upper Saddle River, NJ: Pearson Prentice Hall.
- Kevany, M.J. (2003). GIS in the World Trade Center attack—trial by fire. *Computers, Environment and Urban Systems*, 27, 571-583. DOI: 10.1016/S0198-9715(03)00015-2
- Kienberger, S., & Tiede, D. (2008). ArcGIS Explorer Review [Electronic version]. *GeoInformatics*. Retrieved October 10, 2008,

from <http://www.esri.com/library/reprints/pdfs/geoinformatics-arcgis-explorerer.pdf>

Lam, K.W.K, Lau, W.L., & Li, Z.L. (1999). Effects of JPEG compression on accuracy of image classification. *Proceedings* [Asian Conference on Remote Sensing]. November 1999.

Lau, W.-L., Li, Z.-L., & Lam, W.-K. (2003). Effects of JPEG compression on image classification. *International Journal of Remote Sensing*, 24, 1535-1544. DOI: 10.1080/01431160210142842

MODIS Rapid Response Project. (2004). *The MODIS Rapid Response Project: Near-Real-Time Processing for Fire Monitoring and Other Applications* [PowerPoint slides]. Retrieved October 5, 2008
from http://modis.gsfc.nasa.gov/sci_team/meetings/200407/presentations/land/Descloitres.pdf

MODIS Rapid Response System. (2008a). *Near-Real-Time Level-2 Browse*. Retrieved October 5, 2008 from <http://rapidfire.sci.gsfc.nasa.gov/realtime/2008279/>

MODIS Rapid Response System. (2008b). *Subsets*. Retrieved October 12, 2008
from <http://rapidfire.sci.gsfc.nasa.gov/subsets/>

Narumalani, S., Hlady, J.T., & Jensen, J.R. (2002). Information Extraction from Remotely Sensed Data. In *Manual of Geospatial Science and Technology*. New York: Taylor and Francis.

National Aeronautics and Space Administration. (2008a). *About MODIS*. Retrieved September 19, 2008, from <http://modis.gsfc.nasa.gov/about/>

National Aeronautics and Space Administration. (2008b). *Specifications*. Retrieved October 5, 2008, from <http://modis.gsfc.nasa.gov/about/specifications.php>

National Aeronautics and Space Administration. (2008c). *LAADS Web, Level 2 Browser*. Retrieved October 5, 2008,
from http://ladsweb.nascom.nasa.gov/browse_images/l2_browser.html

National Aeronautics and Space Administration. (2008d). *Data*. Retrieved October 5, 2008, from <http://modis.gsfc.nasa.gov/data/>

National Aeronautics and Space Administration. (2008e). *Data Products*. Retrieved October 5, 2008, from <http://modis.gsfc.nasa.gov/data/dataproduct/index.php>

National Aeronautics and Space Administration, Goddard Flight Center. (2003). *MODIS Rapid Response Project* [Fact sheet]. Retrieved October 5, 2008
from http://eosps0.gsfc.nasa.gov/ftp_docs/MODIS_Rapid_Response.pdf

- National Geospatial-Intelligence Agency. (2008a). *About*. Retrieved September 19, 2008, from <http://www1.nga.mil/About/Pages/default.aspx>
- National Geospatial-Intelligence Agency. (2008b). *GEOINT Analysis*. Retrieved September 19, 2008, from <http://www1.nga.mil/ProductsServices/GeointAnalysis/Pages/default.aspx>
- Ostir, K., Veljanovski, T., Podobnikar, T., & Stancic, Z. (2003). Application of satellite remote sensing in natural hazard management: The Mount Mangart landslide case study. *International Journal of Remote Sensing*, 24, 3983-4002. DOI: 10.1080/0143116031000103826
- Sirikulchayanon, P., Oyana, T., & Sun, W. (2008). Assessing the impact of the 2004 tsunami on mangroves using remote sensing and GIS techniques. *International Journal of Remote Sensing*, 29, 3553-3576. DOI: 10.1080/01431160701646332
- Townshend, J.R.G., & Justice, C.O. (2002). Towards operational monitoring of terrestrial systems by moderate-resolution remote sensing. *Remote Sensing of Environment*, 83, 351-359. Retrieved October 12, 2008, from http://www.gpa.uq.edu.au/courses/geom/3001/literature/Townshend_2002.pdf
- Tralli, D.M., Blom, R.G., Zlotnicki, V., Donnellan, A., & Evans, D.L. (2005). Satellite remote sensing of earthquake, volcano, flood, landslide and coastal inundation hazards. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59, 185-198. DOI: 10.1016/j.isprsjprs.2005.02.002
- Tso, B., & Mather, P.M. (2001). *Classification Methods for Remotely Sensed Data*. London: Taylor & Francis.
- USDA Foreign Agricultural Service. (2007). *Global Agriculture Monitoring at USDA/FAS (2007 Update)* [PowerPoint slides]. Retrieved October 5, 2008 from http://www.earthobservations.com/documents/cop/ag_gams/usda_fas_glam.pdf
- U.S. Geological Survey. (2008). *Imagery for Everyone...Timeline Set to Release Entire USGS Landsat Archive at No Charge* [Press release]. Retrieved October 12, 2008, from http://landsat.usgs.gov/documents/USGS_Landsat_Imagery_Release.pdf
- U.S. Naval Academy. (2007). MICRODEM [Software]. Available from the U.S. Naval Academy: <http://www.usna.edu/Users/oceano/pguth/website/microdem.htm>
- Webb, E.L., Robinson, J.A., & Evangelista, M.A. (2004). Scanning, compression and land cover classification of astronaut-acquired orbital photographs. *International Journal of Remote Sensing*, 10, 653-667. DOI: 10.1080/0143116031000139881

