Comparison of Segment and Pixel Based Non-Parametric Classification of Land Cover in the Amazon Region of Brazil Using Multitemporal Landsat TM/ETM+ Imagery

Katherine A. Joseph

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Forestry

Randolph H. Wynne, Chair
John O. Browder
James B. Campbell

May 9, 2005
Blacksburg, VA

Keywords: remote sensing, contextual classification, non-parametric, *k*NN, CART, multitemporal, Amazon, Brazil, Landsat

Comparison of Segment and Pixel Based Non-Parametric Classification of Land Cover in the Amazon Region of Brazil Using Multitemporal Landsat TM/ETM+ Imagery

Katherine A. Joseph

(ABSTRACT)

This study evaluated the ability of segment-based classification paired with non-parametric methods (CART and *k*NN) to classify a chronosequence of Landsat TM/ETM+ imagery spanning from 1992 to 2002 within the state of Rondônia, Brazil. Pixel-based classification was also implemented for comparison. Interannual multitemporal composites were used in each classification in an attempt to increase the separation of primary forest, cleared, and re-vegetated classes within a given year. The *k*NN and CART classification methods, with the integration of multitemporal data, performed equally well with overall accuracies ranging from 77% to 91%. Pixel-based CART classification, although not different in terms of mean or median overall accuracy, did have significantly lower variability than all other techniques (3.2% vs. an average of 13.2%), and thus provided more consistent results. Segmentation did not improve classification success over pixel-based methods and was therefore an unnecessary processing step with the used dataset. Through the appropriate band selection methods of the respective non-parametric classifiers, multitemporal bands were chosen in 38 of the 44 total classifications, strongly suggesting the utility of interannual multitemporal data for the separation of cleared, re-vegetated, and primary forest classes. The separation of the primary forest class from the cleared and re-vegetated classes was particularly successful and may be a possible result of the incorporation of multitemporal data. The land cover maps from this study allow for an accurate annualized analysis of land cover and can be coupled with household data to gain a better understanding of landscape change in the region.

ACKNOWLEDGEMENTS

This research was funded by the US National Science Foundation (BCS-0136965), NASA (NAG5-10548 and NNS-04AB26G), the Rondônia Agroforestry Pilot Project, supported by the John and Teresa Heinz Charitable Trust (1992-1997), and the US McIntire-Stennis Research Program (VA-136589), and I thank these organizations for making this study possible. I also thank my committee members, Dr. Randolph H. Wynne, Dr. John O. Browder, and Dr. James B. Campbell, for their support and cooperation over the course of this project. I would especially like to extend my gratitude to my advisor, Randy Wynne, for his guidance, continual encouragement, and extraordinary insight. His dedication to high-quality scientific research and education is the example I hope to follow in all of my future endeavors. Additionally, several of my fellow graduate students have provided assistance and support over the past two years, in particular Christine Blinn, Jan Van Aart, and Aaron Bernard. There are many people that I have learned from and enjoyed over my long stay here at Virginia Tech, including old friends that have stayed with me in Blacksburg and those who have moved on, my fellow forestry graduate students, and the faculty and staff of the Virginia Tech College of Natural Resources and Department of Forestry. Finally, I would like to thank my parents Robert and Felisbela Joseph for their love and the valuable life lessons that they have extended to me. They have been and will always be my first teachers in all that I have done and will do.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 The Brazilian Amazon and the State of Rondônia

The rainforests of Rondônia, Brazil have been subject to rapid landscape change over the past 30 years due to various rural development programs and the construction of BR 364, an interstate highway providing access into the region and the state of Rondônia in particular (Browder and Godfrey 1997, Perz 2002). The intent of these regional programs (POLONOROESTE, and PLANALFORO) was to open areas to settlement for landless sharecroppers and to promote sustainable farming systems such as perennial agriculture. As a result, the population in Rondônia grew from approximately 111,000 in 1970 to 1,588,600 in 1990, near the start date of this study (Browder and Godfrey 1997, Perz 2002). The immigration of farmers into the region has led to forest clearing intended for agriculture, resulting in extensive deforestation (INPE 2001). Implications of such considerable ecological alteration include decreases in global carbon sequestration and increases in greenhouse gas emissions (Grace et al. 1996, Fearnside 1997), potential loss of biodiversity (Lugo 1988, Fearnside 1999), and degradation of soil (De Souza et al. 1996). Understanding the factors that influence land cover conversion over time in the Amazonian frontier is necessary to address these ecological issues while promoting sustainable development in the region.

One common pattern of settlement in Rondônia begins with clearing for perennial and annual agriculture followed by conversion to pasture and cattle production (Browder 1996, McCracken et al. 2002). This kind of trajectory, or land use pathway through time, can be associated with socioeconomic conditions, household labor capacity, market prices of agricultural and forest resources, regulatory prices, and landowner perceptions to further the understanding of the smallholder decision process on farm practices and land use management (Browder 1996). Likewise, it is valuable to understand the impacts of these land use changes on the landscape both at a farm level and a more regional scale.

The prominent categories of land use in the Amazon include primary forest, pasture, perennial agriculture (cocoa, coffee), secondary growth, and annual agriculture.

The amount of remaining primary forest in the region has been a focus of many studies because of its global ecological importance in terms of carbon sequestration and biodiversity (Tardin et al. 1979, Skole and Tucker 1993, Alves et al. 1998, Alves 1999). While the intent of the development programs was to promote sustainable agricultural systems such as perennial production, pasture has become the dominant land use in the region (Browder and Godfrey 1997, Porro 2002). Pasture areas tend to be large in size, in some instances encompassing an entire 100 hectare property. On satellite imagery they usually appear as cleared, barren land during the dry season, and occasionally as re-vegetated land when few cattle are being managed. Perennial agriculture includes coffee, cocoa, and agroforestry plantations; it appears as cleared in the first year or two of establishment and will gradually transition to a re-vegetated shrub stage. Secondary growth is re-vegetated land that has been abandoned following agricultural practices or clearing (Perz and Walker 2002). Once permitted to regrow, these abandoned areas can return to forest and can serve many of the same functions as primary forest (Perz and Walker 2002). Secondary growth can range in land cover from cleared to late successional secondary forest. Annual agriculture consists of crops such as maize, beans, and rice managed primarily for subsistence agriculture and for sale in small local markets (McCracken et al. 2002). Areas in annuals tend to be very small in size and represent the lowest proportion of land in production of the aforementioned land uses. During the dry season, areas in annual agriculture are fallow and therefore appear cleared in satellite imagery.

The significant anthropogenic land uses in the Brazilian Amazon do not directly correspond to distinct land cover classes that are identifiable in satellite imagery. For this reason it is imperative that ground level information be utilized in conjunction with remote sensing techniques to understand both the factors influencing land use change and the impacts of land use changes of smallholder farmers on the landscape.

1.2. Classification in the Amazon

Remote sensing techniques have become prominent in landscape classification in the Amazon Basin largely due to the expanse of the region and its inaccessibility (Skole and Tucker 1993, Roberts et al. 2003). Several studies have proven the efficiency of remote sensing in image classification for estimating forest area versus non-forest area with high overall accuracies (› 90% or kappa › 0.9) and dominated the remote sensing research activity in the Amazon through the 1980s and 1990s (Tardin et al. 1979, Skole and Tucker 1993, Alves et al. 1998, Alves 1999). A more recent focus of remote sensing applications in the Amazon has been on land cover classification beyond forest and non-forest classes (Donnely-Morrison 1994, Brondizio et al. 1996, Moraes et al. 1998, Roberts et al. 2002, Roberts et al. 2003, Guild et al. 2004). Generally clearing, re-vegetation, and forest can be mapped using Landsat imagery. Guild et al. (2004) were able to perform a pathway analysis using three different image years (1984, 1986, and 1992) in Rondônia, Brazil. The Landsat TM imagery was combined using both the tasseled cap and principal components transformations. Unsupervised techniques were used to train a maximum likelihood classification that produced 17 total land cover classes of varying combinations of forest, cleared, flooded, dry, and regrowth. The tasseled cap land cover change classification produced an overall accuracy of 79.3% (k=0.78) with individual class accuracies ranging from 54% to 100%. Using Landsat MSS/TM imagery, Roberts et al. (2002) were able to classify primary forest, pasture and green pasture, second growth, soil/urban, water, and cloud using spectral mixture analysis and a decision tree classifier and were able to obtain an overall accuracy of 85% (kappa = 0.76)

1.3. Multitemporal Classification

The introduction of multitemporal imagery is valuable in land use/land cover classification with Landsat TM data (Lo et al. 1986, Wynne et al. 2000, Ippoliti-Ramilo et al. 2003, Guild et al. 2004). Among the most accurate land use classification techniques are those using multitemporal imagery throughout a single year to exploit seasonal transitions in land cover (Ippoliti-Ramilo 2003). Although intra-annual multitemporal classification is a successful technique, it is not practical for use in the tropical region of Brazil. Within the Amazon Basin acquisition of multiple cloud-free images within the same year is rarely feasible. Multitemporal classification is thus limited to the use of interannual imagery for the majority of the Amazon region. Several studies have used interannual multitemporal imagery to strengthen land cover identification in tropical regions (Helmer et al. 2000) including the Brazilian Amazon (Lucas et al. 1993, Adams et al. 1995, Alves and Skole 1996, Alves et al. 2003, Guild et al. 2004). However, most of these studies have commented on the post classification utility of interannual imagery in distinguishing between land covers and have not attempted to use multitemporal imagery in the classification phase. For example, Lucas et al. (1993) used post classification techniques to determine patterns of forest re-growth and ultimately a secondary forest classification was developed.

1.4. Non-Parametric Classification

Most traditional classifiers are parametric, based upon statistical assumptions, including the multivariate normal distribution within spectral classes. This assumption does not fit all applications, and is difficult to implement in complex landscapes with classes of high variance (Hansen et al. 1996). Alternatively, non-parametric methods are not limited by such assumptions and are not based upon class statistics such as mean vectors and covariance matrices. Traditionally, land use/land cover classification in the Amazon has been done using parametric algorithms including minimum distance (Alves et al. 2003) and maximum likelihood (Alves and Skole 1996, Guild et al. 2004, Pan et al. 2004). The heterogeneity of land cover within the Amazon region of Brazil has caused classification difficulty in the past (Alves and Skole 1996, Brondizio et al. 1996). As a result, some have turned to the use of non-parametric algorithms (Roberts et al. 2002). Non-parametric techniques, including artificial neural networks (ANN) and classification and regression trees (CART) have become prominent in recent literature to eliminate the restriction of parametric statistical assumptions (Hansen et al. 1996, Friedl and Brodley 1997, Lawrence and Wright 2001, Pal and Mather 2003, Rogan et al. 2003, Krishnaswamy et al. 2004). Although ANN classification has been shown to greatly improve accuracy over traditional parametric methods with reduced training sets, the process to implement classification is not straightforward and can be time consuming (Pal and Mather 2003). Pal and Mather (2003) found that CART classification provided higher accuracy than ANN classification for Landsat ETM+ imagery in Eastern England for seven land cover types (wheat, potato, sugar beet, onion, peas, lettuce, and beans). In addition to its non-parametric nature, CART is gaining increasing attention due to its ease of use and computational efficiency (Lawrence and Wright 2001, Pal and Mather 2003, Lawrence et al. 2004). During the CART classification process a binary tree is created where decision boundaries are estimated empirically from the training data. A test in the form of $x_i > c$ is performed at each node where x_i is the feature or spectral band and c is the threshold estimated from the distribution of x_i (Breiman et al. 1984). Several studies have found CART to be an acceptable classification method (Hansen et al. 1996, Lawrence and Wright 2001, Rogan et al. 2003, Krishnaswamy et al. 2004) and have shown improvements in accuracy over traditional parametric classifiers (Friedl and Brodley 1997, Pal and Mather 2003).

Another non-parametric classification technique that has been recently explored and improved upon in many forest cover classifications is *k*-nearest neighbor (*k*NN). *k*NN uses a fuzzy non-parametric supervised classification (minimum distance) to assign pixels to informational classes. Rather than assigning a pixel to the mean of the closest spectral class, the classifier assigns a pixel to the majority of the *k* closest training pixels in spectral space. Serpico et al. (1996) compared traditional *k*NN methods with three neural network techniques and found *k*NN to have the highest overall accuracy, however, the difference was not significant. While *k*NN was the most accurate classifier and the simplest during the training phase, the authors claimed it to be more difficult during the classification stage than the neural network classifiers, likely due to the multiple classifications needed to determine the optimal *k* value. The *k*NN algorithm has been successfully implemented in several national forest inventory programs, including in Finland, where it was popularized (Katila and Tomppo 2001, Tomppo and Halme 2004) and in the United States (Franco-Lopez et al. 2001, McRoberts et al. 2002, Haapanen et al. 2004). In several of these studies forest inventory information is coupled with satellite imagery to predict forest and land use attributes of a continuous digital surface (McRoberts et al. 2002, Tomppo and Halme 2004).

1.5. Segment-Based Classification

The majority of remote sensing classifications in the Amazon have used point or pixel-specific methods in which each pixel is classified individually without regard to the spatial relationship of neighboring pixels (Lo et al. 1986, Donnelly-Morrison 1994, Alves and Skole 1996, Ippoliti-Ramilo et al. 2003, Guild et al. 2004) although segment based approaches have gained recent attention (Palubinkas et al. 1995, Brondizio et al. 1996, Lu et al. 2004). Pixel-based classification often results in high heterogeneity giving low classification accuracy for complex landscapes and a salt and pepper appearance to the classified image. Contextual or segment-based classification incorporates spatial relationships among pixels in an attempt to extract homogeneous objects on the landscape and thus decrease the heterogeneity that is an artifact of pixel-based methods (Richards and Jia 1999, Campbell 2002). Many studies have shown an improvement over pixel-based land use classification accuracy through the incorporation of segment-based methods (Palubinkas et al. 1995, Lobo et al. 1996, Shandley, et al. 1996). One of the primary benefits of segment-based classification, as noted by De Wit and Clevers (2004), is the ability to capture the spectral variability within land use types such as shadowing, moisture conditions, and species variability.

Segment based classification has been used in several studies within the Amazon region of Brazil (Palubinkas et al. 1995, Brondizio et al. 1996, Lu et al. 2004). Palubinkas et al. (1995) compared the results from eight different texture based methods (based on a Markov random fields model) with traditional minimum distance and maximum likelihood classifiers to segment Landsat TM imagery for classification of regenerating forest. Six of the varying segment extraction approaches used maximum likelihood classification, while two used minimum distance classification. Those classifiers that implemented a maximum likelihood algorithm following segmentation consistently out-performed the pixel-based classification methods in terms of overall accuracy.

This study evaluated the ability of segment-based classification paired with non-parametric methods (CART and *k*NN) to classify a chronosequence of Landsat

TM/ETM+ imagery spanning from 1992 to 2002 within the state of Rondônia, Brazil. Pixel-based classification was also implemented for comparison. Interannual multitemporal composites were used in each classification in an attempt to increase the separation of primary forest, cleared, and re-vegetated classes within a given year.

OBJECTIVES

The principal aim of this study was to produce accurate annual maps of Amazonian land cover (cleared, re-vegetated, and primary forest) using non-parametric techniques applied to interannual, multitemporal Landsat TM/ETM+ imagery. The specific objectives were:

(i) To determine the preferred non-parametric classification technique between the *k*-nearest neighbor (*k*NN) and classification and regression trees (CART) methods;

(ii) To determine whether land cover classification using segment-based methods is more accurate than comparable efforts using per-pixel methods; and

(iii) To assess the utility of interannual multitemporal data in the classification of a single year.

2. METHODS

A flowchart outlining the basic processing steps of this study is shown in Figure 1.

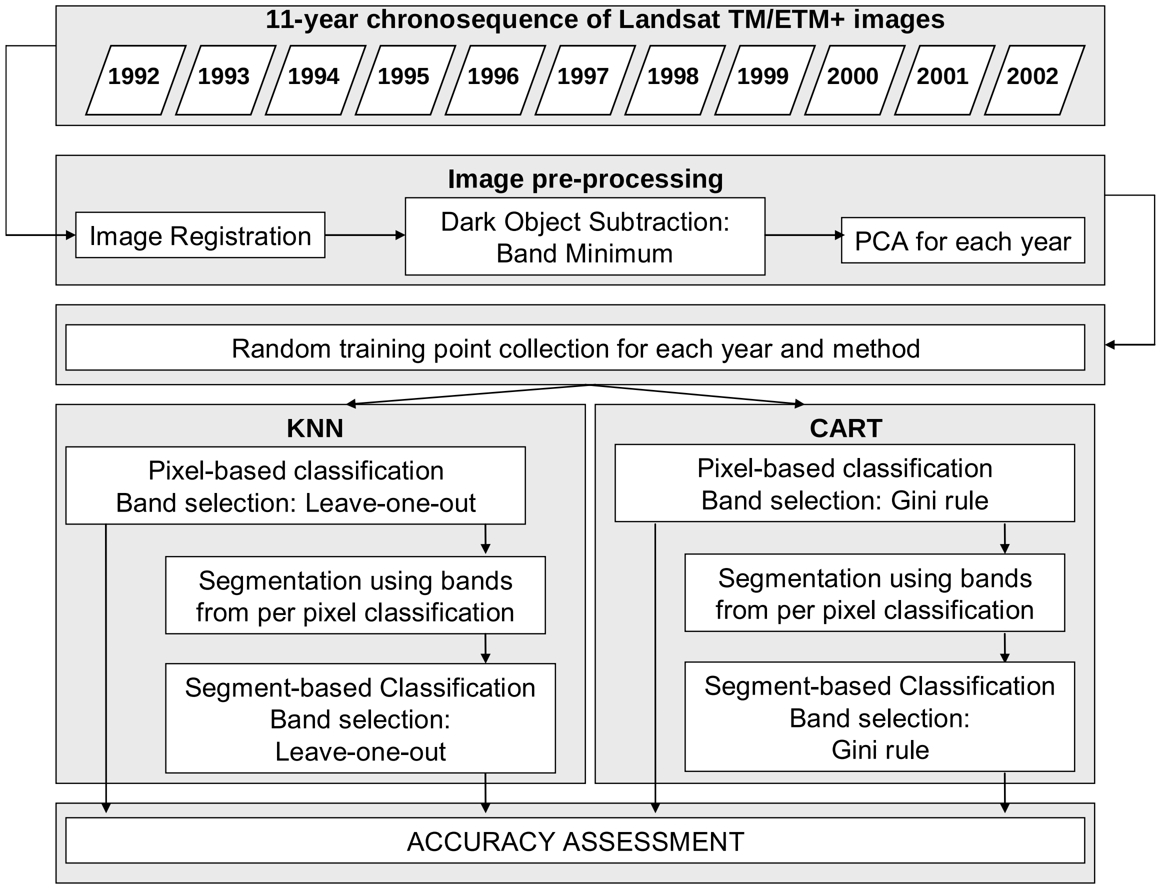

Figure 1. Flowchart of processing steps

2.1. Study Area

The analysis of land cover pathways was performed in two study sites located in western Brazil in the state of Rondônia. The first site is located in the municipio of Nova União (UL 62°41'Wx10°48'S, LR 62°29'W x 10°56'S) and the second in the municipio of Rolim de Moura (UL 62°57'Wx10°29'S, LR 62°49'W x 10°48'S) (Figure 2). While the colonization of Rondônia has been occurring since the 17th century, a series of regional development programs in the 1970s, along with the construction of a highway providing access to the region, spurred extensive migration into the region (Mahar 1979, Fearnside 1986, Goodman 1990, Browder and Godfrey 1997). The development plan included grid-like settlement areas with individual plots totaling approximately 100 hectares each. As farmers have colonized the region the conversion from forest to various agricultural systems has generally originated near the road and extended toward the rear of the property. Prominent agricultural systems include coffee, cocoa, annuals, and pasture.

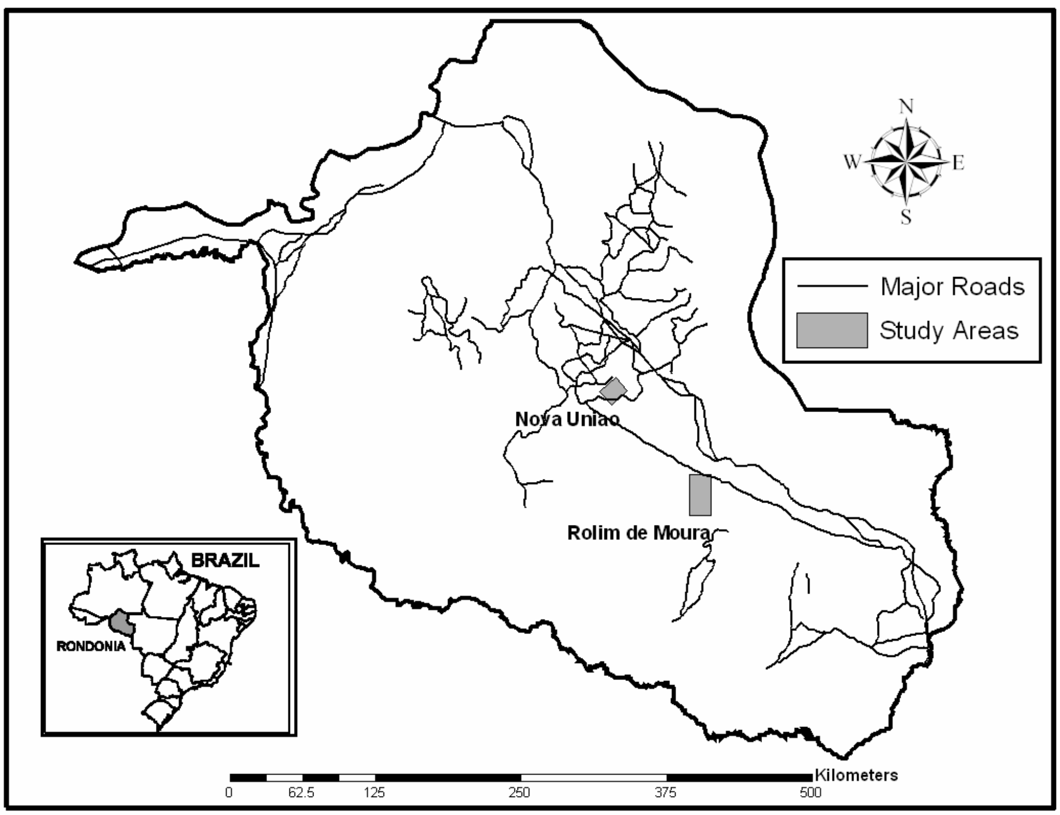

Figure 2. Study areas in Rondônia, Brazil

The major natural vegetation cover within the study region is transitional tropical seasonal moist forest (TTSMF). The dry season extends from June to September, with an average annual rainfall in Nova União and Rolim de Moura ranging from 1600 to 1700 mm and 2000-2250 respectively. Elevation ranges from 100-225 meters in Nova União and is 250 meters in Rolim de Moura (IBGE elevation maps 1974). The main soil type in Nova União is a eutrophic yellow-red podsol with patches of eutrophic litolic soil. In Rolim de Moura the prominent soil type is a eutrophic yellow-red podsol and non-hydromorphic cambisol (Projeto Radambrasil Mapa Exploratório de Solos, 1:1,000,000, 1979).

2.2…Data and Pre-Processing

The digital data consist of a multitemporal series of Landsat Thematic Mapper (TM) and Enhanced Mapper (TM/ETM+) images (path 231/row 68) from 1992 through 2002 (Table 1). Figure 3 provides an example of a Landsat TM image (1992) for path 231, row 68. All images were registered to the 2002 image, which was rectified by EROS Data Center prior to purchase. Registration was performed in Erdas Imagine 8.7 software. At least 75 control points were created for each image pair, 1/3 of which were randomly selected as check points. A control point root mean squared error (RMSE) of less than one-fifth pixel for a 1^{st} order transformation was met for all registrations with check point error not exceeding 0.21 pixels. Dai and Khorram (1998) found that an RMSE of less than one-fifth pixel was necessary to achieve a change error of less than 10%. Using these same criteria for use with multitemporal image chronosequences is just as, if not more, needed. Nearest neighbor resampling was used.

Table 1. Landsat TM/ETM+ image dates used in study

Sensor	Date
TM	25-Jul-92
TM	25-May-93
TM	15-Jul-94
TM	3-Aug-95
TM	20-Jul-96
TM	21-Jun-97
TM	23-May-98
ETM+	6-Aug-99
ETM+	24-Aug-00
ETM+	11-Aug-01
ETM+	26-May-02

Although conversion to reflectance is desirable when working with multitemporal imagery, the variety of sources, preprocessing methods, and the lack of metadata associated with the imagery precluded it. Due to this limitation, band minimum dark object subtraction was implemented as a technique to radiometrically normalize the chronosequence from 1992 to 2002 (Chavez 1989).

A principal component analysis (PCA) was performed on all images to reduce the amount of data and to highlight the greatest variability within the images.

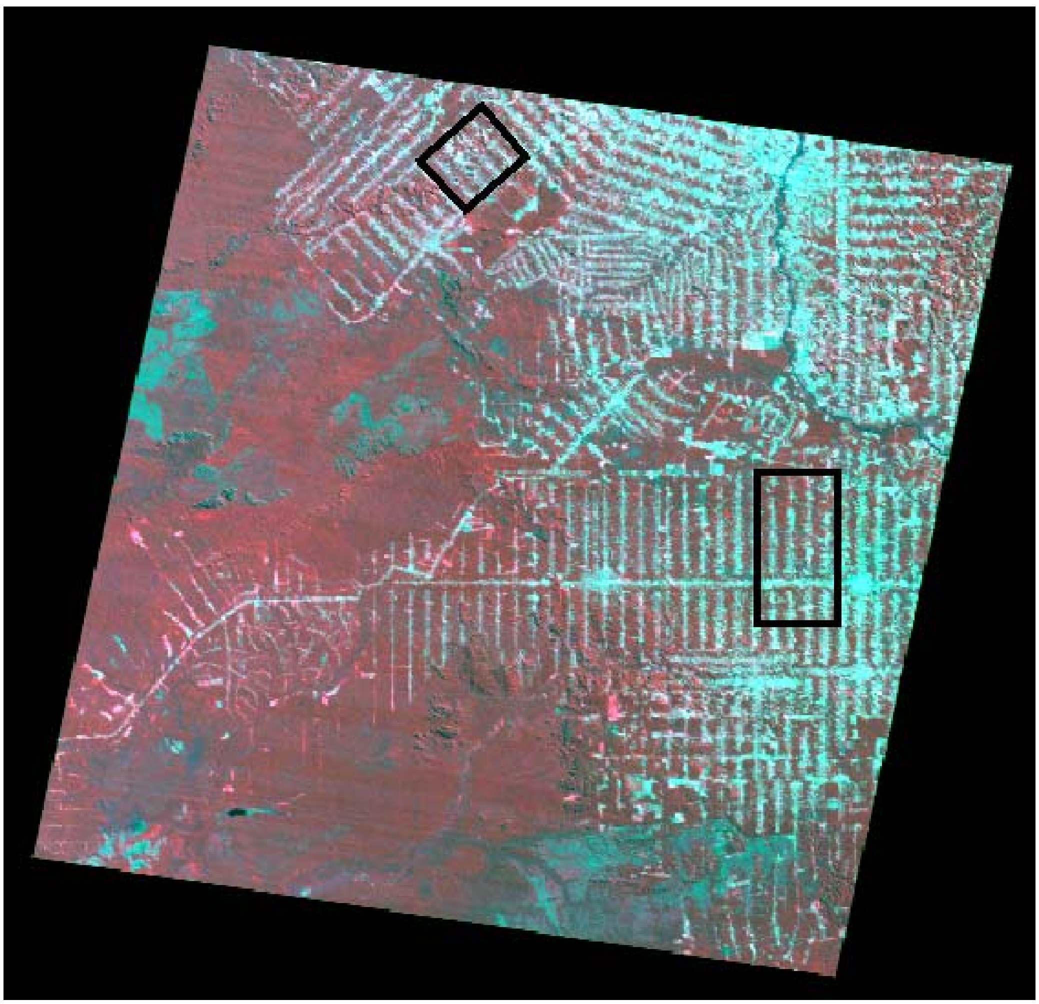

Figure 3. Full Landsat TM image for path 231, row 68 (July 25, 1992) in false color (bands 4, 3, 2 in R, G, B).   The Nova União study area is outlined in the upper left portion of the image and the Rolim de Moura study area is outlined in the lower right portion of the image.

2.3. Training Data and Classes

There are several different sources that can be used in collecting training pixels including in situ data, ancillary data such as topographic maps, aerial photographs, or satellite imagery (Richards and Jia 1999). In remote regions, such as the Amazon, inaccessibility limits the ability to collect in situ field data. When collecting training data in a similar remote environment, Frizzelle et al. (2003) incorporated several different sources to create a training data set including GPS data, field sketch maps, a longitudinal social survey, and satellite imagery. In conjunction with ancillary data the imagery to be classified has also been used to collect training and validation points (Cohen et al. 1998, Sader et al. 2003).

As a part of the larger study, research crews collected detailed information through field interviews with farmers within both study areas in 1992 and 2002. In addition, detailed interviews were conducted in conjunction with GPS point collection in 2003 to gain information on land conversion through years 1993-2001. In both 1992 and 2002 detailed maps were created of each farm included in the study (Figure 4). Using information from the 1992 and 2002 maps, detailed interviews in 2003, and Landsat imagery, a random sample of 838 training pixels was labeled for each year. 50 points per class were randomly removed from each annual training set prior to classification for use as a validation sample. A random sample of training points ensures the inclusion of edge and mixed pixels, which are necessary in avoiding inflated accuracies, particularly in the validation sample (Powell et al. 2004).

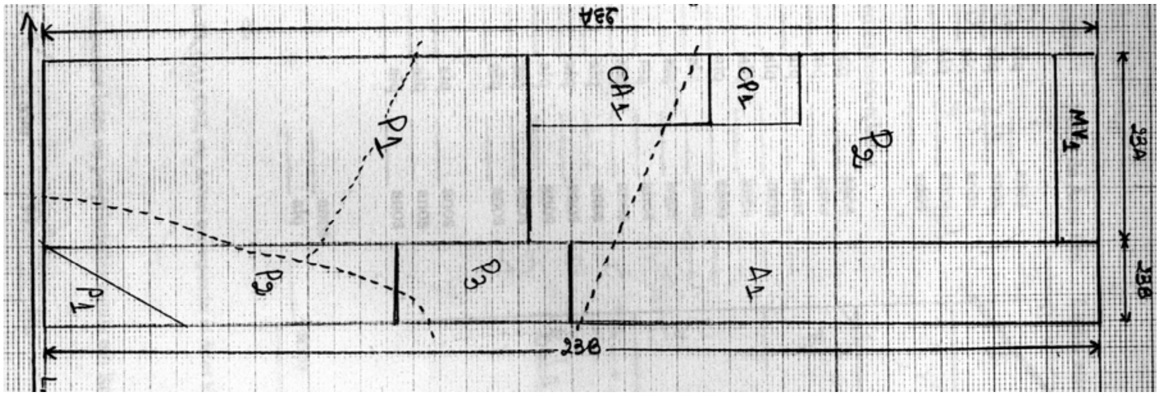

Figure 4. Example of a detailed map constructed during the interview processes in 1992 and 2002. *P* represents pasture, *CA* represents coffee, *A* represents annuals, *MV* represents *mata virgem* or primary forest.

Based on the literature, the two non-parametric classification methods used in this study have differing requirements for training data for optimal classification. The same training data were used for both pixel-based and segment-based classifications to enable an accurate comparison of each technique. Hardin (1994) performed several nearest neighbor classifications, including *k*NN, using a large training set where the proportions of samples within classes represented the proportions within the population. A reduced training set was also used with all classification methods. The results indicated that when large training data sets are used where class proportions are the same as the population to be classified, nearest neighbor techniques are statistically superior to the best parametric classifiers. The results from the reduced training set were not as consistent and were often inferior to parametric methods. CART alternatively, performs best when the training data set has approximately the same number of samples in each class to prevent over- representation in class assignment for classes with more samples (Rogan et al. 2003, Lawrence et al. 2004).

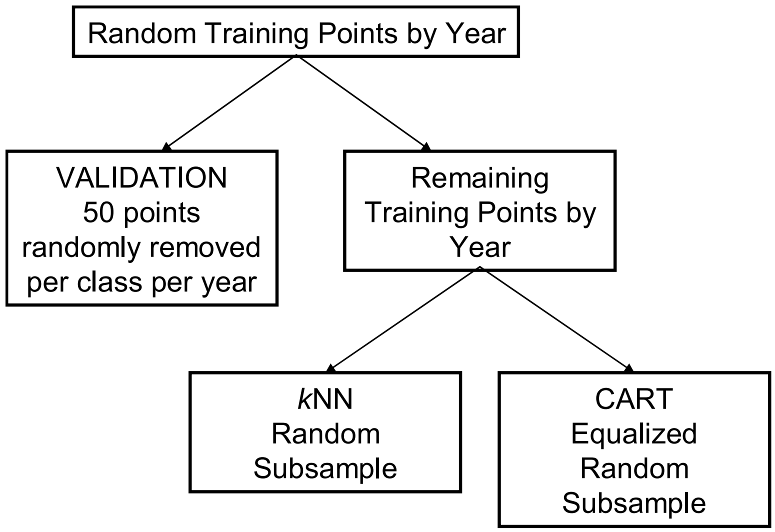

Figure 5. Breakdown of training data subsets for the validation sample, *k*NN subsample, and CART subsample

To optimize results for each classification method, the original dataset was reduced to obtain both a random sample of proportional classes to the population (for use in *k*NN classification) and a random sample of equal classes (for use in CART classification) (Figure 5). Each dataset contained the same number of training pixels and was limited by the number of points in the class with the minimum samples by year (Table 2).

Table 2. Training points by year for the validation sample, the *k*NN random subsample, and the CART equalized random subsample. Class 1 is cleared, Class 2 is re-vegetated, and Class 3 is primary forest.

	Validation			*k* NN			CART		
YEAR	**Class 1**	**Class 2**	**Class 3**	**Class 1**	**Class 2**	**Class 3**	**Class 1**	**Class 2**	**Class 3**
1992	50	50	50	131	77	179	129	129	129
1993	50	50	50	173	140	218	177	177	177
1994	50	50	50	199	150	203	184	184	184
1995	50	50	50	226	92	138	152	152	152
1996	50	50	50	216	103	140	153	153	153
1997	50	50	50	226	85	121	144	144	144
1998	50	50	50	260	134	125	173	173	173
1999	50	50	50	217	75	77	123	123	123
2000	50	50	50	245	64	78	129	129	129
2001	50	50	50	197	63	40	100	100	100
2002	50	50	50	162	74	37	91	91	91

Table 3. Confused training points removed for each method and year for segment-based classification. Class 1 is cleared, Class 2 is re-vegetated, and Class 3 is primary forest.

	k NN			CART		
YEAR	**Class 1**	**Class 2**	**Class 3**	**Class 1**	**Class 2**	**Class 3**
1992	4	3	0	4	4	2
1993	5	8	1	7	6	2
1994	15	14	2	9	5	0
1995	5	5	1	5	9	3
1996	7	8	2	1	2	1
1997	4	4	0	5	7	4
1998	12	7	3	5	6	2
1999	4	7	5	2	3	1
2000	1	1	0	1	3	2
2001	2	2	1	2	5	5
2002	2	3	1	1	2	2

Three different land use classes were included in the study; primary forest, cleared, and re-vegetated, following the classification scheme of Guild et al. (2004). Forest areas consist of primary forest. Re-vegetated areas include perennial agriculture following clearing (e.g., cocoa and coffee plantations), secondary growth, and occasionally "dirty" poorly managed pasture. The cleared class contains areas that are rotated for cattle production, annual agriculture (which are not vegetated during dry season months), recently abandoned areas, and first year perennial agriculture plots.

2.4. Classification

Two non-parametric classification methods, *k*-nearest neighbor (*k*NN) and classification and regression trees (CART) were applied to both image pixels and image segments.

2.4.1. Multitemporal Band Selection

An interannual multitemporal image from the available Landsat TM/ETM+ imagery was created to perform land use classification of each single date, using the target year, the three previous (backward) years, and three forward years where applicable. Previous literature emphasizes the utility of incorporating multitemporal imagery in land use/land cover classification (Lo et al. 1986, Alves and Skole 1996, Wynne et al. 2000, Guild et al. 2004); however, it was unknown what combination of years and bands would be most beneficial to the analysis. For this reason all principal components (PCs) for the target year were evaluated along with the first PC from each of the three backward years and three forward years from the target date (Figure 6).

