

Improving the Accuracy of Pixel Classification by Including Texture Information

Robert M. Rohland

Department of Resource Analysis, Saint Mary's University of Minnesota, Minneapolis, MN 55404

Keywords: GIS, Remote Sensing, Texture Analysis, Spatial Variations, Classification, Segmentation, Photointerpretation, Laws filter, GLCM statistics, Hurst filter, Gabor filter

Abstract

Environmental Systems Research Institute (ESRI) ArcGIS 9.x has limited image processing capabilities in the form of the Spatial Analyst Extension. Spatial Analyst contains a tool called the Maximum Likelihood (ML) classification tool, which can be used to derive a thematic map from an air photo or satellite image. Working with multispectral image data (such as a Landsat MSS or TM+) or with multitemporal image data (such as data obtained at multiple times during a growing season) it is possible to derive thematic maps with their associated legends. However, RGB and color IR air photos having only three layers often contain too little information for ML pixel classification to have an acceptable level of accuracy. The accuracy of classification can be improved by using context information in the form of texture measures. Custom texture filters have been designed for that purpose.

Introduction

One of the most common imagery analysis tasks in remote sensing is classification. A remotely sensed image having multiple layers is used to develop a thematic map in which areas in the image are classified according to landcover type, or some other categorical scheme (Figure 1).

Accompanying the thematic map is a legend, which identifies the various areas in the map. When performed by a human, this

process is known as photointerpretation, and it is very labor-intensive. Automated (computer-based) classification is much faster, but it is far from trivial, and much research has been expended trying to find the most efficient, accurate, and effective method of classification.

Segmentation

What the photointerpreter is attempting to do is more properly known as segmentation because the whole image is subdivided into mutually exclusive areas. Each pixel presumably belongs to only one area (segment) at a time, and each segment belongs to a class. Delineation involves defining boundaries of each area. Therefore, the overall process is more accurately called segmentation, which involves both classification and delineation (Figure 2).

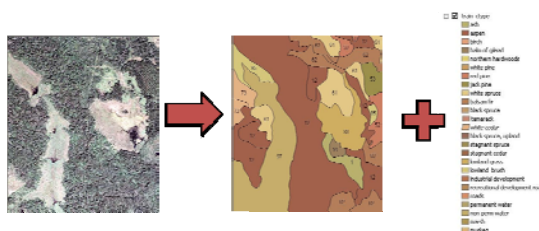


Figure 1. The Process of Classification.

However, the whole process is referred to more loosely as classification by Spatial Analyst Tools in Environmental Systems Research Institute (ESRI) ArcGIS 9.x. This discussion focuses on the classification of pixels rather than delineation between regions. Because the ESRI ML classifier is being used, the whole process will be referred to as classification for the purpose of this project.

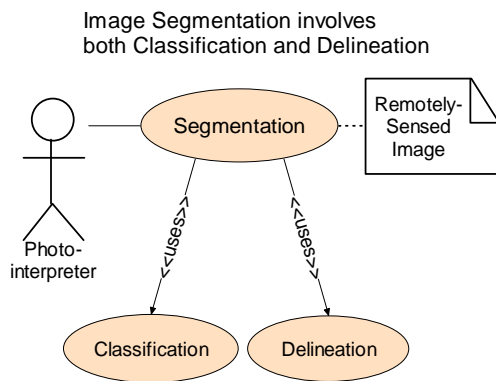


Figure 2. The Process of Segmentation.

Multilayer Classification of Remotely Sensed Images

A trained human interpreter may be able to classify grayscale images, but automated classification, especially a tool like the ESRI ArcGIS ML classifier requires multilayer images (Landgrebe, 2003; Richards and Jia, 2006; Tso and Mather, 2001). Ordinarily, pixel-by-pixel classification is performed on multispectral or multitemporal images. An example of multispectral data is that which is produced by the Landsat series of satellites. These images include red, green, and blue layers and also extra spectral bands in the IR (infrared), giving additional information about vegetation and geology.

An example of multitemporal data would be a series of multispectral images taken of agricultural land at different times during the growing season. As crops grow and mature, their spectral signatures change.

The Need for Greater Classification Accuracy and the Purpose of this Project

When working with six or seven spectral layers, e.g. Landsat TM+ datasets, there are enough bands to give a definitive spectral signature that allows for accurate classification. However, in cases where there are only three layers in a dataset, such as RGB color or color IR images, there is too little information to perform an accurate classification using the ML classifier tool in ESRI ArcGIS 9.x. In that case, additional information must be extracted from the pixel's context.

In this project, custom software tools have been developed to extract additional information based upon the texture within each pixel's neighborhood. Typically, texture information is derived from the value (brightness) of an RGB image. There is such high correlation between the red, green, and blue bands (usually over 95%) that it does not make sense to extract texture information for each spectral band (Maenpaa and Pietikainen, 2004).

Kinds of Classifiers

Classifiers are broadly divided into two kinds: unsupervised and supervised. An unsupervised classifier looks for natural clustering of data in feature space. These clusters in feature space are called *spectral classes*, and they may or may not have informational value.

Supervised classifiers can be "trained." Classifiers can be taught to classify samples according to categories known as *informational classes*. Supervised classification is performed by presenting training samples, which have class labels attached to them, to the classifier. After the classifier is taught to assign labeled sample vectors to the appropriate classes, then the classifier is allowed to classify samples whose class is not known.

Supervised classified classifiers are further divided into statistical classifiers, neural network classifiers, fuzzy logic classifiers, and rule-based classifiers.

Statistical classifiers can be yet further subdivided into parametric and non-parametric classifiers. Parametric classifiers assume classes can be modeled by a multivariate normal distribution (Gaussian distribution). Non-parametric classifiers do not make that assumption.

The ML classifier is a parametric statistical classifier because it uses statistics and it assumes the classes are normally distributed. It is therefore important to remember that the performance of the ML classifier is compromised if the data are not normally distributed.

Goals of Supervised Classification

There are three goals in the supervised classification of pixels (Landgrebe, 2003):

- The classes must have informational value. They must correspond to a category useful to humans, such as a land cover type.
- The classes must be separable. A pixel must be unambiguously assignable to one class or another.
- The set of classes must be exhaustive. That is, every pixel must have a class into which it can fall – including a class labeled “other” where pixels may be assigned if they do not neatly fit into an existing informational class.

In practice, it is rarely possible to achieve all three goals at the same time. Classes may overlap, and some pixels could be assigned to more than one class. In those cases, one is forced to pick the most likely class assignment (ML classifier). Other pixels may have such a low probability of

belonging to any class that they are labeled “unknown.”

The Multivariate Normal Distribution

When the random variable has more than one dimension (a vector) one is dealing with multivariate statistics. The ML classifier makes the assumption that the sample data are multivariate normal. The performance of the ML classifier depends upon how well this assumption holds in reality.

Points to Remember about the Maximum Likelihood Classifier

- It must be kept firmly in mind that the ML classifier is a parametric statistical classifier, and its effectiveness depends on the class sample data being multivariate normal.
- The training samples must be a good and faithful representation of all the samples.
- The class covariance matrices must not be singular. The ML classifier crashes if it encounters a singular covariance matrix.
- There must be very little overlap between classes.

What is Texture?

Researchers agree that texture is a spatial variation in brightness or tone from one pixel to the next within an image. However, there is no absolute or universal consensus as to how texture should be modeled or measured (Davies, 2005; Haralick, et al., 1973; Haralick and Shanmugan, 1974; Parker, 1997; Richards and Jia, 2006).

Texture is not an attribute of a single point or pixel. Texture is a variation between or among several pixels in the neighborhood of a pixel. The texture filter computes a value based on variations in the

neighborhood pixel values. This value is written to the output target pixel.

Methods

The ESRI ArcGIS 9.x Maximum Likelihood Classifier Tool

The ESRI Spatial Analyst Extension must be installed in order to perform classification in ArcGIS 9.x. ESRI ArcGIS 9.x provides only one kind of supervised classifier, the Maximum Likelihood Classification tool in the Spatial Analyst / Multivariate toolbox. It performs a pixel-by-pixel classification of a multilayer dataset.

Some of the Focal Statistics tools in Spatial Analyst may be used to perform limited extraction of context and texture information. ML classification in ArcGIS 9.x is a two-step process (Figure 3). With the addition of some customized VBA code (the core of this project), much more can be done.