Wikipedia. (2009a). *Red River of the North*. Retrieved May 14, 2009,
from http://en.wikipedia.org/wiki/Red_River_of_the_North

Wikipedia. (2009b). *2009 Red River Flood*. Retrieved May 14, 2009,
from http://en.wikipedia.org/wiki/2009_Red_River_Flood

Programming References

The following references are not cited in the written report. They are references used in developing code for the project and are therefore referenced as comments in the original programming code.

Jayman. (2006, August 16). Open a Folder Browse Dialog window [Msg 1]. Message posted to dreamincode.net: Code Snippets»VB.NET Source Code, archived at <http://www.dreamincode.net/code/snippet518.htm>

jo0ls. (2006, January 18). LockBits [Msg 5]. Message posted to vbforums.com: VBForums»Visual Basic»Visual Basic .NET, archived at <http://www.vbforums.com/showthread.php?t=382179>

Microsoft Developer Network Library. (2008). *Directory.CreateDirectory Method*. Retrieved November 7, 2008, from [http://msdn.microsoft.com/en-us/library/system.io.directory.createdirectory\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/system.io.directory.createdirectory(VS.71).aspx)

Microsoft Developer Network Library. (2008). *WebClient.DownloadFile Method*. Retrieved November 7, 2008, from [http://msdn.microsoft.com/en-us/library/system.net.webclient.downloadfile\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/system.net.webclient.downloadfile(VS.71).aspx)

Microsoft Developer Network Library. (2009). *Arrays in Visual Basic*. Retrieved January 14, 2009, from [http://msdn.microsoft.com/en-us/library/wak0wfyf\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/wak0wfyf(VS.80).aspx)

Powell, R.W. (n.d.). *Using the LockBits method to access image data*. Retrieved January 16, 2009, from <http://www.bobpowell.net/lockingbits.htm>