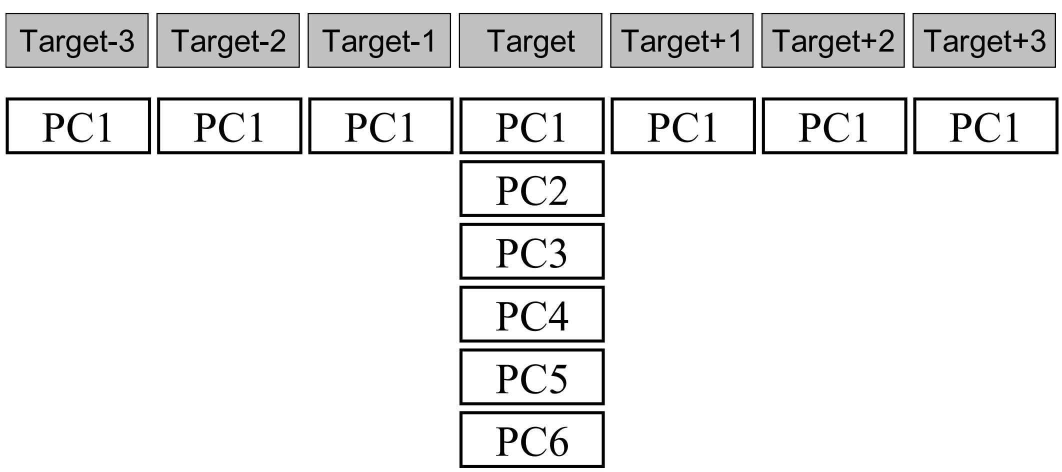

Figure 6. Principal component bands evaluated for each target year and the three forward and backward years

The band selection technique used in the *k*NN algorithm is a leave-one-out approach that tests each possible band (and *k*) combination using the input training data. The best band combination is the one that has the highest overall accuracy. During this band selection process, all PCs that increase accuracy are selected, even if the increase is only incremental. CART 5.0 inherently chooses the best possible bands to classify the data in a binary tree, and has been used as a method of band selection (Brieman et al. 1984, Bittencourt and Clarke 2004). The Gini splitting rule was applied to all CART classifications, which attempts to separate classes by focusing on one class at a time, and is given by:

$$Gini(c) = 1 - \sum_{j} p_j^2$$

where p_j is the probability of class j in c. The Gini rule sequentially looks for the largest class in the training data and strives to isolate it from all other classes. CART 5.0 grows trees until it is not possible to grow them any further. Once each full tree is generated, smaller trees are obtained by pruning away branches. The CART 5.0 algorithm uses a 10-fold cross validation approach to prune the full tree. Pruning is performed so bands that add only small incremental increases in accuracy are eliminated, resulting in the "optimal" tree.

2.4.2. CART

CART classification was performed using CART 5.0 (Salford Systems 2002, Lawrence et al. 2004). There are numerous attribute selection methods that can be used in the creation of decision boundaries. Pal and Mather (2003) tested the effects of five different attribute selection methods on CART classification accuracy and found differences to be minimal and not important to overall accuracy. Lawrence et al. (2004) used CART 5.0 (Salford Systems 2002) to perform CART classification and tested each of the available attribute selection methods for each application. The authors used the optimal method, based on overall accuracy, in final classification. The available selection methods in CART 5.0 include Gini, Symmetric Gini, Entropy, Class

Probability, Twoing, and Ordered Twoing. The Gini splitting rule and the pruned optimal tree were used in all CART classifications to standardize the procedure

2.4.3. *k*NN

Non-parametric *k*NN classification was performed using a program developed in specifically for this project in Fortran 95 (APPENDIX VI). A Euclidean distance metric was implemented to locate the *k* nearest neighbors (Serpico et al. 1996, Franco-Lopez et al. 2001, McRoberts et al. 2002, Haapanen et al. 2004) and constant weighting of the nearest neighbors was used giving all neighbors equal influence over class assignment, which has been shown to outperform other weighting schemes (Franco-Lopez et al. 2001, McRoberts et al. 2002, Haapanen et al. 2004).

McRoberts et al. (2002) provide a careful discussion on the selection criterion for *k*. The authors state that an objective criterion should be chosen, implemented, and reported. In addition, the resulting *k* value should not be extended to other studies with differing data sets. They found in their own study predicting forest land proportion that an optimum *k*-value, using the same objective criterion was $7 \leq k \leq 13$ for one study area and data set while it was $21 \leq k \leq 33$ for another. Our objective criteria for *k* selection was the *k*-value that minimizes overall classification error using a leave-one-out evaluation (integrated into the band selection process) of the training data with a maximum possible *k* of 20. This process iterated through all possible band and *k* combinations for the given training data to select both the optimal *k* and band combination. The maximum threshold (*k* of 20) was determined through preliminary testing and no classifications exceeded a *k* of 17. This objective criterion was applied separately for each dataset, therefore the *k* values varied by year.

The *k*NN program was implemented on a SGI Altix 3300 supercluster to reduce processing time.

2.4.4. Image Segmentation

An object-oriented, multiresolution segmentation algorithm (eCognition 3.0) was used to segment each multitemporal image into image segments for further classification. The eCognition algorithm is a hierarchical classifier that uses spectral and shape information to perform segmentation of imagery at a constant scale throughout the image (Baatz and Schäpe 2000). The developers of this method claim that it is robust for a wide variety of data types, results are repeatable, and the segmentation approach has been successful in many natural resource applications (Schiewe et al. 2001, Antunes et al. 2003, Laliberte et al. 2004, Van Aardt 2004, Van Aardt and Wynne 2004). The segmentation process used in eCognition serves to reduce the heterogeneity of objects at a specified scale. The heterogeneity of image objects is defined by the color and shape of the input image (eCognition User's Manual 2003). The default color/shape ratio in eCognition is 0.8/0.2. This color/shape ratio has been found to be acceptable (Laliberte et al. 2004), and in some cases optimum, for segmentation results in natural resource applications (Van Aardt 2004), and was found to be acceptable for this study through visual inspection.

The optimum segmentation for a given application is also highly dependent on the defined scale parameter. The scale parameter is a measure of the allowed change in heterogeneity between two merging objects and serves as a threshold to terminate segmentation (eCognition User's Manual 2003). The process of determining the optimal scale parameter is not straightforward. The methods in recent literature are still highly subjective and have included a simple visual inspection (Schiewe et al. 2001). The optimal scale parameter for this study was chosen through an evaluation of confused training samples. Confused training samples were training samples of differing classes that fell within the same image segment. The optimal scale parameter was determined to be 5 because it was the smallest scale parameter that could be computed on all images given the large image size, and also the scale parameter that minimized the number of confused training samples. The confused training points that were of differing classes and fell within the same image object or segment were removed prior to each classification (Table 2b).

The bands included in the segmentation process were determined by the bands selected during the pixel-based *k*NN and CART classification (Tables 8-11) to optimize the segmentation for the given data and classes. Multitemporal bands were therefore used in all segmentations. This process resulted in 22 total segmentations for further classification.

2.5. Accuracy Assessment

The validation data set was used to assess the accuracy of each classification. There were 50 validation points for each class per year. Multivariate techniques were used to perform the accuracy assessment, including both an error matrix and a kappa coefficient of agreement (Congalton 1991) by class and overall classification. As noted by Foody (2004), the kappa coefficient (formally estimated by $\hat{K}$) is based on the comparison of predicted and actual class labels for each case in the validation data set, and is calculated as

$$\hat{K} = \frac{p_o - p_c}{1 - p_c}$$

where p_o is the proportion of cases in agreement and p_c is the proportion of agreement expected by chance. Kappa variances and Z-scores were calculated to determine the significance (= 0.05) of each classification (Congalton and Green 1999) given the following hypotheses:

$$H_0 : \hat{K} = 0$$

$$H_1 : \hat{K} \neq 0$$

The error matrix was also used to calculate producer's accuracy, user's accuracy, and overall accuracy. The producer's accuracy relates to the probability that a reference sample was correctly mapped and measures the omission error (1 - producer's accuracy). In contrast, the user's accuracy indicates the probability that a sample from the land cover map actually matches what it was in the reference data and measures the commission error (1- user's accuracy).

11 classifications were performed for each method (1 per year from 1992 through 2002). A series of F-tests (H_0: $\sigma_1^2 - \sigma_2^2 = 0$, H_1: $\sigma_1^2 - \sigma_2^2 \neq 0$) determined that the

variances were not equal among methods. Since this result precluded an analysis of variance, four t-tests were performed (H_0: $\mu_1^2 - \mu_2^2 = 0$, H_1: $\mu_1^2 - \mu_2^2 \neq 0$) to assess whether or not the means differed among techniques (n = 11 in all cases). A Kruskal-Wallis test (H_0: population medians are equal, H_1: population medians are not equal) was included due to the small sample size and low associated power of the t-tests.

3. RESULTS

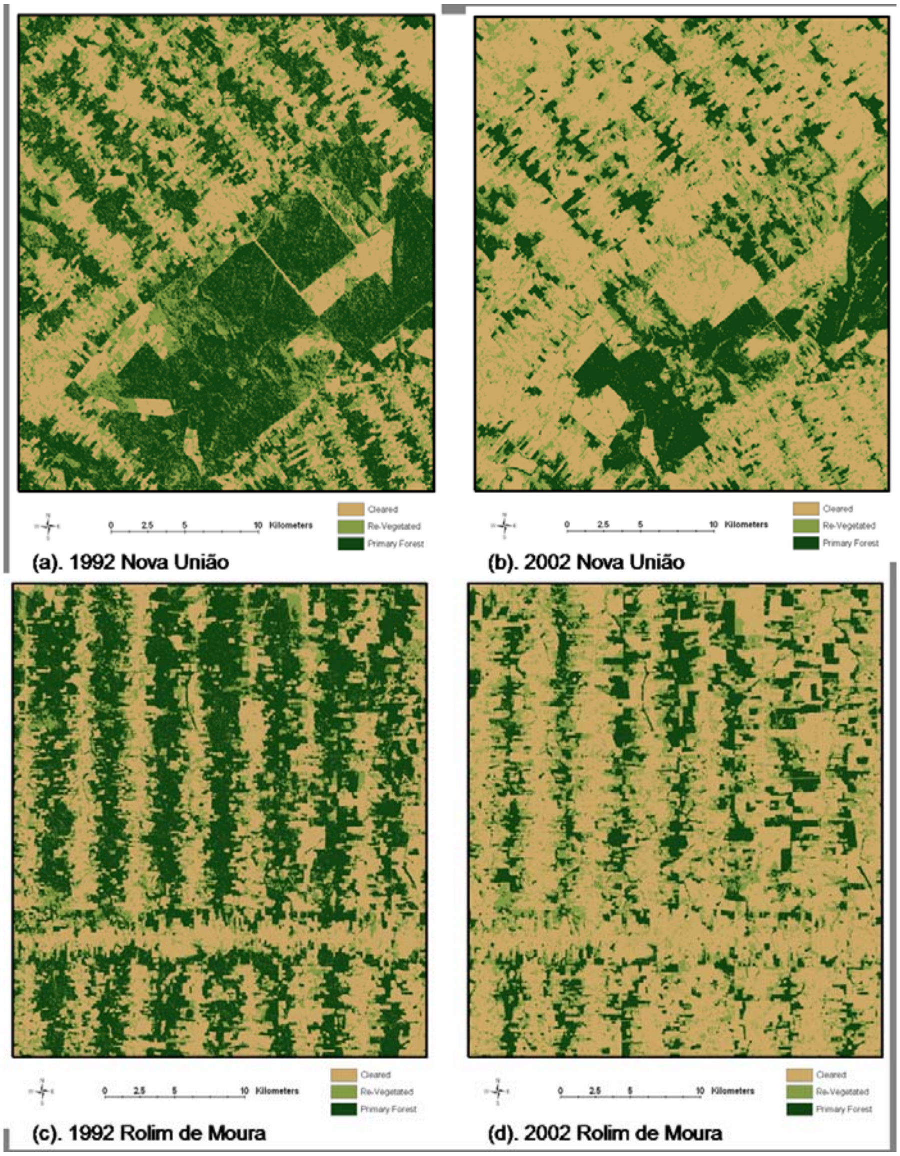

Figure 7. Classification results for the CART pixel-based classification for years 1992 and 2002 and for each study area, Nova União and Rolim de Moura

Classification results for the CART (pixel-based) classification are shown in Figure 7 for 1992 and 2002. Evaluation of summary statistics including overall accuracy, kappa coefficient of agreement, the Kruskal-Wallis test for equal medians, and a series of four t-tests reveal that the classification methods performed equally well and were not significantly different (Table 4, Table 5, Figure 8). The Kruskal-Wallis test results indicate that the classification medians are not different (p = 0.605) and the t-tests indicate that the classification means are not different (Table 5, n = 11 for all methods). All classifications were significant at an alpha level of 0.05 and the overall accuracies (ranging from 77% to 91%) are comparable to other studies (Roberts et al. 2002). Although the overall accuracies were not significantly different between the four classification methods, an F-test revealed that the variances between the CART (pixel-based) technique were significantly lower (σ^2 = 3.60) than all of the three other classification methods (*k*NN pixel-based σ^2 = 11.5, p = 0.04; *k*NN segment-based σ^2 = 12.6, p = 0.03; and CART segment-based σ^2 = 15.5, p = 0.02) indicating that this technique produced more consistent results in this case. The CART (segment-based) classification, however had variances similar to the *k*NN (pixel and segment-based) classifications.

Table 4. Percent overall accuracy by classification method and year

Year	*k* NN (pixel)	CART (pixel)	*k* NN (segment)	CART (segment)
1992	81.33	87.33	79.33	81.33
1993	86.60	87.67	80.67	82.00
1994	88.00	86.00	86.00	84.67
1995	84.67	84.00	78.67	80.00
1996	86.00	85.33	82.00	83.33
1997	89.33	84.00	86.00	90.67
1998	84.00	82.00	86.00	81.33
1999	85.33	85.33	86.00	90.67
2000	84.67	84.67	85.33	90.00
2001	91.33	88.00	86.67	86.00
2002	79.33	83.33	77.33	86.00
Mean	85.51	85.24	83.09	85.09
σ^2	11.55	3.60	12.56	15.50

Table 5. t-tests and associated *p*-values for the classification of cleared, re-vegetated, and primary forest using four methods in the Amazon region of Brazil

t-test	*p*-value (t-test)
*k*NN (pixel) vs. CART (pixel)	0.7831
CART (pixel) vs. CART (seg)	0.9454
*k*NN (seg) vs. CART (seg)	0.2250
*k*NN (pixel) vs. *k*NN (seg)	0.1181

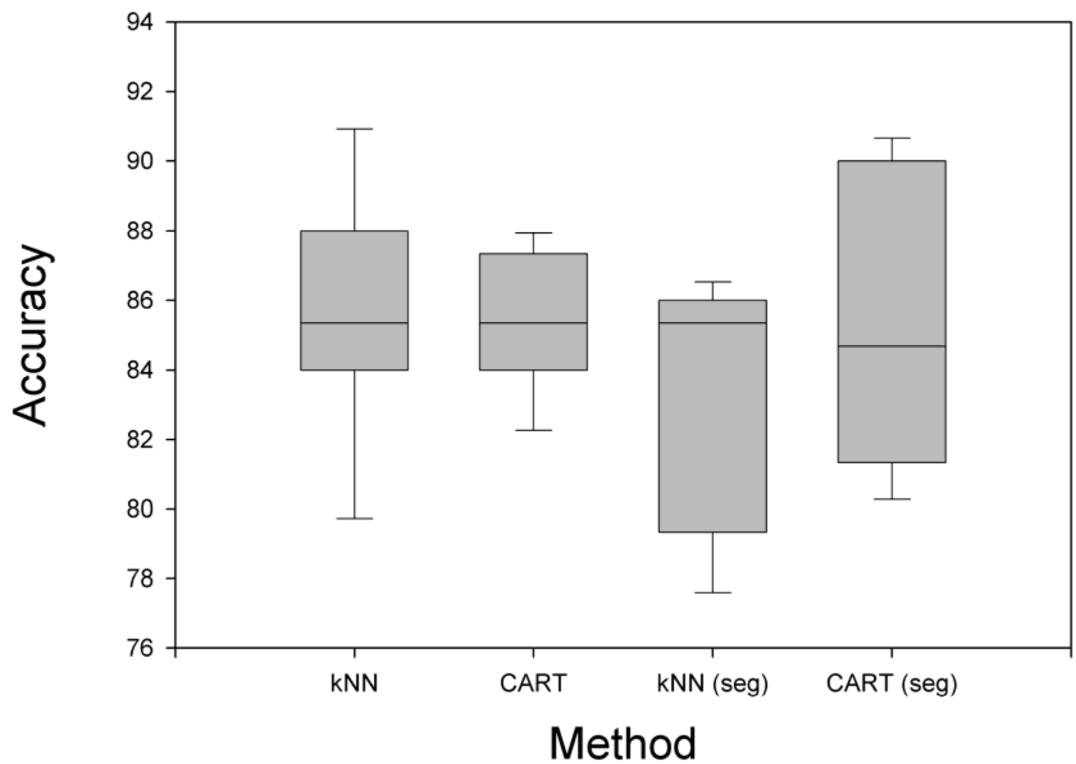

Figure 8. Box and whisker plot of mean overall classification accuracy by method

Tables 6 and 7 summarize the mean producer's and user's accuracies by class for each classification method. All classifications successfully separated primary forest with both producer's and user's accuracies greater than 90%. The classes that were the most difficult to separate were the re-vegetated and cleared classes. All methods tended to over-represent the cleared class by labeling many of the re-vegetated areas as cleared.

This was particularly prevalent with both the pixel and segment-based *k*NN classifications, as can be inferred from the large discrepancies in producer's and user's accuracies between the cleared and re-vegetated classes (Tables 6 and 7).

Table 6. Mean producer's accuracies for all classifications (bounds are given at a 95% confidence level)

	Producer's Accuracy			
	k NN (Pixel-Based)	CART (Pixel-Based)	*k* NN (Segment-Based)	CART (Segment-Based)
Cleared	95.5% ± 2.0%	88.2% ± 5.1%	92.7% ± 3.0%	84.4% ± 5.4%
Re-Vegetated	67.3% ± 5.2%	73.5% ± 5.7%	65.7% ± 7.6%	80.0% ± 4.5%
Primary Forest	93.1% ± 3.1%	93.8% ± 2.1%	91.3% ± 3.6%	90.9% ± 4.0%

Table 7. Mean user's accuracies for all classifications (bounds are given at a 95% confidence level)

	User's Accuracy			
	k NN (Pixel-Based)	CART (Pixel-Based)	*k* NN (Segment-Based)	CART (Segment-Based)
Cleared	78.3% ± 3.5%	82.1% ± 3.1%	75.9% ± 3.5%	85.3% ± 3.3%
Re-Vegetated	87.5% ± 4.2%	82.2% ± 4.3%	83.3% ± 4.7%	72.2% ± 5.8%
Primary Forest	92.7% ± 2.3%	92.5% ± 2.5%	92.9% ± 2.9%	85.2% ± 1.6%

As suggested by McRoberts et al. (2002), the optimal *k* varied drastically by dataset (Tables 8 and 10) ranging from a *k* of 1 to 17. This result reiterates that no one *k* can be applied to all classifications within the same study region or with varying datasets. Similarly the number of nodes selected through the Gini rule and 10-fold cross validation (CART) varied drastically across datasets from 3 to 21.

Table 8. Bands and *k* selected by year using a leave-one-out approach for pixel-based *k*NN classifications by year

	Selected Bands and *k*: kNN Pixel Based Classification							
Target Year	Target Year - 3 (1st PC)	Target Year - 2 (1st PC)	Target Year - 1 (1st PC)	Target Year (all 6 PC's)	Target Year + 1 (1st PC)	Target Year + 2 (1st PC)	Target Year + 3 (1st PC)	*k*
1992	N/A	N/A	N/A	1,6	1	1	0	11
1993	N/A	N/A	0	1,2,3,5	1	1	0	7
1994	N/A	0	0	1,3	0	0	1	3
1995	0	1	1	1,2,5,6	0	1	1	3
1996	0	1	1	1,5,6	0	0	1	17
1997	0	1	0	1,2,3,6	1	0	0	16
1998	0	0	1	2,3	0	1	0	4
1999	0	1	0	1,2,4,5,6	1	0	1	4
2000	0	0	1	1,2,6	1	1	N/A	6
2001	0	0	1	1,3,5,6	0	N/A	N/A	3
2002	0	1	1	2,5,6	N/A	N/A	N/A	9

Table 9. Bands selected and number of nodes in optimal tree using the Gini selection method and 10-fold cross validation for pixel-based CART classification by year

	Selected Bands: CART Pixel-Based Classification							
Target Year	Target Year - 3 (1st PC)	Target Year - 2 (1st PC)	Target Year - 1 (1st PC)	Target Year (all 6 PC's)	Target Year + 1 (1st PC)	Target Year + 2 (1st PC)	Target Year + 3 (1st PC)	Nodes
1992	N/A	N/A	N/A	1,2,3,5,6	1	1	1	21
1993	N/A	N/A	1	1,2,3,4,6	0	1	1	20
1994	N/A	1	1	1,3,4	1	0	0	12
1995	0	0	0	1,2	0	1	0	5
1996	0	0	1	1,2,3,5	1	0	1	17
1997	0	0	1	1	0	0	0	3
1998	0	0	1	1,2,3,4	1	0	0	12
1999	1	1	1	1,2,3	1	1	0	16
2000	1	1	0	1,2	1	0	N/A	14
2001	1	0	0	1	0	N/A	N/A	3
2002	1	0	1	1,2,3	N/A	N/A	N/A	7

Table 10. Bands and *k* selected by year using a leave-one-out approach for segment-based *k*NN classifications by year

	Selected Bands and *k*: kNN Segment Based Classification							
Target Year	Target Year - 3 (1st PC)	Target Year - 2 (1st PC)	Target Year - 1 (1st PC)	Target Year (all 6 PC's)	Target Year + 1 (1st PC)	Target Year + 2 (1st PC)	Target Year + 3 (1st PC)	*k*
1992	N/A	N/A	N/A	1,3,5	0	0	0	14
1993	N/A	N/A	1	1,3,4	1	1	1	1
1994	N/A	0	0	1,3	0	0	0	8
1995	0	0	1	1,2	1	0	0	4
1996	0	0	0	1,3,6	0	1	1	13
1997	0	1	1	2,3,4	0	1	0	9
1998	0	0	0	1,3,4	0	0	0	5
1999	0	0	1	1,2,4,6	0	0	0	15
2000	0	0	0	1,2,3,6	0	1	N/A	6
2001	1	0	0	1,2,3,5,6	0	N/A	N/A	5
2002	0	0	0	1,3	N/A	N/A	N/A	7

Table 11. Bands selected and number of nodes in optimal tree using the Gini selection method and 10-fold cross validation for segment-based CART classifications by year

	Selected Bands: CART Segment-Based Classification							
Target Year	Target Year - 3 (1st PC)	Target Year - 2 (1st PC)	Target Year - 1 (1st PC)	Target Year (all 6 PC's)	Target Year + 1 (1st PC)	Target Year + 2 (1st PC)	Target Year + 3 (1st PC)	Nodes
1992	N/A	N/A	N/A	1	0	0	0	3
1993	N/A	N/A	1	1,2,3	1	1	1	14
1994	N/A	1	0	1,2	0	0	1	6
1995	1	1	0	1,2,3	1	1	0	13
1996	1	0	0	1,2,3	0	0	1	8
1997	0	0	0	1,2	0	0	0	6
1998	0	0	1	1,2,3,4	1	0	0	8
1999	1	0	0	1	0	0	0	4
2000	1	0	0	1	1	0	N/A	7
2001	1	0	0	1,3	0	N/A	N/A	5
2002	0	1	1	1,2,3,6	N/A	N/A	N/A	12

Tables 8-11 summarize the PCs for the target year and the backward and forward years selected for *k*NN and CART classifications using the leave-one-out approach and the Gini splitting rule with pruning respectively. For target years 1992, 1993, and 1994 all backward years could not be evaluated in the band selection processes and for target years 2000, 2001, and 2002 all forward years could not be evaluated in the band selection processes, as indicated by the shaded boxes in Tables 8-11. Multitemporal bands (either forward or backward years) were selected for 38 out of the 44 total classifications, with bands selected for all of the 22 pixel-based classifications and 16 out of the 22 segment-based classifications. Due to the years of imagery included in the study (1992 through 2002), not all of the desired years (3 forward and 3 backward from the target year) were able to be included for all classifications. Figures 9 and 10 illustrate the number of instances each band was selected for those classifications where all desired forward and backward years were used as inputs (20 total). The band that was selected most for both *k*NN and CART methods was the first PC of the target year (8 of 10 and 10 of 10 classifications respectively) followed by the second PC of the target year (7 of 10 and 8 of 10 classifications respectively). This is to be expected, as the first two principal components explain the majority of the variance within most multispectral images (Richards and Jia 1999) and explained over 90 percent of the variance within all images included in this study. The first principal component of the first forward and first backward years was also commonly selected (11 and 8 times out of 20 classifications respectively). The fourth, fifth, and sixth PCs were commonly selected using the *k*NN 'unpruned' leave-one-out method (13 of 20 classifications), while the CART Gini index with pruning did not commonly select those PCs (3 of 20 classifications).

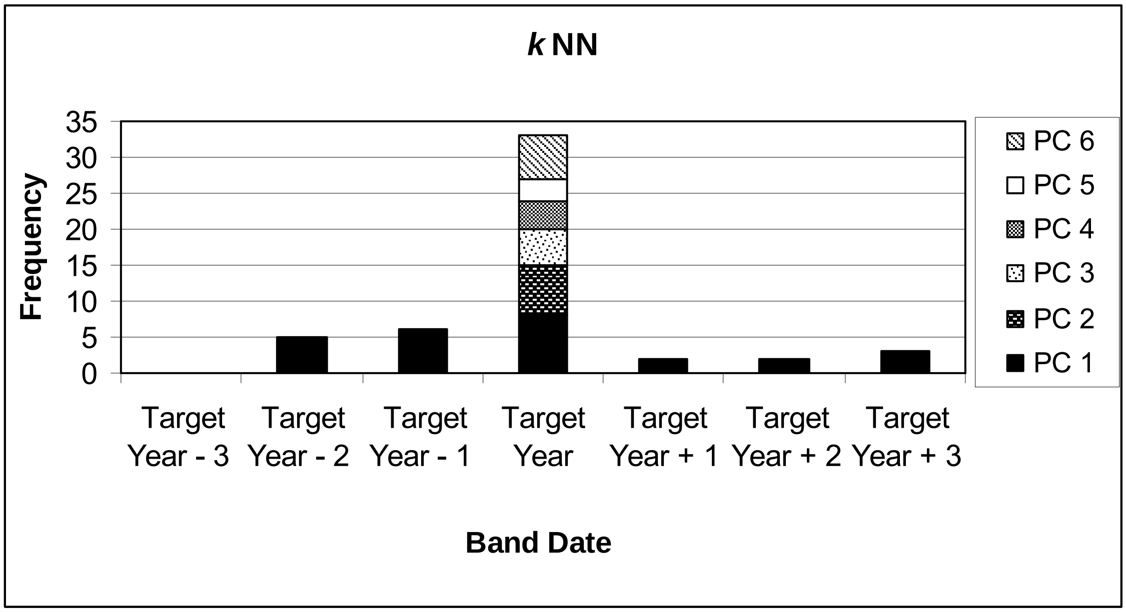

Figure 9. Frequency of bands selected for those *k*NN classifications using all possible multitemporal bands (10 total classifications)

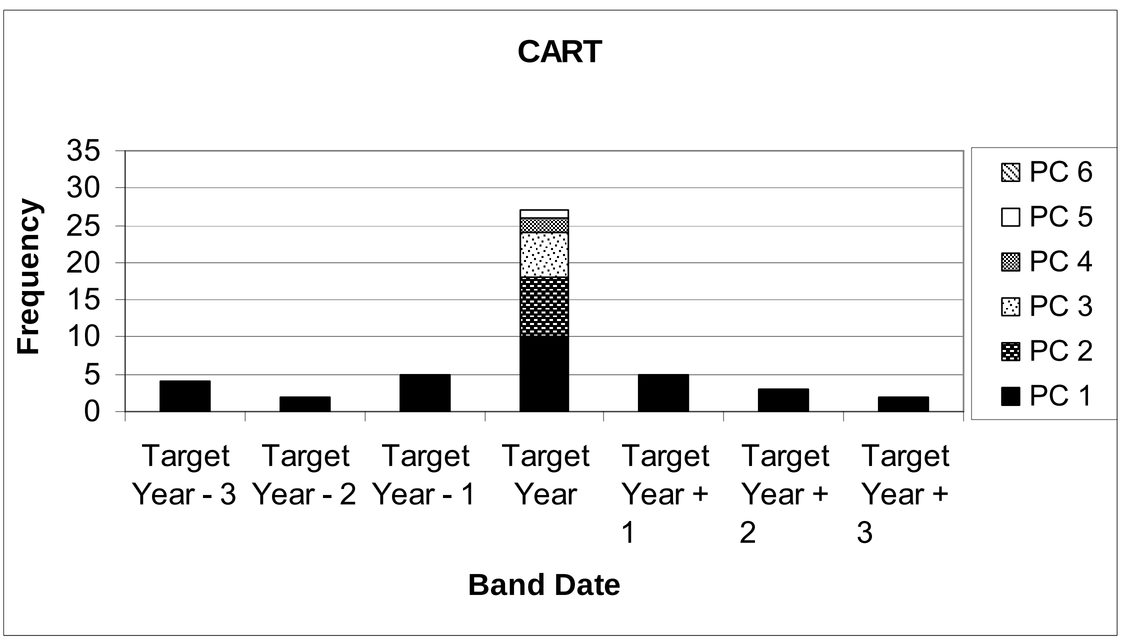

Figure 10. Frequency of bands selected for those CART classifications using all possible multitemporal bands (10 total classifications)

4. DISCUSSION

This study indicates that multitemporal classification using non-parametric *k*-nearest neighbor and classification and regression trees can accurately separate cleared, re-vegetated, and primary forest classes in the Brazilian Amazon with accuracies comparable to past studies (Roberts et al. 2002). The tested non-parametric techniques were relatively simple to implement, particularly since there was no need to meet the statistical assumptions inherent to common parametric decision rules (e.g., that spectral classes have multivariate normal distributions). The CART method is computationally simpler than the *k*NN method and therefore took less time to process.

The *k*NN and CART band selection techniques differed in that the selected CART bands were pruned using 10-fold cross validation while no pruning was involved in the leave-one-out band selection process of *k*NN. Pruning eliminates bands from the classification process that add only small, incremental increases in classification accuracy. Because no 'pruning' was incorporated into the *k*NN classification, the bands that increased classification accuracy, even if only slightly, were included in the classification. This may reveal the reason for the common selection of PCs explaining very little variance within the original image (PC bands 4, 5, and 6, explaining less than four percent of the variance of the original image when combined) for *k*NN classifications, while the CART method rarely used these bands in the optimal classification tree. For example, the 1995 pixel-based *k*NN classification included PCs 5 and 6 of the target year, while the 1995 pixel-based CART classification did not. However, these two PCs *were* included in the un-pruned 1995 pixel-based CART tree. These bands were eliminated from the optimal tree because of the low contribution these PCs made to the classification. Table 12 gives the variable importance for each of the 18 PCs used in the band-selection process for the 1995 pixel-based CART classification with the score reflecting the contribution of each PC to the classification of the data.

Table 12. Variable importance of the 1995 pixel-based CART classification

Year	Band	Score	
Target	PC 1	100	`||||||||||||||||||||||||||||||||||||||||||`
Target + 1	PC 1	87	`|||||||||||||||||||||||||||||||||||||`
Target + 2	PC 1	68	`||||||||||||||||||||||||||||`
Target + 3	PC 1	64	`|||||||||||||||||||||||||||`
Target - 1	PC 1	62	`|||||||||||||||||||||||||`
Target - 2	PC 1	46	`|||||||||||||||||||`
Target - 3	PC 1	20	`||||||||`
Target	PC 2	17	`||||||`
Target	PC 3	8	`|||`
Target	PC 6	5	`|`
Target	PC 5	5	`|`
Target	PC 4	3	

Both *k*NN and CART performed equally well in terms of overall accuracy. However, the pixel-based CART classification had significantly lower variance among the 11 annual classifications. This indicates that the CART classification was thus accurate *and* consistent, which is desirable when evaluating land cover change over several subsequent years. If the terminal years are eliminated (where no multitemporal dates were available either backward in time or forward in time), however, the results from the *k*NN classifications are more consistent (also reflected in Figure 8) indicating that the difference in variance between the *k*NN and CART methods lies primarily within these two years (1992 and 2002).

Although the CART method produced more consistent results in the pixel-based case, indicated by low variability among the 11 annual classifications, this result did not carry through to the segment-based case. Furthermore, segment-based classification was not significantly better than pixel-based classification for either the *k*NN or CART non-parametric methods.

Segment-based classification, however, has been successful and has been an improvement over pixel-based classification in many previous studies (Lobo et al. 1996, Shandley, et al. 1996) including in the Amazon region (Palubinkas et al. 1995, Lu et al. 2004). This current research differs from these previous studies in that multitemporal imagery was used in segment creation and classification. The use of interannual multitemporal imagery in the segmentation process may not be appropriate due to changes occurring in the landscape over time and the inability for segments to act as continuous units through several consecutive years. Figure 10 shows a small portion of the segmentation used in the 1995 CART classification overlaid on the 1992 Landsat TM image and the 1995 Landsat TM image. The bands used in segmentation did not include the 1992 PC; however, the 1992 PC was used in the segment-based CART classification (Tables 9 and 11). From inspection of the Landsat TM images, it is apparent that the highlighted segment was not one continuous land cover in both years.

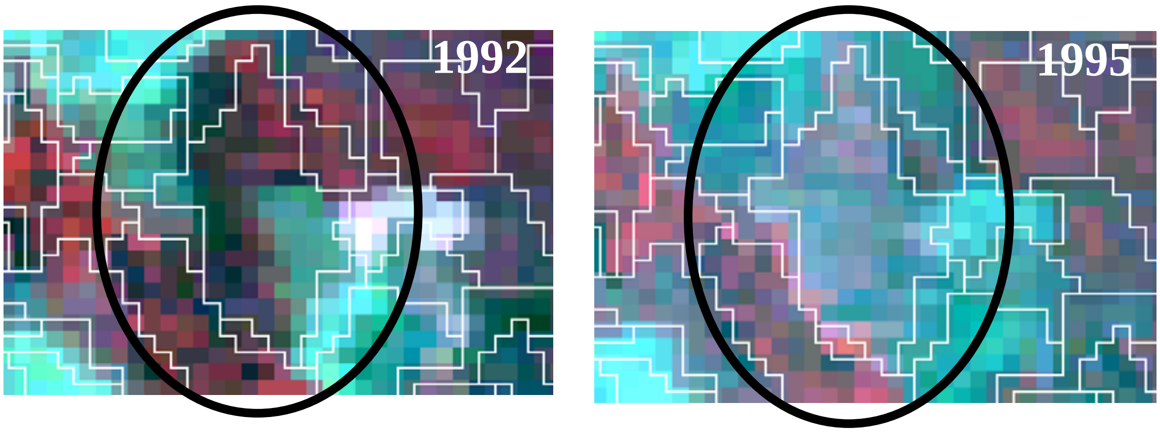

Figure 11. 1995 CART segmentation overlaid on the 1992 and 1995 Landsat TM images

All classifications accurately separated primary forest from cleared and re-vegetated classes. Differentiating primary forest from secondary growth has been difficult in past studies due to the spectral similarities in late successional secondary growth or re-vegetation to primary forest (Brondizio et al. 1996). The ability to accurately separate these two classes may be a result of the incorporation of multitemporal data in classification. Many of the CART classifications selected a backward multitemporal year to specifically separate re-vegetated areas from primary forest. Figure 11 illustrates the 2002 CART pixel-based classification and corresponding Landsat TM (1998) and Landsat ETM+ (2001) images. The areas highlighted in white were classified as re-vegetated in 2001. From inspection of the 2001 Landsat image alone, these areas are indistinguishable from surrounding primary forest. Through evaluation of the 1998 image, it is more evident that these areas had been cleared in the past. The 1998 PC was used in the 2001 CART pixel-based classification specifically to separate primary forest and re-vegetated classes (Table 9), and may explain why these areas were classified as re-vegetated rather than primary forest in this case. It is possible that the addition of more interannual multitemporal data may further increase the separability of re-vegetated and primary forest classes.

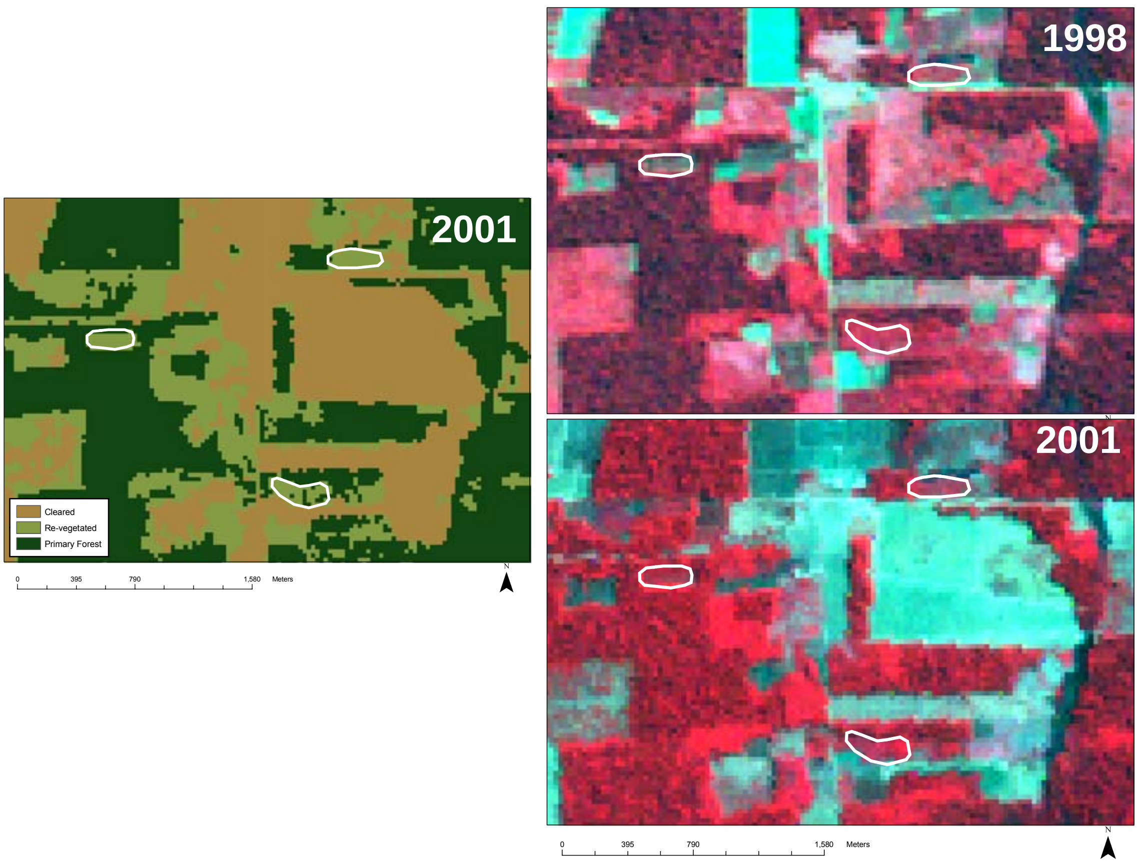

Figure 12. 2001 CART pixel-based classification and corresponding Landsat TM (1998) and Landsat ETM+ (2001) images

Additionally, interannual multitemporal data was useful in the separation of cleared, re-vegetated, and primary forest classes as indicated by Tables 8-12. Table 12 reveals the how important interannual multitemporal data were for the 1995 pixel-based

CART classification. With exception of the first PC of the target year, all forward and backward PCs were more important than all of the other target year PCs.