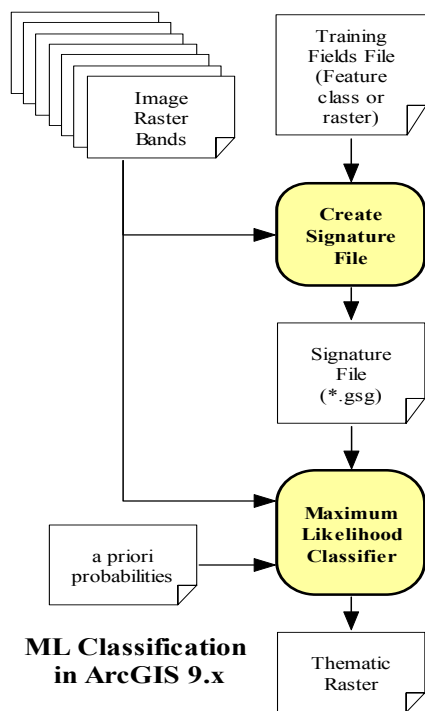


Figure 3. ML Classification in ArcGIS 9.x.

The first step is to use the Create Signatures tool, which takes the image raster bands as input. Figure 4 shows an RGB orthophoto of some agricultural land which includes a lake. The three bands, red, green, and blue are specified as inputs for the Create Signatures tool. Also required is a file, either a polygon vector file or a raster, which defines which areas are the training samples. The attributes of the polygons or pixels (depending on the file type) indicate the class assignment. Figure 5 shows an example of a file marking the training fields.



Figure 4. FSA Orthophoto of a Rural Scene.

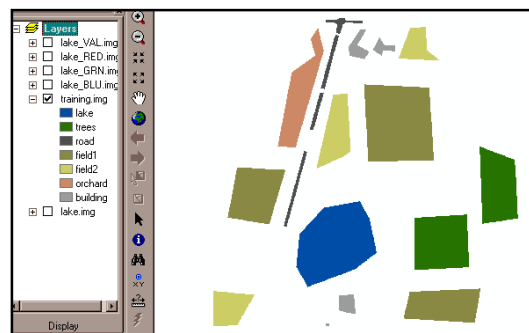


Figure 5. Raster File Marking Training Areas.

The names of the input layers and training field file are entered into the Create Signatures dialog (Figure 6). The attribute field is selectable (in the case of a vector training file). The name of the output file needs to be specified. Also, if an ML classification is desired, one must check the Compute covariance matrices checkbox. Otherwise, the classifier will behave as a

Minimum Distance to Class Means classifier.

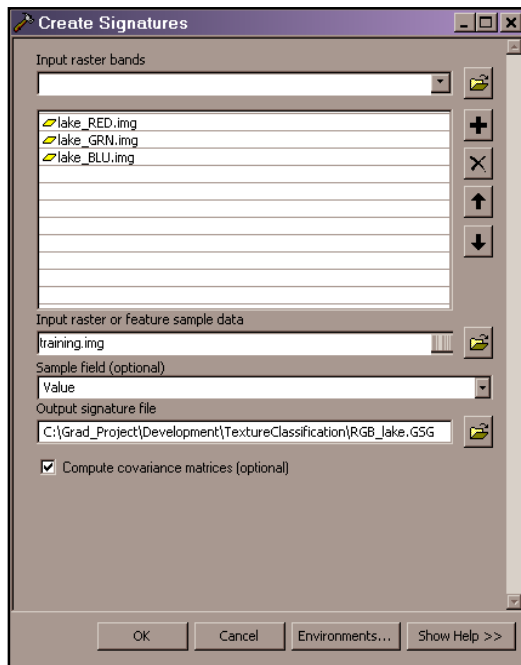


Figure 6. The Create Signatures Dialog.

The output of the Create Signatures tool is a signature file, having a “.gsg” extension, which is ESRI’s name for the statistical information connected with the training samples.

The signature file contains a mean vector (or centroid) and a covariance matrix for each class, which are computed from the training samples. An example of a signature file is shown in Table 1.

For each of the seven classes in the signature file (Table 1), there is a mean vector and a covariance matrix. The mean vector describes the centroid (or center of gravity) of the class in feature space, while the covariance matrix describes the shape of the class distribution. Since there are three layers, the feature space is three-dimensional. Therefore, the mean vector has three elements, and the covariance matrix is a 3×3 matrix.

The second major step in classification (Figure 3) is the Maximum Likelihood Classification tool, which takes

the image raster bands (red, green, and blue) as well as the signature file as inputs.

Table 1. Signature File (*.GSG extension).

# Signatures Produced by ClassSig from				
# Class-Grid c:\grad_p-1\develo-1\textur-1\g_g3				
# and Stack c:\grad_p-1\develo-1\textur-1\z_z2				
# Number of selected grids				
/* 3				
# Layer-Number Grid-name				
/* 1 z_z2c1				
/* 2 z_z2c2				
/* 3 z_z2c3				
# Type Classes Layers Parametric Layers				
# 1 7 3 3				
#				
=====				
# Class ID	1	Number of Cells	49968	Class Name
# Layers	1	2	3	
# Means		36.49442	42.06434	55.47977
# Covariance				
1		31.39861	25.81174	26.44564
2		25.81174	26.72456	24.57453
3		26.44564	24.57453	26.12691
#				
# Class ID	2	Number of Cells	55353	Class Name
# Layers	1	2	3	
# Means		68.11989	77.66493	78.84958
# Covariance				
1		669.31216	672.81481	659.58840
2		672.81481	686.66257	666.17543
3		659.58840	666.17543	657.70707
#				
# Class ID	3	Number of Cells	8186	Class Name
# Layers	1	2	3	
# Means		187.28769	185.37821	188.71696
# Covariance				
1		360.62987	365.95728	343.63416
2		365.95728	380.99781	351.61775
3		343.63416	351.61775	341.83887
#				
# Class ID	4	Number of Cells	100714	Class Name
# Layers	1	2	3	
# Means		104.70264	115.37721	108.02117
# Covariance				
1		226.54026	177.41603	176.99374
2		177.41603	161.81516	151.80360
3		176.99374	151.80360	150.89548
#				
# Class ID	5	Number of Cells	36061	Class Name
# Layers	1	2	3	
# Means		138.20612	134.79920	126.17437
# Covariance				
1		282.72032	229.04577	244.32651
2		229.04577	209.81378	214.96280
3		244.32651	214.96280	225.62384
#				
# Class ID	6	Number of Cells	22597	Class Name
# Layers	1	2	3	
# Means		125.98721	129.06519	121.07687
# Covariance				
1		873.85446	812.21016	819.62357
2		812.21016	775.81930	777.87519
3		819.62357	777.87519	784.89297
#				
# Class ID	7	Number of Cells	8139	Class Name
# Layers	1	2	3	
# Means		178.38088	177.58975	175.01941
# Covariance				
1		3802.93724	3692.17544	3645.04618
2		3692.17544	3599.42777	3558.13379
3		3645.04618	3558.13379	3544.59608

The ML Classification dialog is shown in Figure 7. A name needs to be specified in the dialog for the output raster file. In this example, IMAGINE Image format was chosen by specifying the “.img” extension.

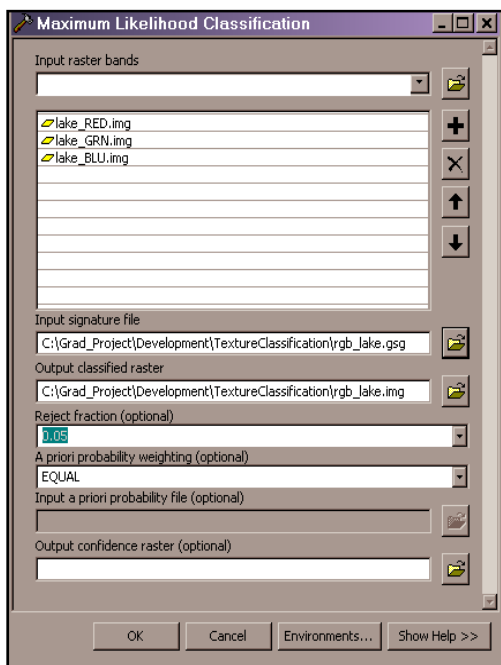


Figure 7. ML Classification Dialog.