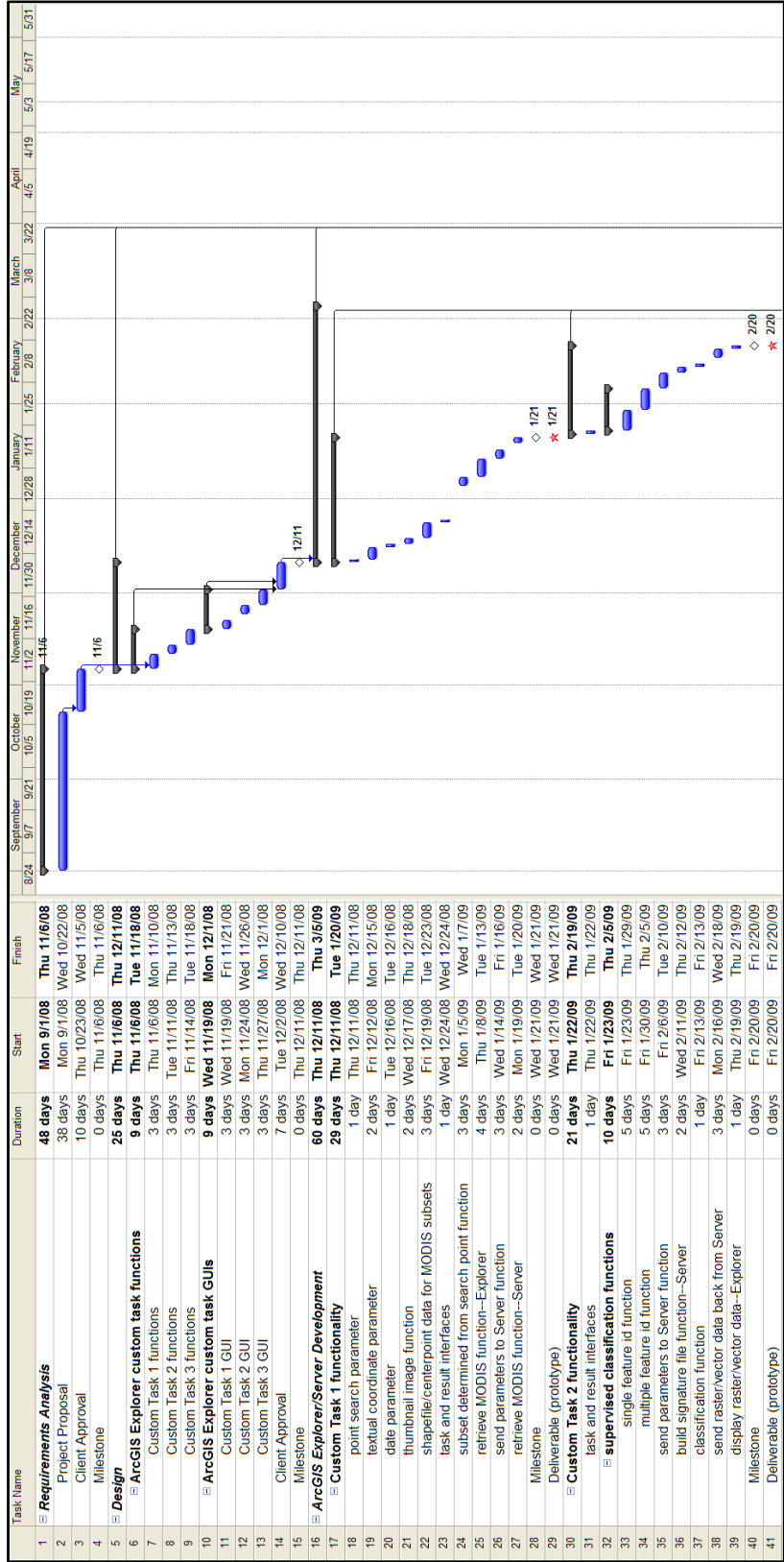
Salimindream. (2006, July 11). TextBox validation In Vb.Net [Msg 5]. Message posted to kdkeys.net: Data Mining»NET Framework, C#, Visual Basic .NET (VB.NET), Web Services»VB.NET, archived at <http://www.kdkeys.net/forums/thread/836.aspx>

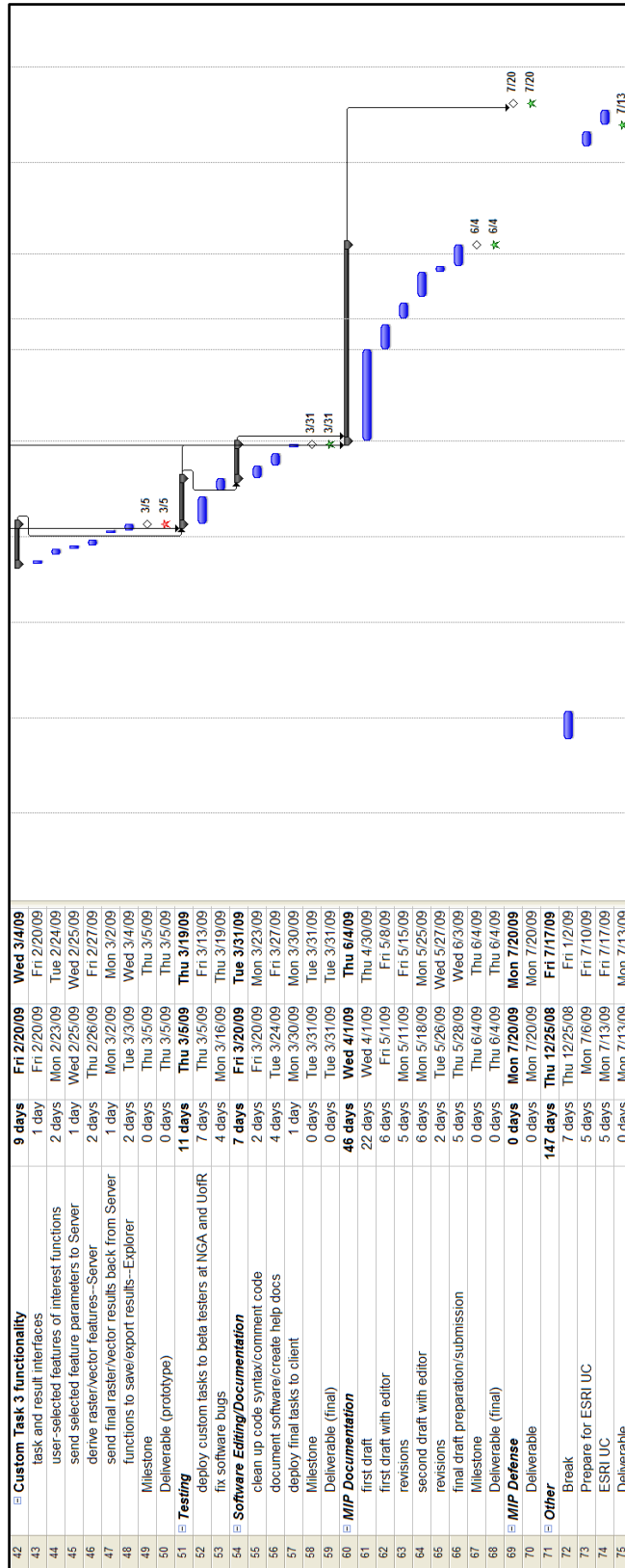
skizznott. (2007, June 5). pause/wait [Msg 7]. Message posted to a1vbcode.com: Home»Visual Basic (VB 4/5/6)»General Visual Basic, archived at <http://www.a1vbcode.com/a1vbcode/vbforums/Topic3320-3-1.aspx#top>