With the dataset used in this study, CART (pixel-based) was the preferred classifier due to its consistency and computational efficiency. Additionally, this classifier did not over-represent the cleared class (through labeling re-vegetated areas as cleared) to the extent of the *k*NN classifier. Although the differences were not significant, the increase in accuracy between these classes is valuable.

5. CONCLUSIONS

The Amazon basin remains a major hotspot of tropical deforestation (Lepers et al. 2005), presenting a clear need for timely, accurate, consistent data on land cover change. Annual classifications produced using interannual multitemporal imagery were quite accurate using both *k*NN and CART. The results are comparable to previous studies, and are of a quality that enables the use of these maps in subsequent analysis of landscape change and deforestation processes. While the non-parametric classifiers performed equally well in terms of overall accuracy, the consistency among pixel-based CART classifications, coupled with the computational efficiency of the technique, suggest that it is preferred. Additionally, slight improvements over *k*NN methods were observed (although not significant) in the separation of cleared and re-vegetated land covers.

Although segment-based classifiers have resulted in improved classifications over traditional pixel-based methods in past research in the Amazon (e.g., Palubinkas et al. 1995), this finding was not observed with the data used in this study. The ability for individual segments to act as a continuous unit through time is not always feasible at the smallest scale parameter that could be used, indicating that the use of segments in classification when also incorporating multitemporal data may be inappropriate. The creation and use of segments rather than individual pixels in classification did not significantly change overall accuracy, and was therefore an unnecessary step in the classification process.

The band-selection methods embedded in both the CART and *k*NN classifications found that PCs of additional years (supplementary to the PCs of the target year), increased accuracy for all pixel-based classifiers and for 16 of the 22 segment-based classifiers, strongly implying the utility of interannual multitemporal data in separating cleared, re-vegetated, and primary forest in the Brazilian Amazon.

There is a potential to extend these methods to other areas of Rondônia as well as other tropical regions to understand differences of land cover change at varying scales. Additionally, the utility of multitemporal bands should be evaluated further through the inclusion of more forward and backward years and also the inclusion of additional principal component bands or raw image bands. The inclusion of added data, however, stretches the computational efficiency of *k*NN when using a leave-one-out band and *k* selection approach. As such, using 10-fold cross validation is recommended for larger datasets.

The accurate, annual land cover maps produced in this study are essential to a fuller understanding of the patterns and processes of deforestation in the Amazon basin. Similar datasets have been successfully used in the Amazon to understand and predict population-environment dynamics at the household level through the incorporation of small farm holder surveys and landscape metrics (Pan et al. 2002), and in other tropical regions to analyze socioeconomic drivers of land use and land cover change at multiple scales using a geographic approach (e.g., Overmars and Verburg 2005).

LITERATURE CITED

Adams, J.B., D.E. Sabol, V. Kapos, R.A. Filho, D.A. Roberts, M.O. Smith, and A.R. Gillespie, 1995. Classification of multispectral images based on fractions of endmembers: Application to land-cover change in the Brazilian Amazon, *Remote Sensing of Environment,* 52: 137-154.

Alves, D.S., 1999. Characterizing landscape changes in central Rondônia using Landsat TM imagery, *International Journal of Remote Sensing*, 20(14): 2877-2882.

Alves, D.S., and D.L. Skole, 1996. Characterizing land cover dynamics using multi-temporal imagery, *International Journal of Remote Sensing,* 17(4): 835-839.

Alves, D.S., J.L.G. Pereira, C.L. Souza, J.V. Soares, and F. Yamaguchi, 1998. Classification of the deforested area in central Rondônia using TM imagery, *Proceedings of the Brazilian Symposium on Remote Sensing*, *Santos, Brazil, 11-18 September 1998* (Sao Jose dos Campos: Instituto Espaciais), pp. 12.

Alves, D.S., M.I.S Escada, J.L.G. Pereira, and C. De Albuquerque Linhares, 2003. Land use intensification and abandonment in Rondônia, Brazilian Amazonia, *International Journal of Remote Sensing,* 24(4): 899-903.

Antunes, A.F.B., C. Lingnau, and J.C. Da Silva, 2003. Object-oriented analysis and semantic network for high resolution image classification, *Anais XI SBSR, Belo Horizonte, Brasil, 05-10 April 2003*, INPE, p. 273-279.

Bittencourt, H.R. and R.T. Clarke, 2004. Feature selection by using classification and regression trees (CART), *Proceedings: XXth ISPRS Congress* (July 12-23, 2004, Istanbul, Turkey) Vol. XXXV: Commission 7, unpaginated CD-ROM.

Breiman, L., J.H. Friedman, R.A. Olshen, C.J. Stone, 1984. *Classification and Regression Trees,* Wadsworth, Inc., Pacific Grove, CA, 358 pp.

Brondizio, E., E. Moran, P. Mausel, and Y. Wu, 1996. Land cover in the Amazon estuary: Linking of the Thematic Mapper with botanical and historical data. *Photogrammetric Engineering and Remote Sensing,* 62(8): 921-929.

Browder, J.O., 1996. Reading colonist landscapes: Social interpretations of tropical forest patches in an Amazonian agricultural frontier, *Forest Patches in Tropical*

Landscapes, (Schelhas, J. and R. Greenberg, editors.), Island Press, Washington, pp. 285-299.

Browder, J. O., and B. J. Godfrey, 1997. *Rainforest Cities: Urbanization, Development, and Globalization of the Brazilian Amazon*, Columbia University Press, New York, 429 pp.

Campbell, J. B., 2002. *Introduction to Remote Sensing, 3rd ed.*, The Guilford Press, New York, 621 pp.

Chavez, P.S., Jr., 1989. Radiometric calibration of Landsat Thematic Mapper multispectral images, *Photogrammetric Engineering and Remote Sensing*, 55: 1285-1294.

Cohen, W.B, M. Fiorella, J. Gray, E. Helmer, and K. Anderson, 1998. An efficient and accurate method for mapping clearcuts in the Pacific Northwest using Landsat imagery, *Photogrammetric Engineering and Remote Sensing*, 64(4): 293-300.

Congalton, R.G., 1991. A review of assessing accuracy of classifications of remotely sensed data, *Remote Sensing of Environment,* 37: 35-46.

Congalton, R.G., and K. Green, 1999. Assessing the Accuracy of Remotely Sensed Data: Principles and Practices. Lewis, 137 pp.

Dai, X., and S. Khorram, 1998. The effects of image misregistration on the accuracy of remotely sensed change detection, *IEEE Transactions on Geoscience and Remote Sensing*, 36(5): 1566-1577.

De Souza, J.R.S., F.M.A Pinheiro, R.L.C. de Araujo, H.S. Pinheiro Jr., and M.G. Hodnett, 1996. Temperature and moisture profiles in soil beneath forest and pasture areas in eastern Amazonia, *Amazonian Deforestation and Climate,* (J.H.C. Gash, C.A. Nobre, J.M. Roberts and R.L. Victoria, editors) John Wiley and Sons, NY, pp. 125-137.

De Wit, A.J.W, and J.G.P.W. Clevers, 2004. Efficiency and accuracy of per-field classification for operational crop mapping, *International Journal of Remote Sensing,* 25(20): 4091-4112.

Donnely-Morrison, D.N., 1994. Defining agricultural land use in Rondônia, Brazil by examination of SPOT multispectral data, Master's Thesis, Department of Geography, Virginia Polytechnic Institute and State University, Blacksburg, VA, 69 pp.

eCognition User's Manual, 2003. Definiens Imaging (www.definiens.com).

Fearnside, P.M., 1986. *Human Carrying Capacity of the Brazilian Rainforest,* New York: Columbia University Press, 293 pp.

Fearnside, P.M., 1997. Greenhouse gases from deforestation in Brazilian Amazonia: Net committed emissions, *Climate Change*, 35(3): 321-360.

Fearnside, P.M., 1999. Biodiversity as an environmental service in Brazil's Amazonian forests: Risk, value and conservation, *Environmental Conservation,* 26(4): 305-321.

Foody, G.M., 2004. Thematic map comparison: Evaluating statistical significance of differences in classification accuracy, *Photogrammetric Engineering & Remote Sensing,* 70(5): 627-633.

Franco-Lopez, H., A.R. Ek, and M.E. Bauer, 2001. Estimation and mapping of forest stand density, volume, and cover type using the k-nearest neighbors method, *Remote Sensing of Environment,* 77: 251-274.

Friedl, M.A. and C.E. Brodley, 1997. Decision tree classification of land cover from remotely sensed data, *Remote Sensing of Environment,* 61: 399-409.

Friedman J.H., 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5): 1189-1232.

Frizzelle, B.G., S.J. Walsh, C.M. Erlien, and C.F. Mena, 2003. Collecting control data for remote sensing applications in the frontier environment of the Ecuadorian Amazon, *Earth Observation Magazine*, 12(7): 20-24.

Goodman, D. and A. Hall, eds., 1990. *The Future of Amazonia: Destruction or Sustainable Development?* New York: St. Martin's Press, Inc., 441 pp.

Grace, J., J. Lloyd, J. McIntyre, A.C. Miranda, P. Meir, and H.S. Miranda, 1996. Carbon dioxide flux over Amazonian rain forest in Rondônia, *Amazonian Deforestation and Climate*, (J.H.C. Gash, C.A. Nobre, J.M. Roberts and R.L. Victoria, editors), John Wiley and Sons, NY, pp. 319-329.

Guild, L.S., W.B. Cohen, and J.B. Kauffman, 2004. Detection of deforestation and land conversion in Rondônia, Brazil using change detection techniques, *International Journal of Remote Sensing,* 25(4): 731-750.

Haapanen, R., A.R. Ek, M.E. Bauer, and A.O. Finley, 2004. Delineation of forest/nonforest land use classes using nearest neighbor methods, *Remote Sensing of Environment,* 89: 265-271.

Hansen, M., R. Dubayah, and R. Defries, 1996. Classification trees: An alternative to traditional land cover classifiers, *International Journal of Remote Sensing,* 17(5): 1075-1081.

Hardin, P.J., 1994. Parametric and nearest-neighbor methods for hybrid classification: A comparison of pixel assignment accuracy, *Photogrammetric Engineering and Remote Sensing*, 60(12): 1439-1448.

Helmer, EH., S Brown, and W.B. Cohen, 2000. Mapping montane tropical forest successional stage and land use with multi-date Landsat imagery, *International Journal of Remote Sensing,* 21(11): 2163-2183.

Instituto Nacional de Pesquisas Espaciais (INPE), 2001. Monitoramento da Floresta Amazonica por Satelite 1999-2000. Deparata (Sao Jose dos Campos, Brazil: Instituto Nacional de Pesquisas Espaciais), 24 pp.

Ippoliti-Ramilo, G. A., J. C. N. Epiphanio, and Y. E. Shimabukuro, 2003. Landsat-5 Thematic Mapper data for pre-planting crop area evaluation in tropical countries, *International Journal of Remote Sensing,* 24(7): 1521-1534.

Katila, M., and E. Tomppo, 2001. Selecting estimation parameters for the Finnish multisource National Forest Inventory, *Remote Sensing of Environment*, 76: 16–32.

Krishnaswamy, J., M.C. Kiran, and K.N. Ganeshaiah, 2004. Tree model based ecoclimatic vegetation classification and fuzzy mapping in diverse tropical deciduous ecosystems using multi-season NDVI, *International Journal of Remote Sensing*, 25(6): 1185-1205.

Laliberte, A.S., A. Rango, K.M. Havstad, J.F. Paris, R.F. Beck, R. McNeely, and A.L. Gonzalez, 2004. Object-oriented image analysis for mapping shrub encroachment from 1937 to 2003 in southern New Mexico, *Remote Sensing of Environment*, 93: 198-210.

Lawrence, R.L., and A. Wright, 2001. Rule-based classification systems using classification and regression tree (CART) analysis, *Photogrammetric Engineering and Remote Sensing,* 67(10): 1137-1142.

Lawrence, R., A. Bunn, S. Powell, and M. Zambon, 2004. Classification of remotely sensed imagery using stochastic gradient boosting as a refinement of classification tree analysis, *Remote Sensing of Environment,* 90: 331-336.

Lo, T.H. C., F.L. Scarpace, and T.M. Lillesand, 1986. Use of multitemporal spectral profiles in agricultural land-cover classification, *Photogrammetric Engineering and Remote Sensing,* 52(4): 535-544.

Lobo, A., O. Chic, and A. Casterad, 1996. Classification of Mediterranean crops with multisensor data: Per-pixel versus per-object statistics and image segmentation, *International Journal of Remote Sensing,* 17(12): 2385-2400.

Lu, D., P. Mausel, M. Batistella, and E. Moran, 2004. Comparison of land-cover classification methods in the Brazilian Amazon Basin, *Photogrammetric Engineering and Remote Sensing,* 70(6): 723-731.

Lucas, R.M., M. Honzak, G.M. Foody, P.J. Curran, and C. Corves, 1993. Characterizing tropical secondary forests using multi-temporal Landsat sensor imagery, *International Journal of Remote Sensing*, 14(16): 3061-3067.

Lugo, A. E., 1988. Diversity of tropical species: Questions that elude answers, *Biology International: Special Issue*, 30(7): 16-20, pp. 41-45.

McCracken, S.D., A.D. Siqueira, E.F. Moran, and E.S. Brondízio, 2002. Land use patterns on an agricultural frontier in Brazil, *Deforestation and Land Use in the*

Amazon, (C.H. Wood and R. Porro, editors), University Press of Florida, Gainesville, FL, pp. 162-192.

McRoberts, R.E., M.D. Nelson, and D.G. Wendt, 2002. Stratified estimation of forest area using satellite imagery, inventory data, and the k-nearest neighbors technique, *Remote Sensing of Environment,* 82: 457-468.

Mahar, D. J., 1979. *Frontier Development Policy in Brazil: A Study of Amazonia.* Praeger, New York, pp. 182.

Moraes, J.F.L., F. Seyler, C.C. Cerri, and B. Volkoff, 1998. Land cover mapping and carbon pools estimates in Rondônia, Brazil, *International Journal of Remote Sensing*, 19(5):921-934.

Overmars, K.P., and P.H. Verburg, 2005. Analysis of land use drivers at the watershed and household level: Linking two paradigms at the Philippine forest fringe, *International Journal of Geographical Information Science* 19(2): 125-152.

Pal M., and Mather P.M., 2003. An assessment of the effectiveness of decision tree methods for land cover classification, *Remote Sensing of Environment,* 86: 554-565.

Palubinkas, G., R.M. Lucas, G.M. Foody, and P.J. Curran, 1995. An evaluation of fuzzy and texture-based classification approaches for mapping regenerating tropical forest classes from Landsat-TM data, *International Journal of Remote Sensing,* 16(4): 747-759.

Pan, W.K.Y., S.J. Walsh, R.E. Bilsborrour, B.G. Frizzelle, C.M Erlien, and F. Baquero, 2004. Farm-level models of spatial patterns of land cover dynamics in the Equadorian Amazon, *Agriculture, Ecosystems, and Environment,* 101: 117-134.

Perz, S.G., 2002. Population growth and net migration in the Brazilian legal Amazon, 1970-1996, *Deforestation and land use in the Amazon,* (C.H. Wood and R. Porro, editors), University Press of Florida, Gainsville, FL, pp. 107-129.

Perz S.G. and R.T. Walker, 2002. Household life cycles and secondary forest cover among small farm colonists in the Amazon, *World Development*, 30(6): 1009-1027.

Porro, R., 2002. Land use, cattle ranching, and the concentration of landownership in Maranhão, Brazil, *Deforestation and Land Use in the Amazon,* (C.H. Wood and R. Porro, editors), University Press of Florida, Gainesville, FL, pp. 315-337.

Powell, R.L., N. Matzke, C. de Souza, M. Clark, I. Numata, L.L. Hess, D.A. Roberts, and M. Clark, 2004. Sources of error in accuracy assessment of thematic land cover maps in the Brazilian Amazon, *Remote Sensing of Environment,* 90(2):221-234.

Richards, J. A. and X. Jia, 1999. *Remote Sensing and Digital Image Analysis: An Introduction, 3rd ed.*, Springer-Verglag, New York, 363 pp.

Roberts, D.A., I. Numata, K. Holmes, G. Batista, T. Krug, A. Monteiro, B. Powell, and O.A. Chadwick, 2002. Large area mapping of land-cover change in Rondônia using multitemporal spectral mixture analysis and decision tree classifiers, *Journal of Geophysical Research-Atmospheres*, 107(D20), 8073.

Roberts, D.A., M. Keller, and J.V. Soares, 2003. Studies of land–cover, land-use, and biophysical properties of vegetation in the Large Scale Biosphere Atmosphere experiment in Amazônia, *Remote Sensing of Environment*, 87: 377-388.

Rogan, J., J. Miller, D. Stow, J. Franklin, L. Levien, and C. Fischer, 2003. Land-cover change monitoring with classification trees using Landsat TM and ancillary data, *Photogrammetric Engineering and Remote Sensing,* 69(7): 793-804.

Sader, S.A., M. Bertrand, and E.H. Wilson, 2003. Satellite change detection of forest harvest patterns on an industrial forest landscape, *Forest Science,* 49(3): 341-353.

Schiewe, J., L. Tufte, and M. Ehlers, 2001. Potential and problems of multi-scale segmentation methods in remote sensing, *GIS,* 6: 34-39. Serpico, S.B., L. Bruzzone, and F. Roli, 1996. An experimental comparison of neural and statistical non-parametric algorithms for supervised classification of remote-sensing images, *Pattern Recognition Letters,* 17: 1331-1341.

Shandley, J., J. Franklin, and T. White, 1996. Testing the Woodcock-Harward image segmentation algorithm in an area of southern California chaparral and woodland vegetation, *International Journal of Remote Sensing,* 17(5): 983-1004.

Skole, D., and C. Tucker, 1993. Tropical deforestation and habitat fragmentation in the Amazon: satellite data from 1978 – 1988, *Science,* 260(5116): 1905-1910.

Tardin, A.T., et al., 1979. Levantamento de Areas de Desmateamento na Amazonia Legal Atrayes de Imagens de Satellite Landsat, INPE-COM3/NTE, CDU 621.38 SR, Instituto Nacional de Pesquisas Espaciais, Sao Paulo, 9 pp.

Tomppo, E., and M. Halme, 2004. Using coarse scale forest variables as ancillary information and weighting of variables in k-NN estimation: A genetic algorithm approach, *Remote Sensing of Environment,* 92: 1-20.

Van Aardt, J.A.N., 2004. An object-oriented approach to forest volume and aboveground biomass modeling using small-footprint lidar data for segmentation, estimation, and classification, PhD Dissertation. Department of Forestry, Virginia Polytechnic Institute and State University, Blacksburg, VA, pp. 344.

Van Aardt, J., and R.H. Wynne, 2004. A multi-resolution approach to forest segmentation as a precursor to estimation of volume and biomass by species, *Proceedings: ASPRS Annual Conference* (May 23-28, 2004, Denver, CO), unpaginated CD-ROM.

Wynne, R.H., R.G. Oderwald, G.A. Reams, and J.A. Scrivani, 2000. Optical remote sensing for forest area estimation, *Journal of Forestry,* 98(5): 31-36.

APPENDIX I
Example of pathway diagram used in training point creation

NU linha 40 lote 38														
Wilson Jose Pereira														
					Pronafin - R$ 1500.00 for cattle (milk)	AF Heinz abandoned - son who looked after ir left to Nova Canaa (ass.)	Daughter gets married - son-in-law moves in			Bought 42 alq in Nova Esperanca, MT				
#	Cultura	Area	92	93	94	95	96	97	98	99	2000	2001	2002	Cultura	Area
1	P1	8												P1	8
2	P2	2					5							P2	5
3	A1	2			P									P3	2
4	CP	3		A		P									
5	MV	25		22					17	10		6		MV	6
6															
7				D	A		CP							CP	3.5
8						AF								AF	0.5
9									P					P3	5
10										P 5				P4	5
11										A		P		P4	2
12												P		P4	4
13												A	CA	CA	0.4
		40													41.4
Number of shifts			0	2	2	2	1	0	1	2	0	3	1	14	
Percentage of shifts per year			0.0	14.3	14.3	14.3	7.1	0.0	7.1	14.3	0.0	21.4	7.1	100	
area shifted per year			0	6	5	3.5	3.5	0	5	7	0	6	0.4	36.4	
% of area shifted per year of tota			0.0	42.9	35.7	25.0	25.0	0.0	35.7	50.0	0.0	42.9	2.9		
% of area shifted per year of tota			0.0	103.5	86.3	60.4	60.4	0.0	86.3	120.8	0.0	103.5	6.9		

APPENDIX II
Training points where 1 represents cleared, 2 represents re-vegetated, and 3 represents primary forest

PROJECTION INFORMATION

Map Units:	meters
Projection Name:	UTM
Projection Zone:	20
Projection Parameters:	
	6378137.000000
	6356752.314250
	0.000000
	-1.000000
	0.000000
	0.000000
	0.000000
	0.000000
	0.000000
	0.000000
	0.000000
	0.000000
	0.000000
	0.000000
	0.000000
Spheroid Name:	WGS 84
Semi-Major Axis:	6378137.000000
Semi-Minor Axis:	6356752.314245
E-squred:	0.006694
Radius:	6371007.180918

APPENDIX II. All random training point locations, class label by year, and associated panel identification number (property identifier).

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
11	616860	8698534	1	1	1	1	1	1	1	1	1	1	1	163
14	625367	8710203	1	1	1	1	1	1	1	1	1	1	1	NONE
15	617381	8698113	1	1	1	1	1	1	1	1	1	1	1	176
16	625003	8713598	1	1	1	1	1	1	1	1	1	1	1	191
18	617174	8699093	1	1	1	1	1	1	1	1	1	1	1	162
27	616572	8699822	1	1	1	1	1	1	1	1	1	1	1	161
29	617196	8698442	1	1	1	1	1	1	1	1	1	1	1	163
31	624358	8713875	1	1	1	1	1	1	1	1	1	1	1	191
33	617148	8700070	1	1	1	1	1	1	1	1	1	1	1	161
40	625123	8707634	1	1	1	1	1	1	1	1	1	1	1	196
47	623996	8713845	1	1	1	1	1	1	1	1	1	1	1	191
48	616543	8698546	1	1	1	1	1	1	1	1	1	1	1	163
51	624762	8713927	1	1	1	1	1	1	1	1	1	1	1	191
55	616857	8728431	1	1	1	1	1	1	1	1	1	1	1	171
57	623957	8713583	1	1	1	1	1	1	1	1	1	1	1	191
58	624095	8714046	1	1	1	1	1	1	1	1	1	1	1	191
64	615809	8728553	1	2	1	1	1	2	2	1	2	2	2	171
66	625831	8711029	1	1	1	1	1	1	2	1	1	1	1	185
71	625974	8708448	1	1	1	1	1	1	1	1	1	1	1	183
72	626527	8710481	1	1	1	1	2	2	2	2	2	2	2	184
75	625751	8712635	1	1	1	1	1	1	1	1	1	1	1	187
77	626440	8714153	1	2	2	1	1	1	1	1	1	1	1	189
78	617160	8698474	1	1	1	1	1	1	1	1	1	1	1	163
79	616932	8698747	1	1	1	1	1	1	1	1	1	1	1	163
82	615772	8728543	1	2	1	1	2	2	2	1	2	2	2	171
83	625096	8707995	1	1	1	1	1	1	1	1	1	1	1	196
85	617124	8698934	1	1	1	1	1	1	1	1	1	1	1	162
87	625453	8712712	2	1	1	1	1	1	1	1	1	1	1	187
89	625505	8710518	1	1	1	1	1	1	2	1	1	1	2	184
97	625406	8708149	1	1	1	1	1	1	1	1	1	1	1	183
98	617466	8698139	1	1	1	1	1	1	2	1	1	1	1	176
103	625726	8710522	1	1	2	1	2	1	2	2	2	1	2	184
106	624300	8713722	1	1	1	1	1	1	1	1	1	1	1	191
113	625298	8714045	1	1	1	1	1	1	1	1	1	1	1	191
118	624401	8713780	1	1	1	1	1	1	1	1	1	1	1	191

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
119	625449	8712606	2	1	1	1	1	1	1	1	1	1	1	187
120	625222	8713937	1	1	1	1	1	1	2	1	1	1	1	191
122	626315	8708549	1	1	2	1	2	2	1	1	1	2	1	183
124	617094	8698632	1	1	1	1	1	1	1	1	1	1	1	163
130	625070	8713644	1	1	1	1	1	1	1	1	1	1	1	191
132	617033	8696196	1	1	1	1	1	1	1	1	1	1	1	168
138	616897	8728392	1	1	1	1	1	1	1	1	1	1	1	171
142	616918	8698575	1	1	1	1	1	1	1	1	1	1	1	163
143	624391	8713672	1	1	1	1	1	1	1	1	1	1	1	191
153	617406	8701169	1	1	1	1	1	1	1	1	1	1	1	180.2
155	625937	8712582	1	1	1	1	1	1	1	1	1	1	1	187
157	624210	8713860	1	1	1	1	1	1	1	1	1	1	1	191
160	615568	8728489	1	2	1	1	2	2	2	1	2	2	2	171
165	626857	8714464	1	1	1	1	1	1	1	1	1	1	1	189
168	615669	8728555	1	2	1	1	2	2	2	1	2	2	2	171
169	625416	8712781	2	1	1	1	1	1	1	1	1	1	1	187
173	626406	8708545	1	1	1	1	1	1	1	1	1	1	1	183
175	625383	8708319	2	1	1	1	1	1	1	1	1	2	2	183
195	615668	8699841	1	1	1	1	1	1	1	1	1	1	1	161
202	625250	8713839	2	1	1	1	1	1	1	1	1	1	1	191
208	625513	8708420	1	1	1	1	1	1	1	1	1	2	1	183
211	616576	8700036	1	1	1	1	1	1	1	1	1	1	1	161
216	618627	8699374	1	1	1	1	1	1	1	1	1	1	1	179.1
220	624756	8714006	2	1	1	1	1	1	1	1	1	1	1	191
222	626152	8710586	1	1	1	1	1	1	1	1	1	1	1	184
223	626077	8708136	1	1	1	1	2	2	2	1	2	2	2	183
226	617386	8701474	1	1	1	1	1	1	1	1	1	1	1	181.1
227	625408	8714455	1	1	1	1	2	1	1	2	2	2	1	189
228	625074	8707694	1	1	1	1	1	1	1	1	1	1	1	196
229	624273	8714004	1	1	1	1	1	1	1	1	1	1	1	191
234	617217	8698393	1	1	1	1	1	1	1	1	1	1	1	163
235	624984	8713967	1	1	1	1	1	1	1	1	1	1	1	191
239	627190	8714177	1	1	1	1	1	1	1	1	1	1	1	189
241	616707	8700083	1	1	1	1	1	1	1	1	1	1	1	161
242	626750	8714124	1	1	2	2	1	2	2	1	2	2	2	189
244	624397	8707684	1	1	1	1	1	1	1	1	1	1	1	196

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
246	626542	8714220	1	1	1	1	1	1	1	1	1	1	1	189
247	626060	8708287	1	1	1	1	1	1	1	1	1	1	1	183
248	625089	8707866	1	1	1	1	1	1	1	1	1	1	1	196
253	623936	8707993	1	1	1	1	1	1	1	1	1	1	2	196
256	616936	8699109	1	1	1	1	1	1	1	1	1	1	1	162
259	616128	8696113	1	1	1	1	1	1	1	1	1	1	1	168
261	624030	8713812	1	1	1	1	1	1	1	1	1	1	1	191
264	624053	8707967	1	1	1	1	1	1	1	1	1	1	1	196
272	542450	8794023	1	1	1	1	1	1	1	1	1	1	1	137.2
274	547375	8800063	1	1	1	1	1	1	2	1	1	1	1	100
285	538491	8797777	1	1	1	1	1	1	1	1	1	1	1	NONE
287	547327	8800486	1	1	1	1	1	1	1	1	1	1	1	104
288	541807	8799632	1	1	1	1	1	1	1	1	1	1	1	120
290	547849	8801121	1	1	1	1	2	1	1	1	2	2	2	104
292	539178	8798522	1	1	1	1	1	1	1	1	1	1	1	158
296	549548	8797538	1	1	1	1	1	1	1	1	1	1	1	96
297	538398	8797723	1	1	2	2	1	2	2	1	1	1	1	NONE
306	543169	8794615	1	2	2	1	1	1	1	1	1	1	1	138.1
307	542350	8800157	1	1	1	1	1	1	1	1	1	1	1	121
308	546062	8801120	1	1	1	1	1	1	1	1	1	1	1	108
314	544141	8799975	1	2	2	2	2	2	1	1	1	1	1	113
316	545641	8800960	1	1	1	1	1	1	1	1	1	1	1	108
322	541769	8802776	1	1	1	1	1	1	1	1	1	1	1	131
323	543195	8794907	1	2	2	1	2	2	2	1	2	1	2	138.1
325	542268	8797951	1	1	1	1	1	1	1	1	1	1	1	110
335	550410	8797696	1	2	2	2	2	2	2	2	2	2	2	94
339	547299	8800192	1	1	1	1	1	1	1	1	1	1	1	102
340	541814	8795664	1	1	1	2	1	1	1	1	1	1	1	142.2
342	542730	8800491	1	1	1	1	1	1	1	1	1	1	1	121
344	543813	8799438	1	1	1	1	1	1	1	1	1	1	1	113
346	545538	8801020	1	1	1	1	1	1	1	1	1	1	1	108
348	540804	8797021	1	1	1	1	1	2	2	2	2	1	1	152
349	549357	8797306	1	1	1	1	1	1	1	1	1	1	1	93
350	542397	8800401	1	1	1	1	1	1	1	1	1	1	1	121
351	548028	8802114	1	2	2	2	1	1	1	1	1	1	1	107
353	538920	8798232	1	1	1	1	1	1	1	1	1	1	1	158

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
356	541972	8802953	1	1	1	1	1	1	1	1	1	1	1	131
357	541429	8800149	1	1	1	1	1	1	1	1	1	1	1	122
362	546333	8799634	1	1	1	1	1	2	2	2	2	2	2	102
363	542256	8793559	1	1	1	1	2	2	2	1	1	1	1	137.1
365	542189	8797731	1	1	1	1	1	1	1	1	1	1	1	110
367	540135	8800850	1	1	1	1	1	1	1	1	1	1	1	128
368	540106	8800836	1	1	1	1	1	1	1	1	1	1	1	128
370	541770	8793129	1	1	2	1	1	1	1	2	1	1	1	137.1
373	542254	8797618	1	1	2	1	1	1	1	1	1	1	1	110
375	541914	8795632	1	1	1	1	1	1	1	1	1	1	1	142.2
376	543334	8798723	1	1	1	1	1	1	1	1	1	1	1	110
377	547844	8800932	1	2	2	2	2	2	2	2	2	2	2	104
383	549598	8797765	1	1	1	1	1	1	1	1	1	2	1	96
384	543830	8799514	1	1	1	1	1	1	1	1	1	1	1	113
388	549701	8797298	1	2	2	2	2	2	2	2	2	2	2	93
389	542103	8797744	1	1	1	1	1	1	1	1	1	1	1	110
390	553510	8798970	1	1	1	1	1	1	1	1	2	1	1	81
391	542633	8800344	1	1	1	1	1	1	1	1	1	1	1	121
394	541609	8799504	1	2	2	2	1	1	1	1	1	1	1	120
402	540427	8799671	1	1	1	1	1	1	1	1	1	1	1	122
404	549734	8797093	1	1	1	2	2	1	1	1	1	1	1	93
406	544961	8800477	1	1	2	2	2	1	1	1	1	1	1	108
408	537716	8798120	1	1	1	1	1	1	1	1	1	1	1	NONE
410	543592	8799517	1	1	1	1	1	1	1	1	1	1	1	113
412	550222	8797565	1	1	1	1	1	1	1	1	2	2	1	94
418	547242	8800319	1	1	1	1	1	1	1	1	1	1	1	102
419	550527	8797822	1	2	1	1	1	1	1	1	2	2	1	94
425	541126	8794721	1	1	1	1	1	1	1	1	1	1	1	141
426	547975	8802160	1	2	2	2	2	1	1	1	2	2	1	107
427	540658	8799687	1	1	1	1	1	1	1	1	1	1	1	122
432	542030	8802942	1	1	1	1	1	1	1	1	1	1	1	131
435	544385	8800256	1	2	2	2	2	1	1	1	1	1	1	113
437	542433	8797978	1	1	1	1	1	1	1	1	1	1	1	110
440	543385	8798934	1	1	1	1	1	1	1	1	1	1	1	113
442	539074	8798497	1	1	1	1	1	1	1	1	1	1	1	158
450	545390	8800881	1	1	1	1	1	1	1	1	1	1	1	108

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
451	546883	8800042	1	2	2	2	2	.	1	.	1	1	1	102
453	550482	8798070	1	1	1	1	1	1	1	1	1	1	1	94
456	540320	8799584	1	1	1	2	1	1	1	1	1	1	1	122
463	540884	8797122	1	1	1	2	1	1	1	1	2	1	1	152
464	543252	8799130	1	1	1	1	1	1	1	1	1	1	1	113
467	540545	8801013	1	1	1	1	1	1	1	1	1	1	1	128
468	552971	8798725	1	2	2	2	2	2	2	2	2	2	2	81
471	547950	8800774	1	2	2	2	2	2	2	2	2	2	2	104
478	547586	8801787	1	2	1	1	2	1	1	1	1	1	1	107
480	553580	8799276	1	2	2	2	1	1	1	2	2	1	2	81
484	545614	8801208	1	1	1	1	1	1	1	1	1	1	1	108
485	546132	8801401	1	2	2	2	2	1	1	1	1	1	1	108
489	550435	8797370	1	2	2	2	2	2	2	2	2	1	2	92.2
492	541225	8799767	1	1	1	1	1	1	1	1	1	1	1	122
493	543763	8799586	1	1	1	1	1	1	1	1	1	1	1	113
498	543338	8799068	1	1	1	1	1	2	1	1	1	1	1	113
499	552980	8798695	1	2	2	2	2	2	2	2	2	2	2	81
500	547085	8799684	1	2	2	2	2	2	2	2	2	2	2	100
501	547088	8800119	1	1	1	1	1	1	1	2	2	2	2	102
502	543870	8799269	1	1	1	1	1	1	1	1	1	1	1	113
504	549742	8797183	1	2	1	2	2	2	2	2	2	2	2	93
506	553676	8799150	1	1	1	1	1	2	2	1	2	1	2	81
510	538924	8798195	1	1	1	1	1	1	1	1	1	1	1	158
511	538103	8798233	1	1	1	1	1	1	1	1	1	1	2	NONE
512	544189	8799701	1	1	1	1	1	1	1	1	1	1	1	113
513	550683	8797480	1	1	1	1	1	1	1	1	2	2	1	92.2
520	547137	8800701	1	1	1	1	1	1	1	1	1	1	1	106.1
521	541749	8802956	1	2	2	2	1	2	2	1	1	1	1	131
524	540950	8799858	1	1	1	1	1	1	1	1	1	1	1	122
525	539917	8796399	1	1	1	1	1	2	2	1	1	1	1	150
529	542909	8794672	1	1	1	1	1	1	1	1	1	1	1	138.1
538	540933	8800050	1	1	1	1	1	1	1	1	1	1	1	122
539	547681	8800888	1	1	2	1	1	1	1	1	1	1	1	104
540	549673	8797532	1	1	1	1	2	2	2	1	2	2	2	93
542	542663	8794235	1	1	1	1	1	1	1	1	1	1	1	138.1
549	549539	8797038	1	1	1	2	1	1	2	1	1	1	1	93