The Reject Fraction specifies how to handle pixels that do not seem to fit into any class. The Reject Fraction was set to zero, therefore forcing the classifier to make a decision no matter how poorly the pixel fits into any class. The a priori probability setting was set to EQUAL for these experiments. It was also decided not to use the Output Confidence Raster feature. The result of the maximum likelihood classification is shown in Figure 8. There is no way to verify the accuracy of the classification other than to use the training areas themselves, since “ground truth” for the rest of the image is not available. It was decided to develop a tool in Visual Basic to perform error analysis as part of this project. The Create Error Matrix tool, will be discussed later on in this section.

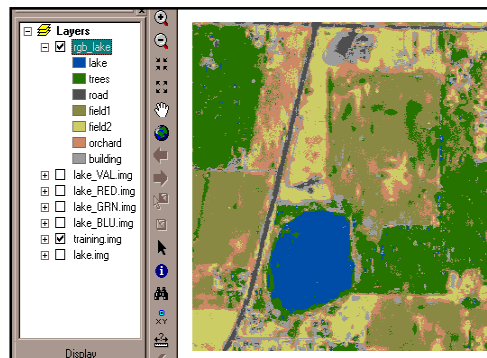


Figure 8. Result of ML Classification.

Custom Software Tools

An extensive set of software tools has been developed in both C#.NET (the Bhattacharyya Matrix tool) and VBA, which is embedded in ArcGIS 9.x (the Thematic Classification Toolbar). It would be impossible to include all the details as to how these tools operate within the scope of this paper. Please see Appendix for several ancillary documents written for those wanting more information.

Custom Texture Filters

There are eight texture filters which can be accessed via the Texture Tools drop-down menu on the Thematic Classification toolbar (Figure 9). Each texture filter takes a raster as input and creates one or more output images based on local texture measures.

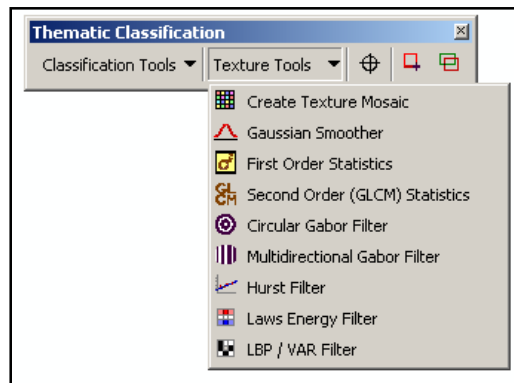


Figure 9. Texture Tool Menu.

Brief descriptions for each texture filter follow.

Gaussian Smoother

This filter is a Gaussian-weighted moving average convolution mask. The size of the square window is four standard deviations of the Gaussian kernel.

First Order Statistics

This filter computes statistics of a square neighborhood around the input pixel. There are six statistics altogether:

- Mean
- Standard Deviation
- Skewness
- Kurtosis
- Median
- Range (Max minus Min)

Second Order (GLCM) Statistics

The Gray-Level Cooccurrence Matrix operators (also called Haralick operators) are based upon statistics calculated upon a two-way cooccurrence matrix (Haralick, et al., 1973; Haralick and Shanmugan, 1974). The statistical operators used here consist of the following operators:

- Angular Second Moment
- Contrast
- Dissimilarity
- Entropy
- Inverse Difference Moment
- Inverse Difference
- Maximum Probability

Gabor Filters

Gabor filters are spatial frequency bandpass filters, and they have many applications in texture recognition matrix (Clausi and Jernigan, 2000; Idrissa and Acheroy, 2002;

Manthalkar, et al., 2003). The Gabor kernel is a sine wave (or cosine wave) modulated by a Gaussian envelope (Figure 10).

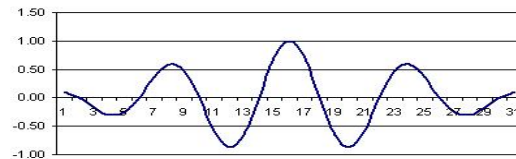


Figure 10. One-dimensional Gabor Kernel.

Circular Gabor Filter

A moving circular Gabor mask is shown in Figure 11. This mask is sometimes called the Mexican Hat filter.

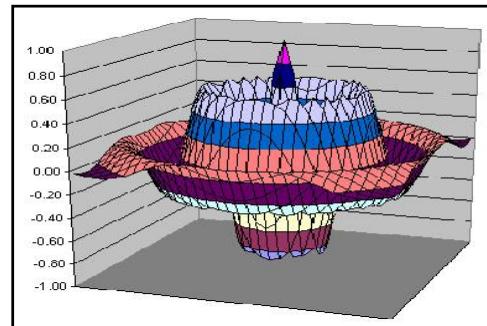


Figure 11. Circular Gabor Mask.

Multidirectional Gabor Filter

The multidirectional Gabor filter applies a directional Gabor filter at various orientation angles. For each orientation a sine and a cosine Gabor mask are applied, squared, and summed (Figure 12).

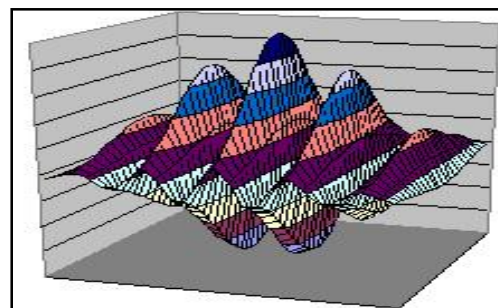


Figure 12. Multidirectional Gabor Mask.

Hurst Filter

The Hurst filter calculates a Hurst coefficient for each input pixel based on a circular neighborhood. The Hurst coefficient is a measure of fractal dimension similar to that defined by Mandelbrot (1983). The Hurst coefficient is the slope of a regression line (Hurst plot, Figure 13) which plots the log distance from the central pixel versus the log range (max - min).

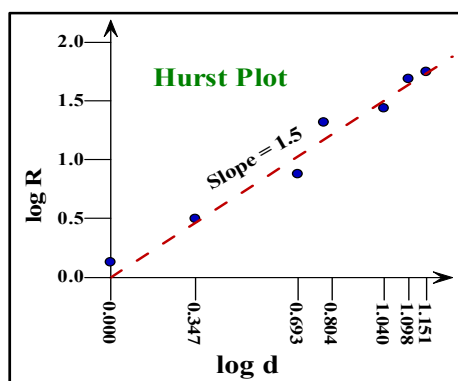


Figure 13. Hurst Plot.

Laws Energy Filter

The Laws Energy filters are integer valued convolution filters of sizes 3×3 , 5×5 , and 7×7 . The 3×3 masks are shown in Figure 14. The results of the convolutions are squared (giving energy) followed by a smoothing filter (Laws, 1980a; Laws, 1980b; Parker, 1997).

L3^TL3	L3^TE3	L3^TS3
1 2 1	-1 0 1	-1 2 -1
2 4 2	-2 0 2	-2 4 -2
1 2 1	-1 0 1	-1 2 -1
E3^TL3	E3^TE3	E3^TS3
-1 -2 -1	1 0 -1	1 -2 1
0 0 0	0 0 0	0 0 0
1 2 1	-1 0 1	-1 2 -1
S3^TL3	S3^TE3	S3^TS3
-1 -2 -1	1 0 -1	1 -2 1
2 4 2	-2 0 2	-2 4 -2
-1 -2 -1	1 0 -1	1 -2 1

Figure 14. 3 x 3 Laws Filters Masks.

LBP / VAR filter

The local binary pattern (LBP) filter and the variance (VAR) filter deal only with the eight immediate neighbors in this particular implementation (Figure 15). These filters were originally developed by the Machine Vision Group at the University of Oulu, Finland (Maenpaa and Pietikainen, 2005; Ojala, et al., 2002; Pietikainen, 2000).

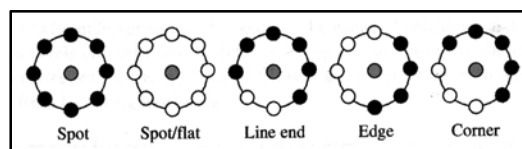


Figure 15. LBP Primitives.

Other Custom Tools

Other custom tools are on the Classification Tools pull-down menu and on the toolbar itself (Figure 16).