VisActualyBasic. (2008, April 19). DateTimePicker [Msg 1]. Message posted to DaniWeb.com: Home»Forums»Software Development»VB.NET, archived at <http://www.daniweb.com/forums/thread119908.html#>

Wagner, H.K. (n.d.). *Opening files, applications, Web documents, and the mail client*.
Retrieved January 16, 2009, from
<http://dotnet.mvps.org/dotnet/faqs/?id=openfileappwebpage&lang=en>

Appendix A. Original Project Workplan





Appendix B. Code Examples

Code Example 1. VB.NET code used to determine the most suitable subset for a given query point using the MODIS Data Retriever custom task.

```
' If layer is connected then determine closest subset to the query
point
If lvlMODISSubsets.IsConnected Then
    fsetSubsets = lvlMODISSubsets.QueryFeatures(point, "", _
        esriE2SpatialRelationship.Intersects)
    If fsetSubsets.FeatureCount > 0 Then
        intNumberOfSubsets = fsetSubsets.FeatureCount
        intNumberOfIterations = intNumberOfSubsets - 1
        ' i matches index values of the GetFeature method
        For i As Integer = 0 To intNumberOfIterations
            ' Convert user point and subset centerpoint lats
            and longs to radians
            subsetLongRadians = (0.0174532925) * _
                fsetSubsets.GetFeature(i).Shape.Envelope.Center.X
            subsetLatRadians = (0.0174532925) * _
                fsetSubsets.GetFeature(i).Shape.Envelope. _
                Center.Y.ToString
            pointLongRadians = (0.0174532925) * _
                point.Envelope.Center.X.ToString
            pointLatRadians = (0.0174532925) * _
                point.Envelope.Center.Y.ToString

            ' Great circle distance
            dblDistance = 6378 * (Acos((Cos(pointLatRadians) _
                * Cos(subsetLatRadians)) * _
                (Cos(pointLongRadians - subsetLongRadians)) _
                + ((Sin(pointLatRadians)) * _
                (Sin(subsetLatRadians))))))

            ' If this is the first iteration then set
            dblShortestDistance to the first distance result
            If i = 0 Then
                dblShortestDistance = dblDistance
                featSubset = fsetSubsets.GetFeature(0). _
                    GetValueAsString(2)
            End If
            ' If distance is shorter than dblShortestDistance,
            reset dblShortestDistance & use new subset name
            If dblDistance <= dblShortestDistance Then
                dblShortestDistance = dblDistance
                featSubset = fsetSubsets.GetFeature(i). _
                    GetValueAsString(2)
                ' used to set up geometry of correct subset for
                display
                subsetIndex = i
            End If
        Next
    End If
End If
```



```

        ' Set the geometry of the correct subset to be
        ' displayed with query point on globe
        _subsetGeometry = fsetSubsets.GetFeature _
            (subsetIndex).Shape
    Else
        featSubset = "No Subsets"
        ' Clears _subsetGeometry if previous queries were
        ' performed
        _subsetGeometry = Nothing
    End If
End If

