

This is to certify that the
dissertation entitled

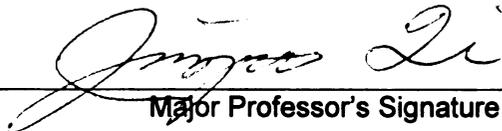
LAND COVER MAPPING AT SUB-PIXEL SCALES

presented by

YASUYO KATO MAKIDO

has been accepted towards fulfillment
of the requirements for the

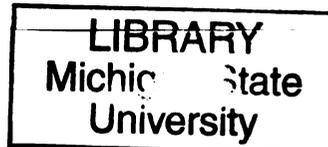
Ph.D. degree in Department of Geography


Major Professor's Signature

12/6/06

Date

MSU is an Affirmative Action/Equal Opportunity Institution



PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.
MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE

LAND COVER MAPPING AT SUB-PIXEL SCALES

By

Yasuyo Kato Makido

A DISSERTATION

**Submitted to
Michigan State University
in partial fulfillment of requirements
for the degree of**

DOCTOR OF PHILOSOPHY

Department of Geography

2006

ABSTRACT

LAND COVER MAPPING AT SUB-PIXEL SCALES

By

Yasuyo Kato Makido

One of the biggest drawbacks of land cover mapping from remotely sensed images relates to spatial resolution, which determines the level of spatial details depicted in an image. Fine spatial resolution images from satellite sensors such as IKONOS and QuickBird are now available. However, these images are not suitable for large-area studies, since a single image is very small and therefore it is costly for large area studies. Much research has focused on attempting to extract land cover types at sub-pixel scale, and little research has been conducted concerning the spatial allocation of land cover types within a pixel. This study is devoted to the development of new algorithms for predicting land cover distribution using remote sensory imagery at sub-pixel level.

The “pixel-swapping” optimization algorithm, which was proposed by Atkinson for predicting sub-pixel land cover distribution, is investigated in this study. Two limitations of this method, the arbitrary spatial range value and the arbitrary exponential model of spatial autocorrelation, are assessed. Various weighting functions, as alternatives to the exponential model, are evaluated in order to derive the optimum weighting function. Two different simulation models were employed to develop spatially autocorrelated binary class maps. In all tested models, Gaussian, Exponential, and IDW, the pixel swapping method improved classification accuracy compared with the initial random allocation of sub-pixels. However the results suggested that equal weight could

be used to increase accuracy and sub-pixel spatial autocorrelation instead of using these more complex models of spatial structure.

New algorithms for modeling the spatial distribution of multiple land cover classes at sub-pixel scales are developed and evaluated. Three methods are examined: sequential categorical swapping, simultaneous categorical swapping, and simulated annealing. These three methods are applied to classified Landsat ETM+ data that has been resampled to 210 meters. The result suggested that the simultaneous method can be considered as the optimum method in terms of accuracy performance and computation time.

The case study employs remote sensing imagery at the following sites: tropical forests in Brazil and temperate multiple land mosaic in East China. Sub-areas for both sites are used to examine how the characteristics of the landscape affect the ability of the optimum technique. Three types of measurement: Moran's I , mean patch size (MPS), and patch size standard deviation (STDEV), are used to characterize the landscape. All results suggested that this technique could increase the classification accuracy more than traditional hard classification.

The methods developed in this study can benefit researchers who employ coarse remote sensing imagery but are interested in detailed landscape information. In many cases, the satellite sensor that provides large spatial coverage has insufficient spatial detail to identify landscape patterns. Application of the super-resolution technique described in this dissertation could potentially solve this problem by providing detailed land cover predictions from the coarse resolution satellite sensor imagery.

Copyright by
YASUYO KATO MAKIDO
2006

To
my daughter Anna with all my love

ACKNOWLEDGEMENTS

I would like to begin by thanking my faculty advisor, Dr. Jiaguo Qi, at Michigan State University. Without his earnest support and encouragement, this paper would never have seen the light of day. I would also like to express my sincere gratitude to my committee members, Dr. Ashton Shortridge, Dr. Joseph P. Messina, and Dr. Runsheng Yin for giving me advice and being patient while I worked through this paper. In addition, I must extend a special thank you to Dr.Shortridge for his guidance and for editing the many drafts of this text.

I would like to thank Eraldo Matricardi and Yuan Zhang for providing valuable data for my research. I would also like to thank my friends and mentors Marian Mitchell and Dr. Catherine Yansa for their moral support and encouragement. Many thanks are given to Sharon Ruggles for her help with my correspondence and in negotiating the University's administrative hurdles.