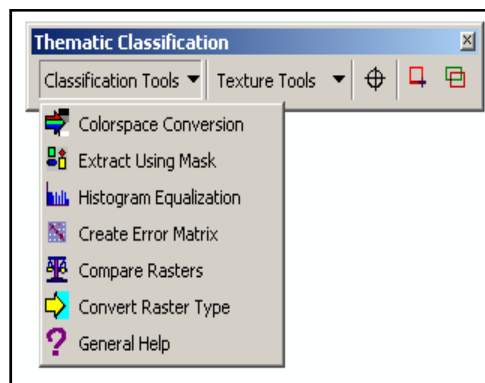


Figure 16. Classification Tool Menu.

The Create Error Matrix Tool

The simplest measure of classification accuracy is the number of correct classifications divided by the total number of pixels. However, the breakdown of errors by category is desired, and this type of display is called an error matrix or confusion matrix (Campbell, 2002; Landgrebe, 2003; Richards and Jia, 2006).

One thematic raster is called the reference, and the other, the target. A reference pixel is assumed to be true, and if the corresponding target pixel matches it, it is said that the classification is correct, otherwise it is in error.

This custom tool was created for this very purpose. The classification matches and mismatches are cross-tabulated (Table 3). The reference classes are shown along the horizontal axis and vertical axis. Tallies along the main diagonal of the matrix are correct, and all off-diagonal tallies are errors.

The correct tallies by column divided by the column totals are called *producer's accuracy*, and the correct tallies by row divided by the row totals are called *user's accuracy*. The total correct divided by the overall total is the overall accuracy.

It is expected that a certain number of pixels will be classified correctly by

random chance. Many statisticians consider κ , or "kappa" to be a better figure of merit than overall accuracy (Congalton, 1991; Congalton, et al., 1983; Rosenfield and Fitzpatrick-Lins, 1986). Kappa, as a population parameter, is:

$$(1.1) \quad \kappa = \frac{p_o - p_c}{1 - p_c}$$

p_o is the probability of observing a correct classification, and p_c is the probability of a correct classification happening by chance. A sampled estimate for kappa is called kappa-hat ($\hat{\kappa}$).

The kappa-hat-variance tells how good that estimate is. The formula for the kappa-hat-variance is rather complicated; see Hudson and Ramm (1987). The Create Error Matrix Tool calculates all of the above, as well as a 95% confidence interval for kappa.

Table 3. An Error Matrix.

Class Legend	Reference = training.img							Target = rgb_lake.img	
1 lake									
2 trees									
3 road									
4 field1									
5 field2									
6 orchard									
7 building									
Target Classes	Reference Classes							Totals	User's
	1	2	3	4	5	6	7		
1	49288	304	0	0	0	0	26	49618	99.33%
2	674	51578	0	3492	0	104	595	56443	91.38%
3	0	3	7473	1	9	71	541	8098	92.28%
4	0	2383	0	82259	72	3266	183	88163	93.30%
5	0	11	0	3028	31691	5031	423	40184	78.86%
6	0	271	14	11187	3460	13386	650	28968	46.21%
7	6	803	699	747	829	739	5721	9544	59.94%
Totals	49968	55353	8186	100714	36061	22597	8139	281018	
Pducer's:	98.64%	93.18%	91.29%	81.68%	87.88%	59.24%	70.29%		
Total Pixels Compared = 281,018									
Correctly Classified = 241,396									
Overall Accuracy = 85.90%									
Estimated Kappa-Hat = 0.8211									
Est. Kappa-Variance = 6.7930E-7									
95% Confidence Interval = [0.8195, 0.8228]									

The Bhattacharyya Matrix Tool

The Bhattacharyya Matrix tool (not on the Thematic Classification toolbar, but under ArcTools) takes an ESRI classification signature file and generates a cross-tabulation matrix of statistical distances between the classes as typified by their signatures. The Bhattacharyya distance is the distance between two distributions in feature space. The formula for the B-distance is:

$$D = \frac{1}{8} [\mu_1 - \mu_2]^T \left[\frac{\Sigma_1 + \Sigma_2}{2} \right]^{-1} [\mu_1 - \mu_2] + \frac{1}{2} \ln \left(\frac{\left| \frac{1}{2} [\Sigma_1 + \Sigma_2] \right|}{\sqrt{|\Sigma_1| |\Sigma_2|}} \right) \quad (1.2)$$

The μ variables are the centroids of the distributions and the Σ variables are the covariance matrices. The B-distance is used in the MultiSpec© software by Landgrebe and Biehl (1995), and is considered to be one of the best measures of class separation (Landgrebe, 2003; Richards and Jia, 2006). For an example of a B-matrix, see Table 4.

Table 4. Bhattacharyya Matrix.

Bhattacharyya matrix for ArcGIS signatures file: C:\Grad_Project\Development\TextureClassification\rgb_lake.gsg						
Type	= 1					
Number of Classes	= 7					
Number of Layers	= 3					
Parameter Layers	= 3					
Class	1	2	3	4	5	6
----	---	---	---	---	---	---
2	3.43					
3	16.47	5.53				
4	13.35	1.56	4.95			
5	37.22	6.64	4.60	1.62		
6	20.12	3.17	2.69	0.61	0.51	
7	4.25	1.70	0.62	1.60	1.34	0.83

The Colorspace Conversion Tool

The Colorspace Conversion tool allows one to extract the red, green, and blue layers

from an RGB raster dataset. It also extracts hue, saturation, and value as separate layers.

Show Raster Cell Values Tool

This modal tool shows the pixel value at the cursor of the topmost active layer. For RGB datasets, the values of all three layers are shown.

Autocorrelation Tools

The two autocorrelation tools allow one to select an area of the image and create an autocorrelation plot of that selection. This can be used to help select the scale for applying texture filters.

Miscellaneous Tools

Additional utility tools are included in the Thematic Classification toolbar:

- Extract Using Mask
- Histogram Equalization
- Compare Rasters
- Convert Raster Type
- General Help

The General Help selection is really not a tool, but rather a set of Help dialogs that gives additional information beyond what is available with each software tool.

Datasets

VisTex Mosaic

The Create Texture Mosaic tool creates a mosaic from a collection of tiles which were originally obtained from the VisTex database at the MIT Media Lab (MIT, 1995). A 12-tile RGB mosaic was created from that tool (Figure 17).

The thematic map for the training areas is shown in Figure 18 and the ground truth thematic map is shown in Figure 19.



Figure 17. 12-Tile RGB Mosaic.

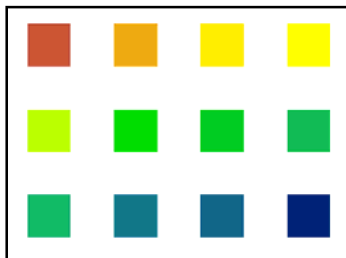


Figure 18. 12-Tile Mosaic Training Areas.

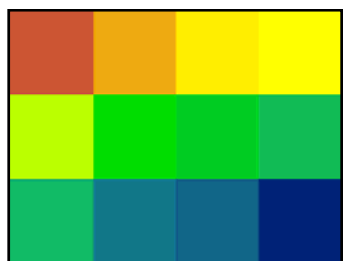


Figure 19. 12-Tile Mosaic "Ground Truth".

Itasca County Forest Dataset

There are two datasets of the same area in Itasca County, Minnesota: RGB and color IR. The area of interest consists of 96 square kilometers of mixed woodland (Figure 20). The RGB dataset is an FSA orthophoto (9/1/2004), at one meter resolution. The color IR orthophoto (10/5/2003) was obtained from the Minnesota DNR, and has a one meter resolution.

Various stands within this area have been ground classified by professional foresters. The ground truth classification information is in the form of a shapefile from the MN DNR, which is shown in Figure 21.

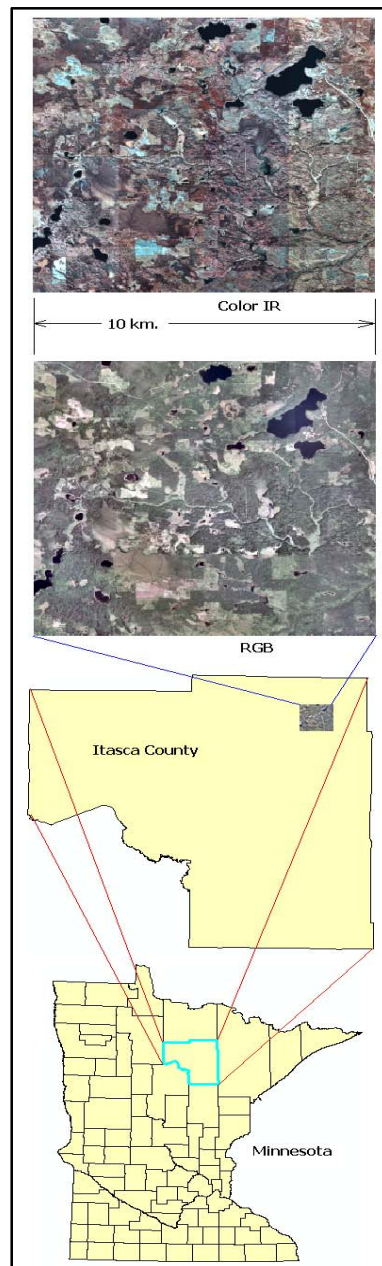


Figure 20. Itasca County Forest Data.