```

Code Example 2. VB.NET code used to establish URI's and filenames from user-defined parameters for retrieving MODIS subset images and world files.

```

' String variables for URI's and local filenames; used to download with
' the WebClient class

' Note: the structure of the URI's below may have to be changed if
' MODIS Rapid Response changes the URL's used to archive data
Dim strImageURI As String = _str_MODIS_RRP_URL & _strMODISSubset & _
    "." & _strYear & _strDay & "." & _strPass & _strBands & _
    ".250m" & ".jpg"
Dim strWorldfileURI As String = _str_MODIS_RRP_URL & _
    _strMODISSubset & "." & _strYear & _strDay & "." & _
    _strPass & _strBands & ".250m" & ".jgw"
Dim strImageFileName As String = _strMODISSubset & "." & _
    _strYear & _strDay & "." & _strPass & _strBands & _
    ".250m" & ".jpg"
Dim strWorldfileName As String = _strMODISSubset & "." & _strYear _
    & _strDay & "." & _strPass & _strBands & ".250m" & ".jgw"

```

Code Example 3. VB.NET code used for retrieving MODIS subsets and world files with the WebClient class

```

' Set up webclients and download MODIS images and worldfiles
Dim webclientRetrieveMODIS As New WebClient()
Dim webclientRetrieveWorldfile As New WebClient
webclientRetrieveMODIS.DownloadFile(strNewImageURI, strNewImageFilename)
webclientRetrieveWorldfile.DownloadFile(strNewWorldfileURI, _
    strNewWorldfileFilename)

```

Note that the variable strNewImageURI holds the same value as strImageURI (etc. for the world file URI and filename variables) from Code Example 2. strImageURI is passed to the Task class to run asynchronously and is renamed to try to avoid confusion

Code Example 4. VB.NET code used to programmatically create spatial reference XML files for every image and thumbnail that is retrieved.

```

' Create the spatial reference for the images. All are WGS-84
Dim intFilenameLength As Integer = strImageFileName.Length
Dim strXMLFilename As String = strImageFileName.Remove_
    (intFilenameLength - 4, 4)

```

```

Dim strSpatialRefFilename As String = _strProgramFilesPath & _
    "\ArcGIS Explorer\MODIS Toolkit\MODISDataRetriever\ _
    MODIS Images\" & strXMLFilename & ".jpg.aux.xml"
Dim charQuote As Char = Chr(34)
Dim strSpatialRefText As String = "<PAMDataset>" & vbCrLf & _
    " <SRS>GEOGCS[" & charQuote & "GCS_WGS_1984" & charQuote & _
    ",DATUM[" & charQuote & "WGS_1984" & charQuote & ",SPHEROID[" & _
    charQuote & "WGS_1984" & charQuote & ",6378137.0," & _
    "298.257223563]],PRIMEM[" & charQuote & "Greenwich" & _
    charQuote & ",0.0],UNIT[" & charQuote & "Degree" & charQuote & _
    ",0.0174532925199433]]</SRS>" & vbCrLf & "</PAMDataset>"
File.WriteAllText(strSpatialRefFilename, strSpatialRefText)

```

Code Example 5. VB.NET code used to programmatically create metadata XML files for every image that is retrieved.

```

' Create a simple metadata file for the image.
Dim strMetadataFilename As String = _strProgramFilesPath & _
    "\ArcGIS Explorer\MODIS Toolkit\MODISDataRetriever\" & _
    "MODIS Images\" & strXMLFilename & ".jpg.xml"
Dim strMetadataText As String = "<?xml version=" & charQuote & "1.0" & _
    charQuote & "?>" & vbCrLf & "<!-- <!DOCTYPE metadata SYSTEM " & _
    charQuote & "http://www.esri.com/metadata/esriprof80.dtd" & _
    charQuote & "-->" & vbCrLf & "<metadata xml:lang=" & _
    charQuote & "en" & charQuote & ">" & vbCrLf & " <idinfo>" & _
    vbCrLf & " <descript>" & vbCrLf & " <langdata Sync=" & _
    charQuote & "TRUE" & charQuote & ">en</langdata>" & vbCrLf & _
    " <abstract>This is a MODIS subset image produced by the " & _
    "MODIS Rapid Response Project at NASA/GSFC and available at " & _
    "http://rapidfire.sci.gsfc.nasa.gov/subsets/" & "</abstract>" & _
    vbCrLf & " </descript>" & vbCrLf & " <citation>" & vbCrLf & _
    " <citeinfo>" & vbCrLf & " <origin>MODIS Rapid Response " & _
    "Project at NASA/GSFC</origin>" & vbCrLf & " <pubdate>" & _
    dtpDate.Value.Date & "</pubdate>" & vbCrLf & " <title Sync=" & _
    charQuote & "TRUE" & charQuote & ">" & strImageFileName & _
    "</title>" & vbCrLf & " <ftname Sync=" & charQuote & _
    "TRUE" & charQuote & ">" & strImageFileName & "</ftname>" & _
    vbCrLf & " <geoform Sync=" & charQuote & "TRUE" & charQuote & _
    ">raster digital data</geoform>" & vbCrLf & " </citeinfo>" & _
    vbCrLf & " </citation>" & vbCrLf & " <timeperd>" & vbCrLf & _
    " <timeinfo>" & vbCrLf & " <sngdate>" & vbCrLf & _
    " <caldate>" & dtpDate.Value.Date & "</caldate>" & vbCrLf & _
    " </sngdate>" & vbCrLf & " </timeinfo>" & vbCrLf & _
    " </timeperd>" & vbCrLf & " <keywords>" & vbCrLf & _
    " <theme>" & vbCrLf & " <themekey>MODIS Rapid Response & _
    Project, " & "NASA/GSFC, subset: " & _strMODISSubset & ", source" & _
    "file: " & strImageFileName & "</themekey>" & vbCrLf & _
    " </theme>" & vbCrLf & " </keywords>" & vbCrLf & _
    " </idinfo>" & vbCrLf & "</metadata>"
File.WriteAllText(strMetadataFilename, strMetadataText)