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
550	537715	8796888	1	2	2	2	2	2	2	2	1	1	1	NONE
553	553596	8798919	1	1	1	1	2	2	2	2	2	2	2	81
564	550279	8798025	1	1	1	1	1	1	1	1	1	1	1	94
567	547609	8800633	1	1	1	2	2	1	1	1	1	1	1	104
568	539459	8795863	1	1	1	1	1	1	1	2	1	1	2	150
571	549981	8797390	1	1	1	1	1	1	1	1	1	1	1	93
572	543521	8798329	1	1	1	1	1	1	1	1	1	1	1	110
574	541033	8799110	1	2	1	2	2	2	2	2	2	2	2	120
577	541746	8802795	1	1	1	2	1	1	1	1	1	1	1	131
578	542763	8800686	1	1	1	1	1	1	1	1	1	1	1	121
581	547012	8799715	1	2	2	2	2	2	2	2	2	2	2	100
586	542626	8800712	1	1	1	1	1	1	1	1	1	1	1	121
589	541652	8802547	1	2	2	2	2	2	2	2	1	1	1	131
594	553296	8799186	1	2	2	2	2	2	2	2	2	2	2	81
598	542738	8800346	1	1	1	1	1	1	2	1	1	1	1	121
601	545864	8801254	1	1	1	1	1	1	1	1	1	1	1	108
602	545656	8801241	1	1	1	2	2	1	1	1	1	1	1	108
604	550834	8797640	1	2	2	2	2	2	2	2	2	2	2	92.2
606	537936	8797193	1	2	2	2	2	2	2	2	1	1	1	NONE
611	541936	8795767	1	2	1	1	1	1	1	1	1	1	1	142.2
613	541649	8802582	1	2	2	1	2	2	2	2	1	1	1	131
614	542313	8797891	1	1	1	1	1	1	1	1	1	1	1	110
619	542362	8797727	1	1	1	1	1	1	1	1	1	1	1	110
624	540053	8800714	1	1	1	1	1	1	1	1	1	1	1	128
626	553424	8799019	1	1	1	1	1	1	1	1	2	1	2	81
628	541598	8800048	1	1	1	1	1	1	1	1	1	1	1	122
629	547201	8800962	1	1	1	1	1	1	1	1	1	1	1	106.1
635	539879	8796323	1	1	1	1	1	1	1	1	1	1	1	150
636	546966	8799645	1	2	2	2	2	2	2	2	.	.	.	100
640	541043	8799209	1	2	2	1	2	2	2	2	2	2	2	120
644	541415	8800345	1	1	1	1	1	1	1	1	1	1	1	122
648	542563	8797678	1	2	1	1	1	1	1	1	1	2	2	110
650	542882	8800354	1	1	1	1	1	1	1	1	1	1	1	121
651	541103	8799794	1	1	1	1	1	1	1	1	1	1	1	122
654	553638	8799064	1	1	1	1	1	2	2	1	2	2	2	81
656	537759	8798104	1	1	1	1	1	1	1	1	1	1	1	NONE

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
657	545764	8801198	1	1	1	1	1	1	1	1	1	1	1	108
658	540493	8802022	1	2	2	1	1	1	1	1	1	1	1	131
659	541235	8799687	2	1	2	2	2	2	2	2	1	1	1	122
664	544909	8800200	1	1	2	1	1	1	2	1	1	2	2	108
665	540316	8799499	1	1	1	1	1	1	1	1	1	1	1	122
666	549492	8797502	1	1	1	1	1	1	1	1	1	1	1	96
674	539142	8798148	1	1	1	1	1	1	1	1	2	2	2	158
679	553460	8798809	1	1	1	1	1	2	2	2	2	2	2	81
684	542185	8800095	1	1	1	1	1	1	1	1	1	1	1	121
686	546970	8799692	1	2	2	2	2	2	2	2	2	2	2	100
688	537945	8798091	1	1	1	1	1	1	1	1	1	2	2	NONE
689	541980	8802867	1	1	1	1	1	1	1	1	1	1	1	131
691	541703	8795516	1	2	2	2	2	2	2	1	1	1	1	142.2
694	541831	8795535	1	2	1	1	1	1	1	1	1	1	1	142.2
697	542356	8800293	1	1	1	1	1	1	1	1	1	1	1	121
701	543962	8799559	1	1	1	1	1	1	1	1	1	1	1	113
702	542372	8794172	1	1	1	1	1	1	1	1	1	1	1	138.1
703	540957	8794974	1	1	1	1	1	1	1	1	1	1	1	NONE
705	545361	8801082	1	1	1	2	2	1	1	2	2	2	1	108
707	542711	8800513	1	1	1	2	1	1	1	1	1	1	1	121
708	539059	8798225	1	1	2	1	2	2	2	1	1	1	1	158
710	538883	8797938	1	1	1	1	1	1	1	1	1	1	1	158
711	541002	8794638	1	1	1	1	1	1	1	1	1	1	1	141
712	553758	8799173	1	1	1	1	1	2	2	1	1	1	1	81
713	547043	8799666	1	.	2	2	2	2	2	2	2	2	2	100
716	539120	8798644	1	1	1	1	1	1	1	1	1	1	1	158
717	547049	8801051	2	1	1	1	1	1	1	1	1	1	1	106.2
721	542854	8800500	1	1	1	1	1	1	1	1	1	1	1	121
722	549755	8797807	1	1	2	1	1	1	1	1	1	1	1	96
723	537976	8798362	1	1	1	1	1	1	1	1	2	2	1	NONE
724	545990	8801376	1	2	1	1	1	1	1	1	1	1	1	108
725	549833	8797347	1	1	1	2	2	2	2	2	2	2	2	93
728	540309	8794424	1	2	2	2	1	1	1	1	1	1	1	NONE
733	549751	8797563	1	1	1	1	1	2	1	1	2	2	2	93
736	540603	8799782	1	1	1	1	1	1	1	1	1	1	1	122
740	540404	8802028	1	2	2	1	1	1	2	1	1	1	2	131

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
743	537781	8797051	1	1	1	1	1	2	2	1	1	1	1	NONE
745	544373	8800226	1	2	2	2	2	1	1	1	1	1	1	113
746	541849	8795666	1	1	1	2	1	1	1	1	1	1	1	142.2
747	543300	8800396	1	2	2	2	2	2	2	2	2	2	2	119
750	544148	8799952	1	2	2	2	2	2	1	1	1	1	1	113
751	546881	8799840	1	2	1	1	1	1	1	2	2	2	2	102
756	540765	8794662	1	2	2	1	1	1	1	1	1	1	1	141
759	542531	8797881	1	2	2	2	2	2	2	2	2	2	2	110
764	547624	8801798	1	2	1	1	2	1	1	1	1	1	1	107
768	547500	8801841	1	2	2	2	2	1	1	1	1	1	1	107
770	547677	8801864	1	2	1	1	2	1	1	1	1	1	1	107
772	541427	8799434	1	2	2	2	1	1	2	2	1	1	1	120
779	546913	8800093	1	2	2	2	2	1	1	1	1	1	1	102
826	549752	8797593	1	1	1	1	1	2	1	1	2	2	2	93
827	543858	8799566	1	1	1	1	1	1	1	1	1	1	1	113
828	542683	8799960	1	1	1	1	1	1	1	1	1	1	1	119
829	542479	8793668	1	2	2	2	2	1	1	1	1	1	1	107
830	541974	8799798	1	1	1	1	1	1	1	1	1	1	1	120
831	540024	8796438	1	1	1	1	1	1	1	1	1	1	1	150
832	542711	8799898	1	1	1	1	1	1	1	1	1	1	1	119
833	546280	8798089	1	1	1	1	1	1	1	1	1	1	1	109
834	545543	8801239	1	1	1	1	1	1	1	1	1	1	1	108
835	543464	8798952	1	1	1	1	1	1	1	1	1	1	1	113
836	543394	8799102	1	1	1	1	1	1	1	1	1	1	1	113
837	538933	8798323	1	1	1	1	1	1	1	1	1	1	1	158
838	545339	8797822	1	1	1	1	1	1	1	1	1	1	1	NONE
839	541606	8799760	1	2	2	1	1	1	2	1	1	1	2	120
840	549656	8797733	1	2	1	1	1	1	1	1	1	2	1	96
841	549612	8797417	1	1	1	2	2	2	2	1	2	2	2	93
842	544921	8797499	1	1	1	1	1	1	1	1	1	1	1	NONE
843	542999	8800128	1	1	1	1	1	1	1	1	1	1	1	119
844	545567	8801061	1	1	1	1	1	1	1	1	1	1	1	108
845	545248	8797303	1	1	1	1	1	1	1	1	1	1	1	NONE
846	543509	8798990	1	1	1	1	1	1	1	1	1	1	1	113
847	543233	8798209	1	1	1	1	1	1	1	1	1	1	1	110
848	542282	8797950	1	1	1	1	1	1	1	1	1	1	1	110

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
849	542581	8798061	1	1	1	1	1	1	1	1	1	1	1	110
850	543178	8798343	1	1	1	1	1	1	1	1	1	1	1	110
851	542280	8797793	1	1	1	1	1	1	1	1	1	1	1	110
853	542354	8797925	1	1	1	1	1	1	1	1	1	1	1	110
858	549288	8797404	1	1	1	1	1	1	1	1	2	2	2	96
1	626153	8708162	2	1	2	1	1	1	1	1	1	1	1	183
26	624068	8707614	2	2	2	1	1	2	1	1	1	1	2	196
28	625539	8710948	2	2	2	2	2	2	2	2	2	2	2	185
35	618808	8699308	2	2	2	2	2	1	2	2	2	2	2	179.1
36	624874	8713643	2	2	1	1	1	1	1	1	1	1	1	191
38	625518	8712332	2	2	2	2	2	2	2	2	2	2	2	186
39	625434	8708528	2	2	1	1	1	1	1	1	1	1	1	183
42	626169	8712309	2	2	2	2	2	2	2	2	2	2	2	186
45	616782	8699115	2	2	2	1	1	1	1	1	1	1	1	162
46	625936	8710503	2	2	2	1	1	1	2	2	2	2	2	184
49	616827	8727857	2	2	2	2	2	2	2	2	1	2	2	172
50	617450	8699360	2	2	2	2	2	1	2	2	2	2	2	179.1
56	624117	8707828	2	2	2	2	1	1	1	1	1	1	1	196
59	616969	8696092	2	2	2	2	1	1	2	1	1	1	1	168
60	616877	8696073	2	2	2	2	1	1	2	2	1	1	1	168
74	625458	8710860	2	2	2	2	2	2	2	2	2	2	2	185
81	617000	8696103	2	2	2	2	1	1	2	1	1	1	1	168
84	616509	8700127	2	2	2	1	2	1	2	2	1	1	1	161
88	623941	8707859	2	2	2	1	2	2	2	1	1	1	1	196
96	625769	8710409	2	2	2	1	1	2	2	2	2	2	2	184
100	617989	8697882	2	2	1	1	1	1	1	1	1	1	1	175
104	616605	8727756	2	2	2	2	2	2	2	2	2	2	2	172
105	617426	8701420	2	1	1	1	1	1	1	1	1	1	1	181.1
135	616218	8698842	2	1	2	1	1	1	1	1	1	1	1	162
136	617695	8697748	2	2	2	2	2	1	1	2	1	1	1	175
158	617411	8701351	2	1	1	1	2	1	1	1	1	1	1	181.1
162	626768	8714177	2	1	1	1	2	1	2	2	2	2	2	189
167	617480	8698219	2	1	1	1	2	1	1	1	1	1	1	176
170	623989	8707653	2	2	2	1	1	1	1	1	1	1	2	196
174	617377	8697735	2	2	2	2	2	1	2	2	1	1	1	175
177	624763	8707492	2	2	2	1	1	1	1	1	1	1	1	196

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
192	618277	8699318	2	2	2	2	2	2	2	1	1	1	1	179.1
199	616008	8728540	2	2	2	1	2	2	2	2	2	2	2	171
201	618090	8698053	2	2	2	2	1	2	1	1	1	1	1	175
203	618155	8697724	2	2	2	2	1	2	1	1	1	1	2	175
204	623698	8714018	2	2	2	1	1	1	1	1	1	1	1	191
206	617839	8697961	2	2	2	1	2	1	1	1	1	1	1	175
207	615649	8699148	2	2	2	1	2	2	1	1	1	1	1	162
210	617504	8697935	2	2	2	1	1	1	1	1	1	1	1	175
218	617633	8699311	2	2	2	1	1	1	1	1	1	1	1	179.1
219	623825	8713807	2	2	2	1	1	1	1	1	1	1	1	191
230	617534	8701426	2	2	1	1	1	1	1	1	1	1	1	181.1
232	616651	8698793	2	1	2	1	1	1	1	1	1	1	1	162
243	623687	8713603	2	2	1	1	1	1	1	1	1	1	1	191
252	616865	8696036	2	2	2	2	2	1	2	2	1	1	1	168
258	617152	8698278	2	2	2	2	1	1	1	1	1	1	1	163
260	617682	8701194	2	2	2	1	1	1	1	1	1	1	1	180.2
262	618874	8699373	2	2	1	1	1	1	1	1	1	1	1	179.1
265	617606	8697897	2	2	2	1	1	1	1	1	1	1	1	175
267	626534	8708471	2	2	1	1	2	1	2	1	1	1	1	183
269	625804	8708511	2	1	2	1	1	2	2	1	1	1	1	183
271	547461	8801154	2	2	2	2	2	2	2	2	2	2	2	106.1
275	548691	8796916	2	2	2	1	2	1	1	1	1	1	1	96
284	541009	8797175	2	2	1	2	2	2	2	1	2	2	1	152
289	540787	8802127	2	2	2	1	1	1	1	1	1	1	2	131
293	544032	8799916	2	2	2	2	2	2	2	2	1	1	1	113
294	542975	8798376	2	2	2	2	2	2	2	2	2	2	2	110
298	547533	8801736	2	2	2	2	2	1	2	2	2	2	2	107
302	548801	8797163	2	2	2	1	2	2	1	1	1	1	1	96
310	543037	8798057	2	2	2	2	2	1	2	1	1	1	1	110
311	541571	8799396	2	2	2	2	1	1	1	1	1	1	1	120
321	548087	8802124	2	2	2	2	1	1	1	1	1	1	1	107
332	542990	8794738	2	2	2	2	2	2	2	2	2	2	2	138.1
333	547949	8801958	2	2	2	1	1	1	1	2	1	1	1	107
337	551468	8798678	2	2	2	2	2	2	2	2	2	2	2	94
341	537519	8798009	2	2	2	2	2	2	2	2	2	2	2	NONE
343	542423	8794238	2	2	2	2	2	2	1	1	1	1	2	138.1

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
347	551518	8798655	2	2	2	2	2	2	2	2	2	2	2	94
358	539967	8800670	2	2	2	2	2	1	1	1	1	1	2	128
360	543055	8794570	2	2	2	1	1	1	2	2	1	1	2	138.1
361	540921	8794488	2	2	2	1	1	2	1	1	1	1	1	141
374	543876	8799750	2	2	2	2	2	2	2	2	2	2	2	113
379	548375	8796856	2	2	2	1	1	1	1	1	1	1	1	96
381	546731	8801235	2	2	2	2	2	2	2	2	2	2	2	107
385	542928	8794399	2	2	2	1	1	1	1	1	1	1	1	138.1
387	551547	8798516	2	2	2	2	2	2	2	2	2	2	2	94
393	542750	8794314	2	2	1	2	2	2	2	1	1	1	1	138.1
395	545233	8800964	2	2	2	2	2	1	1	2	2	2	1	108
398	547921	8801956	2	2	2	1	1	1	1	1	1	1	1	107
399	547533	8800413	2	2	2	2	2	1	1	1	1	1	1	104
407	540977	8799586	2	2	2	2	2	2	2	2	1	1	1	122
413	553031	8798772	2	2	2	1	2	2	2	2	2	2	2	81
415	548511	8796952	2	2	2	1	2	2	1	1	1	1	1	96
417	550943	8797778	2	2	2	2	1	1	1	1	2	2	2	92.2
421	551297	8798733	2	2	1	2	1	2	2	2	1	2	2	94
422	541080	8797246	2	2	1	2	2	1	2	1	2	1	2	152
430	541394	8799315	2	2	2	2	2	2	2	2	2	1	1	120
447	549127	8796830	2	2	2	2	2	2	2	2	2	2	2	93
449	549699	8797763	2	2	2	1	1	1	1	1	1	1	1	96
458	545202	8800776	2	2	2	2	2	2	2	2	2	2	2	108
459	546664	8801218	2	2	2	2	2	2	2	2	2	2	2	107
465	547447	8800518	2	2	2	2	2	2	2	2	2	2	2	104
470	551523	8798584	2	2	2	2	2	2	2	2	2	2	2	94
473	551339	8798835	2	2	1	2	2	2	2	2	2	2	2	94
475	546873	8801464	2	2	2	2	2	2	2	2	2	2	2	107
486	553210	8799012	2	2	2	2	2	2	2	2	2	2	2	81
487	539824	8796149	2	1	1	1	1	1	1	1	1	1	2	150
488	542643	8794477	2	2	2	2	2	2	2	2	1	2	2	138.1
507	549161	8797038	2	2	2	2	2	2	2	2	2	2	2	93
508	548582	8796836	2	2	2	1	1	1	1	1	1	1	1	96
515	540514	8794202	2	2	2	2	2	1	1	1	2	2	2	141
518	540538	8801882	2	2	2	2	2	1	2	2	1	1	1	131
519	549158	8797479	2	2	2	1	2	1	1	1	1	1	1	96

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
523	543011	8797982	2	2	2	2	2	1	2	1	1	1	1	110
532	548453	8796754	2	2	2	1	2	2	1	1	1	1	1	96
541	546613	8799877	2	2	2	2	2	2	2	1	1	1	1	102
544	551053	8797850	2	2	2	2	2	1	2	2	2	2	2	92.2
547	541397	8799277	2	2	2	2	2	2	2	2	2	1	1	120
551	547364	8801759	2	2	2	2	2	2	2	2	2	2	2	107
561	543839	8799787	2	2	2	2	2	2	2	2	2	2	2	113
562	538750	8798199	2	2	1	1	1	1	1	1	2	2	2	158
563	539850	8796102	2	2	2	1	2	1	2	2	2	2	2	150
566	548494	8796745	2	2	2	1	2	2	1	1	1	1	1	96
569	542863	8798120	2	2	2	2	2	2	2	2	2	2	2	110
583	541265	8799353	2	2	2	2	2	2	2	2	2	1	1	120
585	547006	8801625	2	2	2	2	2	2	2	2	2	2	2	107
592	546156	8799514	2	2	2	2	2	2	1	2	2	2	2	102
603	538824	8798195	2	2	1	1	1	2	1	1	2	2	2	158
609	549772	8797337	2	2	2	2	2	2	2	2	2	2	2	93
610	538879	8798103	2	2	1	1	1	1	1	2	1	1	1	158
615	548967	8797313	2	2	2	1	1	1	2	1	2	2	2	96
618	542862	8798251	2	2	2	2	2	2	2	2	2	2	2	110
623	536969	8797501	2	2	2	1	1	1	1	2	2	2	2	NONE
627	542985	8794456	2	2	2	1	1	1	1	1	1	1	1	138.1
632	547714	8801023	2	2	2	1	1	2	2	2	1	1	2	104
638	546186	8801449	2	2	2	2	2	2	2	2	2	2	2	108
641	551714	8798638	2	2	2	2	2	2	2	2	2	2	2	94
645	553549	8799358	2	2	2	2	2	2	2	1	2	1	2	81
653	549278	8797455	2	2	2	1	1	1	1	1	2	2	2	96
668	537669	8797864	2	2	1	2	2	1	1	1	1	1	1	NONE
669	540617	8794692	2	2	2	2	2	1	1	1	1	2	2	NONE
670	551414	8798912	2	2	2	2	2	2	2	2	2	2	2	94
676	542735	8797987	2	2	2	2	2	2	2	2	2	2	2	110
678	547600	8801267	2	2	2	2	2	2	2	2	2	2	2	106.1
692	549327	8797577	2	2	2	2	2	1	1	1	1	1	1	96
693	546709	8799917	2	2	2	2	2	2	1	1	1	1	1	102
699	537663	8796930	2	2	2	2	2	2	2	2	1	1	1	NONE
700	547860	8802018	2	2	2	1	1	1	1	1	1	1	1	107
709	541915	8803082	2	2	1	1	1	2	2	1	1	1	1	131

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
718	540894	8794647	2	1	1	1	1	1	1	1	1	1	1	141
726	538118	8798349	2	2	2	2	2	2	2	2	2	2	2	NONE
727	542881	8798394	2	2	2	2	2	2	2	2	2	2	2	110
738	547817	8801910	2	2	2	1	1	1	1	1	1	1	1	107
749	543234	8794832	2	2	2	1	2	2	2	1	2	2	2	138.1
753	547296	8801805	2	2	2	2	2	2	2	2	2	2	2	107
754	546932	8801627	2	2	2	2	2	2	2	2	2	2	2	107
755	549691	8797187	2	2	2	2	2	2	2	2	2	2	2	93
757	542679	8798056	2	2	2	2	2	2	2	2	2	2	2	110
758	542650	8797953	2	2	2	2	2	2	2	2	2	2	2	110
760	543914	8799836	2	2	2	2	2	2	2	2	2	2	2	113
762	547907	8802333	2	1	2	2	1	1	1	1	1	1	1	107
763	546687	8801352	2	2	2	2	2	2	2	2	2	2	2	107
765	547296	8801805	2	2	2	2	2	2	2	2	2	2	2	107
766	545736	8800947	2	2	2	1	1	1	1	1	1	1	1	108
767	542648	8794420	2	2	1	2	2	2	1	1	1	1	1	138.1
771	541642	8799663	2	2	2	2	2	1	2	1	1	1	1	120
773	547952	8802272	2	1	2	2	1	1	1	1	1	1	1	107
774	549667	8797641	2	2	2	2	1	1	1	1	1	1	1	96
775	546651	8801435	2	2	2	2	2	2	2	2	2	2	2	107
776	546785	8801142	2	2	2	2	2	2	2	2	2	2	2	107
777	543444	8800754	2	2	2	2	2	2	2	2	2	2	2	119
778	546515	8801377	2	2	2	2	2	2	2	2	2	2	2	107
780	548809	8797089	2	2	1	1	2	2	1	1	1	1	1	96
854	545202	8800742	2	2	2	2	2	2	2	2	2	2	2	108
855	545169	8800773	2	2	2	2	2	2	2	2	2	2	2	108
856	540496	8794187	2	2	2	2	2	1	1	1	2	2	2	141
857	547427	8800526	2	2	2	2	2	2	2	2	2	2	2	104
859	546193	8801505	2	2	2	2	2	2	2	2	2	2	2	108
860	545268	8800882	2	2	2	2	2	1	1	2	2	2	1	108
861	543020	8794711	2	2	2	2	2	2	2	2	2	2	2	138.1
863	542976	8798059	2	2	2	2	2	1	2	2	2	2	2	110
0	616258	8728340	3	2	2	2	2	2	2	2	2	2	2	171
2	618528	8698141	3	3	3	1	1	2	2	1	1	1	1	176
3	627284	8712359	3	3	3	3	3	3	3	3	3	3	3	186
4	616372	8695810	3	3	3	3	3	3	3	3	3	3	3	168

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
5	615544	8727842	3	3	3	3	3	3	3	3	3	3	3	172
6	626872	8708529	3	3	3	1	1	1	1	1	1	1	1	183
7	616234	8698517	3	3	3	1	1	1	1	1	1	1	1	163
8	627039	8710897	3	3	3	3	3	3	3	3	3	3	3	185
9	618408	8701675	3	3	3	3	3	3	3	3	3	3	3	181.2
10	626541	8708312	3	3	3	3	3	1	1	1	1	1	1	183
12	615696	8700215	3	3	3	1	1	1	1	1	1	1	1	161
13	625896	8712494	3	3	3	3	3	3	1	2	1	1	1	186
17	616697	8728386	3	2	2	2	1	2	2	1	1	1	2	171
19	626671	8708528	3	3	1	1	1	1	1	1	1	1	1	183
20	616616	8695880	3	3	3	3	3	3	3	3	3	3	3	168
21	626672	8710572	3	3	3	3	3	3	3	3	3	3	3	184
22	616211	8695943	3	3	3	3	3	3	3	3	3	3	3	168
23	618779	8701495	3	3	3	3	3	3	3	3	3	3	3	181.2
24	624679	8713652	3	1	2	1	1	1	1	1	1	1	1	191
25	618024	8701312	3	3	3	1	2	2	2	1	1	1	1	181.1
30	627218	8708148	3	3	3	3	3	3	3	3	1	1	1	183
32	626872	8708262	3	3	3	3	3	3	3	1	1	1	1	183
34	615983	8699138	3	3	3	3	3	2	1	1	1	1	1	162
37	627081	8711837	3	3	3	3	3	3	3	3	3	3	3	186
41	615643	8727692	3	3	3	3	3	3	3	3	3	3	3	172
43	615660	8698867	3	3	3	3	3	3	3	3	3	3	2	162
44	617860	8701405	3	3	3	3	3	3	3	1	1	1	2	181.1
52	615784	8698713	3	3	3	3	3	3	3	3	3	3	3	163
53	615801	8727927	3	3	3	3	3	3	3	3	3	3	3	172
54	617892	8701277	3	3	3	1	2	2	1	1	1	1	1	181.1
61	618292	8701488	3	3	3	3	3	3	3	3	3	3	3	181.1
62	626486	8710944	3	3	3	3	3	3	3	1	1	1	1	185
63	616155	8727941	3	3	3	3	3	3	3	3	3	3	3	172
65	618753	8697711	3	3	3	3	3	3	3	3	1	1	2	175
67	615810	8727955	3	3	3	3	3	3	3	3	3	3	3	172
68	619164	8701693	3	3	3	3	3	3	3	3	3	3	3	181.2
70	615628	8698296	3	3	3	3	3	3	1	2	1	1	1	163
73	627246	8712561	3	3	3	3	3	3	3	3	3	3	3	187
76	615600	8728344	3	3	3	3	3	3	3	3	3	3	3	171
80	626363	8710885	3	3	3	3	3	3	3	1	1	1	1	185

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
86	618685	8701617	3	3	3	3	3	3	3	3	3	3	3	181.2
90	616235	8727830	3	3	3	3	3	3	3	3	3	3	3	172
91	627138	8711899	3	3	3	3	3	3	3	3	3	3	3	186
92	618924	8697733	3	3	3	3	3	3	3	3	1	1	2	175
93	616201	8699192	3	3	3	3	3	1	1	1	1	1	1	162
94	616424	8728539	3	3	2	2	2	2	2	1	1	1	2	171
95	615729	8698443	3	3	3	3	3	3	1	2	1	1	2	163
99	615189	8727726	3	3	3	3	3	3	3	3	3	3	3	172
102	618941	8701455	3	3	3	3	3	3	3	3	3	3	3	181.2
107	616430	8727694	3	3	3	3	3	3	3	3	3	3	3	172
108	615319	8698408	3	3	3	3	1	1	2	1	1	1	1	163
110	615419	8695928	3	3	3	3	3	3	3	2	2	2	2	168
111	627217	8708401	3	3	3	1	1	1	1	1	1	1	1	183
112	615379	8698268	3	3	3	3	3	1	2	1	1	1	1	163
114	616369	8698364	3	3	1	1	1	1	1	1	1	1	1	163
115	615517	8698292	3	3	3	3	3	3	1	2	1	1	1	163
116	615671	8698289	3	3	3	3	3	1	2	1	1	1	1	163
117	625914	8712800	3	3	2	1	1	1	1	1	1	1	1	187
121	616271	8698425	3	3	2	1	1	1	1	1	1	1	1	163
123	618018	8701181	3	3	1	1	1	1	1	1	1	1	1	180.2
125	615270	8728553	3	3	3	3	3	3	3	3	2	2	2	171
126	623354	8707846	3	3	3	3	3	2	2	1	1	1	1	196
127	616497	8698919	3	3	1	1	1	1	1	1	1	1	1	162
129	623645	8707951	3	3	2	1	1	1	1	1	1	1	2	196
131	615542	8727929	3	3	3	3	3	3	3	3	3	3	3	172
133	626677	8714483	3	1	1	1	1	1	1	1	1	1	1	189
134	615713	8698843	3	3	3	3	3	3	3	3	3	3	2	162
137	627032	8708244	3	3	3	3	3	3	3	3	1	1	1	183
139	615372	8698637	3	3	3	3	2	1	2	1	1	1	1	163
140	615592	8698446	3	3	3	3	3	3	1	1	1	1	2	163
141	623668	8707970	3	3	2	1	1	1	1	1	1	1	2	196
144	616435	8698767	3	3	1	1	1	1	1	1	1	1	1	162
145	625904	8708179	3	3	3	3	3	3	3	1	1	1	1	183
147	616082	8727904	3	3	3	3	3	3	3	3	3	3	3	172
148	623715	8707989	3	3	3	1	1	1	1	1	1	1	2	196
149	617975	8701190	3	3	3	1	1	1	1	1	1	1	1	180.2

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
150	623785	8707793	3	2	2	1	1	1	1	1	1	1	2	196
152	615309	8727826	3	3	3	3	3	3	3	3	3	3	3	172
154	625943	8712648	3	3	2	1	1	1	1	1	1	1	1	187
156	615730	8699103	3	3	3	3	3	2	2	1	1	1	1	162
159	616398	8698817	3	3	1	1	1	1	1	1	1	1	1	162
161	626212	8711892	3	3	3	3	3	3	3	3	3	3	3	186
163	616134	8727668	3	3	3	3	3	3	3	3	3	3	3	172
164	625870	8712401	3	3	3	3	1	2	2	2	2	2	2	186
166	627191	8710505	3	3	3	3	3	3	3	3	3	3	3	184
171	615553	8695791	3	3	3	3	3	1	2	2	2	2	2	168
176	618012	8701541	3	3	3	3	3	3	3	3	3	3	3	181.2
178	626428	8712756	3	3	2	1	1	1	1	1	1	1	1	187
179	616525	8728300	3	3	3	3	3	3	3	1	2	2	2	171
180	616455	8698263	3	3	1	1	1	1	1	1	1	1	1	163
181	616238	8728528	3	3	1	1	1	1	1	1	2	2	2	171
182	626551	8708196	3	3	3	3	3	1	1	1	1	1	1	183
183	626373	8711830	3	3	3	3	3	3	3	3	3	3	3	186
185	615641	8727774	3	3	3	3	3	3	3	3	3	3	3	172
186	617798	8701351	3	3	3	1	1	1	2	1	1	1	1	181.1
187	617655	8701612	3	3	3	3	1	1	1	1	1	1	1	181.2
188	616180	8727923	3	3	3	3	3	3	3	3	3	3	3	172
189	615634	8727681	3	3	3	3	3	3	3	3	3	3	3	172
190	616371	8698466	3	3	3	1	1	1	1	1	1	1	1	163
191	623395	8707630	3	3	3	3	3	3	2	1	1	1	1	196
193	615407	8728300	3	3	3	3	3	3	3	3	3	3	3	171
194	626083	8710421	3	3	3	3	3	3	3	3	3	3	3	184
196	615976	8698463	3	3	3	3	3	3	1	1	1	2	2	163
197	615438	8700029	3	3	3	3	3	2	2	1	1	1	1	161
198	616345	8727849	3	3	3	1	2	2	2	2	2	2	2	172
200	616317	8727699	3	3	3	3	3	3	3	3	3	3	3	172
205	618937	8697814	3	3	3	3	3	3	3	3	1	1	2	175
209	618571	8698049	3	3	3	1	1	2	2	2	2	2	2	176
212	615705	8698588	3	3	3	3	3	3	3	3	3	3	3	163
214	615995	8700161	3	3	1	2	1	2	1	1	1	1	1	161
215	616682	8696015	3	3	3	3	3	3	3	3	3	3	3	168
217	618818	8698062	3	3	3	3	3	3	3	3	2	2	2	176

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
221	623543	8707682	3	3	3	3	1	1	1	1	1	1	1	196
224	626232	8711078	3	3	3	3	3	3	3	1	1	1	1	185
225	615955	8695887	3	3	3	3	3	3	3	3	1	1	1	168
231	626234	8710915	3	3	3	3	3	3	3	1	1	1	1	185
233	626597	8711097	3	3	3	3	3	3	3	3	3	3	3	185
236	615401	8699183	3	3	3	3	3	3	3	3	3	3	3	162
237	626247	8711953	3	3	3	3	3	3	3	3	3	3	3	186
238	618448	8701172	3	3	3	3	3	3	3	3	3	3	3	180.2
240	626307	8708315	3	3	3	1	1	1	1	1	1	1	1	183
245	626591	8714366	3	3	1	1	1	1	1	1	1	1	1	189
249	623406	8707499	3	3	3	3	3	3	2	1	1	1	2	196
250	615504	8698774	3	3	3	3	3	3	3	3	3	3	3	162
251	617683	8701522	3	3	3	3	3	2	1	1	1	1	1	181.2
254	618637	8701293	3	3	3	3	3	3	3	3	3	3	3	181.1
255	626177	8711887	3	3	3	3	3	3	3	3	3	3	3	186
257	626268	8712481	3	3	3	1	1	2	2	1	1	1	1	186
263	626720	8712345	3	3	3	3	3	3	3	3	3	3	3	186
266	626708	8708187	3	3	3	3	3	3	1	1	1	1	1	183
268	626074	8712547	3	3	2	1	2	1	1	1	1	1	1	187
270	619050	8699317	3	3	2	1	2	2	2	2	2	2	2	179.1
273	540747	8799437	3	3	3	3	3	3	3	3	3	2	2	122
276	540541	8799339	3	3	3	3	3	3	3	3	3	1	2	122
277	539110	8795752	3	3	3	1	1	1	1	1	1	1	1	150
278	540236	8794149	3	3	3	3	3	3	3	3	1	1	1	141
279	537953	8797330	3	3	1	2	2	2	2	1	1	1	1	NONE
281	539965	8799112	3	3	3	3	3	3	3	3	3	3	3	158
282	543142	8800659	3	3	3	3	3	3	3	3	3	1	2	121
283	543305	8801330	3	3	3	3	3	3	2	1	1	1	1	121
286	541272	8797418	3	1	2	2	1	1	2	2	2	2	2	152
291	540214	8796441	3	3	3	3	3	3	3	3	3	3	3	152
295	544589	8800278	3	3	1	2	1	1	1	1	1	1	1	113
299	541013	8792928	3	3	3	3	3	3	3	3	3	3	3	137.2
300	544196	8800052	3	1	1	1	1	1	1	1	1	1	1	113
301	542332	8796021	3	3	3	1	1	1	1	1	1	1	1	142.2
303	541203	8792841	3	3	3	3	3	3	3	3	3	3	3	137.1
304	544431	8799944	3	3	3	2	1	1	1	1	1	1	1	113

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
305	541320	8797322	3	1	2	2	1	1	2	1	1	2	2	152
309	542063	8793807	3	3	3	3	3	3	1	2	1	2	1	137.2
312	548192	8800883	3	3	3	3	3	3	3	3	3	3	3	104
313	544696	8800417	3	1	1	1	1	1	1	1	1	1	1	108
315	541291	8802341	3	3	3	3	3	3	3	3	3	3	3	131
317	539634	8798710	3	2	2	2	2	2	2	2	1	1	1	158
318	540638	8796879	3	3	3	3	3	3	3	3	3	3	3	152
319	542725	8796465	3	3	3	3	3	3	3	1	1	1	2	142.2
320	552607	8798625	3	3	3	3	3	3	3	3	3	3	3	81
324	542013	8797509	3	3	3	3	3	3	3	3	1	1	1	110
326	537342	8797623	3	3	3	3	3	3	3	3	2	1	2	NONE
328	543175	8800659	3	3	3	3	3	3	3	3	3	1	2	121
329	543017	8800861	3	3	3	3	3	3	3	3	3	3	3	121
330	536738	8797306	3	3	3	1	2	2	2	2	2	2	2	NONE
331	539607	8799050	3	3	3	3	3	3	3	1	1	1	1	158
336	541601	8792985	3	3	3	3	3	3	3	3	3	3	3	137.1
338	541147	8799442	3	3	3	3	1	2	2	1	1	1	1	120
345	540214	8794091	3	3	3	3	3	3	3	3	1	1	1	141
352	542062	8797494	3	3	3	3	3	3	3	3	1	1	1	110
354	549080	8797096	3	1	2	2	2	1	1	1	2	2	2	93
355	546869	8801224	3	3	3	3	3	3	3	3	3	3	3	107
359	543128	8800929	3	3	3	3	3	3	3	3	3	3	3	121
364	537256	8796764	3	3	3	3	3	3	3	3	3	3	3	NONE
366	540661	8799531	3	3	3	3	3	3	3	3	3	2	2	122
369	540655	8799274	3	3	3	3	3	3	3	3	3	2	2	122
371	543355	8800901	3	3	3	3	3	3	3	3	3	3	2	121
372	541335	8793358	3	3	3	2	2	2	2	2	1	1	1	137.2
378	541496	8793111	3	3	3	3	3	3	3	3	3	3	3	137.1
380	541412	8793136	3	3	3	3	3	3	3	3	3	3	3	137.1
382	545854	8799181	3	3	3	3	3	3	3	3	3	3	3	102
386	541245	8793079	3	3	3	3	3	3	3	3	3	3	3	137.2
392	540657	8799467	3	3	3	3	3	3	3	3	3	2	2	122
396	548459	8796499	3	3	3	3	3	3	3	3	3	3	3	93
397	539974	8799022	3	3	3	3	3	3	3	3	3	3	3	158
400	548737	8796603	3	3	3	3	3	3	3	1	1	2	2	93
401	540379	8799184	3	3	3	3	3	3	3	3	3	2	2	122