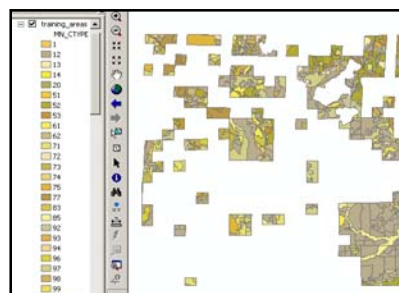


Figure 21. Itasca County Forest Ground Truth.

Alaska Wetland Data

The Alaska Wetland dataset (Figure 22) was obtained from Geospatial Services, St. Mary's University, Winona, MN (GSS). The entire dataset has been classified by professional photointerpreters at GSS into several wetland main types (Figure 23):

- Estuarine and Marine Wetland
- Freshwater Emergent Wetland
- Freshwater Pond
- Lake

The classification data comes in the form of a polygon feature dataset. Therefore, the entire feature dataset can be used as “ground truth” to check the accuracy of our classification. The orthophoto has been clipped to a rectangular area of interest consisting of 125 square kilometers. The cell size is 2.5 meters square, giving a raster of 4000 columns by 5000 rows making 2,000,000 pixels altogether.

Results

Experiment #1: VisTex Mosaic

Accuracy Performance

The VisTex tile mosaic classification accuracy results are summarized in Figure 24. Because each of these classifications employs different numbers of layers, any kind of a direct comparison is somewhat misleading. For example, the Hurst filter produces only one additional layer of information in addition to red, green, and blue, making 4 altogether. As a result, the classification accuracy based upon the Hurst filter is only slightly better than RGB alone.

The 1st order statistics outperformed all the other methods, which was unexpected. That was followed by the Laws Energy Filter, followed by the 2nd order

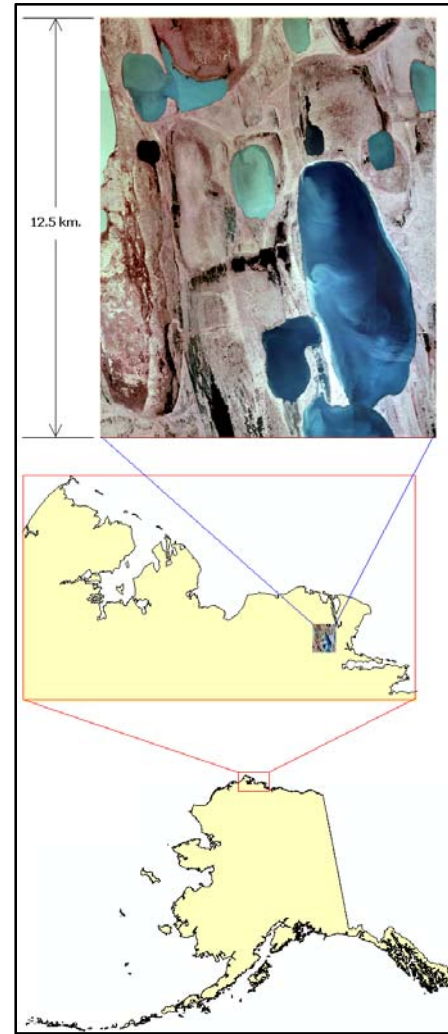


Figure 22. Alaska Wetland Data.

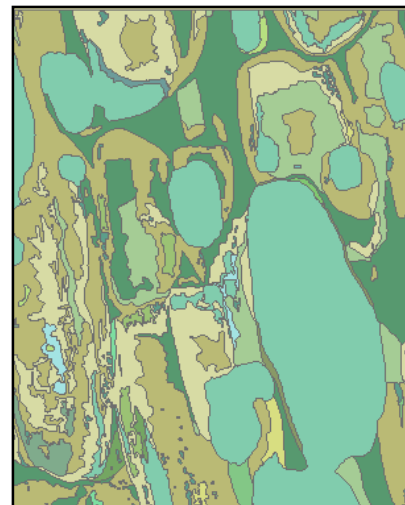


Figure 23. Alaska Wetland Data Ground Truth.

(GLCM) statistics, and the Multidirectional Gabor filter, in descending order.

Experiment #1: Classification Accuracy of Mosaic Dataset

Layers	Method	Overall Accuracy	Kappa-hat
3	RGB Only	70.45%	67.76%
4	Hurst Filter	72.63%	70.14%
7	Circular Gabor	73.73%	71.34%
5	LBP / VAR Filter	78.69%	76.75%
4	Gaussian Smoother	78.91%	76.99%
7	Multi-Dir. Gabor	83.71%	82.23%
10	GLCM Statistics	84.67%	83.27%
9	Laws Energy Filter	87.78%	86.67%
9	1st Order Statistics	89.99%	89.08%

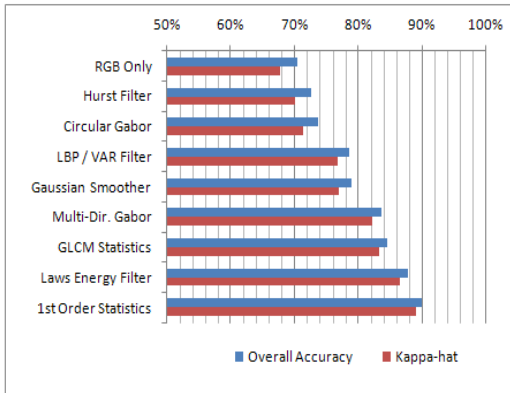


Figure 24. VisTex Mosaic Classification Results.

Speed Performance

The details of filter performance testing are these:

- With the exception of the LBP / VAR filter, the convolution kernels for all the filters were set to 17×17.
- For the multidirectional Gabor filter, four orientations were used.
- Each of the Laws filters used a 17×17 smoothing filter after the primary convolution.

The speed measurements of the various filters are summarized in table 5. When comparing speed with classification accuracy, it is apparent that the most accurate tools are not necessarily the fastest, or vice-versa. The first order statistics were the best performers in accuracy, but were

relatively slow. The primary reason for its slow performance had to do with computing the order statistics: median and range. The software uses the Quicksort algorithm, which has a time complexity of $O(n \log n)$, but is nonetheless slower than computing statistical moments, which are $O(n)$.

Table 5. Summary of Filter Speed Tests.

Filter Name	No. of layers	Total seconds	Seconds per layer
LBP / VAR	2	4.97	2.49
Circular Gabor	4	74.55	18.64
Laws Energy 3 x 3	1	19.19	19.19
Gaussian Smoother	1	19.36	19.36
Laws Energy 5 x 5	1	20.23	20.23
Laws Energy 7 x 7	1	22.02	22.02
Hurst	1	30.36	30.36
GLCM Statistics	7	230.83	32.98
1st Order Statistics	6	593.41	98.90
Multidirect. Gabor	4	792.7	198.18

The Laws Energy filters came in 2nd place in the accuracy test, but were reasonable in terms of speed. The main limitation on speed for the Laws Energy filters is the size of the smoothing filter, not the primary convolution kernel itself.

The 2nd order (GLCM or Haralick) statistics came in 3rd place in the accuracy test, but were also reasonable in speed, although not as speedy as the Laws filters. The multidirectional Gabor filters are very competitive in accuracy, but were very poor performers in regard to speed. The reason for this was because the multidirectional Gabor needed to compute a convolution kernel for both sine and cosine waves in multiple orientations.

Balancing accuracy and speed considerations, it appears that the best choices are the Laws Energy filters and the 2nd order (GLCM) statistics.