```

Code Example 6. VB.NET code used to ensure that only MODIS RRP subsets are able to be selected for classification via the MODIS Data Classifier.

```

' Setup vars for metadata (if it exists)
Dim strMetadata As String = currentLayer.Workspace & _
    currentLayer.Dataset & ".xml"
Dim isMetadata As Boolean = File.Exists(strMetadata)
Dim xmlDocument As XmlDocument = New XmlDocument
Dim strXMLAbstractkey As String
Dim strXMLThemekey As String
Dim isJPEG As Boolean = False
Dim isMODIS As Boolean = False

' If metadata exists, check it to see if the image is a
' jpeg and if it's a MODIS RRP image subset
If isMetadata = True Then
    xmlDocument.Load(strMetadata)
    strXMLAbstractkey = xmlDocument.SelectSingleNode _
        ("/metadata/idinfo/descript/abstract").InnerText
    strXMLThemekey = xmlDocument.SelectSingleNode _
        ("/metadata/idinfo/keywords/theme").InnerText
    isJPEG = Not strXMLAbstractkey.Contains("classified") And _
        Not strXMLAbstractkey.Contains("change")
    isMODIS = strXMLThemekey.Contains("MODIS Rapid Response Project")
End If

' If the layer is a raster dataset (i.e., an image) and it
' is a MODIS RRP subset, then add to the combobox
If currentLayer.Type = "Raster Layer" And isJPEG = True _
    And isMODIS = True Then
    cboSelectImage.Items.Add(currentLayer.DisplayName)
End If

```

Code Example 7. VB.NET code used by the MODIS Data Classifier to create an array of training point row-column coordinates. This array is used to develop band statistics for the training points (see Code Example 9)

```

ReDim _aryTrainingPoints(_intPointCount - 1, 1)

' Iterate through all training points and add
' row, column coordinates to array
For i As Integer = 0 To _intPointCount - 1
    ' Set up variables to hold training point coordinates
    Dim currentResult As ESRI.ArcGIS.E2API.Result = _
        _resultManager.GetTopLevelResult(0).GetChildResult(i)
    Dim dblPointLong As Double = currentResult.GetExtent.Center.X
    Dim dblPointLat As Double = currentResult.GetExtent.Center.Y
    ' Set up variables to determine column, row of the point
    Dim dblPixelSizeX = _aryWorldfileText(0)
    Dim dblPixelSizeY = _aryWorldfileText(3)
    ' get first pixel longitude
    Dim dblFirstPixelX As Double = _aryWorldfileText(4)
    ' get first pixel latitude
    Dim dblFirstPixelY As Double = _aryWorldfileText(5)
    ' Calculate column, row of the training point

```

```

Dim intPointColumn As Integer = ((dblPointLong - dblFirstPixelX)_
    / dblPixelSizeX) + 1
Dim intPointRow As Integer = ((dblPointLat - dblFirstPixelY) _
    / dblPixelSizeY) + 1
' Add to the training point array
_aryTrainingPoints(i, 0) = intPointRow
_aryTrainingPoints(i, 1) = intPointColumn