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
403	540655	8796739	3	3	3	3	3	3	3	3	3	3	3	152
405	541456	8793460	3	3	3	2	1	2	1	1	1	1	1	137.2
409	540564	8798931	3	3	3	3	3	3	3	3	3	3	3	120
411	543021	8800529	3	1	1	1	1	2	2	2	2	2	2	121
414	545154	8800377	3	1	1	1	1	1	1	1	1	1	1	108
416	542098	8797365	3	3	3	3	3	3	3	3	1	1	1	110
420	548542	8796457	3	3	3	3	3	3	3	3	3	3	3	93
423	541856	8793753	3	3	3	1	1	1	1	1	1	1	1	137.2
424	540934	8802200	3	3	3	3	3	3	3	3	1	1	2	131
428	540929	8799157	3	3	3	1	1	1	2	2	2	2	2	120
429	540834	8798899	3	3	3	3	3	3	3	3	3	3	3	120
431	541121	8797379	3	.	2	2	1	1	2	2	2	2	2	152
433	543659	8795250	3	3	3	3	3	3	3	3	3	3	3	138.1
434	546982	8801410	3	3	3	3	3	3	3	3	3	3	3	107
436	540909	8793035	3	3	3	1	1	1	1	1	2	2	2	137.2
438	538689	8798079	3	3	1	1	1	1	1	1	2	2	2	158
439	543504	8801063	3	3	3	3	3	3	3	3	3	3	2	121
443	546395	8799179	3	3	3	3	3	3	3	3	3	3	3	100
444	542357	8793815	3	3	3	3	3	3	2	2	2	2	2	137.1
445	537292	8797710	3	3	3	3	3	3	3	3	3	2	2	NONE
448	548053	8801129	3	3	3	3	3	3	3	3	3	3	3	104
452	543701	8795294	3	3	3	3	3	3	3	3	3	3	3	138.1
454	542370	8796128	3	3	3	1	1	1	1	1	1	2	2	142.2
455	541313	8802431	3	3	3	3	3	3	3	3	3	3	3	131
457	541311	8797377	3	1	2	2	1	1	2	2	2	2	2	152
460	548871	8796629	3	3	3	1	1	2	1	2	2	2	2	93
461	542201	8793939	3	3	3	3	3	3	2	1	1	1	1	137.2
466	543314	8801169	3	3	3	3	3	3	3	3	3	2	2	121
469	552907	8798835	3	3	3	3	3	3	3	3	3	3	3	81
474	543337	8801155	3	3	3	3	3	3	3	3	3	2	2	121
476	547208	8801513	3	3	3	3	3	3	3	3	3	3	3	107
477	541468	8802571	3	3	3	3	3	3	3	2	1	1	1	131
479	541600	8793014	3	3	3	3	3	3	3	3	3	3	3	137.1
481	553149	8798517	3	3	3	3	3	3	3	3	3	3	3	81
482	542765	8796350	3	3	3	3	3	3	3	1	1	1	2	142.2
483	546963	8801266	3	3	3	3	3	3	3	3	3	3	3	107

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
490	542218	8793778	3	3	3	3	3	3	2	2	2	2	2	137.1
491	550683	8798073	3	3	3	3	3	3	1	1	1	2	2	94
494	544503	8800327	3	3	1	1	2	1	1	1	1	1	1	113
495	543431	8794930	3	3	3	3	3	3	3	3	3	1	2	138.1
496	548405	8801858	3	3	3	3	3	3	3	3	3	3	3	106.1
497	540954	8802391	3	3	3	3	3	3	3	3	3	3	3	131
503	548755	8797015	3	3	1	2	1	1	1	1	1	1	1	96
505	546614	8799398	3	1	.	2	2	2	2	2	2	2	1	100
509	542138	8793420	3	3	3	3	3	3	3	1	1	1	2	137.1
514	543596	8795128	3	3	3	3	3	3	3	3	3	2	2	138.1
516	553145	8798670	3	3	3	3	3	3	3	3	3	3	3	81
522	541128	8792938	3	3	3	3	3	3	3	3	3	3	3	137.2
526	540393	8796701	3	3	3	3	3	3	3	3	3	3	3	152
527	546100	8799021	3	3	3	3	3	3	3	3	3	3	3	100
528	543564	8801040	3	3	3	3	3	3	3	3	3	3	2	121
530	544472	8799861	3	3	3	2	1	1	1	1	1	1	1	113
531	540620	8798952	3	3	3	3	3	3	3	3	3	3	3	120
533	537196	8797661	3	3	3	3	3	3	3	3	3	1	1	NONE
535	542091	8797419	3	3	3	3	3	3	3	3	1	1	1	110
536	550755	8798040	3	3	3	3	3	3	3	3	3	3	2	94
537	543642	8795224	3	3	3	3	3	3	3	3	3	3	3	138.1
543	536930	8797238	3	3	3	3	3	3	3	3	3	3	3	NONE
545	540171	8799028	3	3	3	3	3	3	3	3	3	3	3	158
546	548185	8801859	3	3	3	3	3	3	3	3	3	3	3	106.2
548	547705	8801575	3	3	3	1	1	2	2	2	1	1	1	106.2
552	543826	8795406	3	3	3	3	3	3	3	3	3	3	3	138.1
554	545828	8801354	3	3	1	1	1	1	1	1	1	1	1	108
555	539399	8798825	3	3	3	3	3	3	3	3	1	1	1	158
556	541369	8802517	3	3	3	3	3	3	3	2	1	1	1	131
557	548240	8801887	3	3	3	3	3	3	3	3	3	3	3	106.2
558	548361	8802053	3	3	3	3	3	3	3	3	3	3	3	106.2
559	548454	8801739	3	3	3	3	3	2	2	2	2	2	2	106.1
560	542512	8796217	3	3	3	1	1	1	2	1	1	2	2	142.2
565	544704	8800586	3	1	1	1	1	1	1	1	1	1	1	108
570	543349	8800955	3	3	3	3	3	3	3	3	3	2	2	121
573	539995	8794085	3	3	3	3	3	2	2	2	2	2	2	none

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
576	543532	8801448	3	3	3	3	3	3	1	1	1	1	1	121
579	540439	8796849	3	3	3	3	3	3	3	3	3	3	3	152
582	541575	8802617	3	3	3	1	2	2	2	2	1	1	1	131
584	539846	8798996	3	3	3	3	3	3	3	3	3	3	3	158
587	541046	8802188	3	3	3	3	3	3	3	3	1	1	2	131
588	540917	8802093	3	3	3	3	3	3	3	3	1	1	1	131
590	543272	8800709	3	3	3	3	3	3	3	3	3	3	2	121
591	540013	8794060	3	3	3	3	3	3	3	3	3	2	1	141
593	549182	8797167	3	1	2	2	2	1	2	1	2	2	2	93
595	543011	8800617	3	1	1	1	2	2	2	2	2	2	2	121
596	540804	8799287	3	3	3	1	1	1	2	2	2	2	2	120
597	548048	8801771	3	3	3	3	3	3	3	3	3	3	3	106.2
599	543497	8801380	3	3	3	3	3	3	3	1	1	1	1	121
600	540172	8799510	3	3	3	3	3	3	3	3	3	1	1	122
605	541017	8802569	3	3	3	3	3	3	3	3	3	3	3	131
607	543506	8795007	3	3	3	3	3	3	3	3	3	2	2	138.1
608	552740	8798522	3	3	3	3	3	3	3	3	3	3	3	81
612	550762	8797946	3	3	1	2	2	2	2	1	2	2	2	94
616	540997	8802137	3	3	3	3	3	3	3	3	1	1	2	131
617	540804	8802301	3	3	3	3	3	3	3	3	1	1	2	131
621	544315	8800027	3	1	1	1	1	1	1	1	1	1	1	113
622	548913	8796443	3	3	3	3	3	3	3	3	3	3	3	93
625	538574	8798162	3	3	1	2	1	1	1	1	1	1	1	158
630	540920	8799316	3	3	3	1	1	1	2	2	2	2	2	120
631	540214	8799290	3	3	3	3	3	3	3	3	3	1	1	122
633	541785	8793345	3	3	3	3	3	3	1	1	2	2	2	137.1
634	541144	8792728	3	3	3	3	3	3	3	3	3	3	3	137.1
637	541927	8793823	3	3	3	1	1	1	1	1	1	1	1	137.2
639	546218	8799085	3	3	3	3	3	3	3	3	3	3	3	100
642	536935	8797329	3	3	3	3	3	3	3	3	3	3	3	NONE
643	541347	8802674	3	3	3	3	3	3	3	2	1	1	1	131?
646	543251	8800922	3	3	3	3	3	3	3	3	3	3	3	121
647	541114	8792607	3	3	3	3	3	3	3	3	3	3	3	137.1
649	541569	8792996	3	3	3	3	3	3	3	3	3	3	3	137.1
652	539572	8798680	3	2	2	2	2	2	2	2	1	1	2	158
655	543687	8799175	3	3	3	3	3	3	3	3	3	3	3	113

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
660	541565	8793569	3	3	3	2	1	2	1	1	1	1	1	137.2
661	551239	8798498	3	3	3	2	1	1	1	1	2	2	2	94
662	543309	8798441	3	3	3	3	3	3	3	3	3	3	3	110
663	540519	8794390	3	1	1	2	1	1	1	1	1	1	1	141
667	539554	8798674	3	2	2	2	2	2	2	2	1	1	2	158
671	543717	8795319	3	3	3	3	3	3	3	3	3	3	3	138.1
672	540898	8802067	3	3	3	3	3	3	3	3	1	1	2	131
673	540678	8801714	3	3	3	3	3	3	3	3	3	3	1	131
675	552684	8798651	3	3	3	3	3	3	3	3	3	3	3	81
677	539290	8795674	3	3	3	1	2	2	2	2	1	2	2	150
680	545035	8800714	3	3	3	3	3	3	3	2	2	1	2	108
681	540100	8799020	3	3	3	3	3	3	3	3	3	3	3	158
682	543257	8800613	3	3	3	3	3	3	3	2	2	2	2	119
683	540643	8798840	3	3	3	3	3	3	3	3	3	3	3	120
685	539767	8798846	3	3	3	3	3	3	1	1	1	1	1	158
687	540152	8798938	3	3	3	3	3	3	3	3	3	3	3	158
690	542977	8800771	3	3	3	3	3	3	3	3	3	3	3	121
695	553051	8798640	3	3	3	3	3	3	3	3	3	3	3	81
698	543586	8801037	3	3	3	3	3	3	3	3	3	3	2	121
704	547709	8801678	3	3	3	1	1	1	2	2	1	1	1	106.2
706	540672	8794326	3	2	2	2	1	1	1	1	1	1	1	141
715	547061	8801192	3	3	2	1	2	2	2	1	1	1	1	106.2
719	541960	8793476	3	3	3	3	3	3	1	1	2	2	2	137.1
720	541444	8797652	3	3	3	3	3	3	3	3	3	3	3	152
729	548248	8802133	3	3	3	3	3	3	3	3	3	3	3	107
730	548319	8801548	3	3	3	3	3	3	3	3	3	3	3	106.1
732	552802	8798735	3	3	3	3	3	3	3	3	3	3	3	81
734	539691	8798788	3	2	2	2	2	2	2	1	1	1	1	158
735	541364	8802637	3	3	3	3	3	3	3	2	1	1	1	131
737	539160	8800492	3	3	3	3	3	3	3	3	3	3	3	127
739	541313	8793319	3	3	3	2	2	2	2	1	1	1	1	137.2
741	541419	8797580	3	3	3	3	3	3	3	3	3	3	3	152
742	537120	8797611	3	3	3	3	3	3	3	3	3	1	1	NONE
744	541703	8793419	3	1	2	2	2	2	2	1	1	2	2	137.2
761	546146	8798302	3	3	3	3	1	2	2	2	1	2	2	109
781	543527	8795059	3	3	3	3	3	3	3	3	3	2	2	138.1

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
782	543106	8794598	3	3	1	1	1	1	2	2	2	2	2	138.1
783	541962	8793801	3	3	3	1	1	1	1	1	1	1	1	137.2
784	540708	8799034	3	3	3	3	3	3	3	3	3	3	3	120
785	541541	8799592	3	3	3	3	1	2	2	2	2	1	1	120
786	540136	8794091	3	3	3	3	3	3	3	3	3	1	1	141
788	548246	8802157	3	3	3	3	3	3	3	3	3	3	3	107
789	546138	8799052	3	3	3	3	3	3	3	3	3	3	3	100
790	541132	8793014	3	3	3	3	3	3	3	3	3	3	3	137.2
791	541021	8793144	3	3	3	2	1	1	1	1	2	2	2	137.2
792	546795	8801321	3	3	3	3	3	3	3	3	3	3	3	107
793	543489	8795211	3	3	3	3	3	3	3	3	3	2	2	138.1
794	548721	8796992	3	3	1	2	1	1	1	1	1	1	1	96
795	540033	8793840	3	3	3	3	3	3	3	3	3	3	3	141
796	543688	8795183	3	3	3	3	3	3	3	3	3	3	3	138.1
797	540026	8799211	3	3	3	3	3	3	3	3	3	3	3	158
798	541459	8792896	3	3	3	1	1	1	2	2	2	2	2	137.1
799	544476	8800290	3	3	1	1	2	1	1	1	1	1	1	113
800	540563	8796881	3	3	3	3	3	3	3	3	3	3	3	152
801	539218	8795806	3	3	3	1	1	2	1	1	1	1	1	150
802	548712	8796678	3	3	3	2	2	2	2	1	1	2	2	93
803	539775	8798921	3	3	3	3	3	3	2	1	1	1	1	158
804	541423	8793394	3	3	3	2	1	2	1	1	1	1	1	137.2
805	540010	8799088	3	3	3	3	3	3	3	3	3	3	3	158
806	544483	8800399	3	3	2	2	2	1	1	1	1	1	1	113
807	544999	8800660	3	3	3	3	3	3	3	2	1	1	1	108
808	537891	8797306	3	3	1	2	2	2	2	1	1	1	1	NONE
809	544905	8800648	3	3	3	3	3	3	3	1	1	1	1	108
810	548624	8796313	3	3	3	3	3	3	3	3	3	3	3	93
811	537412	8796699	3	3	3	3	3	3	3	3	3	3	3	NONE
812	540021	8793889	3	3	3	3	3	3	3	3	3	3	3	141
813	542061	8793469	3	3	3	3	3	3	3	1	1	1	2	137.1
814	540746	8799024	3	3	3	3	3	3	3	3	3	3	3	120
815	543689	8795202	3	3	3	3	3	3	3	3	3	3	3	138.1
816	542057	8793964	3	3	3	2	1	2	1	1	1	1	1	137.2
817	540958	8793114	3	3	3	2	1	1	1	1	2	2	2	137.2
818	540055	8798965	3	3	3	3	3	3	3	3	3	3	3	158

APPENDIX II. All random training point locations, class label by year, and associated property identification number.

ID	X_COORD	Y_COORD	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	PANEL_ID
819	541818	8793347	3	3	3	3	3	3	2	2	2	2	2	137.1
820	541186	8797385	3	1	2	2	1	1	2	2	2	2	2	152
821	542135	8797401	3	3	3	3	3	3	3	3	1	1	1	110
822	542022	8797596	3	1	2	2	1	1	1	1	1	1	1	110
823	543470	8798486	3	3	3	3	3	3	3	3	3	3	3	110
824	541729	8793475	3	3	3	3	3	3	3	1	2	2	2	137.2
280	545437	8800743	1	1	2	1	1	1	1	1	1	1	1	108
462	547813	8800579	1	1	1	1	2	2	2	1	1	2	2	104
714	547774	8800600	1	1	1	1	2	2	2	2	1	2	2	104
731	546382	8799494	1	1	2	2	2	2	2	2	2	2	2	102

APPENDIX III
Co-registration root mean squared error

Co-registration root-mean squared error (RMSE, pixels) by year. A first order transformation was implemented for all image to image registrations and the 2002 image was used as the reference image in all cases.

Year	Path	Row	Sensor	RMSE Control 1st	RMSE Check 1st	RMSE Control 2nd	RMSE Check 2nd	RMSE Control 3rd	RMSE Check 3rd
1992	231	68	TM	0.1864	0.2190	0.1849	0.2181	0.1761	0.2749
1993	231	68	TM	0.1875	0.1949	0.1790	0.2066	0.1670	0.2323
1994	231	68	TM	0.1989	0.1919	0.1916	0.1950	0.1769	0.2169
1995	231	68	TM	0.1941	0.2052	0.1827	0.2173	0.1686	0.2404
1996	231	68	TM	0.1957	0.1957	0.1889	0.1946	0.1820	0.2018
1997	231	68	TM	0.1708	0.2043	0.1609	0.2100	0.1512	0.2126
1998	231	68	TM	0.1754	0.1781	0.1704	0.1730	0.1674	0.1759
1999	231	68	ETM+	0.1903	0.1868	0.1868	0.1832	0.1742	0.2094
2000	231	68	ETM+	0.1948	0.1899	0.1879	0.1913	0.1737	0.1941
2001	231	68	ETM+	0.1857	0.1767	0.1712	0.1830	0.1658	0.2026
2002	231	68	ETM+	**REFERENCE IMAGE**					

APPENDIX IV
Eigen vectors, eigen matrices, and factor loadings for PCA images

1992 PCA Eigen Values and Explained Variance

PC	Eigen Value	Percentage	Cumulative
1	696.62	89.29%	89.29%
2	64.96	8.33%	97.62%
3	11.37	1.46%	99.07%
4	3.83	0.49%	99.56%
5	2.36	0.30%	99.87%
6	1.04	0.13%	100.00%
Total	780.177869	100%	

1992 PCA Eigen Vectors

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.1438	0.0148	-0.3965	-0.0614	0.8820	-0.2002
2	0.1399	0.0696	-0.3844	-0.1387	-0.0023	0.8992
3	0.2821	-0.0092	-0.7276	-0.1594	-0.4703	-0.3800
4	0.0202	0.9888	0.0075	0.1352	-0.0197	-0.0557
5	0.8686	0.0190	0.3981	-0.2938	0.0139	-0.0118
6	0.3539	-0.1293	-0.0845	0.9203	-0.0156	0.0608

1992 PCA Factor Loadings

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.8926	0.0281	-0.3144	-0.0283	0.3189	-0.0479
2	0.9084	0.1380	-0.3188	-0.0668	-0.0009	0.2252
3	0.9443	-0.0094	-0.3112	-0.0396	-0.0917	-0.0491
4	0.0661	0.9888	0.0032	0.0328	-0.0038	-0.0070
5	0.9967	0.0067	0.0584	-0.0250	0.0009	-0.0005
6	0.9762	-0.1089	-0.0298	0.1882	-0.0025	0.0065

1993 PCA Eigen Values and Explained Variance

PC	Eigen Value	Percentage	Cumulative
1	304.15	70.83%	70.83%
2	87.99	20.49%	91.32%
3	22.97	5.35%	96.66%
4	6.28	1.46%	98.13%
5	4.28	1.00%	99.12%
6	3.77	0.88%	100.00%
Total	429.427584	100%	

1993 PCA Eigen Vectors

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.1850	-0.1922	-0.6602	0.4748	0.5173	-0.0075
2	0.1837	-0.0940	-0.3397	-0.7996	0.2057	0.4004
3	0.2534	-0.2633	-0.3770	-0.2093	-0.4872	-0.6657
4	0.4603	0.8630	-0.1720	0.0472	-0.1070	-0.0052
5	0.7480	-0.2592	0.5183	-0.0031	0.2989	-0.1240
6	0.3105	-0.2701	-0.0916	0.2985	-0.5933	0.6173

1993 PCA Factor Loadings

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.6262	-0.3499	-0.6143	0.2309	0.2076	-0.0028
2	0.7661	-0.2108	-0.3894	-0.4792	0.1017	0.1859
3	0.8124	-0.4540	-0.3321	-0.0964	-0.1851	-0.2375
4	0.6922	0.6980	-0.0711	0.0102	-0.0191	-0.0009
5	0.9574	-0.1784	0.1823	-0.0006	0.0454	-0.0177
6	0.8944	-0.4184	-0.0725	0.1235	-0.2026	0.1979

1994 PCA Eigen Values and Explained Variance

PC	Eigen Value	Percentage	Cumulative
1	601.33	79.27%	79.27%
2	89.07	11.74%	91.01%
3	60.87	8.02%	99.03%
4	3.75	0.49%	99.52%
5	2.82	0.37%	99.89%
6	0.80	0.11%	100.00%
Total	758.632041	100%	

1994 PCA Eigen Vectors

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.3417	-0.1648	0.6887	0.5029	0.3298	-0.1417
2	0.2250	-0.0860	0.2605	-0.3362	0.0661	0.8699
3	0.3762	0.0449	0.3337	-0.7128	-0.1727	-0.4551
4	0.2150	-0.9265	-0.2742	-0.0052	-0.1297	-0.0573
5	0.7165	0.2677	-0.5198	0.0651	0.3748	-0.0065
6	0.3625	0.1825	0.0329	0.3488	-0.8365	0.1129

1994 PCA Factor Loadings

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.8264	-0.1533	0.5299	0.0960	0.0546	-0.0125
2	0.9166	-0.1349	0.3377	-0.1081	0.0185	0.1292
3	0.9512	0.0436	0.2684	-0.1422	-0.0299	-0.0419
4	0.5041	-0.8360	-0.2046	-0.0010	-0.0208	-0.0049
5	0.9644	0.1387	-0.2226	0.0069	0.0346	-0.0003
6	0.9676	0.1875	0.0279	0.0735	-0.1530	0.0110

1995 PCA Eigen Values and Explained Variance

PC	Eigen Value	Percentage	Cumulative
1	707.31	88.87%	88.87%
2	66.92	8.41%	97.28%
3	13.87	1.74%	99.02%
4	3.80	0.48%	99.50%
5	3.07	0.39%	99.89%
6	0.91	0.11%	100.00%
Total	795.881656	100%	

1995 PCA Eigen Vectors

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.1441	-0.0178	-0.5230	-0.2903	0.7631	-0.1968
2	0.1360	0.0378	-0.3683	-0.1254	-0.0913	0.9057
3	0.3008	-0.0432	-0.6483	0.0055	-0.5944	-0.3662
4	-0.0151	0.9870	-0.0661	0.1373	0.0214	-0.0446
5	0.8694	0.0785	0.3993	-0.2799	0.0022	-0.0099
6	0.3378	-0.1270	-0.0822	0.8960	0.2356	0.0690

1995 PCA Factor Loadings

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.8454	-0.0322	-0.4296	-0.1248	0.2947	-0.0414
2	0.9075	0.0775	-0.3443	-0.0613	-0.0401	0.2169
3	0.9487	-0.0419	-0.2864	0.0013	-0.1234	-0.0415
4	-0.0490	0.9864	-0.0301	0.0327	0.0046	-0.0052
5	0.9946	0.0276	0.0640	-0.0235	0.0002	-0.0004
6	0.9734	-0.1125	-0.0332	0.1891	0.0447	0.0071

1996 PCA Eigen Values and Explained Variance

PC	Eigen Value	Percentage	Cumulative
1	658.06	86.67%	86.67%
2	84.69	11.15%	97.83%
3	9.98	1.32%	99.14%
4	3.48	0.46%	99.60%
5	2.27	0.30%	99.90%
6	0.77	0.10%	100.00%
Total	759.249327	100%	

1996 PCA Eigen Vectors

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.1817	-0.0129	-0.4725	-0.1198	0.8310	-0.1966
2	0.1483	0.0426	-0.4061	-0.1723	-0.0792	0.8805
3	0.3197	-0.0736	-0.6477	-0.0686	-0.5465	-0.4116
4	0.0535	0.9846	-0.0523	0.1476	-0.0175	-0.0535
5	0.8209	0.0334	0.4354	-0.3677	0.0132	-0.0098
6	0.4075	-0.1485	-0.0035	0.8912	0.0626	0.1169

1996 PCA Factor Loadings

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.9216	-0.0235	-0.2951	-0.0442	0.2474	-0.0340
2	0.9236	0.0952	-0.3115	-0.0780	-0.0290	0.1872
3	0.9623	-0.0795	-0.2401	-0.0150	-0.0966	-0.0423
4	0.1494	0.9866	-0.0180	0.0300	-0.0029	-0.0051
5	0.9969	0.0145	0.0651	-0.0325	0.0009	-0.0004
6	0.9793	-0.1280	-0.0010	0.1558	0.0088	0.0096

1997 PCA Eigen Values and Explained Variance

PC	Eigen Value	Percentage	Cumulative
1	349.36	76.55%	76.55%
2	88.22	19.33%	95.88%
3	11.19	2.45%	98.33%
4	4.12	0.90%	99.23%
5	2.64	0.58%	99.81%
6	0.85	0.19%	100.00%
Total	456.37862	100%	

1997 PCA Eigen Vectors

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.1753	-0.0570	-0.4872	-0.2862	0.7967	-0.1093
2	0.1622	-0.0060	-0.3851	-0.1832	-0.2191	0.8624
3	0.2633	-0.1431	-0.6196	-0.0731	-0.5392	-0.4798
4	0.2853	0.9475	-0.0679	0.1159	-0.0023	-0.0534
5	0.8160	-0.1771	0.4588	-0.3022	-0.0230	-0.0199
6	0.3553	-0.2172	-0.1237	0.8800	0.1611	0.1043

1997 PCA Factor Loadings

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.8274	-0.1351	-0.4115	-0.1467	0.3267	-0.0255
2	0.8844	-0.0164	-0.3758	-0.1085	-0.1038	0.2326
3	0.8797	-0.2402	-0.3705	-0.0265	-0.1565	-0.0793
4	0.5137	0.8572	-0.0219	0.0227	-0.0004	-0.0048
5	0.9880	-0.1077	0.0994	-0.0397	-0.0024	-0.0012
6	0.9244	-0.2840	-0.0576	0.2486	0.0364	0.0134

1998 PCA Eigen Values and Explained Variance

PC	Eigen Value	Percentage	Cumulative
1	447.14	76.65%	76.65%
2	120.10	20.59%	97.24%
3	10.08	1.73%	98.97%
4	3.33	0.57%	99.54%
5	1.93	0.33%	99.87%
6	0.75	0.13%	100.00%
Total	583.332222	100%	

1998 PCA Eigen Vectors

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.18213	-0.05807	-0.46948	-0.10983	0.84048	-0.15673
2	0.16566	-0.00552	-0.43500	-0.22207	-0.15103	0.84331
3	0.28422	-0.15919	-0.62901	-0.02435	-0.51669	-0.48028
4	0.27139	0.94822	-0.06761	0.14068	-0.02179	-0.04884
5	0.80283	-0.14408	0.43586	-0.37899	0.00387	-0.03293
6	0.37472	-0.22665	0.01275	0.88011	0.05763	0.17356

1998 PCA Factor Loadings

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.8868	-0.1465	-0.3432	-0.0462	0.2691	-0.0313
2	0.9068	-0.0157	-0.3575	-0.1049	-0.0544	0.1893
3	0.9090	-0.2638	-0.3020	-0.0067	-0.1087	-0.0630
4	0.4829	0.8743	-0.0181	0.0216	-0.0025	-0.0036
5	0.9914	-0.0922	0.0808	-0.0404	0.0003	-0.0017
6	0.9374	-0.2938	0.0048	0.1900	0.0095	0.0178

1999 PCA Eigen Values and Explained Variance

PC	Eigen Value	Percentage	Cumulative
1	1358.65	90.43%	90.43%
2	109.85	7.31%	97.75%
3	22.25	1.48%	99.23%
4	7.21	0.48%	99.71%
5	3.10	0.21%	99.91%
6	1.30	0.09%	100.00%
Total	1502.35973	100%	

1999 PCA Eigen Vectors

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.1273	-0.0158	-0.3611	-0.0267	0.7107	-0.5894
2	0.1872	0.0622	-0.4967	-0.1039	0.3454	0.7644
3	0.3754	-0.0456	-0.6586	0.0074	-0.6037	-0.2424
4	-0.0033	0.9805	-0.0219	0.1850	-0.0278	-0.0555
5	0.7360	0.1129	0.3883	-0.5417	0.0282	-0.0234
6	0.5159	-0.1401	0.1944	0.8129	0.0981	0.0775

1999 PCA Factor Loadings

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.9013	-0.0319	-0.3272	-0.0138	0.2404	-0.1293
2	0.9306	0.0880	-0.3160	-0.0376	0.0820	0.1178
3	0.9723	-0.0336	-0.2183	0.0014	-0.0747	-0.0195
4	-0.0119	0.9986	-0.0100	0.0483	-0.0048	-0.0062
5	0.9942	0.0434	0.0671	-0.0533	0.0018	-0.0010
6	0.9885	-0.0763	0.0477	0.1134	0.0090	0.0046

2000 PCA Eigen Values and Explained Variance

PC	Eigen Value	Percentage	Cumulative
1	1406.22	91.16%	91.16%
2	66.46	4.31%	95.47%
3	47.53	3.08%	98.55%
4	12.44	0.81%	99.35%
5	7.88	0.51%	99.87%
6	2.07	0.13%	100.00%
Total	1542.59771	100%	

2000 PCA Eigen Vectors

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.1022	-0.1995	-0.5332	-0.0788	0.6875	-0.4320
2	0.1544	-0.0866	-0.4964	-0.0443	0.0920	0.8437
3	0.3124	-0.2623	-0.5036	0.0457	-0.6976	-0.3020
4	-0.0382	0.8652	-0.3490	0.3457	-0.0297	-0.0881
5	0.7488	0.3254	0.1898	-0.5422	0.0518	-0.0261
6	0.5531	-0.1716	0.2405	0.7591	0.1692	0.0441

2000 PCA Factor Loadings

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.6473	-0.2747	-0.6209	-0.0469	0.3260	-0.1050
2	0.8377	-0.1022	-0.4950	-0.0226	0.0374	0.1757
3	0.9293	-0.1696	-0.2754	0.0128	-0.1553	-0.0345
4	-0.1865	0.9177	-0.3131	0.1587	-0.0108	-0.0165
5	0.9907	0.0936	0.0462	-0.0675	0.0051	-0.0013
6	0.9850	-0.0664	0.0788	0.1272	0.0226	0.0030

2001 PCA Eigen Values and Explained Variance

PC	Eigen Value	Percentage	Cumulative
1	1406.22	91.16%	91.16%
2	66.46	4.31%	95.47%
3	47.53	3.08%	98.55%
4	12.44	0.81%	99.35%
5	7.88	0.51%	99.87%
6	2.07	0.13%	100.00%
Total	1542.59771	100%	

2001 PCA Eigen Vectors

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.1022	-0.1995	-0.5332	-0.0788	0.6875	-0.4320
2	0.1544	-0.0866	-0.4964	-0.0443	0.0920	0.8437
3	0.3124	-0.2623	-0.5036	0.0457	-0.6976	-0.3020
4	-0.0382	0.8652	-0.3490	0.3457	-0.0297	-0.0881
5	0.7488	0.3254	0.1898	-0.5422	0.0518	-0.0261
6	0.5531	-0.1716	0.2405	0.7591	0.1692	0.0441

2001 PCA Factor Loadings

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.6361	-0.2700	-0.6102	-0.0461	0.3204	-0.1032
2	0.6468	-0.0789	-0.3822	-0.0174	0.0288	0.1356
3	0.6767	-0.1235	-0.2005	0.0093	-0.1131	-0.0251
4	-0.1846	0.9082	-0.3098	0.1570	-0.0107	-0.0163
5	0.9275	0.0876	0.0432	-0.0632	0.0048	-0.0012
6	0.9651	-0.0651	0.0772	0.1246	0.0221	0.0030

2002 PCA Eigen Values and Explained Variance

PC	Eigen Value	Percentage	Cumulative
1	457.72	76.59%	76.59%
2	105.33	17.63%	94.22%
3	23.65	3.96%	98.18%
4	5.75	0.96%	99.14%
5	2.92	0.49%	99.63%
6	2.21	0.37%	100.00%
Total	597.590022	100%	

2002 PCA Eigen Vectors

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.1504	-0.0711	0.2481	-0.0090	-0.5748	-0.7618
2	0.2786	0.0012	0.5103	-0.3047	-0.4769	0.5845
3	0.3624	-0.2598	0.6272	0.1328	0.6044	-0.1575
4	0.2351	0.9365	0.1252	0.2091	0.0711	-0.0563
5	0.7261	-0.0270	-0.4673	-0.4807	0.1198	-0.0910
6	0.4312	-0.2227	-0.2249	0.7840	-0.2398	0.2044

2002 PCA Factor Loadings

	Component *p*					
Band *k*	1	2	3	4	5	6
1	0.8436	-0.1913	0.3163	-0.0057	-0.2576	-0.2970
2	0.9027	0.0019	0.3759	-0.1107	-0.1234	0.1316
3	0.8799	-0.3026	0.3461	0.0361	0.1172	-0.0266
4	0.4625	0.8839	0.0560	0.0461	0.0112	-0.0077
5	0.9870	-0.0176	-0.1444	-0.0733	0.0130	-0.0086
6	0.9455	-0.2343	-0.1121	0.1928	-0.0420	0.0311

APPENDIX V
Resulting Optimal CART Trees

The terminal node class assignment for all optimal trees is red for cleared, blue for re-vegetated, and green for primary forest.