Experiment #2: Itasca County Forest Data, All Categories

It was evident from the outset that the custom software tools were not able to

handle a raster dataset as large as 284 megabytes. The tools threw “out of memory” exceptions. Thus, the decision was made to analyze a somewhat smaller section measuring 1600×2500 meters, resulting in an 11.8 megabyte file (Figure 25).



Figure 25. A smaller (1600 x 2500) area of interest.

Principal Components Analysis

Before undergoing classification, the dataset was analyzed using the Principal Components Analysis tool in ArcTools. It was discovered that the three bands, red, green, and blue, are very highly correlated with each other (Table 6).

Table 6. Variance of the Itasca RGB Dataset.

Princ. Comp.:	1	2	3	Total
Eigenvalues:	3113.36	24.64	12.49	3150.49
% Variance:	98.82%	0.78%	0.40%	100.00%
<u>Eigenvectors</u>				
red	0.607	-0.412	-0.680	
green	0.591	-0.337	0.732	
blue	0.531	0.846	-0.039	

An inspection of the eigenvectors reveals that the red, green, and blue bands make a nearly equal contribution to the first principal component, which can be considered to be total image brightness. Therefore, the strategy of deriving texture information from the value (brightness) of the pixels is considered to be valid, as suggested by Maenpaa and Pietikainen (2004). Principal components analysis was

also performed on the color IR dataset, and the result was very similar to Table 6.

It was decided to combine the color IR dataset (IR, red, and green layers) with the blue layer from the RGB dataset. Principal component analysis of the four layers yielded the results in Table 7.

Table 7. Variance of the Itasca IRGB (4-layer) Dataset.

Princ. Comp.	1	2	3	4
Eigenvalues:	8268.59	837.38	302.01	8.42
% Variance:	87.81%	8.89%	3.21%	0.09%
<u>Eigenvectors</u>				
IR	0.652	-0.128	0.735	0.135
red	0.558	-0.014	-0.361	-0.747
green	0.504	-0.001	-0.568	0.651
blue	0.092	0.992	0.089	0.008

IRGB-only Classification

Table 8 illustrates the poor accuracy of the IRGB-only classification. This is to be expected, because of the granularity of forest images. Each tree casts a shadow, so a pixel-by-pixel classification scheme is bound to fail.

Table 8. Classification Results of IRGB Layers.

Overall Accuracy = 19.86%
Estimated Kappa-Hat = 0.1251
Est. Kappa-Variance = 7.5919E-8
95% Confidence Interval = [0.1245, 0.1256]

Scale

In Figure 26, it is evident that the brightness varies over the course of several meters, but the variance in brightness pattern varies from one stand of trees to another. Note that if the imaging device had been the Landsat TM+ instrument, the pixel resolution would be 30 meters, and the variance seen here would be lost. In our dataset the spatial resolution is one meter, and the crowns of trees are several meters wide. It is precisely

this spatial variance of brightness between one stand of trees and another that is the basis for our expectation that texture measurement should improve land cover classification.

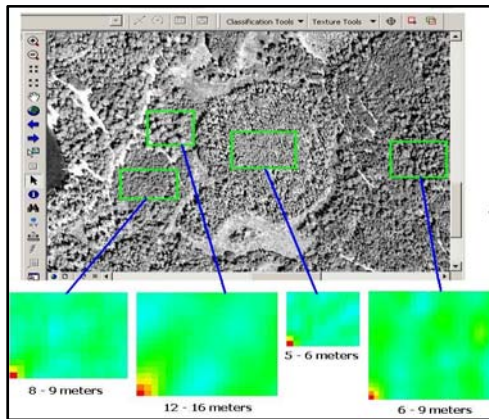


Figure 26. Autocorrelation analysis of forest.

It was decided that the texture filters should be set so that the raster is subsampled by a factor of 4 before filtering and supersampled by 4 afterwards to restore the raster to full scale.

Registration

A second reason for poor classifier performance could be that there may be alignment or registration problems between the ground truth dataset and the image (Figure 27). The green double-headed arrow shows a displacement of about 50 meters between the stand of trees and its boundary in the ground truth layer.

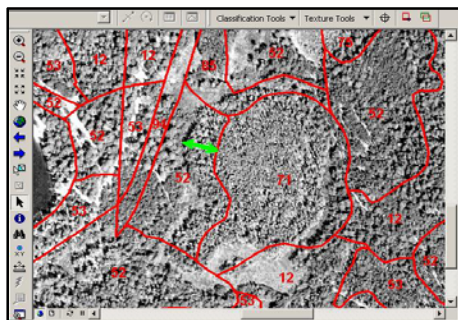


Figure 27. Possible Registration Problem.

Categories

In experiment #2, the results of using four texture methods were compared with the results of using IRGB-only. Categories used for training areas are shown in Table 9, with the training areas shown in Figure 28.

Table 9. Experiment #2 Landcover Categories.

Category	Code	Description
Hardwood	1	ash
“	12	aspen
“	13	birch
Conifers	52	red pine
“	53	jack pine
“	62	balsam fir
“	71	black spruce
“	72	tamarack
Non-productive	75	stagnant spruce
Non-forest	85	lowland brush
“	96	permanent water

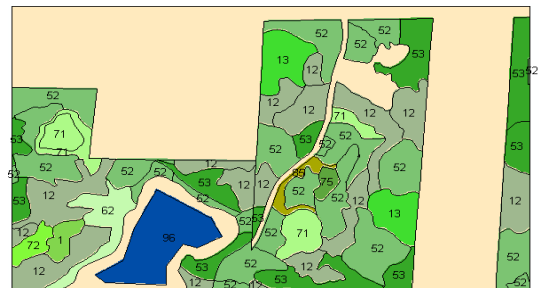


Figure 28. Experiment #2: Training Areas.

The four texture filtering methods that yielded the greatest accuracy in the VisTex mosaic experiment were used (Figure 29). It is apparent that the 1st order statistics and the Laws Energy filter provide much more accuracy than IRGB alone.

Generally speaking, the ML classifier has a hard time distinguishing among the various species of trees.

Experiment #3: Itasca County Forest Data, Collapsed Categories

The same filters were used as in experiment #2. The rasters were subsampled in the same fashion as well. But experiment #3 differs in

that the hardwood and conifer categories were collapsed (Table 10 and Figure 30).

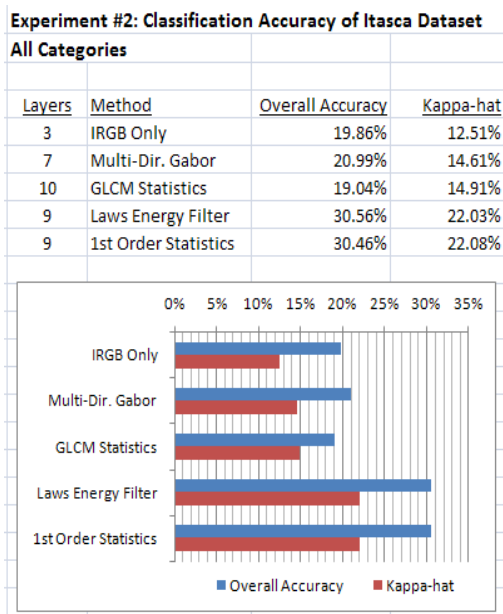


Figure 29. Experiment #2 Classification Results.

Table 10. Experiment #3 - Collapsed Landcover Categories.

Category	Code
Hardwood	1
Conifers	2
Stagnant	3
Brush	4
Permanent Water	5

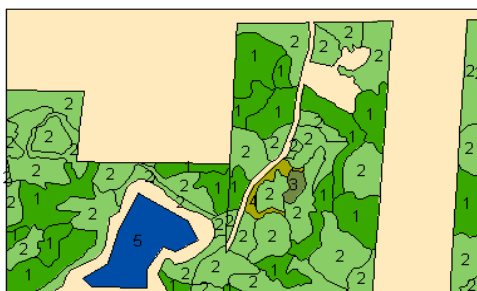


Figure 30. Experiment #3 Training Areas.

It appears that the classifier can distinguish hardwood from conifer much better than it can discriminate between individual species, like red pine from jack pine, for example. The front runner performer by far is the Laws Energy filter

with an overall accuracy of 56.1% and a kappa-hat of 32.58%.