```

Next

Code Example 8. VB.NET code used to copy individual byte values from MODIS images into an image data array for further use in the MODIS Data Classifier custom task.

```

' Redimension image data array using height and width of selected image
' to set array extent. 5 layers for 5 bands.
ReDim _aryImageData(4, _imageOriginal1.Height - 1, _
    _imageOriginal1.Width - 1)

' Set up parameters for lockbits method
Dim imageRectangle As Rectangle = New Rectangle(0, 0, _
    _imageOriginal1.Width, _imageOriginal1.Height)
Dim pixelFormat As PixelFormat = pixelFormat.Format24bppRgb
' true color (bands 1-4-3)
Dim imageData1 As BitmapData = _imageOriginal1.LockBits _
    (imageRectangle, ImageLockMode.ReadWrite, pixelFormat)
' false color (bands 7-2-1)
Dim imageData2 As BitmapData = _imageOriginal2.LockBits _
    (imageRectangle, ImageLockMode.ReadWrite, pixelFormat)
Dim ptr1 As IntPtr = imageData1.Scan0 ' true color
Dim ptr2 As IntPtr = imageData2.Scan0 ' false color

' Determine total number of bytes in the image
Dim intNumBytes As Integer = _imageOriginal1.Width * _
    _imageOriginal1.Height * 3

Dim aryByteValues1(intNumBytes - 1) As Byte
Dim aryByteValues2(intNumBytes - 1) As Byte
Marshal.Copy(ptr1, aryByteValues1, 0, intNumBytes)
Marshal.Copy(ptr2, aryByteValues2, 0, intNumBytes)

' Since data was copied to aryByteValues 1 and 2, we can unlock
' bits of the original images
_imageOriginal1.UnlockBits(imageData1)
_imageOriginal2.UnlockBits(imageData2)

' Iterate through the image byte value arrays, adding band values
' to the appropriate layer in _aryImageData. Iteration is set up
' to add the bands sequentially (1-2-3-4-7).
Dim intByteIndex As Integer = 0
For j As Integer = 0 To _imageOriginal1.Height - 1
    For k As Integer = 0 To _imageOriginal1.Width - 1
        ' band 1
        _aryImageData(0, j, k) = aryByteValues1(intByteIndex + 2)
        ' band 2
        _aryImageData(1, j, k) = aryByteValues2(intByteIndex + 1)
        ' band 3
        _aryImageData(2, j, k) = aryByteValues1(intByteIndex)
    
```

```

        'band 4
        _aryImageData(3, j, k) = aryByteValues1(intByteIndex + 1)
        'band 7
        _aryImageData(4, j, k) = aryByteValues2(intByteIndex + 2)
        intByteIndex = intByteIndex + 3
    Next k
Next j

```

Code Example 9. VB.NET code used by the MODIS Data Classifier to iterate through the training point row-column coordinate array to calculate the mean training point value for each MODIS image band.

```

Private Function calculateMean(ByVal bandNumber As Integer) As Double

    Dim sum As Integer = 0
    Dim mean As Double
    Dim row As Integer
    Dim column As Integer

    ' Iterate through the training point row-column coordinate array,
    ' using the rows and columns to access image data
    For i As Integer = 0 To _intPointCount - 1
        row = _aryTrainingPoints(i, 0)
        column = _aryTrainingPoints(i, 1)
        If _isNDVI = False Then
            sum = sum + _aryImageData(bandNumber, row - 1, _
                                     column - 1)
        Else
            sum = sum + _aryNDVIData(bandNumber, row - 1, _
                                     column - 1)
        End If
    Next i

    mean = (sum / _intPointCount)
    Return mean
End Function

```

Note that this function calculates the mean for each band. Similar code is used to calculate the standard deviations for each band of the training point data.

Code Example 10. VB.NET code using nested For loops to test each element of the image data array for inclusion in the image class of interest. If array (subset pixel) values fall within the window of inclusion, their equivalent pixel in a new .png bitmap is set to red—otherwise they are set to transparent.

```

' Iterate through the image byte value arrays, testing each pixel
' to see if it meets the min-max envelope of each band as determined
' from the training point set.
Dim i As Integer = 0
For j As Integer = 0 To _imageOriginal1.Height - 1
    For k As Integer = 0 To _imageOriginal1.Width - 1
        If _isNDVI = False Then ' classify image data
            If (band1Min <= _aryImageData(0, j, k) _

```

```

And _aryImageData(0, j, k) <= band1Max) _
And (band2Min <= _aryImageData(1, j, k) _
And _aryImageData(1, j, k) <= band2Max) _
And (band3Min <= _aryImageData(2, j, k) _
And _aryImageData(2, j, k) <= band3Max) _
And (band4Min <= _aryImageData(3, j, k) _
And _aryImageData(3, j, k) <= band4Max) _
And (band7Min <= _aryImageData(4, j, k) _
And _aryImageData(4, j, k) <= band7Max) Then
    aryByteValues(i) = 0 ' blue channel
    aryByteValues(i + 1) = 0 ' green channel
    aryByteValues(i + 2) = 255 ' red channel--set
    aryByteValues(i + 3) = 255 ' Alpha channel
Else
    aryByteValues(i) = 0 ' blue channel
    aryByteValues(i + 1) = 0 ' green channel
    aryByteValues(i + 2) = 0 ' red-don't set
    aryByteValues(i + 3) = 0 ' Alpha-set transp.
End If
i = i + 4
Next k
Next j