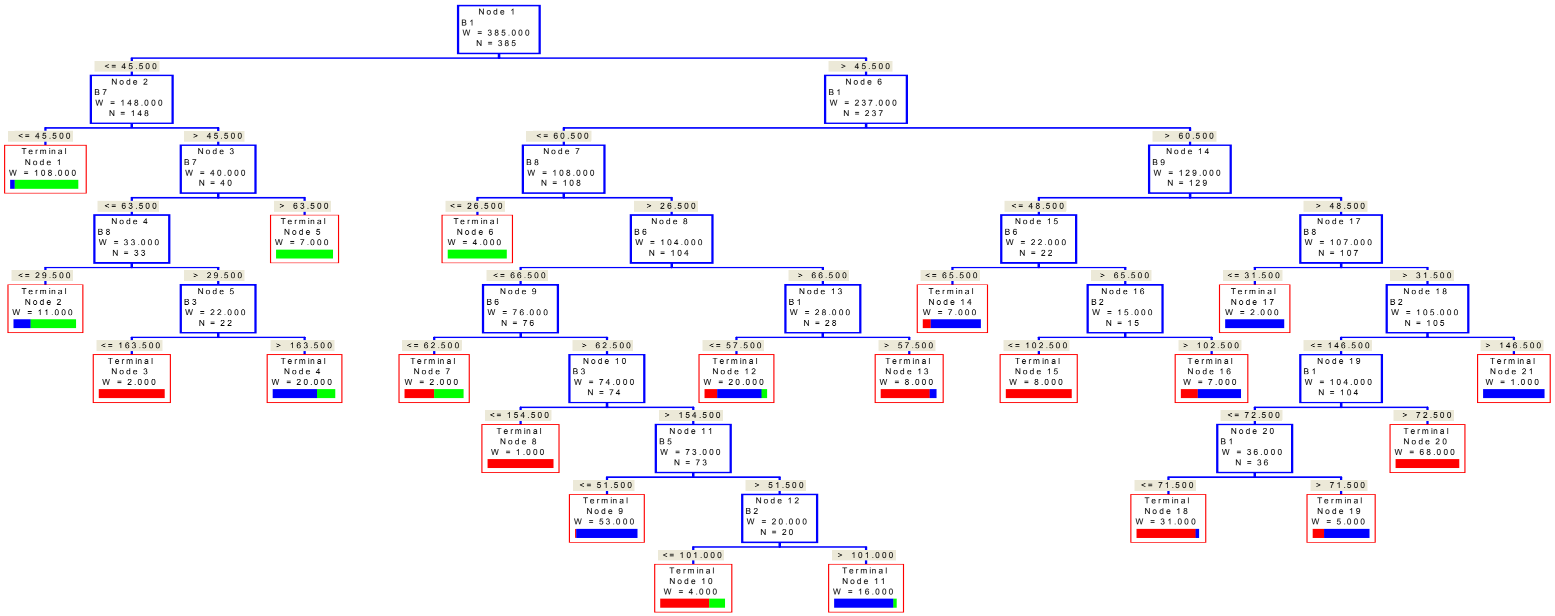

Figure 1. Optimal tree for 1992 pixel-based CART classification based on the Gini rule and 10-fold cross validation

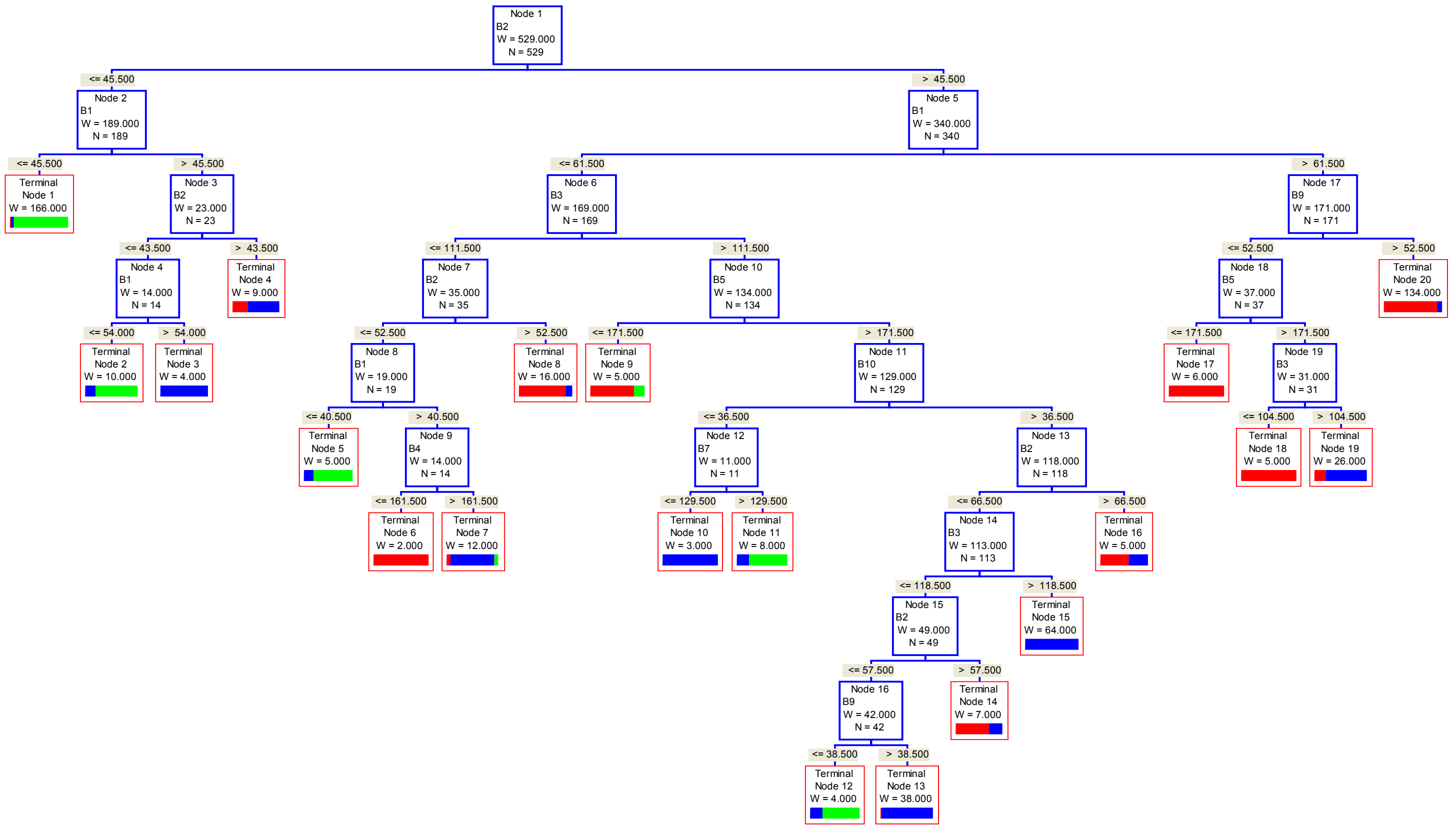

Figure 2. Optimal tree for 1993 pixel-based CART classification based on the Gini rule and 10-fold cross validation

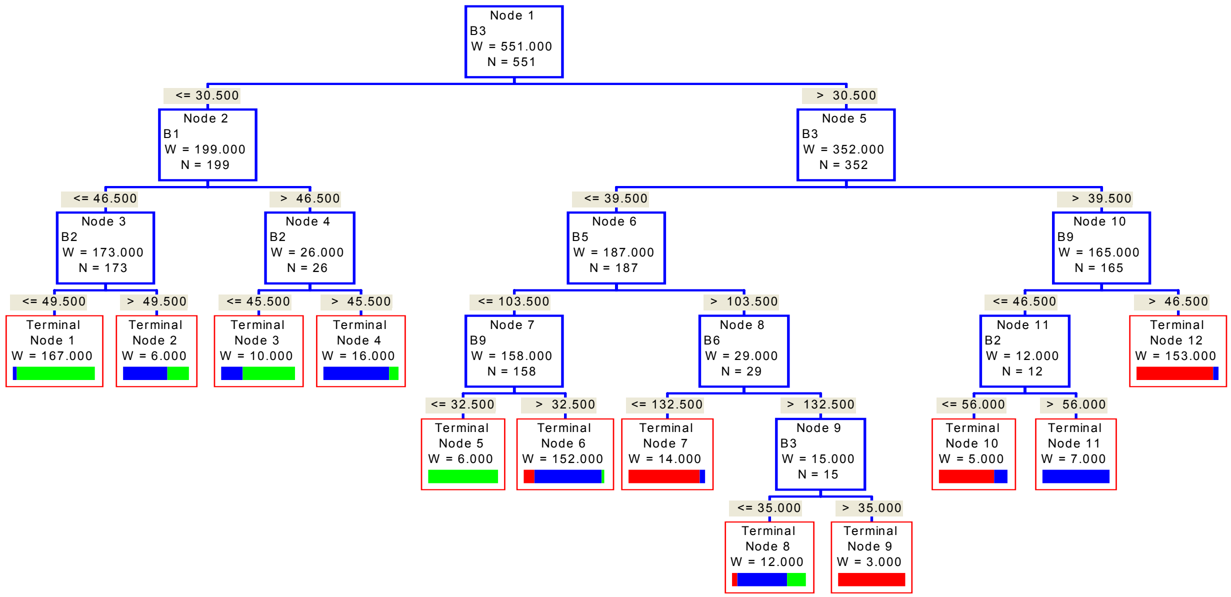

Figure 3. Optimal tree for 1994 pixel-based CART classification based on the Gini rule and 10-fold cross validation

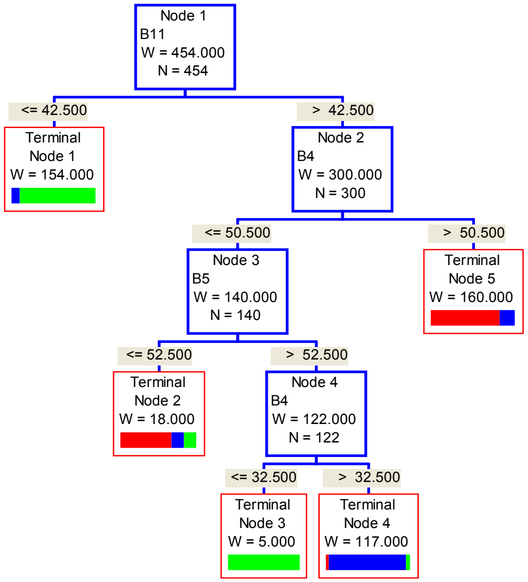

Figure 4. Optimal tree for 1995 pixel-based CART classification based on the Gini rule and 10-fold cross validation

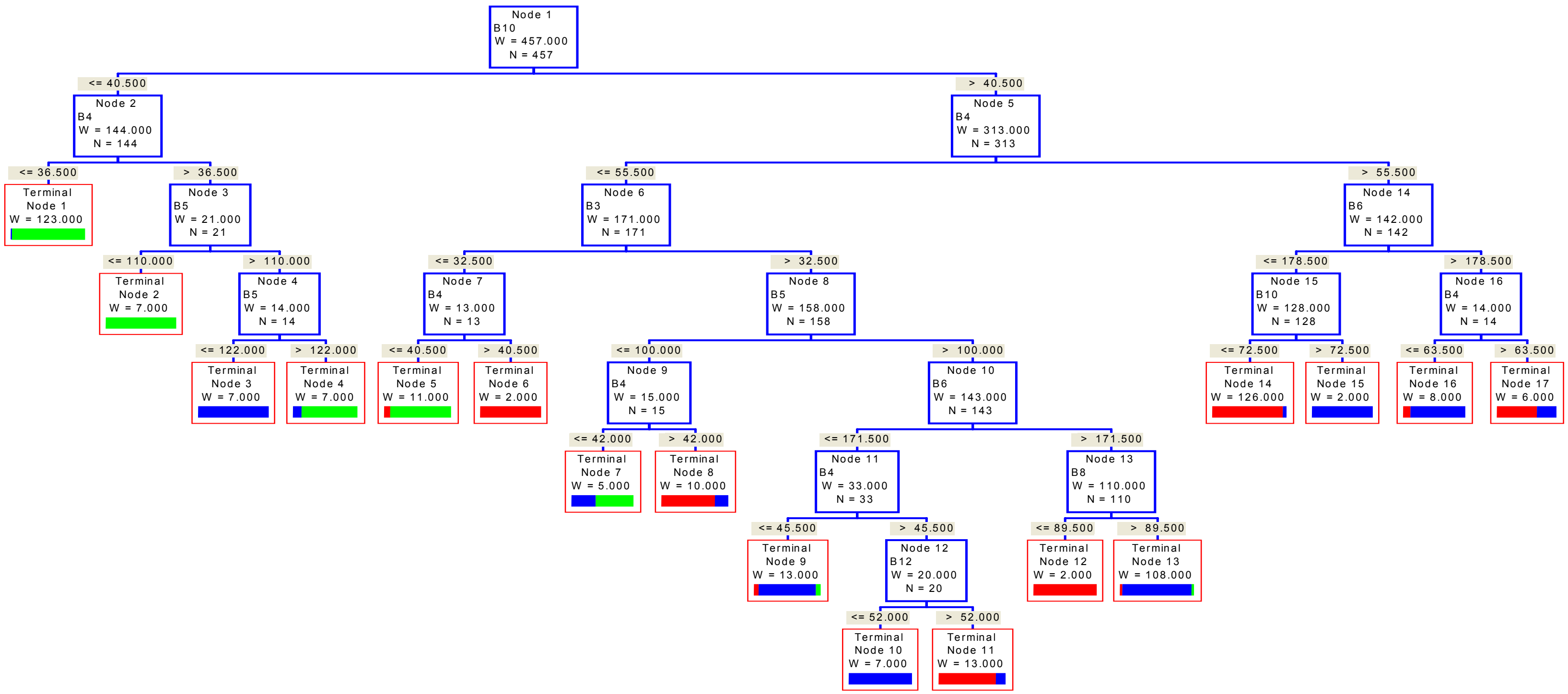

Figure 5. Optimal tree for 1996 pixel-based CART classification based on the Gini rule and 10-fold cross validation

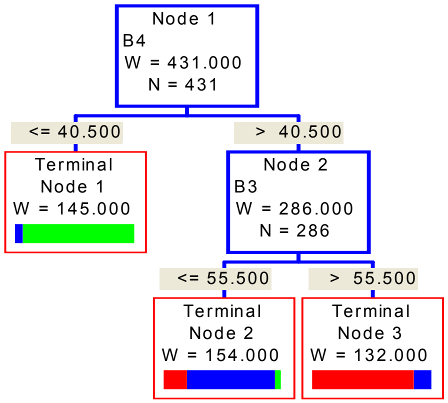

Figure 6. Optimal tree for 1997 pixel-based CART classification based on the Gini rule and 10-fold cross validation

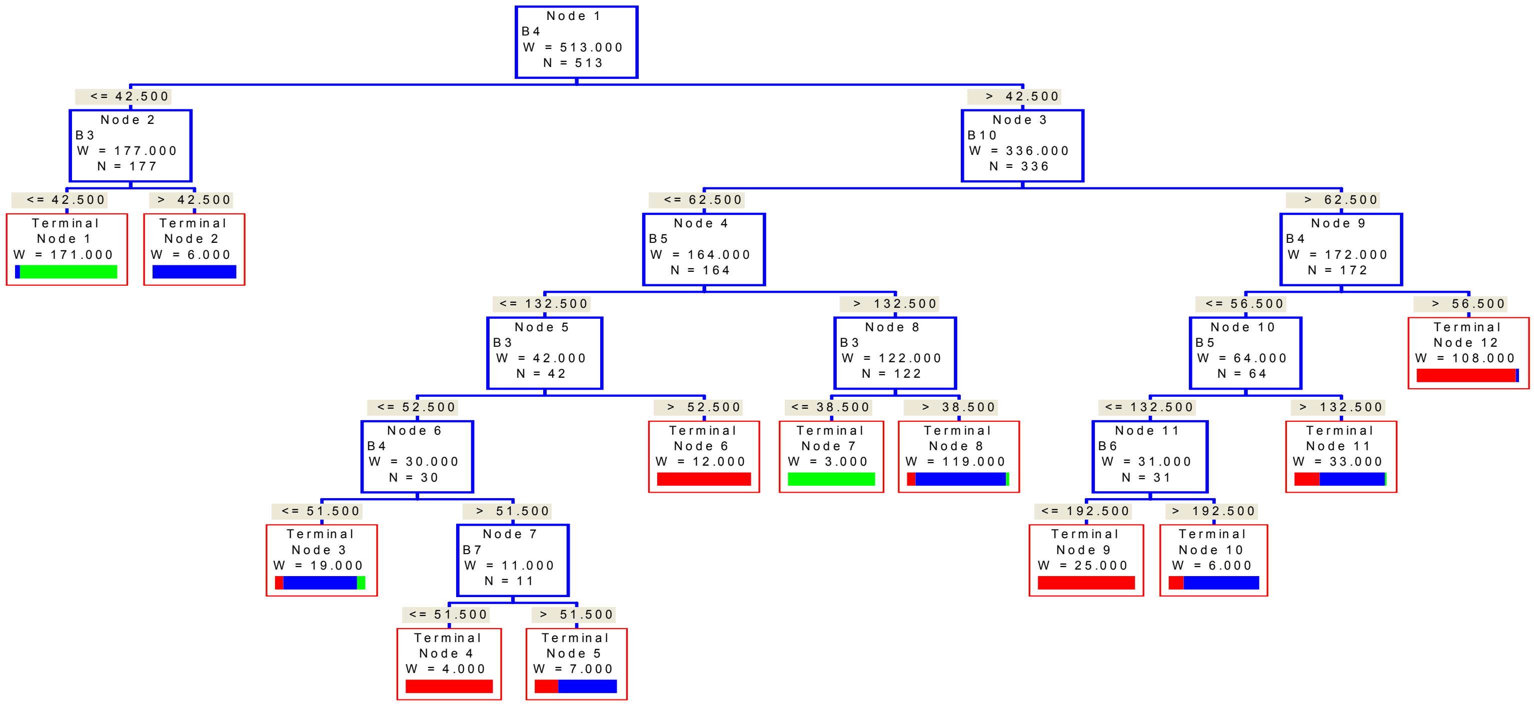

Figure 7. Optimal tree for 1998 pixel-based CART classification based on the Gini rule and 10-fold cross validation

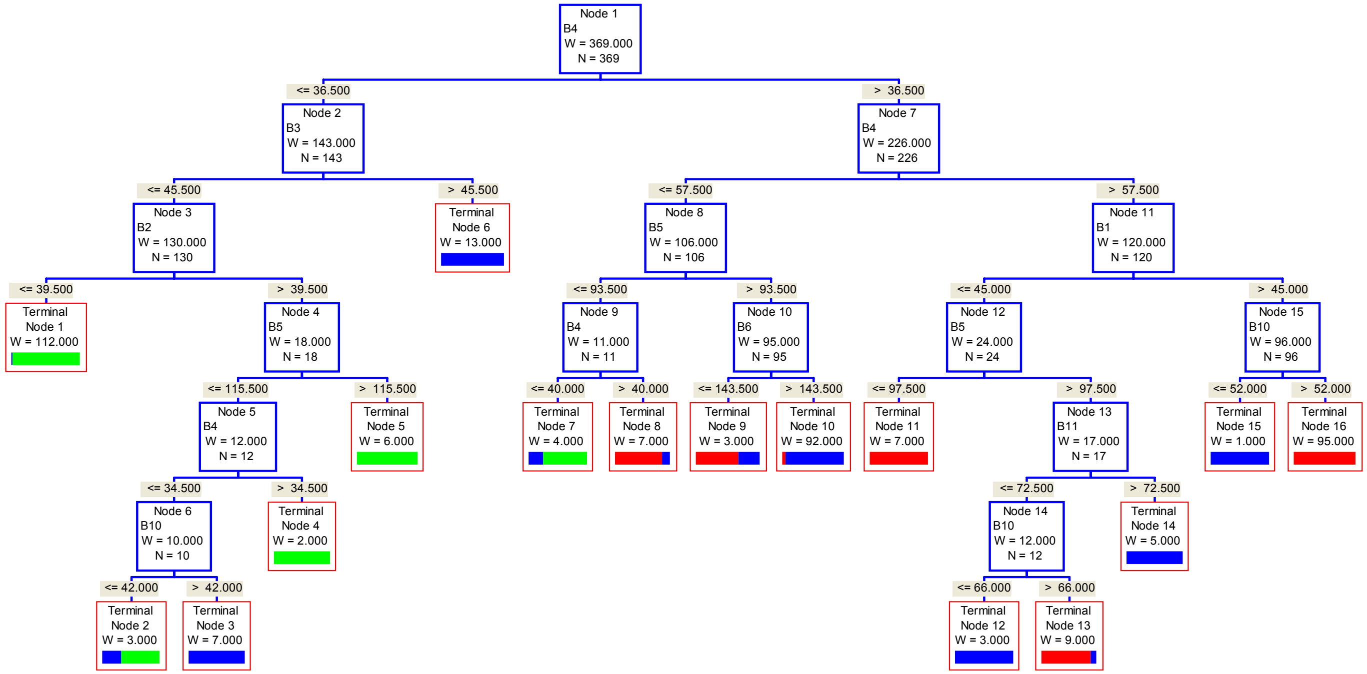

Figure 8. Optimal tree for 1999 pixel-based CART classification based on the Gini rule and 10-fold cross validation

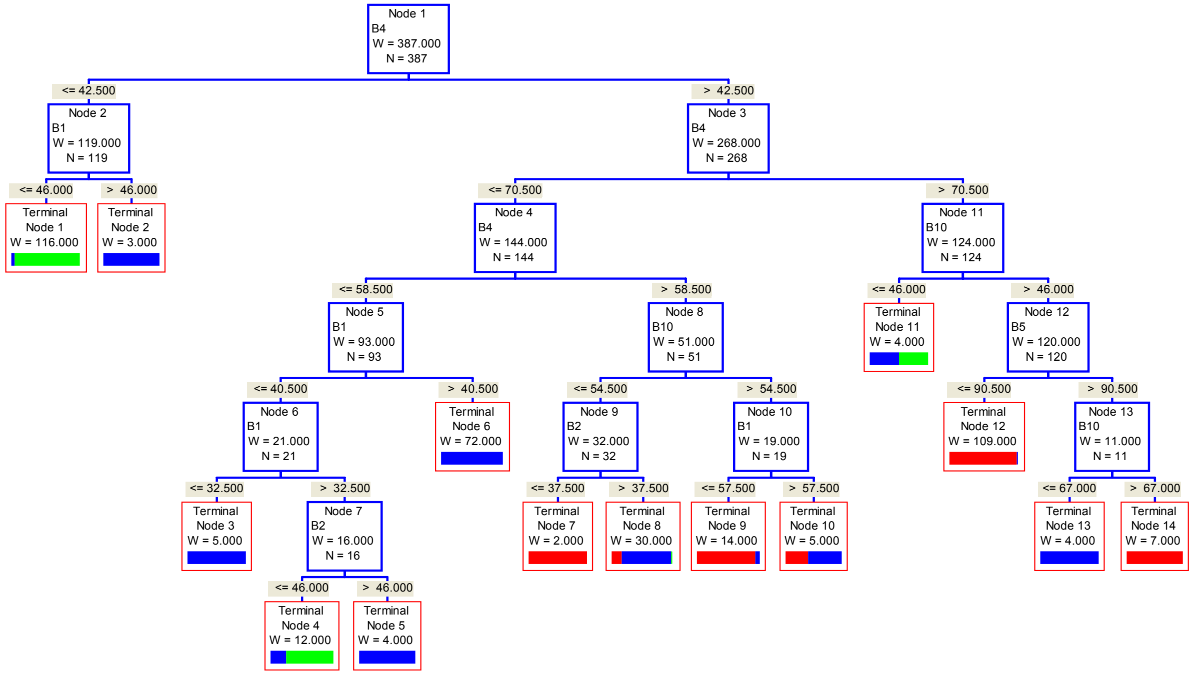

Figure 9. Optimal tree for 2000 pixel-based CART classification based on the Gini rule and 10-fold cross validation

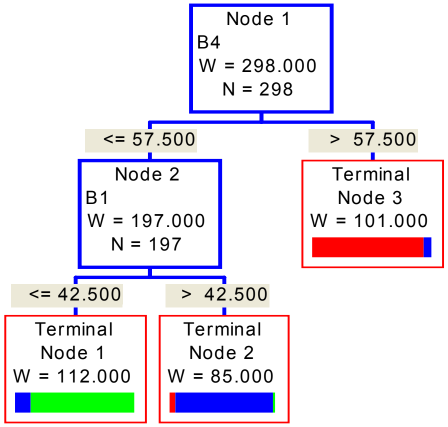

Figure 10. Optimal tree for 2001 pixel-based CART classification based on the Gini rule and 10-fold cross validation

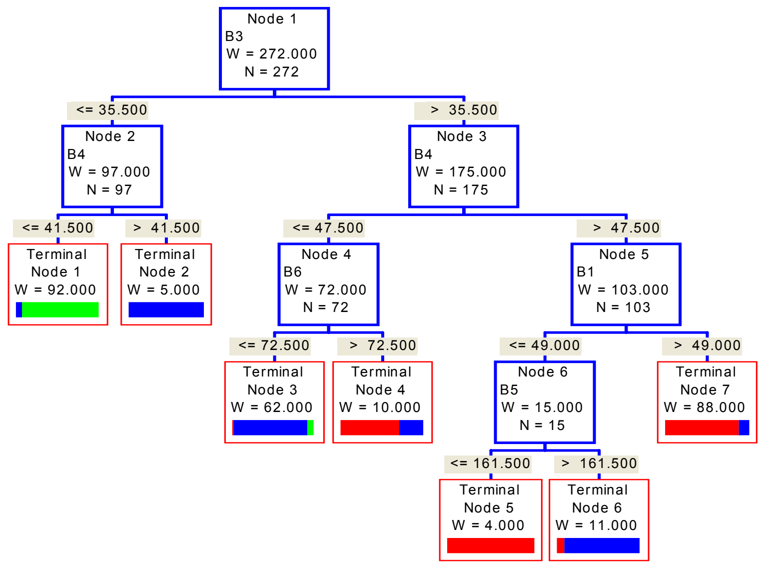

Figure 11. Optimal tree for 2002 pixel-based CART classification based on the Gini rule and 10-fold cross validation

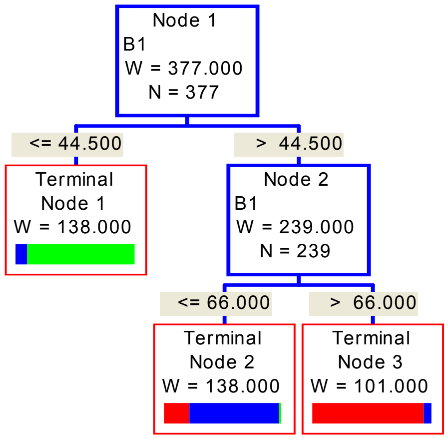

Figure 12. Optimal tree for 1992 segment-based CART classification based on the Gini rule and 10-fold cross validation

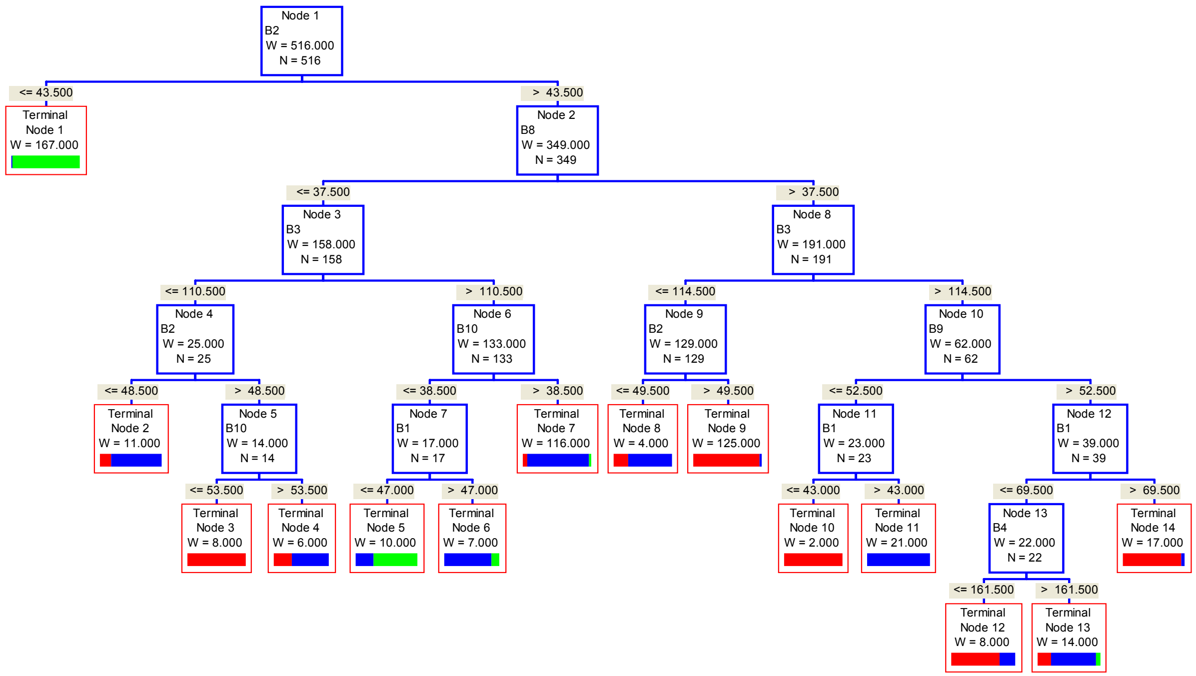

Figure 13. Optimal tree for 1993 segment-based CART classification based on the Gini rule and 10-fold cross validation

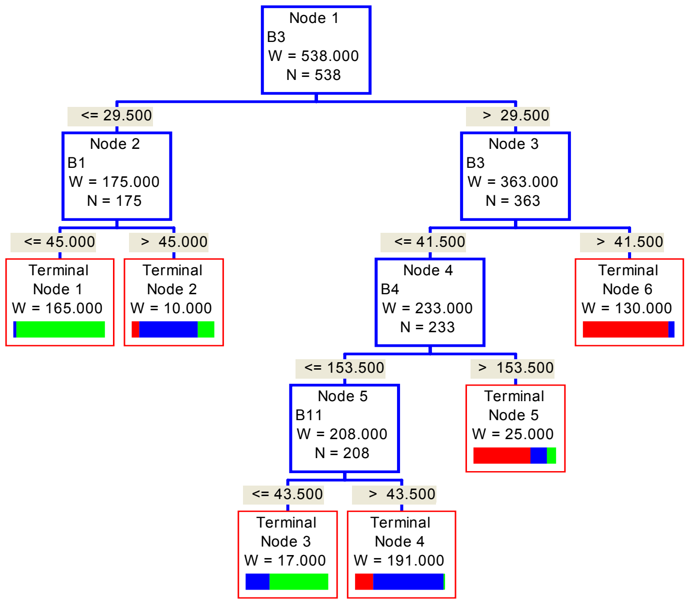

Figure 14. Optimal tree for 1994 segment-based CART classification based on the Gini rule and 10-fold cross validation

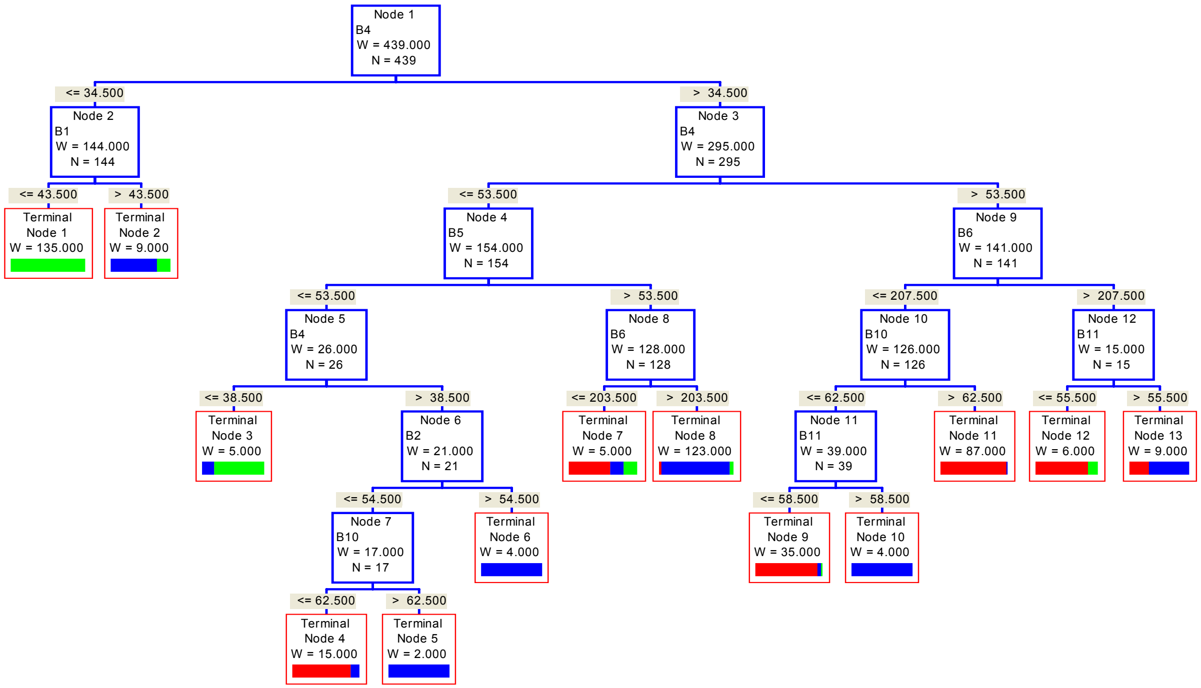

Figure 15. Optimal tree for 1995 segment-based CART classification based on the Gini rule and 10-fold cross validation

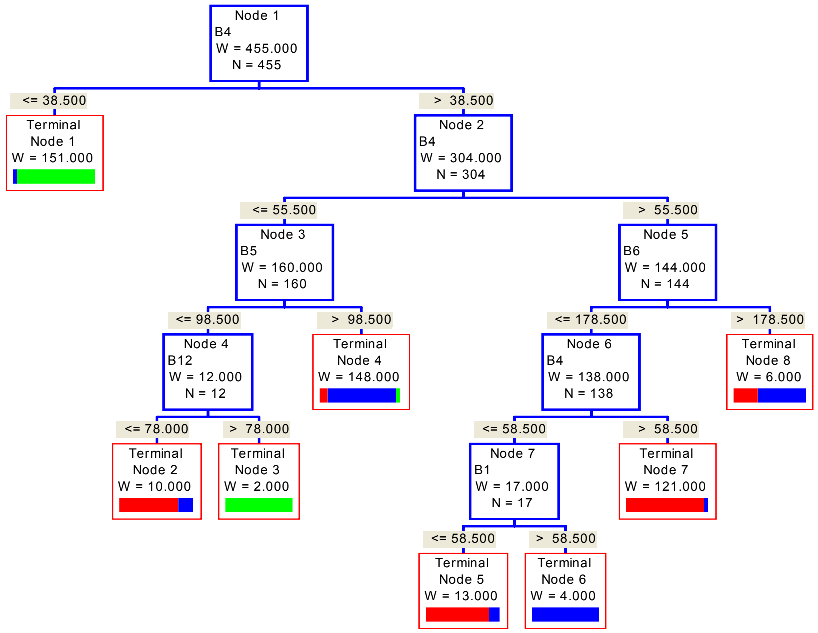

Figure 16. Optimal tree for 1996 segment-based CART classification based on the Gini rule and 10-fold cross validation

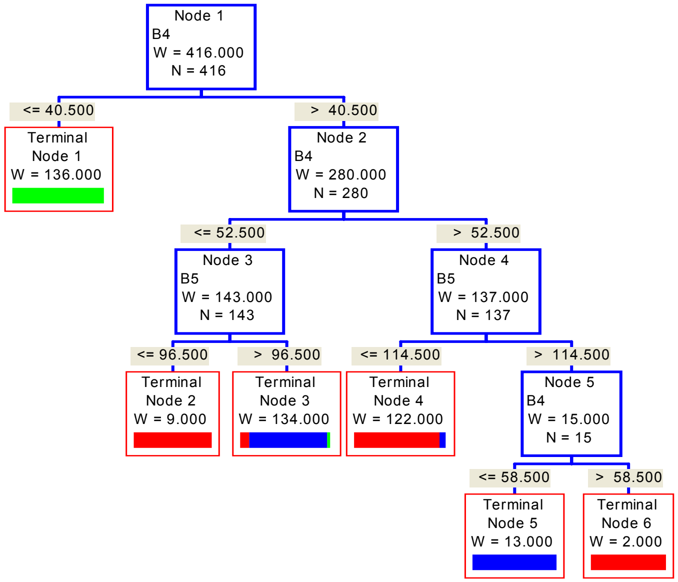

Figure 17. Optimal tree for 1997 segment-based CART classification based on the Gini rule and 10-fold cross validation

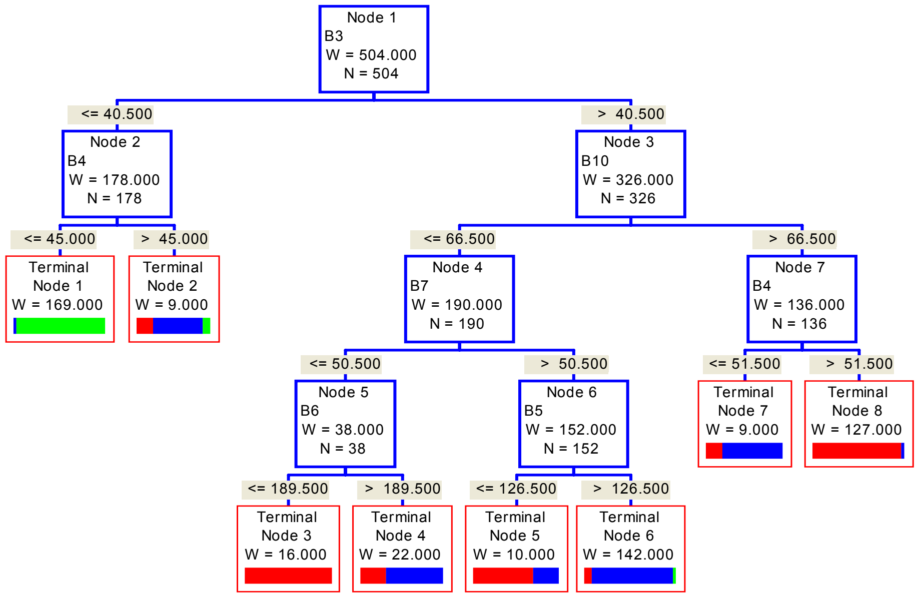

Figure 18. Optimal tree for 1998 segment-based CART classification based on the Gini rule and 10-fold cross validation

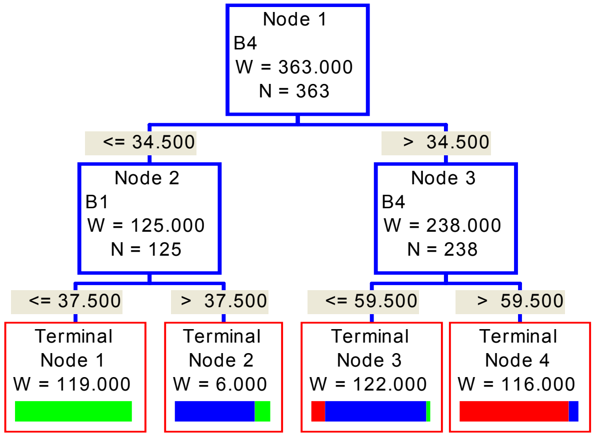

Figure 19. Optimal tree for 1999 segment-based CART classification based on the Gini rule and 10-fold cross validation

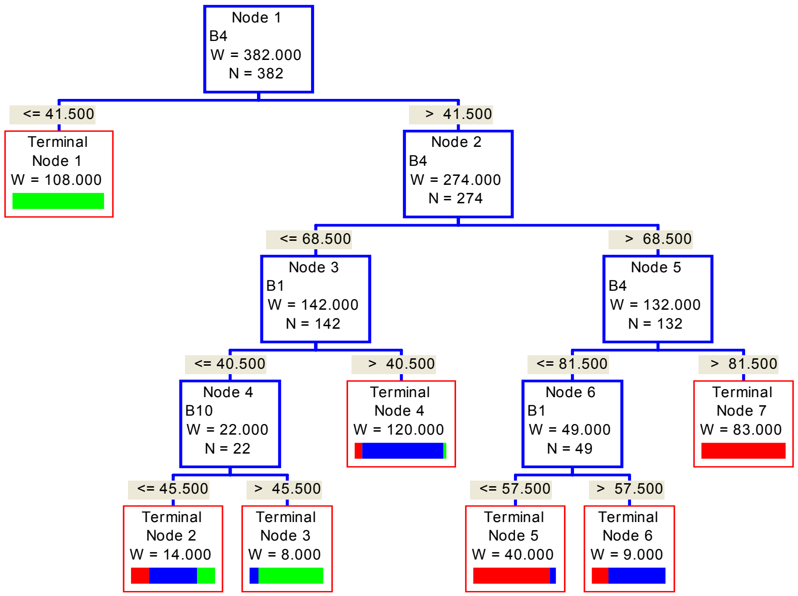

Figure 20. Optimal tree for 2000 segment-based CART classification based on the Gini rule and 10-fold cross validation