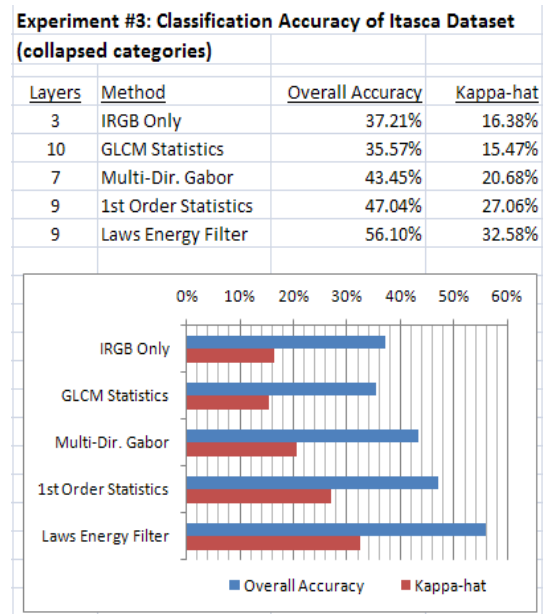


Figure 31. Experiment #3 Classification Results.

Experiment #4: Alaska Wetland Dataset, Full Categories

Principal Components Analysis

As was done with the Itasca forest dataset, a principal components analysis of the Alaska wetland dataset was performed first (Table 11). Unlike the Itasca dataset, the first principal component, which carried over 90% of the variance, was heavily weighted in the red band by over 80%. So, rather than using the value band derived from an HSV color transform, it was decided to use the

Table 11. Variance of the Alaska RGB Dataset.

Prin. Comp.:	1	2	3	Total
Eigenvalues:	5082.61	511.90	26.50	5621.02
% Variance:	90.42%	9.11%	0.47%	100.00%
<u>Eigenvectors</u>				
red	0.816	-0.557	0.156	
green	0.478	0.499	-0.723	
blue	0.325	0.664	0.673	

first principal component as the basis for texture filtering.

Scale

The next task was to perform an autocorrelation on the 1st principal component at various places in the image. Based upon this test it was decided that the 1st principal component should be subsampled by a factor of 10 before applying texture filters, and afterwards supersampled by 10 to bring the result back to full scale.

Full Categories

Table 12 show the full set of categories used. As with the Itasca dataset, the four texture filters associated with the most accurate classifications in previous experiments were used. The results are displayed in Figure 32.

It appears that first order statistics yielded the best classification at 50.51% overall accuracy with a kappa-hat of 41.81%. The Laws Energy filter came in second with 44.66% overall accuracy and a kappa-hat of 38.42%. Apparently, the classifier performed much better with full Alaska wetland categories than with the Itasca forestry dataset with full arboreal species categories.

Experiment #5: Alaska Wetland Dataset with Collapsed Categories

The same experiments were run with four collapsed categories of lake, freshwater emergent wetland, and freshwater pond. The best classification results were obtain with the red, green, and blue bands only, yielding 90.55% overall accuracy and kappa-hat of 79.88%. Classification involving any of the texture filters yielded poorer results than RGB alone (Figure 33). This was quite unexpected.

Table 12. Alaska Dataset Landcover Types.

Wetland_Type	Code	Attribute
Lake	5	L1UBH
“	6	L2EM2/UBH
“	7	L2EM2H
“	8	L2USE
Freshwater Emergent Wetland	9	PEM1/2F
“	10	PEM1/SS1B
“	11	PEM1/SS1E
“	12	PEM1/SS1F
“	13	PEM1/UBF
“	14	PEM1/UBH
“	15	PEM1B
“	16	PEM1E
“	17	PEM1F
“	18	PEM1H
“	19	PEM2/1F
“	20	PEM2/1H
“	21	PEM2H
Freshwater Pond	22	PUB/EM1H
“	23	PUBH

Experiment #4: Classification Accuracy of Alaska Wetland

Full Categories			
Layers	Method	Overall Accuracy	Kappa-hat
3	RGB Only	33.28%	25.63%
7	Multi-Dir. Gabor	40.75%	32.12%
10	GLCM Statistics	42.86%	35.35%
9	Laws Energy Filter	46.66%	38.42%
9	1st Order Statistics	50.51%	41.81%

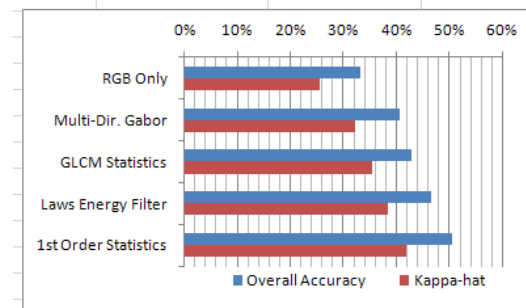


Figure 32. Experiment #4 Classification Results.

Discussion

Experiment #1: VisTex Mosaic

A step-by-step description of the classification performed here is given in the ancillary document *Texture Classification Tools for ArcGIS 9.3* (Rohland, 2008a) .

**Classification Accuracy of Alaska Wetland Dataset
(collapsed categories)**

Layers	Method	Overall Accuracy	Kappa-hat
3	RGB Only	90.55%	79.88%
7	Multi-Dir. Gabor	90.08%	79.50%
9	1st Order Statistics	88.20%	76.26%
9	Laws Energy Filter	88.28%	76.20%
10	GLCM Statistics	80.84%	55.91%

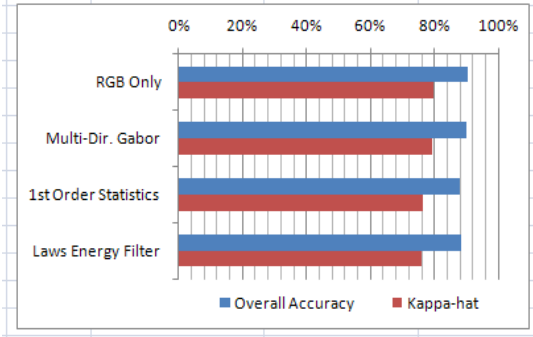


Figure 33. Experiment #5 Classification Results.

This was the only experiment performed with a texture mosaic. While the software tools have the ability to perform texture filtering at scales coarser than 1:1, this particular feature was not used. It is very possible that better classification results could have been obtained with a 2:1 or greater scaling.

In this project the results obtained from one section of the project were used to inform choices made for sections following. Because the results obtained from the VisTex Mosaic showed that the Laws Energy filter and the GLCM statistics offered the best balance of accuracy and speed performance, these tools were chosen for analyzing the Itasca and Alaska datasets. First order statistics and Multidirectional Gabor filters were also used for later experiments even though they were slow.

Experiments 2 and 3: Itasca County Forest Dataset

The inability of the custom software tools to handle the entire dataset consisting of $9600 \times 10,000 = 96,000,000$ pixels reveals a critical weakness of the software tools at this

stage of development. The reason for the “out of memory” error is that the tools require the entire dataset to be buffered in RAM at once, rather than handling it piecemeal. For the time being, we will have to deal with the data in smaller pieces. A section of size $2500 \times 1600 = 4,000,000$ pixels was chosen.

It was noted that classification accuracy improved when the training area categories were collapsed so that all hardwoods were grouped together, and all conifers were grouped together. Apparently, species of trees cannot be reliably identified by texture plus near IR plus RGB. Only when the trees are collapsed into hardwoods and conifers do accuracies start to approach 50%. It is possible that there might not be a strong correlation between arboreal species and texture.

Experiments 4 and 5: Alaska Wetland Data

Apparently the Alaska wetland data was photographed during the Arctic summer. The dataset did not come with metadata, so the date of acquisition is unknown. This area near the shore of the Arctic Ocean appears to be much more than just a frozen wasteland. Texture based classification fares much better here. With 19 categories of wetland / lake / pond the overall accuracies approach 50%, and in the case of 1st order statistics, exceed it.

In experiment #5, with three collapsed categories, the classifier overall accuracy performance exceeded 90% without texture filtering and without subsampling. The use of texture filtering actually worsened classifier performance. The researcher is unable to offer an explanation for this unexpected result.

Statistical Significance

To demonstrate the statistical significance that the employment of a certain texture

filter improves the accuracy of ML classification is quite easy because of the very narrow confidence intervals for kappa-hat. An example follows:

$$(1.3) \quad H_0 : t_{0.05} = 0 \quad \text{where} \quad t = \frac{(\hat{\kappa}_1 - \hat{\kappa}_2)}{\sqrt{\hat{\sigma}_{\kappa 1}^2 + \hat{\sigma}_{\kappa 2}^2}}$$

$$H_A : t_{0.05} \neq 0$$

Table 13. Classification Accuracy of Alaska Wetland, RGB-only.