I am also grateful to my fellow students, the faculty, and staff members for their friendship and support throughout my time in the Department of Geography. Finally, special thanks to my daughter Anna for understanding and tolerance of my need to focus on work. Without her love and support, I could not have completed this work.

TABLE OF CONTENTS

LIST OF TABLES..... ix

LIST OF FIGURES.....x

CHAPTER 1

Introduction1

1.1 Up- and Down- Scaling1

1.2 Literature Review5

1.3 Research Objectives9

CHAPTER 2

Modeling Binary Landscapes at Sub-pixel Scales.....12

2.1 Current method12

2.2 Research Issues13

2.3 New Modeling Methods14

2.3.1 Simulation of Autocorrelated Images.....14

2.3.2 Spatial Resolution.....16

2.3.3 Identifying Neighbors.....17

2.3.4 Spatial Weighting Functions18

2.4 Results20

2.5 Discussions31

CHAPTER 3

Modeling Multiple Class Land Cover at Sub-pixel Scales33

3.1 Introduction33

3.2 Methods33

3.2.1 Sequential categorical swapping34

3.2.2 Simultaneous categorical swapping.....35

3.2.3 Simulated Annealing37

3.3 Case study.....39

3.4 Results and Discussion43

3.5 Conclusions53

CHAPTER 4

Application to Real Landscapes	57
4.1 Introduction	57
4.2 Study area	58
4.2.1 Tropical Forests in state, Brazil	58
4.2.2 Temperate land mosaic in East China	61
4.3 Methods	64
4.3.1 Characterizing landscape pattern	64
4.3.2 Accuracy Assessment	67
4.4 Results	70
4.4.1 Brazil	70
4.4.2 China.....	75
4.5 Discussion and Conclusions	82

CHAPTER 5

Conclusions and Future Research	85
--	-----------

APPENDICES.....

Appendix A (IDL program for pixel-swapping algorithm using exponential weighting function).....	90
Appendix B (IDL program for pixel-swapping algorithm using exponential weighting function).....	96
Appendix C (IDL program for Sequential Categorical Swapping).....	103
Appendix D (IDL program for Simultaneous Categorical Swapping).....	119
Appendix E (IDL program for Simulated Annealing).....	132

REFERENCES

LIST OF TABLES

Table 2-1. Mean accuracy and student t-test using 20 simulated neutral images	30
Table 3-1. Decision Rule Table	37
Table 3-2. Accuracy comparison for Simultaneous method	50
Table 3-3. Accuracy comparison for Sequential method	51
Table 3-4. Moran's <i>I</i> value for each class	52
Table 4-1. Various extent and grains for entire and sub-areas	65
Table 4-2. Confusion matrix for the Brazil Data (actual values in rows, predictions in columns)	73
Table 4-3. Accuracy of Simultaneous method for the Brazil Data per class.....	73
Table 4-4. Accuracy of Simultaneous method for the Brazil Data	73
Table 4-5. Correlation coefficients between PCC and MPS, STDEV and Moran's <i>I</i> using 60 sub-areas (315 x 315 pixels).....	75
Table 4-6. Confusion matrix for China 1978 (actual values in rows, predictions in columns)	78
Table 4-7. Accuracy of the Simultaneous Method for China 1978 per class.....	78
Table 4-8. Confusion matrix for China 2004 (actual values in rows, predictions in columns)	79
Table 4-9. Accuracy of Simultaneous Method for China 2004 per class.....	79
Table 4-10. Accuracy of Simultaneous Method.....	79
Table 4-11. Correlation coefficient between PCC and MPS, STDEV and Moran's <i>I</i>	79
Table 4-12. Correlation coefficient between Increase rate of PCC from random allocation and MPS, STDEV and Moran's <i>I</i>	81
Table 4-13. Correlation coefficient between three indices (MPS, STDEV and Moran's <i>I</i>)	81