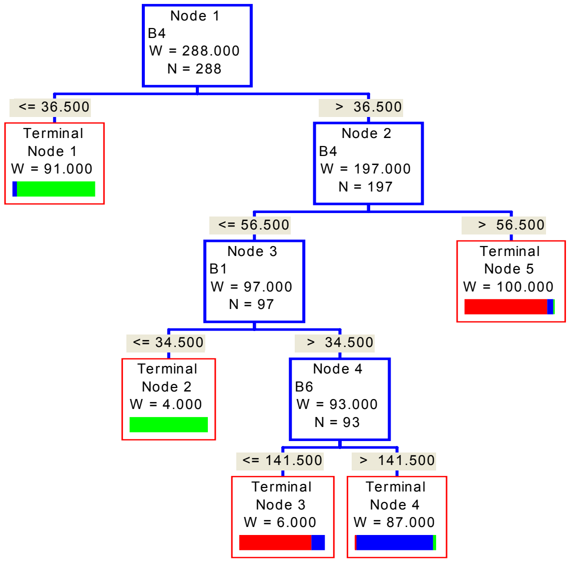

Figure 21. Optimal tree for 2001 segment-based CART classification based on the Gini rule and 10-fold cross validation

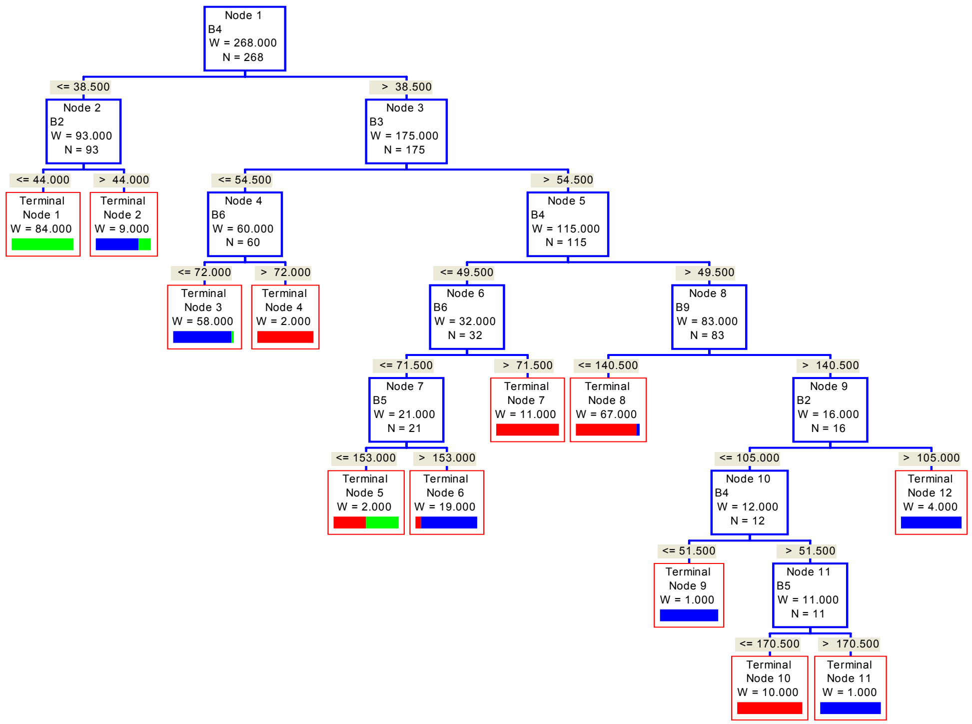

Figure 22. Optimal tree for 2002 segment-based CART classification based on the Gini rule and 10-fold cross validation

APPENDIX VI
Error Matrices

Table 1. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1992 pixel-based kNN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	45	15	1	90.00%	73.77%	0.6066
Re-vegetated	5	30	2	60.00%	81.08%	0.7162
Primary Forest	0	5	47	94.00%	90.38%	0.8558

Table 2. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1993 pixel-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	48	7	1	96.00%	85.71%	0.7857
Re-vegetated	1	37	4	74.00%	88.10%	0.8214
Primary Forest	1	6	45	90.00%	86.54%	0.7981

Table 3. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1994 pixel-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	47	9	1	94.00%	82.46%	0.7368
Re-vegetated	3	35	3	70.00%	85.37%	0.7805
Primary Forest	0	6	46	92.00%	88.46%	0.8269

Table 4. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1995 pixel-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	47	14	0	94.00%	77.05%	0.6557
Re-vegetated	3	33	3	66.00%	84.62%	0.7692
Primary Forest	0	3	47	94.00%	94.00%	0.91

Table 5. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1996 pixel-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	47	13	1	94.00%	77.05%	0.6557
Re-vegetated	3	34	1	68.00%	89.47%	0.8421
Primary Forest	0	3	48	96.00%	94.12%	0.9118

Table 6. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1997 pixel-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	49	11	0	98.00%	81.67%	0.7250
Re-vegetated	0	36	1	72.00%	97.30%	0.9595
Primary Forest	1	3	49	98.00%	92.45%	0.8868

Table 7. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1998 pixel-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	46	10	1	92.00%	80.70%	0.7105
Re-vegetated	4	38	7	76.00%	77.55%	0.6633
Primary Forest	0	2	42	84.00%	95.45%	0.9318

Table 8. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1999 pixel-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	48	13	0	96.00%	78.69%	0.6803
Re-vegetated	2	32	2	64.00%	88.89%	0.8333
Primary Forest	0	5	48	96.00%	90.57%	0.8585

Table 9. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 2000 pixel-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	49	18	0	98.00%	73.13%	0.5970
Re-vegetated	1	29	1	58.00%	93.55%	0.9032
Primary Forest	0	3	49	98.00%	94.23%	0.9135

Table 10. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 2001 pixel-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	50	10	0	100.00%	83.33%	0.7500
Re-vegetated	0	39	2	78.00%	95.12%	0.9268
Primary Forest	0	1	48	96.00%	97.96%	0.9694

Table 11. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 2002 pixel-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	49	21	2	98.00%	68.06%	0.5208
Re-vegetated	1	27	5	54.00%	81.82%	0.7273
Primary Forest	0	2	43	86.00%	95.56%	0.9333

Table 12. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1992 pixel-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	44	7	1	88.00%	84.62%	0.7692
Re-vegetated	6	41	3	82.00%	82.00%	0.7300
Primary Forest	0	2	46	92.00%	95.83%	0.9375

Table 13. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1993 pixel-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	46	7	1	92.00%	85.19%	0.7778
Re-vegetated	3	38	3	76.00%	86.36%	0.7955
Primary Forest	1	5	46	92.00%	88.46%	0.8269

Table 14. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1994 pixel-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	44	8	0	88.00%	84.62%	0.7692
Re-vegetated	6	41	6	82.00%	77.36%	0.6604
Primary Forest	0	1	44	88.00%	97.78%	0.9667

Table 15. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1995 pixel-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	47	12	0	94.00%	79.66%	0.6949
Re-vegetated	3	32	3	64.00%	84.21%	0.7632
Primary Forest	0	6	47	94.00%	88.68%	0.8302

Table 16. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1996 pixel-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	43	7	1	86.00%	84.31%	0.7647
Re-vegetated	7	38	2	76.00%	80.85%	0.7128
Primary Forest	0	5	47	94.00%	90.38%	0.8558

Table 17. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1997 pixel-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	43	11	0	86.00%	79.63%	0.6944
Re-vegetated	7	35	2	70.00%	79.55%	0.6932
Primary Forest	0	4	48	96.00%	92.31%	0.8846

Table 18. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1998 pixel-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	34	4	0	68.00%	89.47%	0.8421
Re-vegetated	16	44	5	88.00%	67.69%	0.5154
Primary Forest	0	2	45	90.00%	95.74%	0.9362

Table 19. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1999 pixel-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	43	9	0	86.00%	82.69%	0.7404
Re-vegetated	7	36	1	72.00%	81.82%	0.7273
Primary Forest	0	5	49	98.00%	90.74%	0.8611

Table 20. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 2000 pixel-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	46	11	1	92.00%	79.31%	0.6897
Re-vegetated	4	33	1	66.00%	86.84%	0.8026
Primary Forest	0	6	48	96.00%	88.89%	0.8333

Table 21. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 2001 pixel-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	47	9	1	94.00%	82.46%	0.7368
Re-vegetated	3	36	0	72.00%	92.31%	0.8846
Primary Forest	0	5	49	98.00%	90.74%	0.8611

Table 22. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 2002 pixel-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	48	19	0	96.00%	71.64%	0.5746
Re-vegetated	2	30	3	60.00%	85.71%	0.7857
Primary Forest	0	1	47	94.00%	97.92%	0.9688

Table 23. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1992 segment-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	42	14	0	84.00%	75.00%	0.6250
Re-vegetated	8	30	3	60.00%	73.17%	0.5976
Primary Forest	0	6	47	94.00%	88.68%	0.8302

Table 24. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1993 segment-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	44	13	3	88.00%	73.33%	0.6000
Re-vegetated	4	31	1	62.00%	86.11%	0.7917
Primary Forest	2	6	46	92.00%	85.19%	0.7778

Table 25. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1994 segment-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	45	11	1	90.00%	78.95%	0.6842
Re-vegetated	5	38	3	76.00%	82.61%	0.7391
Primary Forest	0	1	46	92.00%	97.87%	0.9681

Table 26. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1995 segment-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	48	21	0	96.00%	69.57%	0.5435
Re-vegetated	2	23	3	46.00%	82.14%	0.7321
Primary Forest	0	6	47	94.00%	88.68%	0.8302

Table 27. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1996 segment-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	45	13	3	90.00%	73.77%	0.6066
Re-vegetated	5	34	3	68.00%	80.95%	0.7143
Primary Forest	0	3	44	88.00%	93.62%	0.9043

Table 28. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1997 segment-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	47	13	0	94.00%	78.33%	0.6750
Re-vegetated	1	34	2	68.00%	91.89%	0.8784
Primary Forest	2	3	48	96.00%	90.57%	0.8585

Table 29. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1998 segment-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	46	6	1	92.00%	86.79%	0.8019
Re-vegetated	4	43	9	86.00%	76.79%	0.6518
Primary Forest	0	1	40	80.00%	97.56%	0.9634

Table 30. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1999 segment-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	47	14	0	94.00%	77.05%	0.6557
Re-vegetated	3	33	1	66.00%	89.19%	0.8378
Primary Forest	0	3	49	98.00%	94.23%	0.9135

Table 31. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 2000 segment-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	49	15	1	98.00%	75.38%	0.6308
Re-vegetated	1	31	1	62.00%	93.94%	0.9091
Primary Forest	0	4	48	96.00%	92.31%	0.8846

Table 32. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 2001 segment-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	49	12	1	98.00%	79.03%	0.6855
Re-vegetated	1	37	5	74.00%	86.05%	0.7907
Primary Forest	0	1	44	88.00%	97.78%	0.9667

Table 33. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 2002 segment-based *k*NN classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Cleared	48	23	0	96.00%	67.61%	0.5141
Re-vegetated	2	25	7	50%	73.53%	0.6029
Primary Forest	0	2	43	86.00%	95.56%	0.9333

Table 34. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1992 segment-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Class 1	36	6	0	72.00%	85.71%	0.7857
Class 2	14	39	3	78.00%	69.64%	0.5446
Class 3	0	5	47	94.00%	90.38%	0.8558

Table 35. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1993 segment-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Class 1	43	8	3	86.00%	79.63%	0.6944
Class 2	7	38	5	76.00%	76.00%	0.6400
Class 3	0	4	42	84.00%	91.30%	0.8696

Table 36. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1994 segment-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Class 1	39	6	1	78.00%	84.78%	0.7717
Class 2	10	41	2	82.00%	77.36%	0.6604
Class 3	1	3	47	94.00%	92.16%	0.8824

Table 37. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1995 segment-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Class 1	39	10	2	78.00%	76.47%	0.6471
Class 2	11	38	5	76.00%	70.37%	0.5556
Class 3	0	2	43	86.00%	95.56%	0.9333

Table 38. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1996 segment-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Class 1	40	4	1	80.00%	88.89%	0.8333
Class 2	10	41	5	82.00%	73.21%	0.5982
Class 3	0	5	44	88.00%	89.80%	0.8469

Table 39. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1997 segment-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Class 1	46	7	0	92.00%	86.79%	0.8019
Class 2	4	41	1	82.00%	89.13%	0.8370
Class 3	0	2	49	98.00%	96.08%	0.9412

Table 40. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1998 segment-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Class 1	38	4	0	76.00%	90.48%	0.8571
Class 2	12	44	10	88.00%	66.67%	0.5000
Class 3	0	2	40	80.00%	95.24%	0.9286

Table 41. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 1999 segment-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Class 1	44	4	0	88.00%	91.67%	0.8750
Class 2	6	44	2	88.00%	84.62%	0.7692
Class 3	0	2	48	96.00%	96.00%	0.9400

Table 42. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 2000 segment-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Class 1	46	7	0	92.00%	86.79%	0.8019
Class 2	3	41	2	82.00%	89.13%	0.8370
Class 3	1	2	48	96.00%	94.12%	0.9118

Table 43. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 2001 segment-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Class 1	49	13	0	98.00%	79.03%	0.6855
Class 2	1	32	2	64.00%	91.43%	0.8714
Class 3	0	5	48	96.00%	90.57%	0.8585

Table 44. Error matrix including Producer's and User's Accuracies and kappa coefficients by class for 2002 segment-based CART classification

	Reference Data			Producers	Users	
Classified Data	Cleared	Re-vegetated	Primary Forest	Accuracy	Accuracy	Kappa
Class 1	44	6	0	88.00%	88.00%	0.82
Class 2	5	41	6	82.00%	78.85%	0.6827
Class 3	1	3	44	88.00%	91.67%	0.875