Overall Accuracy = 33.28% Estimated Kappa-Hat = 0.2563 Est. Kappa-Variance = 7.5516E-9 95% Confidence Interval = [0.2561, 0.2565]
--

Table 14. Classification Accuracy of Alaska Wetland, Multidirectional Gabor Filter.

Overall Accuracy = 40.75% Estimated Kappa-Hat = 0.3212 Est. Kappa-Variance = 1.0004E-8 95% Confidence Interval = [0.3210, 0.3214]
--

Compare the RGB-only data with the Gabor filtered data in tables 13 and 14:

$$(1.4) \quad t = \frac{(0.3212 - 0.2563)}{\sqrt{(1.0004 \times 10^{-8})^2 + (7.5516 \times 10^{-9})^2}}$$

$$= \frac{.0649}{1.2534 \times 10^{-8}} = 5.178 \times 10^6$$

...which is significant at any level one would like to pick.

Summary

Finding statistical significance is extremely easy because the kappa-hat-variances are so small, and that is a result of the N (number of pixels) being so large. But the issue of statistical significance is not the same as the issue of usefulness. Is it worth the time and trouble to employ texture filters in conjunction with the ML classifier in ESRI ArcGIS when the overall accuracy is about 50% at best? Let the reader judge.

Perhaps an ML pixel classifier is not the appropriate instrument for landcover segmentation, at least not in the problem domains tried here. What this study *does* demonstrate is that texture filters can be used to significantly improve the accuracy of an ML pixel classifier when used with real-world data.

Suggestions for Future Research and Development

The software tools (texture filters and all) in their present form suffice as a proof of concept. However, before they could be used in a full production setting or be sold commercially, several things would have to happen:

- They need to handle large datasets. In their present form, the software tools buffer the entire raster dataset in memory, as well as allocate memory for an entire output data array. This places a severe restriction on the size of datasets that can be processed. ESRI raster tools apparently work on datasets piecemeal. Our software tools would have to do likewise. It is more complicated to do things this way, and there is slightly more overhead.
- The tools must have an ArcTools interface. Porting the code to C#.NET would allow us to build an ArcTools style interface. This would be advantageous because one could set up scripts in Python or one could construct models in ModelBuilder resulting in assembly line processing for large datasets and less tedium for the human user.
- The tools must be able to handle geodatabase data, both feature classes and raster datasets. This requires rewriting some of the data input/output routines and a little

extra overhead handling cells containing “NoData.”

- One could build a new classifier. The ML classifier is not the highest performing classifier for landcover classification (Rohland, 2008b; Tso and Mather, 2001). For classes that have a multimodal distribution, a non-parametric Parzen window classifier could be used, for example. Higher-level segmentation could be performed by using a Hidden Markov Model and simulated annealing (Tso and Mather, 2001). There are many approaches to classification that could be tried as an adjunct to ESRI ArcGIS imaging.

Acknowledgements

The author wishes to express his gratitude to the following persons for this project:

- Mr. John Ebert, committee chairman, faculty of St. Mary’s University of MN Resource Analysis Dept.,
- Mr. Jeff Knopf from Geospatial Services, St. Mary’s University in Winona, MN, who provided the Alaska Wetland dataset.
- Mr. Tim Loesch, Mr. Tim Aunan, and Mr. Jim Rack of the Minnesota Department of Natural Resources, who provided the Itasca County Forestry dataset.
- Mr. Kurt Swendson, alumnus of St. Mary’s University GIS Program, fellow student and comrade, for his insight and advice regarding programming with ESRI ArcObjects.
- The faculty of St. Mary’s University of MN Department of Resource Analysis, especially Dr. David McConville, department chair, without whom this project would not have been possible.

Appendix: Ancillary Documents

Details of texture based classification and the particular software tools used here can be found in the following ancillary documents:

- *Texture Classification Tools for ESRI ArcGIS 9.3: A Technical Walkthrough*: “how-to” information about the software tools described in this study (Rohland, 2008a).
- *An Overview of Classification Methods for Remote Sensing*: information on theory and issues related to classification (Rohland, 2008b).
- *Texture Classification Tools Help Dialogs*: an emphasis on the actual methods and formulas used in each software tool (Rohland, 2008c).

References

- Campbell, J. B. 2002. *Introduction to remote sensing*. The Guilford Press, New York.
- Clausi, D. A. and Jernigan, E. M. 2000. Designing Gabor filters for optimal texture separability. *Pattern Recognition* 33: 1835-1849.
- Congalton, R. G. 1991. A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sensing of Environment* 37: 35-46.
- Congalton, R. G., Oderwald, R. G. and Mead, R. A. 1983. Assessing Landsat classification accuracy using discrete multivariate analysis statistical techniques. *Photogram. Eng, Remote Sens.* 49: 1671-1678.
- Davies, E. R. 2005. *Machine vision: theory, algorithms, practicalities*. Elsevier, San Francisco, CA.
- Haralick, R. M., Shanmugan, K. and Dinstein, I. 1973. Textural features for image classification. *IEEE Transactions on Systems, Man & Cybernetics* 3: 610-621.

- Haralick, R. M. and Shanmugan, K. S. 1974. Combined spectral and spatial processing of ERTS imagery data. *Remote Sensing of the Environment* 3: 3-13.
- Hudson, W. D. and Ramm, C. W. 1987. Correct formulation of the Kappa coefficient of agreement. *Photogram. Eng., Remote Sens.* 53: 421-422.
- Idrissa, M. and Acheroy, M. 2002. Texture classification using Gabor filters. *Pattern Recognition Letters* 23: 1095-1102.
- Landgrebe, D. A. 2003. *Signal theory methods in multispectral remote sensing*. John Wiley & Sons, Inc, Hoboken, NJ.
- Landgrebe, D. A. and Biehl, L. 1995. *MultiSpec software*. Multispectral classification software. West Lafayette, IN: Laboratory for Applications in Remote Sensing (LARS).
http://cobweb.ecn.purdue.edu/~biehl/MultiSpec/download_win.html
- Laws, K. I. 1980a. Rapid texture identification. *Proceedings of the SPIE* 238 Image Processing for Missile Guidance: 376-380.
- Laws, K. I. 1980b. Textured Image Segmentation, University of Southern California: Los Angeles. 178pp.
- Maenpaa, T. and Pietikainen, M. 2004. Classification with color and texture: jointly or separately? *Pattern Recognition* 37: 1629-1640.
- Maenpaa, T. and Pietikainen, M. 2005. Texture analysis with local binary patterns. *Handbook of Pattern Recognition and Computer Vision*, 3rd ed., pp. 197-216. World Scientific, Singapore.
- Mandelbrot, B. B. 1983. *The fractal geometry of nature*. W.H. Freeman & Co., New York.
- Manthalkar, R., Biswas, P. K. and Chatterji, B. N. 2003. Rotation invariant texture classification using even symmetric Gabor filters. *Pattern Recognition Letters* 24: 2061-2068.
- MIT Media Lab. 1995. *VisTex texture database*.
<http://vismod.media.mit.edu/pub/VisTex/>
- Ojala, T., Pietikainen, M. and Maenpaa, T. 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 24: 971-987.
- Parker, J. R. 1997. *Algorithms for image processing and computer vision*. John Wiley & Sons, New York.
- Pietikainen, M. K., ed. 2000. *Texture analysis in machine vision*. World Scientific Publishing Co. Pte. Ltd., Singapore.
- Richards, J. A. and Jia, X. 2006. *Remote sensing digital image analysis*. Springer-Verlag, Berlin.
- Rohland, R. M. 2008a. Texture Classification Tools for ESRI ArcGIS 9.3: A Technical Walkthrough. Volume: 63 pp.
<http://www.rmrohland.com>
- Rohland, R. M. 2008b. An Overview of Classification Methods for Remote Sensing. Volume: 84 pp.
<http://www.rmrohland.com>
- Rohland, R. M. 2008c. Texture Classification Tools Help Dialogs. Volume: 39 pp.
<http://www.rmrohland.com>
- Rosenfield, G. H. and Fitzpatrick-Lins 1986. A coefficient of agreement as a measure of thematic classification accuracy. *Photogram. Eng., Remote Sens.* 52: 223.
- Tso, B. and Mather, P. M. 2001. *Classification methods for remotely-sensed data*. Taylor & Francis, London.