LIST OF FIGURES

Figure 1-1. The pixel view of the world (From Fisher, 1997).....	3
Figure 1-2. Four causes of mixed pixels (From Fisher, 1997)	3
Figure 2-1. Neutral A: Cell size 1.....	15
Figure 2-2. Neutral B: Cell size 1.....	15
Figure 2-3. Neutral A: Cell size 10.....	16
Figure 2-4. Neutral B: Cell size 10.....	16
Figure 2-5. Resampled cell size 3.....	17
Figure 2-6. Moran's <i>I</i> for various cell sizes.....	21
Figure 2-7. Moran's <i>I</i> and accuracy (PCC (%)) for various iterations (Exponential, cell 7 to 1, range=15).....	24
Figure 2-8. Moran's <i>I</i> and accuracy (PCC (%)) for various range values (Exponential, cell 7 to 1).....	25
Figure 2-9. Moran's <i>I</i> and accuracy (PCC (%)) for various range values (Gaussian, cell 7 to 1).....	26
Figure 2-10. Moran's <i>I</i> and accuracy (PCC (%)) for various K-values (IDW, cell 7 to 1)	27
Figure 2-11. Moran's <i>I</i> and accuracy (PCC (%)) for various iterations (Equal weight, cell 7 to 1).....	28
Figure 2-12. Neutral A result (Nearest Neighbors function, cell 7to1, after 20 iterations)	29
Figure 2-13. Neutral B result (Nearest Neighbors function, cell 7to1, after 20 iterations)	29
Figure 3-1. Initial random allocation.....	35
Figure 3-2. Binary matrix	36
Figure 3-3. Attractiveness O_i (Grey color indicates the pixel is occupied by the class)...36	36
Figure 3-4. Classified image (Mode 5).....	42
Figure 3-5. Input images for five classes (Mode 5).....	42

Figure 3-6. Left: Subset of the reference data, Right: Subset of degraded image for vegetation class (Mode 5).....	43
Figure 3-7. Initial random allocation (Mode 5).....	44
Figure 3-8. Output of Simultaneous method (after 30 iterations) (Mode 5)	45
Figure 3-9. Example hard classification output (Mode 5).....	45
Figure 3-10. Overall accuracy of the methods (Non-filter).....	47
Figure 3-11. Overall accuracy of the methods (Mode 3)	47
Figure 3-12. Overall accuracy of the methods (Mode 5)	48
Figure 3-13. Overall accuracy of the methods (Mode 7)	48
Figure 3-14. Images for Non-filter. (a) reference image (b) output for simultaneous method after 30 iterations (c) hard classification	50
Figure 4-1. Study site location (Brazil: Landsat scene, path 226 and row 068).....	59
Figure 4-2. Left: Brazil, classified Landsat image, Right: Brazil sub image	60
Figure 4-3. Study site location for China	62
Figure 4-4. China reference image 1978	63
Figure 4-5. China reference image 2004	63
Figure 4-6. Systematic samples of sub-areas: 60 samples (Left), 8 samples (Right).....	66
Figure 4-7. China 1978 sub-areas (315x315) (Left: STDEV=11,482, MPS = 2,177, Right: STDEV = 3,748, MPS =2,000).....	67
Figure 4-8. Mixed pixels used for adjusted Kappa statistics for Brazil	69
Figure 4-9. Output of simultaneous method (after 20 iterations)	71
Figure 4-10. Brazil sub set (2 different areas).....	72
Figure 4-11. Brazil: simultaneous method and hard classification results	73
Figure 4-12. Simultaneous method: top (MPS at PCC), middle (STDEV at PCC), bottom (Moran's <i>I</i> at PCC)	74
Figure 4-13. China 1978, 2004: simultaneous method and hard classification results	77
Figure 4-14. China 1978 (sub-area=98x98): Moran's <i>I</i> at PCC	80

Figure 4-15. Sub-area (98x98) for China 1978: Moran's $I = 0.72$, PCC =99.6%, MPS = 3072, STDEV = 528680

Figure 4-16. Patch size distribution at sub-area (315x315) at China 197881

Images in this dissertation are presented in color

Chapter 1

Introduction

1.1 Up- and Down- Scaling

Scale is one of the fundamental attributes in describing geographic data and yet the word is ambiguous. The term “scale” has a variety of meanings and has been used in various disciplines. There are at least four meanings of scale within the spatial domain: the cartographic or map scale, the geographic (observational) scale, the operational scale, and the measurement scale (resolution) (Cao and Lam, 1997). The fourth scale, the spatial resolution, refers to the smallest distinguishable parts of an object (Tobler, 1988). Pixels in a remote sensed imagery are defined by the combination of the height and instantaneous field-of-view (IFOV) of the sensor (Atkinson and Curran, 1995), and limit the quantitative potential of land cover information from remote sensing imagery. Land cover is a fundamental variable that underpins much scientific research. Accurate land cover information is both difficult and expensive to obtain. Remote sensing has the potential to provide such information. Many researchers have conducted research that focused on increasing accuracy of land cover classification from remote sensing imagery (e.g., Justice and Townshend, 1981). The issue remains that land cover data provided by remote sensing are limited by the spatial resolution of the sensor. Increasing the spatial resolution generally reveals greater detail (Atkinson, 2005), and spatial resolution has been the subject of research in remote sensing for many years (Woodcock and Strahler, 1987, Atkinson and Tate, 2000). The spatial variation observed in remote sensing imagery is a function of both the property of interest and the sampling framework