APPENDIX VII
Fortran 95 *k*-nearest neighbor program library

```
!   Last change:  RHW  21 Jan 2005   11:20 pm
!   Background pixel check only enabled for nns, PercentOverallAccuracy,
and Recode
!
module CEARSwLib
USE quicksort_mod

contains


!     Last change:  RHW  28 Feb 2004    6:30 am
!
! ComputeNN
!
! Purpose:
!
!     From input of cluster means and an array of brightness value
vectors,
!     find the cluster mean nearest to each brightness value vector in the
image
!
! Inputs:
!
!       Cluster means array
!       Brightness value vector array
!
! Outputs:
!
!       Nearest neighbor array
!
! Assumptions:
!
!       Index values in cluster means array are the cluster numbers

subroutine knn (k, c, x, m, icx, kNN_k, sc_ic, NumInfoClasses)

implicit none
        INTEGER (kind = 4), INTENT (IN) :: k,c
        INTEGER (kind = 2), INTENT (IN) :: m (:,:)
        REAL (KIND = 8) :: XminusM (size (m,1),1)
        INTEGER (KIND = 2), INTENT (IN) :: kNN_k
        INTEGER (KIND = 2), INTENT (IN) :: sc_ic (:)
        INTEGER (KIND = 2), INTENT (IN) :: x (:,:)
        INTEGER (KIND = 2), INTENT (IN) :: NumInfoClasses
          INTEGER (KIND = 2), INTENT (OUT) :: icx (:)
        INTEGER (KIND = 4) :: h,i,j
        INTEGER (kind = 4) :: MinimumDistance1x1Array (1,1)
        INTEGER (kind = 4) :: MinimumDistanceArray (2,c)
        INTEGER (kind = 4) :: MinimumDistanceArray3D (3,c)

          icx = 0

!       PRINT *, "c = ", c
```

```
!       PRINT *, "m (", j,i, ") = ", m (j,i)
!       PRINT *, "m (:,5) = ", m (:,5)
!        PRINT *, "x (:,5) = ", x (:,5)
!        PRINT *, "m = ", m

!$OMP do
    NumPixelsLoop: do i = 1, k

                ProgressStatements: if ((i / (k/4) .EQ. 1) .AND.
(modulo(i,(k/4)) .EQ. 0)) then
                    PRINT *, "One-fourth of image has been processed."
                    else if ( ((i / (k/2) ) .EQ. 1) .AND. (modulo(i,(k/2))
.EQ. 0) ) then
                        PRINT *, "One-half of image has been processed."
                    else if ( ((i / ((3*k)/4) ) .EQ. 1) .AND.
(modulo(i,((3*k)/4)) .EQ. 0)) then
                        PRINT *, "Three-fourths of image has been
processed."
                end if ProgressStatements

                  IsNotBackground: if ( maxval(x (:,i)) .NE. 0 ) then

                  NumClustersLoop: do j = 1, c

                        XminusM (:,1) = x(:,i) - m(:,j)

 !                      if (i .EQ. 5) then
 !                          PRINT *, "XminusM at i = 5 and j = ", j
 !                          PRINT *, int(XminusM (:,1),2)
 !                      end if

                        MinimumDistanceArray (1,j) = j

 !                      if (i .EQ. 5) then
 !                          PRINT *, "MinimumDistanceArray (1,j) at i = 5"
 !                          PRINT *, MinimumDistanceArray (1,j)
 !                      end if

                        MinimumDistance1x1Array = &
                                    sqrt(MATMUL (TRANSPOSE (XminusM),
XminusM))

  !                     if (i .EQ. 5) then
  !                         PRINT *, "MinimumDistance at i = 5 and j = ",
j
  !                         PRINT *, MinimumDistance1x1Array
  !                     end if

                    MinimumDistanceArray (2,j) = MinimumDistance1x1Array
(1,1)

                end DO NumClustersLoop

  !            if (i .EQ. 5) then
```

```
  !                PRINT *, "MinimumDistanceArray (2,:) before QuickSort
at i = 5."
  !                PRINT *, MinimumDistanceArray (2,:)
  !            end if

                call QuickSort(MinimumDistanceArray, 1, c)

  !           if (i .EQ. 5) then
  !               PRINT *, "MinimumDistanceArray (2,:) after QuickSort at
i = 5."
  !               PRINT *, MinimumDistanceArray (2,:)
  !           end if

                MinimumDistanceArray3D (1,:) = MinimumDistanceArray (1,:)
                MinimumDistanceArray3D (2,:) = MinimumDistanceArray (2,:)

                AssignInfoClass: do h = 1, c

                    MinimumDistanceArray3D (3,h) =  sc_ic (
MinimumDistanceArray3D (1,h) )

                end do AssignInfoClass

  !             if (i .EQ. 5) then
  !                 PRINT *, "MinimumDistanceArray3D (1,:), (2,:).  and
(3,:) after QuickSort at i = 5."
  !                 PRINT *, MinimumDistanceArray3D (1,:)
  !                 PRINT *, MinimumDistanceArray3D (2,:)
  !                 PRINT *, MinimumDistanceArray3D (3,:)
  !             end if

                icx (i) = Find_kNN_Winner (kNN_k, NumInfoClasses,
MinimumDistanceArray3D)

  !             if (i .EQ. 5) then
  !                 PRINT *, "icx(5) = ", icx(5)
  !             end if

!               if (icx (i) .EQ. 2) then
!                   PRINT *, "icx (", i, ") = ", 2
!               end if

            end if IsNotBackground

        end DO NumPixelsLoop
!$OMP end do

        PRINT *, "All of image has been processed."

end subroutine knn

! end of good knn

function knnk (c, m, MaxkNN_k, sc_ic, NumInfoClasses) result (BestkNN_k)

implicit none
        INTEGER (kind = 4), INTENT (IN) :: c
```

```
        INTEGER (kind = 2), INTENT (IN) :: m (:,:)
        REAL (KIND = 8) :: XminusM (size (m,1),1)
        INTEGER (KIND = 2), INTENT (IN) :: MaxkNN_k
        INTEGER (KIND = 2) :: kNN_k
        INTEGER (KIND = 2), INTENT (IN) :: sc_ic (:)
        INTEGER (KIND = 2), INTENT (IN) :: NumInfoClasses
          INTEGER (KIND = 2) BestkNN_k
          INTEGER (KIND = 2) :: BestkNN_kArray(1)
        INTEGER (KIND = 4) :: g, h,i,j
        INTEGER (kind = 4) :: MinimumDistance1x1Array (1,1)
        INTEGER (kind = 4) :: MinimumDistanceArray (2,c)
        INTEGER (kind = 4) :: MinimumDistanceArray3D (3,c)
        INTEGER (KIND = 2) :: icx (c)
        INTEGER (KIND = 2) :: Accuracy (MaxkNN_k)
        INTEGER (KIND = 2) :: NumCorrect

          icx = 0
          Accuracy = 0

!       PRINT *, "c = ", c


!       PRINT *, "m (", j,i, ") = ", m (j,i)
!       PRINT *, "m (:,5) = ", m (:,5)
!        PRINT *, "m = ", m

        kNN_k_Loop: do kNN_k = 1, MaxkNN_k

        kNot2: if (kNN_k .NE. 2) then

            NumPixelsLoop: do i = 1, c

            MinimumDistanceArray (1,:) = 0
            MinimumDistanceArray (2,:) = 10000

!           PRINT *, MinimumDistanceArray

!                    ProgressStatements: if ((i / (k/4) .EQ. 1) .AND.
(modulo(i,(k/4)) .EQ. 0)) then
!                        PRINT *, "One-fourth of image has been
processed."
!                        else if ( ((i / (k/2) ) .EQ. 1) .AND.
(modulo(i,(k/2)) .EQ. 0) ) then
!                            PRINT *, "One-half of image has been
processed."
!                        else if ( ((i / ((3*k)/4) ) .EQ. 1) .AND.
(modulo(i,((3*k)/4)) .EQ. 0)) then
!                            PRINT *, "Three-fourths of image has been
processed."
!                    end if ProgressStatements

                  IsNotBackground: if ( m (1,i) .NE. 0 ) then

                  NumClustersLoop: do j = 1, c

!                 PRINT *, "j = ", j
```

```
                MinimumDistanceArray (1,j) = j

              LeaveOneOut: if (j .NE. i) then

                        XminusM (:,1) = m(:,i) - m(:,j)

!                   if (i .EQ. 1) then
!                       PRINT *, "XminusM at i = 1 and j = ", j
!                       PRINT *, int(XminusM (:,1),2)
!                   end if

                        MinimumDistanceArray (1,j) = j

!                       if (i .EQ. 1) then
!                       PRINT *, "MinimumDistanceArray (1,j) at i = 1"
!                       PRINT *, MinimumDistanceArray (1,j)
!                   end if

                        MinimumDistance1x1Array = &
                                    sqrt(MATMUL (TRANSPOSE (XminusM),
XminusM))

!                   if (i .EQ. 1) then
!                       PRINT *, "MinimumDistance at i = 1 and j = ",
j
!                       PRINT *, MinimumDistance1x1Array
!                   end if

                    MinimumDistanceArray (2,j) =
MinimumDistance1x1Array (1,1)

 !                  PRINT *, "MinimumDistanceArray (2,j) = ",
MinimumDistanceArray (2,j)

!                   if (i .EQ. 1) then
!                               PRINT *, "MinimumDistanceArray (:,:)
before      QuickSort at i = 1."
!                               PRINT *, MinimumDistanceArray
!                         end   if

                end if LeaveOneOut

            end DO NumClustersLoop

!             if (i .EQ. 1) then
!                 PRINT *, "MinimumDistanceArray (:,:) before QuickSort
at i = 1."
!                 PRINT *, MinimumDistanceArray
!             end if

!                 PRINT *, "j = ", j

                call QuickSort(MinimumDistanceArray, 1, c)

!           if (i .EQ. 1) then
!               PRINT *, "MinimumDistanceArray (2,:) after QuickSort
at i = 1."
```

```
!                   PRINT *, MinimumDistanceArray (2,:)
!               end if

                    MinimumDistanceArray3D (1,:) = MinimumDistanceArray
(1,:)
                    MinimumDistanceArray3D (2,:) = MinimumDistanceArray
(2,:)

                    AssignInfoClass: do h = 1, c

                        MinimumDistanceArray3D (3,h) =  sc_ic (
MinimumDistanceArray3D (1,h) )

                    end do AssignInfoClass

!               if (i .EQ. 1) then
!                   PRINT *, "MinimumDistanceArray3D (1,:), (2,:).  and
(3,:) after QuickSort at i = 1."
!                   PRINT *, MinimumDistanceArray3D (1,:)
!                   PRINT *, MinimumDistanceArray3D (2,:)
!                   PRINT *, MinimumDistanceArray3D (3,:)
!               end if

                    icx (i) = Find_kNN_Winner (kNN_k, NumInfoClasses,
MinimumDistanceArray3D)

!                if (i .EQ. 1) then
!                    PRINT *, "icx(1) = ", icx(1)
!                end if

!               if (icx (i) .EQ. 2) then
!                   PRINT *, "icx (", i, ") = ", 2
!               end if

                end if IsNotBackground

            end DO NumPixelsLoop

        end if kNot2

        NumCorrect = 0

!        PRINT *, "icx = ", icx

        FindAccuracy: do g = 1, c

            if (icx (g) .EQ. sc_ic (g)) then

                NumCorrect = NumCorrect + 1
            end if

        end do FindAccuracy

  !      PRINT *, "kNN_k = ", kNN_k
  !      PRINT *, "NumCorrect = ", NumCorrect
       Accuracy (kNN_k) = int((100 * NumCorrect / c), 2)
```

```
   !     PRINT *, "Accuracy (kNN_k) = ", Accuracy (kNN_k)

        end do kNN_k_Loop

        PRINT *, "Accuracy = ", Accuracy

        BestkNN_kArray = maxloc (Accuracy)
        BestkNN_k = BestkNN_kArray(1)

        PRINT *, "BestkNN_k = ", BestkNN_k

!        PRINT *, "All of image has been processed."

end function knnk

subroutine knnkb (c, m, NumberBands, MaxkNN_k, sc_ic, NumInfoClasses,
BestkNN_k, BestBandMask)

implicit none
        INTEGER (kind = 4), INTENT (IN) :: c
        INTEGER (kind = 2), INTENT (IN) :: m (:,:)
        REAL (KIND = 8) :: XminusM (size (m,1),1)
        REAL (KIND = 8) :: XminusM_NoMask (size (m,1),1)
        INTEGER (KIND = 2), INTENT (IN) :: MaxkNN_k
        INTEGER (KIND = 2) :: kNN_k
        INTEGER (KIND = 2), INTENT (IN) :: sc_ic (:)
        INTEGER (KIND = 2), INTENT (IN) :: NumInfoClasses
          INTEGER (KIND = 2), INTENT (OUT) :: BestkNN_k
          INTEGER (KIND = 2), INTENT (OUT) :: BestBandMask (:)
        INTEGER (KIND = 4) :: g, h,i,j
        INTEGER (kind = 4) :: MinimumDistance1x1Array (1,1)
        INTEGER (kind = 4) :: MinimumDistanceArray (2,c)
        INTEGER (kind = 4) :: MinimumDistanceArray3D (3,c)
        INTEGER (KIND = 2) :: icx (c)
        INTEGER (KIND = 2), INTENT (IN) :: NumberBands
        INTEGER (KIND = 2) :: Accuracy
        INTEGER (KIND = 2) :: BestAccuracy_SoFar
        INTEGER (KIND = 2) :: Bestk_SoFar
        INTEGER (KIND = 2) :: NumCorrect
        INTEGER (KIND = 2) :: BandMask (size(BestBandMask))
        INTEGER (KIND = 8) :: NumSets
        INTEGER (KIND = 8) :: ClassNumber
        INTEGER (KIND = 8) :: ClassNumberofBestAccuracy_SoFar
        CHARACTER (LEN = 40) :: TempString

        BandMask = 0
        ClassNumber = 0
!         PRINT *, BandMask

          BestAccuracy_SoFar = 0
        Bestk_SoFar = 0

          icx = 0
          Accuracy = 0
        NumSets = 2**NumberBands

        PRINT *, " "
```

```
        CALL i_to_s_left (NumSets, TempString)
        write (unit = *, fmt = *) "Number of possible band combinations =
", TempString

!       PRINT *, "c = ", c

!       PRINT *, "m (", j,i, ") = ", m (j,i)
!       PRINT *, "m (:,5) = ", m (:,5)
!        PRINT *, "m = ", m

!$OMP do
        BandCombinationLoop: do ClassNumber = 2, NumSets

            BandMask = ComputeBandMask (ClassNumber, NumberBands)

!        PRINT *, "BandMask for 64 = ", BandMask

            kNN_k_Loop: do kNN_k = 1, MaxkNN_k

            kNot2: if (kNN_k .NE. 2) then

                NumPixelsLoop: do i = 1, c

                MinimumDistanceArray (1,:) = 0
                MinimumDistanceArray (2,:) = 10000

    !           PRINT *, MinimumDistanceArray

    !                    ProgressStatements: if ((i / (k/4) .EQ. 1) .AND.
(modulo(i,(k/4)) .EQ. 0)) then
    !                        PRINT *, "One-fourth of image has been
processed."
    !                        else if ( ((i / (k/2) ) .EQ. 1) .AND.
(modulo(i,(k/2)) .EQ. 0) ) then
    !                            PRINT *, "One-half of image has been
processed."
    !                        else if ( ((i / ((3*k)/4) ) .EQ. 1) .AND.
(modulo(i,((3*k)/4)) .EQ. 0)) then
    !                            PRINT *, "Three-fourths of image has been
processed."
    !                    end if ProgressStatements

                      IsNotBackground: if ( m (1,i) .NE. 0 ) then

                      NumClustersLoop: do j = 1, c

    !                   PRINT *, "j = ", j

                        MinimumDistanceArray (1,j) = j

                      LeaveOneOut: if (j .NE. i) then

    !                            PRINT *, BandMask
    !                            PRINT *, m (:,i)

                                XMinusM_NoMask (:,1) = m(:,i) - m(:,j)
```

```
                              XminusM (:,1) = BandMask * XMinusM_NoMask
(:,1)

    !                       if ((i .EQ. 1) .AND. (j .EQ. 2)) then
    !                           PRINT *, "XminusM masked and not at i = 1
and j = 2"
    !                           PRINT *, int(XminusM (:,1),2)
    !                           PRINT *, int (XMinusM_NoMask (:,1),2)
    !                       end if

                                MinimumDistanceArray (1,j) = j

    !                             if (i .EQ. 1) then
    !                           PRINT *, "MinimumDistanceArray (1,j) at i
= 1"
    !                           PRINT *, MinimumDistanceArray (1,j)
    !                       end if

                                MinimumDistance1x1Array = &
                                            sqrt(MATMUL (TRANSPOSE
(XminusM), XminusM))

    !                       if ((i .EQ. 1) .AND. (j .EQ. 2)) then
    !                           PRINT *, "MinimumDistance after and before
mask at i = 1 and j = 2"
    !                           PRINT *, MinimumDistance1x1Array
    !                           PRINT *, sqrt(MATMUL (TRANSPOSE
(XMinusM_NoMask), XMinusM_NoMask))
    !                       end if

                            MinimumDistanceArray (2,j) =
MinimumDistance1x1Array (1,1)

    !                       PRINT *, "MinimumDistanceArray (2,j) = ",
MinimumDistanceArray (2,j)

    !                       if (i .EQ. 1) then
    !                               PRINT *, "MinimumDistanceArray (:,:)
before      QuickSort at i = 1."
    !                               PRINT *, MinimumDistanceArray
    !                         end   if

                        end if LeaveOneOut

                    end DO NumClustersLoop

    !                if (i .EQ. 1) then
    !                    PRINT *, "MinimumDistanceArray (:,:) before
QuickSort at i = 1."
    !                    PRINT *, MinimumDistanceArray
    !                end if

    !                    PRINT *, "j = ", j

                        call QuickSort(MinimumDistanceArray, 1, c)

    !               if (i .EQ. 1) then
```

```
    !                   PRINT *, "MinimumDistanceArray (2,:) after
QuickSort at i = 1."
    !                   PRINT *, MinimumDistanceArray (2,:)
    !               end if

                        MinimumDistanceArray3D (1,:) =
MinimumDistanceArray (1,:)
                        MinimumDistanceArray3D (2,:) =
MinimumDistanceArray (2,:)

                        AssignInfoClass: do h = 1, c

                            MinimumDistanceArray3D (3,h) =  sc_ic (
MinimumDistanceArray3D (1,h) )

                        end do AssignInfoClass

    !               if (i .EQ. 1) then
    !                   PRINT *, "MinimumDistanceArray3D (1,:), (2,:).
and (3,:) after QuickSort at i = 1."
    !                   PRINT *, MinimumDistanceArray3D (1,:)
    !                   PRINT *, MinimumDistanceArray3D (2,:)
    !                   PRINT *, MinimumDistanceArray3D (3,:)
    !               end if

                        icx (i) = Find_kNN_Winner (kNN_k, NumInfoClasses,
MinimumDistanceArray3D)

    !                if (i .EQ. 1) then
    !                    PRINT *, "icx(1) = ", icx(1)
    !                end if

    !               if (icx (i) .EQ. 2) then
    !                   PRINT *, "icx (", i, ") = ", 2
    !               end if

                    end if IsNotBackground

                end DO NumPixelsLoop

            end if kNot2

            NumCorrect = 0

    !        PRINT *, "icx = ", icx

            FindAccuracy: do g = 1, c

                if (icx (g) .EQ. sc_ic (g)) then

                    NumCorrect = NumCorrect + 1
                end if

            end do FindAccuracy

!          PRINT *, "kNN_k = ", kNN_k
!          PRINT *, "ClassNumber = ", ClassNumber
```

```
    !      PRINT *, "NumCorrect = ", NumCorrect
           Accuracy = int((100 * NumCorrect / c), 2)

            if (Accuracy .GT. BestAccuracy_SoFar) then

                BestAccuracy_SoFar = Accuracy
                Bestk_SoFar = kNN_k
                ClassNumberofBestAccuracy_SoFar = ClassNumber

            end if

    !     PRINT *, "Accuracy (kNN_k, ClassNumber) = ", Accuracy (kNN_k)

            end do kNN_k_Loop

        end do BandCombinationLoop
!$OMP end do

!        PRINT *, "Accuracy = ", Accuracy

        BestkNN_k = Bestk_SoFar
        BestBandMask = ComputeBandMask (ClassNumberofBestAccuracy_SoFar,
NumberBands)

        write (unit = *, fmt = "(A19,I2,A9)") "Best accuracy was ",
BestAccuracy_SoFar, " percent."
        PRINT *, "Best k was", BestkNN_k
        PRINT *, "Best band mask is", BestBandMask

!        PRINT *, "All of image has been processed."

end subroutine knnkb

subroutine nn (b, k, c, x, m, scx, DistanceImage)

implicit none
        INTEGER (kind = 2), INTENT (IN) :: b
        INTEGER (kind = 4), INTENT (IN) :: k,c
        REAL (kind = 8), INTENT (IN) :: m (b,c)
        REAL (KIND = 8) :: XminusM (b,1)
        REAL (kind = 8) :: MinimumDistance1x1Array (1,1)
        REAL (kind = 8) :: MinimumDistance
        INTEGER (kind = 2) :: ClosestSpectralClass
        INTEGER (KIND = 2), INTENT (IN) :: x (b,k)
          INTEGER (KIND = 2), INTENT (OUT) :: scx (k)
        INTEGER (KIND = 4) :: i,j
        REAL (KIND = 8), INTENT (OUT) :: DistanceImage (k)

      MinimumDistance = 1000000.0
      scx = 0
      DistanceImage = 0.0

        NumPixelsLoop: do i = 1, k

                ClosestSpectralClass = 0

                  IsNotBackground: if ( x (1,i) .NE. 0 ) then
```

```
                  NumClustersLoop: do j = 1, c

                        XminusM (:,1) = x(:,i) - m(:,j)
                        MinimumDistance1x1Array = &
                                    MATMUL (TRANSPOSE (XminusM), &
                                        XminusM)

                        if (MinimumDistance1x1Array (1,1) .LT. &
                                    MinimumDistance ) then

                                    MinimumDistance =
MinimumDistance1x1Array (1,1)
                                    ClosestSpectralClass = j
                              end if

                  end DO NumClustersLoop

                    DistanceImage (i) = MinimumDistance
                  MinimumDistance = 1000000.0

                  end if IsNotBackground

                  scx (i) = ClosestSpectralClass
        end DO NumPixelsLoop

end subroutine nn

recursive function Find_kNN_Winner (kNN_k, NumInfoClasses,
MinimumDistanceArray3D) result (Winner)

implicit none

        INTEGER (KIND = 2) :: Winner
        INTEGER (KIND = 2), INTENT (IN) :: kNN_k
        INTEGER (KIND = 2), INTENT (IN) :: NumInfoClasses
        INTEGER (kind = 4), INTENT (IN) :: MinimumDistanceArray3D (:,:)
        INTEGER (KIND = 4) :: g, h, Maximum, NumberMaxima
        INTEGER (KIND = 2) :: NewkNN_k
        INTEGER (KIND = 2) :: kNN_Count (NumInfoClasses)
        INTEGER (KIND = 2) :: Winner1x1Array (1)

!       PRINT *, "kNN_k = ", kNN_k

        kNN_Count = 0

        CountInfoClasses: do h = 1, kNN_k

            kNN_Count (MinimumDistanceArray3D (3,h)) = kNN_Count
(MinimumDistanceArray3D (3,h)) + 1

        end DO CountInfoClasses

         Maximum = maxval (kNN_Count)
         NumberMaxima = 0

         LookforTie: do g = 1, NumInfoClasses
```

```
            if (kNN_Count (g) .EQ. Maximum) then

                        NumberMaxima = NumberMaxima + 1

            end IF

            end DO LookforTie

               CheckIfTie: if (NumberMaxima .EQ. 1) then

                    Winner1x1Array = maxloc (kNN_Count)
                    Winner = Winner1x1Array (1)

               else

                    NewkNN_k = kNN_k -1
                    Winner = Find_kNN_Winner (NewkNN_k, NumInfoClasses,
MinimumDistanceArray3D)

               end IF CheckIfTie

end function Find_kNN_Winner

subroutine Recode (k, scx, sc_ic, icx)

        implicit none

        INTEGER (KIND = 4), INTENT (IN) :: k
        INTEGER (KIND = 2), INTENT (OUT) :: icx (k)
        INTEGER (KIND = 2), INTENT (IN) :: scx (k)
        INTEGER (KIND = 4), INTENT (IN) :: sc_ic (:)
        INTEGER (KIND = 4) :: i

        icx = 0

        NumPixelsLoop: do i = 1, k

            IsNotBackground: if ( scx (i) .NE. 0 ) then

                icx (i) = sc_ic (scx (i))

            end if IsNotBackground

        end DO NumPixelsLoop

end subroutine Recode

subroutine ComputeAccuracyPlusAreaEstimate (k, x, NumPoints, AccPairs,
Acres,&
    ErrorMatrix, ForestAreaEstimate, ForestMapMarginal,
PercentOverallAccuracy,&
    StandardError, PrecisionFAE, AAPoints)

        implicit none

        INTEGER (kind = 4), INTENT (IN) :: k, NumPoints
```

```
  INTEGER (KIND = 2), INTENT (INOUT) :: x (k)
  INTEGER (KIND = 4), INTENT (INOUT) :: AccPairs (NumPoints,2)
  INTEGER (KIND = 4), INTENT (OUT) :: ErrorMatrix (2,2)
  REAL (KIND = 8), INTENT(IN) :: Acres
  REAL (KIND = 8), INTENT (OUT) :: ForestAreaEstimate, StandardError
  REAL (KIND = 8), INTENT (OUT) :: ForestMapMarginal
  REAL (KIND = 8), INTENT (OUT) :: PercentOverallAccuracy
  REAL (KIND = 8), INTENT (OUT) :: PrecisionFAE
  INTEGER (KIND = 4), INTENT (OUT) :: AAPoints (NumPoints,2)
  INTEGER (KIND = 2) :: DenomF
  INTEGER (KIND = 2) :: DenomNf
  REAL (KIND = 8) :: NumPointsReal = 0.0, NumAccurateReal = 0.0
  REAL (KIND = 8) :: NumNonzeroPixels = 0.0
  INTEGER (KIND = 4) :: i, CaseProduct = 0
  REAL (KIND = 8) :: NumNonforestPixels = 0.0
  REAL (KIND = 8) :: NumForestPixels = 0.0
  REAL (KIND = 8) :: NonforestMapMarginal = 0.0
  REAL (KIND = 8) :: PfRf = 0.0
  REAL (KIND = 8) :: PnfRf = 0.0
  REAL (KIND = 8) :: FmmxPf = 0.0
  REAL (KIND = 8) :: NfmmxPnf = 0.0

  ErrorMatrix = 0
  ForestAreaEstimate = 0.0
PrecisionFAE = 0.0
  StandardError = 0.0
  ForestMapMarginal = 0.0
  PercentOverallAccuracy = 0.0
  PfRf = 0.0
  PnfRf = 0.0
  FmmxPf = 0.0
  NfmmxPnf = 0.0
  DenomF = 0
  DenomNf = 0
  AAPoints = 0.0

  ConvertToThreesAndFours: do i = 1, NumPoints


      OneToThree: if ( AccPairs (i,1) .EQ. 1 ) then

          AccPairs (i,1) = 3

      end if OneToThree

      TwoToFour: if (AccPairs (i,1) .EQ. 2) then

          AccPairs (i,1) = 4

      end if TwotoFour

  end do ConvertToThreesAndFours

  NumPointsLoop: do i = 1, NumPoints

      IsNotBackground: if ( x ( AccPairs (i,2) ) .NE. 0 ) then
```

```
        CaseProduct =  x ( AccPairs (i,2) ) * AccPairs (i,1)

          select case (CaseProduct)

            case (3)

                ErrorMatrix (1,1) = ErrorMatrix (1,1) + 1

            case (4)

                ErrorMatrix (2,1) = ErrorMatrix (2,1) + 1

            case (6)

                ErrorMatrix (1,2) = ErrorMatrix (1,2) + 1

            case (8)

                ErrorMatrix (2,2) = ErrorMatrix (2,2) + 1

        end select

        AAPoints (i,1) = AccPairs (i,2)
        AAPoints (i,2) = CaseProduct


    end if IsNotBackground

end DO NumPointsLoop

NumAccurateReal = ErrorMatrix (1,1) + ErrorMatrix (2,2)
NumPointsReal = NumPoints
PercentOverallAccuracy = 100 * NumAccurateReal / NumPointsReal

PRINT *, "NumAccurateReal =  ", NumAccurateReal
PRINT *, "NumPointsReal =  ", NumPointsReal


NumNonzeroPixels = 0.0
NumForestPixels = 0.0
NumNonforestPixels = 0.0

CountForestPixels: do i = 1, k

          if ( x (i) .NE. 0 ) then

    NumNonzeroPixels = NumNonzeroPixels + 1.0

          IfForestPixel: if (x(i) .EQ. 2) then

                NumForestPixels = NumForestPixels + 1

                end if IfForestPixel

          IfNonforestPixel: if (x(i) .EQ. 1) then

                NumNonforestPixels = NumNonforestPixels + 1
```

```
                    end if IfNonforestPixel

        end if

        end do CountForestPixels

        ForestMapMarginal = NumForestPixels / (NumForestPixels +
NumNonforestPixels)

        NonforestMapMarginal = NumNonforestPixels / (NumForestPixels +
NumNonforestPixels)

        DenomF = ErrorMatrix (1,2) + ErrorMatrix (2,2)
        DenomNf = ErrorMatrix (1,1) + ErrorMatrix (2,1)

        if (DenomF .NE. 0) then
            PfRf = ErrorMatrix (2,2) / real (DenomF, 4)
        end if

        if (DenomNf .NE. 0) then
            PnfRf = ErrorMatrix (2,1) / real (DenomNf, 4)
        end if

        FmmxPf = ForestMapMarginal * PfRf
        NfmmxPnf = NonforestMapMarginal * PnfRf

        ForestAreaEstimate = (ForestMapMarginal * PfRf) &
            + (NonforestMapMarginal * PnfRf)

        StandardError = (((ForestMapMarginal - FmmxPf) * FmmxPf) / &
                (ForestMapMarginal * NumPoints)) + (((NonforestMapMarginal
- NfmmxPnf) * &
                 NfmmxPnf) / (NonforestMapMarginal * NumPoints))

        PrecisionFAE =
(SQRT(StandardError)*(SQRT(((ForestAreaEstimate)*Acres)/1000000)))

end subroutine ComputeAccuracyPlusAreaEstimate

! Copy of above added for use when using previously calculated forest map
marginal
! with a new error matrix.

subroutine ComputeAccPlusAreaEstP2 (k, x, NumPoints, AccPairs,
ForestMapMarginal,  &
   ErrorMatrix, ForestAreaEstimate, PercentOverallAccuracy, StandardError,
AAPoints)

        implicit none

        INTEGER (kind = 4), INTENT (IN) :: k, NumPoints
        INTEGER (KIND = 2), INTENT (INOUT) :: x (k)
        INTEGER (KIND = 4), INTENT (INOUT) :: AccPairs (NumPoints,2)
        INTEGER (KIND = 4), INTENT (OUT) :: ErrorMatrix (2,2)
        REAL (KIND = 8), INTENT (OUT) :: ForestAreaEstimate, StandardError
        REAL (KIND = 8), INTENT (INOUT) :: ForestMapMarginal
```

```
  REAL (KIND = 8), INTENT (OUT) :: PercentOverallAccuracy
  INTEGER (KIND = 4), INTENT (OUT) :: AAPoints (NumPoints,2)
  INTEGER (KIND = 2) :: DenomF
  INTEGER (KIND = 2) :: DenomNf
  REAL (KIND = 8) :: NumPointsReal = 0.0, NumAccurateReal = 0.0
  INTEGER (KIND = 4) :: i, CaseProduct = 0
  REAL (KIND = 8) :: NonforestMapMarginal = 0.0
  REAL (KIND = 8) :: PfRf = 0.0
  REAL (KIND = 8) :: PnfRf = 0.0
  REAL (KIND = 8) :: FmmxPf = 0.0
  REAL (KIND = 8) :: NfmmxPnf = 0.0

  ErrorMatrix = 0
  ForestAreaEstimate = 0.0
StandardError = 0.0
PercentOverallAccuracy = 0.0
  PfRf = 0.0
  PnfRf = 0.0
  FmmxPf = 0.0
  NfmmxPnf = 0.0
  DenomF = 0
  DenomNf = 0
  AAPoints = 0.0

  ConvertToThreesAndFours: do i = 1, NumPoints


      OneToThree: if ( AccPairs (i,1) .EQ. 1 ) then

          AccPairs (i,1) = 3

      end if OneToThree

      TwoToFour: if (AccPairs (i,1) .EQ. 2) then

          AccPairs (i,1) = 4

      end if TwotoFour

  end do ConvertToThreesAndFours

  NumPointsLoop: do i = 1, NumPoints

      IsNotBackground: if ( x ( AccPairs (i,2) ) .NE. 0 ) then

          CaseProduct =  x ( AccPairs (i,2) ) * AccPairs (i,1)

            select case (CaseProduct)

              case (3)

                  ErrorMatrix (1,1) = ErrorMatrix (1,1) + 1

              case (4)

                  ErrorMatrix (2,1) = ErrorMatrix (2,1) + 1
```

```
                case (6)

                    ErrorMatrix (1,2) = ErrorMatrix (1,2) + 1

                case (8)

                    ErrorMatrix (2,2) = ErrorMatrix (2,2) + 1

            end select

            AAPoints (i,1) = AccPairs (i,2)
            AAPoints (i,2) = CaseProduct


        end if IsNotBackground

    end DO NumPointsLoop

    NumAccurateReal = ErrorMatrix (1,1) + ErrorMatrix (2,2)
    NumPointsReal = NumPoints
    PercentOverallAccuracy = 100 * NumAccurateReal / NumPointsReal

    PRINT *, "NumAccurateReal =  ", NumAccurateReal
    PRINT *, "NumPointsReal =  ", NumPointsReal

    NonforestMapMarginal = (1 - ForestMapMarginal)

    DenomF = ErrorMatrix (1,2) + ErrorMatrix (2,2)
    DenomNf = ErrorMatrix (1,1) + ErrorMatrix (2,1)

    if (DenomF .NE. 0) then
        PfRf = ErrorMatrix (2,2) / real (DenomF, 4)
    end if

    if (DenomNf .NE. 0) then
        PnfRf = ErrorMatrix (2,1) / real (DenomNf, 4)
    end if

    FmmxPf = ForestMapMarginal * PfRf
    NfmmxPnf = NonforestMapMarginal * PnfRf

    ForestAreaEstimate = (ForestMapMarginal * PfRf) &
        + (NonforestMapMarginal * PnfRf)

    StandardError = (((ForestMapMarginal - FmmxPf) * FmmxPf) / &
            (ForestMapMarginal * NumPoints)) + (((NonforestMapMarginal
- NfmmxPnf) * &
             NfmmxPnf) / (NonforestMapMarginal * NumPoints))


end subroutine ComputeAccPlusAreaEstP2


subroutine ComputeAccuracyOnly (k, x, NumPoints, AccPairs, ErrorMatrix, &
    PercentOverallAccuracy, AAPoints)

    implicit none
```

```
INTEGER (kind = 4), INTENT (IN) :: k, NumPoints
INTEGER (KIND = 2), INTENT (INOUT) :: x (k)
INTEGER (KIND = 4), INTENT (INOUT) :: AccPairs (NumPoints,2)
INTEGER (KIND = 4), INTENT (OUT) :: ErrorMatrix (2,2)
REAL (KIND = 8), INTENT (OUT) :: PercentOverallAccuracy
REAL (KIND = 8) :: NumPointsReal = 0.0, NumAccurateReal = 0.0
INTEGER (KIND = 4) :: i, CaseProduct = 0
INTEGER (KIND = 4), INTENT (OUT) :: AAPoints (NumPoints,2)

ErrorMatrix = 0
PercentOverallAccuracy = 0.0
AAPoints = 0.0

ConvertToThreesAndFours: do i = 1, NumPoints

    OneToThree: if ( AccPairs (i,1) .EQ. 1 ) then

        AccPairs (i,1) = 3

    end if OneToThree

    TwoToFour: if (AccPairs (i,1) .EQ. 2) then

        AccPairs (i,1) = 4

    end if TwotoFour

end do ConvertToThreesAndFours

NumPointsLoop: do i = 1, NumPoints

    IsNotBackground: if ( x (i) .NE. 0 ) then

        CaseProduct =  x (i) * AccPairs (i,1)

          select case (CaseProduct)

            case (3)

                ErrorMatrix (1,1) = ErrorMatrix (1,1) + 1

            case (4)

                ErrorMatrix (2,1) = ErrorMatrix (2,1) + 1

            case (6)

                ErrorMatrix (1,2) = ErrorMatrix (1,2) + 1

            case (8)

                ErrorMatrix (2,2) = ErrorMatrix (2,2) + 1

        end select

        AAPoints (i,1) = AccPairs (i,2)
```

```
                AAPoints (i,2) = CaseProduct

            end if IsNotBackground

        end DO NumPointsLoop

!               PRINT *, "Error Matrix 1,1 = ", ErrorMatrix (1,1)
!               PRINT *, "Error Matrix 2,1 = ", ErrorMatrix (2,1)
!               PRINT *, "Error Matrix 1,2 = ", ErrorMatrix (1,2)
!               PRINT *, "Error Matrix 2,2 = ", ErrorMatrix (2,2)

        NumAccurateReal = ErrorMatrix (1,1) + ErrorMatrix (2,2)
        NumPointsReal = NumPoints
        PercentOverallAccuracy = 100 * NumAccurateReal / NumPointsReal

        PRINT *, "NumAccurateReal =  ", NumAccurateReal
        PRINT *, "NumPointsReal =  ", NumPointsReal
        PRINT *, "Percent Overall Accuracy = ", PercentOverallAccuracy

end subroutine ComputeAccuracyOnly

subroutine ComputeMcRobertsAreaEstimate (k, x, NumPoints, AccPairs,
ForestMapMarginal, &
        PlotProportionForest, Acres, ForestCount, NonforestCount,
WeightedProportion, &
        ForestArea, VariancesSum, VarianceForestArea, FAEPrecision)

        INTEGER (kind = 4), INTENT (IN) :: k, NumPoints
        INTEGER (KIND = 2), INTENT (INOUT) :: x (k)
        INTEGER (KIND = 4), INTENT (INOUT) :: AccPairs (NumPoints,2)
        REAL (KIND = 8), INTENT (INOUT) :: ForestMapMarginal
        REAL (KIND = 8), INTENT (INOUT) :: PlotProportionForest
(NumPoints)
        REAL (KIND = 8), INTENT (IN) :: Acres
          INTEGER (kind = 4), INTENT (IN) :: ForestCount
        INTEGER (kind = 4), INTENT (IN) :: NonforestCount
        INTEGER (KIND = 4) :: i, j
        INTEGER (KIND = 4) :: FCount, NfCount
        REAL (KIND = 8) :: ForestCountReal
          REAL (KIND = 8) :: NonforestCountReal
        REAL (KIND = 8) :: NonforestMapMarginal = 0.0
        REAL (KIND = 8) :: PlotsInForest (ForestCount)
        REAL (KIND = 8) :: PlotsInNonforest (NonforestCount)
        REAL (KIND = 8) :: ForestSumPlotProp = 0.0
        REAL (KIND = 8) :: NonforestSumPlotProp = 0.0
        REAL (KIND = 8) :: MeanPlotPropForest = 0.0
        REAL (KIND = 8) :: MeanPlotPropNonforest = 0.0
        REAL (KIND = 8), INTENT (OUT) :: WeightedProportion
        REAL (KIND = 8), INTENT (OUT) :: ForestArea
        REAL (KIND = 8) :: SqDiffPlotsInForest (ForestCount)
        REAL (KIND = 8) :: SqDiffPlotsInNonforest (NonforestCount)
        REAL (KIND = 8) :: ForestSumSqDiff = 0.0
        REAL (KIND = 8) :: NonforestSumSqDiff = 0.0
        REAL (KIND = 8) :: ForestStratumVariance = 0.0
        REAL (KIND = 8) :: NonforestStratumVariance = 0.0
        REAL (KIND = 8), INTENT (OUT) :: VariancesSum
        REAL (KIND = 8), INTENT (OUT) :: VarianceForestArea
```

```
        REAL (KIND = 8), INTENT (OUT) :: FAEPrecision

        FCount = 0
        NfCount = 0
        ForestCountReal = 0.0
        NonforestCountReal = 0.0
        WeightedProportion = 0.0
        ForestArea = 0.0
        VariancesSum = 0.0
        VarianceForestArea = 0.0
        FAEPrecision = 0.0

        NonforestMapMarginal = (1 - ForestMapMarginal)

        PlotsByClass: do j = 1, NumPoints

                  if ( x (AccPairs (j,2)) .NE. 0 ) then

                  IfForestPixel: if (x (AccPairs (j,2)) .EQ. 2) then

                                FCount = FCount + 1
                        PlotsInForest (FCount) = PlotProportionForest (j)


                        end if IfForestPixel

                  IfNonforestPixel: if (x (AccPairs (j,2)) .EQ. 1) then

                                NfCount = NfCount + 1
                        PlotsInNonforest (NfCount) = PlotProportionForest
(j)

                        end if IfNonforestPixel

                  end if

        end do PlotsByClass

        PRINT *, "FCount = ", FCount
        PRINT *, "NfCount = ", NfCount

                    do i = 1, ForestCount

                  ForestSumPlotProp = ForestSumPlotProp + PlotsInForest
(i)

                    end do

                    do i = 1, NonforestCount

                  NonforestSumPlotProp = NonforestSumPlotProp +
PlotsInNonforest (i)

                    end do

       PRINT *, "ForestSumPlotProp = ", ForestSumPlotProp
       PRINT *, "NonforestSumPlotProp = ", NonforestSumPlotProp
```

```
       ForestCountReal = ForestCount
       NonforestCountReal = NonforestCount

       MeanPlotPropForest = ForestSumPlotProp / ForestCountReal
       MeanPlotPropNonforest = NonforestSumPlotProp / NonforestCountReal

       PRINT *, "MeanPlotPropForest = ", MeanPlotPropForest
       PRINT *, "MeanPlotPropNonforest = ", MeanPlotPropNonforest

       WeightedProportion = ((ForestMapMarginal * MeanPlotPropForest) + &
                  (NonforestMapMarginal * MeanPlotPropNonforest))

       PRINT *, "WeightedProportion = ", WeightedProportion

       ForestArea = Acres * WeightedProportion

       PRINT *, "ForestArea = ", ForestArea

        do i = 1, ForestCount

              SqDiffPlotsInForest (i) = ((PlotsInForest (i) -
MeanPlotPropForest) * &
                (PlotsInForest (i) - MeanPlotPropForest))

          end do

        do i = 1, NonforestCount

              SqDiffPlotsInNonforest (i) = ((PlotsInNonforest (i) -
MeanPlotPropNonforest) * &
                (PlotsInNonforest (i) - MeanPlotPropNonforest))

          end do

                    do i = 1, ForestCount

                  ForestSumSqDiff = ForestSumSqDiff + SqDiffPlotsInForest
(i)

                    end do

                    do i = 1, NonforestCount

                  NonforestSumSqDiff = NonforestSumSqDiff +
SqDiffPlotsInNonforest (i)

                    end do

          PRINT *, "ForestSumSqDiff = ", ForestSumSqDiff
          PRINT *, "NonforestSumSqDiff = ", NonforestSumSqDiff

          ForestStratumVariance = ((1 / (ForestCountReal - 1)) *
ForestSumSqDiff)
          NonforestStratumVariance = ((1 / (NonforestCountReal - 1)) *
NonforestSumSqDiff)
```

```
          PRINT *, "Forest Map Marginal = ", ForestMapMarginal
          PRINT *, "Nonforest Map Marginal = ", NonforestMapMarginal
          PRINT *, "Forest Stratum Variance = ", ForestStratumVariance
          PRINT *, "Nonforest Stratum Variance = ",
NonforestStratumVariance
          PRINT *, "Forest Count Real = ", ForestCountReal
          PRINT *, "Nonforest Count Real = ", NonforestCountReal

          VariancesSum = (((ForestMapMarginal * ForestMapMarginal) *
ForestStratumVariance) / &
                ForestCountReal) + (((NonforestMapMarginal *
NonforestMapMarginal) *  &
                NonforestStratumVariance) /NonforestCountReal)

          VarianceForestArea = (Acres * Acres) * VariancesSum

          FAEPrecision = ((SQRT(VarianceForestArea)) / ForestArea) *
SQRT(ForestArea / 1000000)

          PRINT *, "Acres = ", Acres
          PRINT *, "ForestArea = ", ForestArea
          PRINT *, "VariancesSum = ", VariancesSum
          PRINT *, "VarianceForestArea = ", VarianceForestArea
          PRINT *, " FAEPrecision = ", FAEPrecision

end subroutine ComputeMcRobertsAreaEstimate

function ComputeBandMask (ClassNumber, NumBands) result (BandMask)

    implicit none

    INTEGER (KIND = 8), INTENT (IN) :: ClassNumber
    INTEGER (KIND = 2), INTENT (IN) :: NumBands
    INTEGER (KIND = 2) :: BandMask(NumBands)
    INTEGER (KIND = 8) :: Quotient
    INTEGER (KIND = 8) :: Remainder
    INTEGER (KIND = 8) :: i

    BandMask = 0
    Quotient = ClassNumber - 1

    do i = 1, NumBands

 !           PRINT *, "i = ", i
 !           PRINT *, "Quotient = ", Quotient

            Remainder = mod (Quotient,int(2,8))

 !           PRINT *, "Remainder = ", Remainder

            BandMask (NumBands - i + 1) = Remainder

 !           PRINT *, "BandMask = ", BandMask

            Quotient = Quotient / 2

    end do
```

```
end function ComputeBandMask

subroutine i_to_s_left ( intval, s )

!**************************************************************************
*****
!
!! I_TO_S_LEFT converts an integer to a left-justified string.
!
!  Examples:
!
!    Assume that S is 6 characters long:
!
!    INTVAL  S
!
!         1  1
!        -1  -1
!         0  0
!      1952  1952
!    123456  123456
!   1234567  ******  <-- Not enough room!
!
!  Modified:
!
!    28 July 2000
!
!  Author:
!
!    John Burkardt
!
!  Parameters:
!
!    Input, integer INTVAL, an integer to be converted.
!
!    Output, character ( len = * ) S, the representation of the integer.
!    The integer will be left-justified.  If there is not enough space,
!    the string will be filled with stars.
!
  implicit none

  character :: c
  integer :: i
  integer :: idig
  integer :: ihi
  integer :: ilo
  integer (KIND = 8), INTENT (IN) :: intval
  integer ipos
  integer ival
  character (len = 40), INTENT (OUT) :: s

  s = ' '

  ilo = 1
  ihi = len ( s )

  if ( ihi <= 0 ) then
```

```
    return
  end if
!
!  Make a copy of the integer.
!
  ival = intval
!
!  Handle the negative sign.
!
  if ( ival < 0 ) then

    if ( ihi <= 1 ) then
      s(1:1) = '*'
      return
    end if

    ival = -ival
    s(1:1) = '-'
    ilo = 2

  end if
!
!  The absolute value of the integer goes into S(ILO:IHI).
!
  ipos = ihi
!
!  Find the last digit of IVAL, strip it off, and stick it into the
string.
!
  do

    idig = mod ( ival, 10 )
    ival = ival / 10

    if ( ipos < ilo ) then
      do i = 1, ihi
        s(i:i) = '*'
      end do
      return
    end if

    call digit_to_ch ( idig, c )

    s(ipos:ipos) = c
    ipos = ipos - 1

    if ( ival == 0 ) then
      exit
    end if

  end do
!
!  Shift the string to the left.
!
  s(ilo:ilo+ihi-ipos-1) = s(ipos+1:ihi)
  s(ilo+ihi-ipos:ihi) = ' '
```

```
  return
end subroutine i_to_s_left

subroutine digit_to_ch ( digit, c )

!************************************************************************
******
!
!! DIGIT_TO_CH returns the character representation of a decimal digit.
!
!  Example:
!
!    DIGIT   C
!    -----  ---
!      0    '0'
!      1    '1'
!    ...    ...
!      9    '9'
!     17    '*'
!
!  Modified:
!
!    04 August 1999
!
!  Author:
!
!    John Burkardt
!
!  Parameters:
!
!    Input, integer DIGIT, the digit value between 0 and 9.
!
!    Output, character C, the corresponding character.
!
  implicit none

  character, INTENT (OUT) :: c
  integer, INTENT (IN) :: digit

  if ( 0 <= digit .and. digit <= 9 ) then

    c = char ( digit + 48 )

  else

    c = '*'

  end if

  return
end subroutine digit_to_ch

end module CEARSwLib
```

APPENDIX VIII
Fortran 95 *k*-nearest neighbor program

```
!     Last change:  RHW  3 Feb 2005    8:40 pm
!
! kNN
!
! Randolph Hamilton Wynne, Department of Forestry, Virginia Polytechnic
Institute and State University
! Christine Elizabeth Blinn, Department of Forestry, Virginia Polytechnic
Institute and State University
! Katherine Amanda Joseph, Department of Forestry, Virginia Polytechnic
Institute and State University
!
! Version 0.1
!
! Program start date: February 3, 2005 (based on ksbootfa8)
! Inputs:
!
! Outputs:

program kNearestNeighborkb

        USE CEARSwLib

        implicit none

        INTEGER, parameter :: S_LEN = 32
        INTEGER (KIND = 4) :: i, j ! Loop counters
        INTEGER (KIND = 4) :: NumTrainingPoints, k, NumRows, NumCols,
Pixel
        INTEGER (KIND = 4) :: FileX, FileY
        INTEGER (KIND = 2) :: b, kNN_k, NumInfoClasses, MaxkNN_k
        INTEGER (KIND = 2), ALLOCATABLE :: icx (:)
        INTEGER (KIND = 2), ALLOCATABLE :: x (:,:)
        INTEGER (KIND = 2), ALLOCATABLE :: InfoClasses (:)
        INTEGER (KIND = 2), ALLOCATABLE :: m (:,:)
        INTEGER (KIND = 2), ALLOCATABLE :: BestBandMask (:)
        REAL (KIND = 8) :: UpperLeftX, UpperLeftY, PixelSize, TempX, TempY
        CHARACTER (LEN = S_LEN) :: InputFileNameImage
        CHARACTER (LEN = S_LEN) :: OutputFileNameRoot
        CHARACTER (LEN = S_LEN) :: OutputImageFileName
        CHARACTER (LEN = S_LEN) :: OutputHeaderFileName
        CHARACTER (LEN = S_LEN) :: InputFileNameTrainingData

            open (unit = 10, file='parameters.in', status = "old", action
= "read")
            read (unit = 10, fmt = *) InputFileNameImage
!       PRINT *, "Input File Name Raw Image: ", InputFileNameImage

            read (unit = 10, fmt = *) NumRows
!         PRINT *, "Number of Rows in Raw Image: ", NumRows

            read (unit = 10, fmt = *) NumCols
!         PRINT *, "Number of Columns in Raw Image: ", NumCols
```

```
            read (unit = 10, fmt = *) b
!       PRINT *, "Number of Bands: ", b

        read (unit = 10, FMT = *) InputFileNameTrainingData
!       PRINT *, "Training Data File Name: ", InputFileNameTrainingData

            read (unit = 10, fmt = *) NumTrainingPoints
!       PRINT *, "Number of Training Points: ", NumTrainingPoints

            read (unit = 10, fmt = *) NumInfoClasses
!       PRINT *, "Number of Information Classes: ", NumInfoClasses


            read (unit = 10, fmt = *) MaxkNN_k
!       PRINT *, "Maximum number of nearest neighbors: ", kNN_k

            read (unit = 10, fmt = *) OutputFileNameRoot
!         PRINT *, "Output File Name: ", OutputFileNameRoot

            read (unit = 10, fmt = *) UpperLeftX
!         PRINT *, "Output File Name: ", OutputFileNameRoot

            read (unit = 10, fmt = *) UpperLeftY
!         PRINT *, "Output File Name: ", OutputFileNameRoot

            read (unit = 10, fmt = *) PixelSize
!         PRINT *, "Output File Name: ", OutputFileNameRoot

            k = NumRows * NumCols
        OutputImageFileName = TRIM (OutputFileNameRoot) // ".bip"
        OutputHeaderFileName = TRIM (OutputFileNameRoot) // ".hdr"

        write (unit = *, FMT = *) "Input File Names: "
        write (unit = *, FMT = *) "    ", InputFileNameImage, " (Unit =
11)"
        write (unit = *, FMT = *) "    ", InputFileNameTrainingData, "
(Unit = 1)"
        write (unit = *, FMT = *) "    ", "parameters.in (Unit = 10)"

        write (unit = *, FMT = *) "Number of Bands: "
        write (unit = *, FMT = *) "    ", b

        write (unit = *, FMT = *) "Number of Pixels: "
        write (unit = *, FMT = *) "    ", k

        write (unit = *, FMT = *) "Number of Information Classes: "
        write (unit = *, FMT = *) "    ", NumInfoClasses

        write (unit = *, FMT = *) "Number of Training Points: "
        write (unit = *, FMT = *) "    ", NumTrainingPoints

        write (unit = *, FMT = *) "Upper Left X, Upper Left Y, Pixel
Size:"
        write (unit = *, FMT = *) "    ", UpperLeftX
        write (unit = *, FMT = *) "    ", UpperLeftY
        write (unit = *, FMT = *) "    ", PixelSize
```

```
        write (unit = *, FMT = *) "Maximum Number of Nearest Neighbors: "
        write (unit = *, FMT = *) "    ", MaxkNN_k

        write (unit = *, FMT = *) "Output File Names: "
        write (unit = *, FMT = *) "    ", OutputImageFileName, " (Unit =
2)"
        write (unit = *, FMT = *) "    ", OutputHeaderFileName, " (Unit =
3)"

        ALLOCATE (x (b,k))

        open (unit = 11, file = InputFileNameImage, &
              form = "binary", status = "old", action = "read")

        write (unit = *, fmt = *) "Reading input image into memory..."

        ReadRawPixelsLoop: do i = 1, k

            ReadRawBandsLoop: do j = 1, b

                  read (unit = 11) x (j,i)

            end do ReadRawBandsLoop

      end do ReadRawPixelsLoop

        close (unit = 11)

        ALLOCATE (InfoClasses (NumTrainingPoints))
        ALLOCATE (m (b,NumTrainingPoints))
        ALLOCATE (BestBandMask (b))

!        PRINT *, "Arrays allocated."

!        open (unit = 1, file = InputFileNameTrainingData, access =
"sequential", &
!             form = "formatted", status = "old", action = "read")

        open (unit = 1, file = InputFileNameTrainingData, access =
"sequential", &
              form = "formatted")

        ReadTrainingPoints: do i = 1, NumTrainingPoints

                read (unit = 1, fmt = "(I6)") InfoClasses(i)

!               PRINT *, "InfoClasses (", i, ") = ", InfoClasses (i)

!               ReadXYs

                read (unit = 1, fmt = "(F)") TempX
                read (unit = 1, fmt = "(F)") TempY

                if (i .EQ. 1) then

                   PRINT *, "TempX = ", TempX
                   PRINT *, "TempY = ", TempY
```

```
                end if

!               end ReadXYs

!               ComputeBVs
!               Convert to File Coordinates

                FileX = int(( ( (TempX - UpperLeftX) / (PixelSize) ) +
1),4)
                FileY = int(( ( (UpperLeftY - TempY) / (PixelSize) ) +
1),4)

                if (i .EQ. 1) then

                   PRINT *, "FileX = ", FileX
                   PRINT *, "FileY = ", FileY

                end if

!               end Convert to File Coordinates
!               Find Correct Pixel

                Pixel = ((FileY - 1) * NumCols) + FileX

                if (i .EQ. 1) then

                   PRINT *, "Pixel = ", Pixel

                end if

!               end Find Correct Pixel
!               GetBV

                m (:,i) = x(:,Pixel)

                if (i .EQ. 1) then

                   PRINT *, "m (:,i) = ",  m (:,i)

                end if

!               end GetBV
!               end ComputeBVs

        end do ReadTrainingPoints

        close (unit = 1)

!        PRINT *, "Training points read successfully"

!       Beginning of section to select bands and k for kNN

        CALL knnkb (NumTrainingPoints, m, b, MaxkNN_k, InfoClasses,
NumInfoClasses, kNN_k, BestBandMask)

!        PRINT *, "kNN_k = ", kNN_k
```

```
!         PRINT *, "BestBandMask = ", BestBandMask

!         PRINT *, "BestBandMask = ", BestBandMask

        Mask_m: do i = 1, NumTrainingPoints

            m (:,i) = m (:,i) * BestBandMask

        end do Mask_m

        Mask_x: do i = 1, k

            x (:,i) = x (:,i) * BestBandMask

        end do Mask_x

        write (unit = *, fmt = *) "x (:,1) = ", x (:,1)
        write (unit = *, fmt = *) "x (:,2) = ", x (:,2)
        write (unit = *, fmt = *) "x (:,3) = ", x (:,3)

        write (unit = *, fmt = *) "Raw image written into memory..."

        ALLOCATE (icx (k))
        icx = 0

        CALL knn (k, NumTrainingPoints, x, m, icx, kNN_k, InfoClasses,
NumInfoClasses)

        open (unit = 2, file = OutputImageFileName, &
            form = "binary", status = "replace", action = "write")

        PRINT *, "Writing output files..."

          WriteOutputFile: do i = 1, k

                  write (unit = 2) icx (i)

          end do WriteOutputFile

        close (unit = 2)

        open (unit = 3, file = OutputHeaderFileName, &
            form = "formatted", status = "replace", action = "write")

        write (unit = 3, fmt = "(A)") "ENVI"
        write (unit = 3, fmt = "(A)") "description = "
        write (unit = 3, fmt = "(A,I)") "samples = ", NumCols
        write (unit = 3, fmt = "(A,I)") "lines   = ", NumRows
        write (unit = 3, fmt = "(A)") "bands   = 1"
        write (unit = 3, fmt = "(A)") "header offset = 0"
        write (unit = 3, fmt = "(A)") "file type = ENVI Standard"
        write (unit = 3, fmt = "(A)") "data type = 2"
        write (unit = 3, fmt = "(A)") "interleave = bip"
        write (unit = 3, fmt = "(A)") "sensor type = Unknown"
        write (unit = 3, fmt = "(A)") "byte order = 0"

        close (unit = 3)
```

```
        PRINT *, "Output files successfully written and closed."

        DEALLOCATE (InfoClasses)
        DEALLOCATE (x)
        DEALLOCATE (m)
        DEALLOCATE (icx)

end program kNearestNeighborkb
```

APPENDIX IX
Fortran 95 code to convert CART Ascii output to a BIP or BSQ

This program was used to convert CART output with minimal variables included to a BIP or BSQ file.

```
!     Last change:  CL   29 Mar 2005    3:17 pm
!
! ReadAscii
!
! Randolph Hamilton Wynne, Department of Forestry, Virginia Polytechnic
Institute and State University
! Katherine Amanda Joseph, Department of Forestry, Virginia Polytechnic
Institute and State University
!
! Version 0.1
!
! Program start date: March 29, 2005)
! Inputs:       ASCII text file list formatted
!
! Outputs:      "BIP" or "BSQ" single band binary image file as signed 16-
bit integers

program ReadAscii

        implicit none

        INTEGER, parameter :: S_LEN = 32
        INTEGER (KIND = 4) :: i ! Loop counters
        INTEGER (KIND = 4) :: k, t1, t2
        INTEGER (KIND = 2), ALLOCATABLE :: x (:)
        CHARACTER (LEN = S_LEN) :: InputFileName
        CHARACTER (LEN = S_LEN) :: OutputFileName

                write (unit = *, fmt = *) "InputFileName?"
            read (unit = *, fmt = *) InputFileName
!       PRINT *, "Input File Name Raw Image: ", InputFileNameImage

                write (unit = *, fmt = *) "Number of Pixels?"
            read (unit = *, fmt = *) k
!       PRINT *, "Number of Training Points: ", NumTrainingPoints

                  write (unit = *, fmt = *) "Output File Name?"
            read (unit = *, fmt = *) OutputFileName
!         PRINT *, "Output File Name: ", OutputFileName

        write (unit = *, FMT = *) "Input File Names: "
        write (unit = *, FMT = *) "    ", InputFileName, " (Unit = 11)"

        write (unit = *, FMT = *) "Number of Pixels: "
        write (unit = *, FMT = *) "    ", k

        write (unit = *, FMT = *) "Output File Names: "
```

```
        write (unit = *, FMT = *) "    ", OutputFileName, " (Unit = 2)"

        ALLOCATE (x (k))
        x = 0

        open (unit = 11, file = InputFileName, &
              form = "formatted", status = "old", action = "read")

        write (unit = *, fmt = *) "Reading ASCII file into memory..."

      read(11,25)
        ReadLoop: do i = 1, k
25 format(I, ",", I, ",", I, ",")
                  read (11, *) t1, x (i), t2


      end do ReadLoop

        close (unit = 11)

        open (unit = 2, file = OutputFileName, &
            form = "binary", status = "replace", action = "write")

        PRINT *, "Writing output files..."

          WriteOutputFile: do i = 1, k

                  write (unit = 2) x (i)

          end do WriteOutputFile

        close (unit = 2)

        PRINT *, "Output file successfully written and closed."

        DEALLOCATE (x)
100   continue

end program ReadAscii
```