```

Code Example 11. VB.NET code used to validate entries in the change detection user interface of the MODIS Data Classifier custom task.

```

Private Sub ValidateEntries()
    btnCompare.Enabled = False 'reset to disabled then check again
    '2 datasets selected
    If Not lblDataset1.Text = "" And Not lblDataset2.Text = "" Then
        'not the same dataset
        If Not lblDataset1.Text = lblDataset2.Text Then
            'same image subset for comparison
            If _strSubset1Name = _strSubset2Name Then
                'same image class
                If _strImageClass1 = _strImageClass2 Then
                    btnCompare.Enabled = True
                End If
            End If
        End If
    End If
End Sub

```

Code Example 12. Python script used on ArcGIS Server to conduct maximum likelihood classification of a MODIS subset image. Required inputs include a .txt file with training polygon coordinates and descriptions, and the MODIS subset ID as text.

```

import string, os, sys, locale, arcgisscripting, urllib, time,
traceback
gp = arcgisscripting.create(9.3)
gp.overwriteoutput = 1

def retrieveMODISImage(imageName):
    gp.AddMessage("\tRetrieving MODIS image....")

```

```

outputImage = os.path.join(gp.scratchworkspace, 'Image.jpg')
outputWorldfile = os.path.join(gp.scratchworkspace, 'Image.jgw')

# setup necessary variables
worldfileName = imageName[:-4] + ".jgw"
imageURL = 'http://rapidfire.sci.gsfc.nasa.gov/subsets/?subset=' +
           imageName
worldfileURL = 'http://rapidfire.sci.gsfc.nasa.gov/subsets/?subset='
              + worldfileName

# retrieve image data from NASA's MODIS Rapid Response web server
image = urllib.URLopener()
image.retrieve(imageURL,outputImage) # retrieve image
image.retrieve(worldfileURL,outputWorldfile) # retrieve worldfile

return outputImage, outputWorldfile

if __name__ == '__main__':
try:
# Check out Spatial Analyst extension license
gp.CheckOutExtension("Spatial")

# Get inputs
inputTxtFile = gp.getparameterastext(0)
imageName = gp.getparameterastext(1)
OutClassifiedRaster = gp.getparameterastext(2)
OutClassifiedRaster = os.path.join(gp.scratchworkspace,
                                   "mlcoutput")
TrainingPolygonFC = os.path.join(gp.scratchworkspace,
                                  "TrainingPolygons.shp")

##-----
##Create Feature Class from the trainingpolygons
##-----
outputSR = "GEOGCS['GCS_WGS_1984',DATUM['D_WGS_1984',SPHEROID
           ['WGS_1984',6378137.0,298.257223563]],PRIMEM
           ['Greenwich',0.0],UNIT['Degree',0.0174532925199433]];
           IsHighPrecision"
gp.CreateFeaturesFromTextFile_samples(inputTxtFile, ".",
                                       TrainingPolygonFC, outputSR)
gp.AddField(TrainingPolygonFC, "TrainingID", "LONG")
gp.CalculateField(TrainingPolygonFC, "TrainingID", "[File_ID]")

##-----
##Delete "Id" field from TrainingPolygon.shp
##-----
gp.AddMessage("\tDeleting 'File_ID' and 'Id' fields from training
              polygon feature class....")
gp.deletefield_management(TrainingPolygonFC, "File_ID;Id")

##-----
##Retrieve MODIS image
##-----
InRasters, WorldFile = retrieveMODISImage(imageName)

##-----

```

```

##Create signature file
##-----
gp.AddMessage("\tCreating signature file....")
SignatureFile = os.path.join(gp.scratchworkspace, "sigfile.gsg")
SampleField = "TrainingID"
gp.CreateSignatures_sa(InRasters, TrainingPolygonFC, SignatureFile,
                       "", SampleField)

##-----
##Run maximum likelihood classification
##-----
gp.AddMessage("\tConducting maximum likelihood classification....")
gp.MLClassify_sa(InRasters, SignatureFile, OutClassifiedRaster)

except:
    tb = sys.exc_info()[2]
    tbinfo = traceback.format_tb(tb)[0]
    pymsg = "PYTHON ERRORS:\nTraceback Info:\n" + tbinfo + "\nError
            Info:\n      " + \ str(sys.exc_type)+ ": " +
            str(sys.exc_value) + "\n"
    gp.AddError(pymsg)
    gp.AddError("GP ERRORS:\n" + gp.getmessages())

```