(Atkinson, 2005). Researchers attempt to evaluate the effect of spatial resolution on detectable spatial variation as characterized by functions, such as local variance and the variogram (Woodcock and Strahler, 1987, Curran and Atkinson, 1998). Such research confirmed that spatial resolution has a fundamental effect on the spatial variation in remotely sensed imagery (Atkinson, 2005). Because of the limitation of data storage capacity, large area studies are often associated with coarse resolution imagery (such as MODIS and National Oceanic and Atmospheric Administration (NOAA) Advanced Very High Resolution Radiometer (AVHRR)) and fine resolution imagery (such as IKONOS and QuickBird) is the characteristic of small area studies (Cao and Lam, 1997). Such coarse resolution imagery has high temporal frequency, but may not be fine enough for monitoring environment. For example, the 1.1 km data from AVHRR are adequate for mapping large scale phenomena but too coarse for mapping finer scale phenomena, such as changes in wetland dynamics (Pelky *et al.*, 2003), small burn scars (Hlavka and Livingston, 1997), and typical agricultural fields in UK (Atkinson, *et al.*, 1997).

Early techniques for land cover classification from remotely sensed imagery employed hard classification in which each pixel was classified into one of many land cover types, implying that land cover classes exactly fit within the bounds of one or more pixels (Figure 1-1). However, many pixels consist of a mixture of multiple land cover classes. Thus, remote sensing images contain a combination of pure and mixed pixels. Fisher (1997) lists four cases of mixed pixels (Figure 1-2).

- (a) small sub-pixel objects (a house or tree)
- (b) boundaries between two or more mapping units (field-woodland boundary)
- (c) the intergrade between central concepts of mappable phenomena (ecotone)

(d) linear sub-pixel objects (a road)

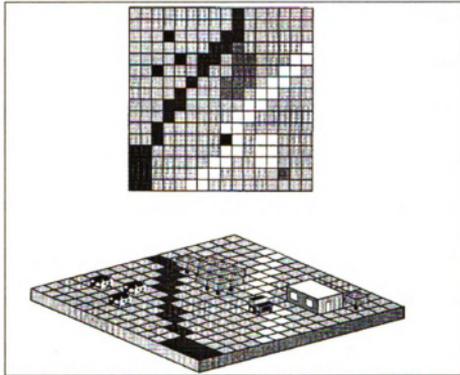


Figure 1-1. The pixel view of the world (From Fisher, 1997)

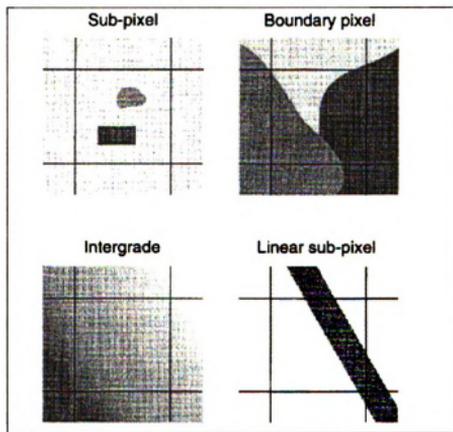


Figure 1-2. Four causes of mixed pixels (From Fisher, 1997)

The solution to the mixed pixel problem typically centers on soft (often termed fuzzy in the remote sensing literature) classification, which allows proportions of each pixel to be partitioned between classes. Sub-pixel class composition is estimated through the use of techniques, such as mixture modeling (e.g., Kerdiles and Grondona, 1995), supervised fuzzy c-means classification (e.g., Foody and Cox, 1994) and artificial neural networks (e.g., Kanellopoulos *et al.*, 1992). The output of these techniques generally produces images that display the proportion of a certain class within each pixel. For example, these techniques may predict that a certain pixel is comprised of 70 percent forest and 30 percent non-forest. In most cases, this results in a more informative and less error prone representation of land cover than that produced using a hard, one-class-per-pixel classification (McKelvey and Noon, 2001). However, the spatial distribution of these class components within the pixel remains unknown. It would be useful to know *where* the class components are located spatially within the pixel, and this is a goal of this thesis.

Much previous research has been focused on attempting to extract class proportion of sub-pixel scale features, and very little research has been conducted concerning the allocation of class proportions within a pixel. The objective of this study is to overcome the mixed pixel problem by investigating a method for predicting sub-pixel land cover distribution for multiple land cover classes. This innovative method can produce a fine resolution land cover map without a need for any additional data. In the following section, various sub-pixel mapping algorithms are discussed. This is followed by the research objectives and the outlines of dissertation.

1.2 Literature Review

Several algorithms have been proposed for allocating classes of sub-pixels. Foody (1998) introduced a simple regression based approach to create a sharpened fuzzy classification image through the use of an additional finer spatial resolution image. The approach was illustrated by refining a fuzzy classification with a sharpening image at a resolution one half of that of the image used to derive the classification. This approach was applied to a small lake with islands. The resulting sharpened fuzzy classification provided a visually accurate representation of land cover. However, the areal extent of the lake was not maintained, and it was not always possible to obtain two images at same area of different spatial resolutions. Gavin and Jennison (1997) adopted a Bayesian approach which incorporates prior information about the true image in a stochastic model that attached higher probability to images with shorter total edge length. The model produced accurate results, but the multistage operation was computationally intensive and was most suitable for small objects. Aplin *et al.* (1999) developed a set of techniques to classify land cover on a per-field basis, rather than a traditional per-pixel basis, by utilizing the Ordnance Survey land line vector data. They concluded that the per-field classification technique was generally more accurate than the per-pixel classification. However, the necessity of accurate vector data sets limited this technique.

Atkinson (1997) originally proposed super-resolution mapping using only the output from a soft classification. The idea was to maximize the spatial autocorrelation between neighboring sub-pixels while honoring the original pixel proportions. The approach, which comprised two stages, was proposed for preparing remotely sensed images so that sub-pixel vector boundaries in land cover might be mapped. The first stage

involved the application of a technique for estimating the land cover proportions for individual pixels. The second stage involved a new technique to determine where the relative proportions of each class occurred within each pixel. The algorithm worked by assuming spatial dependence within and among pixels. Verhoeye and De Wulf (2002) adapted this assumption and introduced an approach that formulated the sub-pixel mapping concept as a linear optimization problem to maximize spatial autocorrelation within the image. They produced a sharpened crisp land cover map without the need for finer spatial resolution data. However, this non-iterative solution produced linear artifacts in the final map. Mertens *et al.* (2003) employed the same optimization function as Verhoeye and De Wulf (2002), but used a procedure based on Genetic Algorithms (GA), a fast search technique, based on principles of natural selection. This method can only be used for small images with a few land cover classes and a small upscaling factor. Kasetkasem *et al.* (2005) introduced a Markov random field (MRF) model based approach to generate super-resolution land cover maps from remote sensing data. It was assumed that a super-resolution map has MRF properties, i.e., two adjacent pixels are more likely to belong to the same land cover class than different classes. The results showed a considerable increase in the accuracy of land cover maps over those obtained from a linear optimization approach suggested by Verhoeye and De Wulf (2002). An advantage of this method was that the algorithm used the image directly without requiring the output of soft classification techniques, since the method included the step to generate fraction images from the coarse resolution multi-spectral images. Tatem *et al.* (2001, 2002) examined the application of a Hopfield neural network (HNN) technique to predict the spatial pattern of land cover features smaller and larger than the scale of a

pixel by using information about class composition determined from soft classification. A Hopfield neural network was used as an optimization tool to make the output of a neuron similar to that of its neighboring neuron in order to maximize the spatial autocorrelation within the image. Tatem *et al.* (2003) applied the HNN technique to Landsat Thematic Mapper (TM) agricultural imagery to derive accurate estimates of land cover and reduce uncertainty inherent in such imagery, and demonstrated that the spatial resolution of satellite sensor imagery did not necessarily represent a limit to the spatial detail obtainable within land cover maps derived from such imagery. Boucher and Kyriakidis (2006) introduced a non-iterative super-resolution land cover mapping using indicator cokriging, that approximated the probability that a pixel at the fine resolution belonged to a particular class, given the coarse resolution fractions and a sparse set of class labels at some informed fine pixels. Such kriging-derived probabilities were used in sequential indicator simulation to generate synthetic maps of class labels at the fine resolution pixels. As authors stated this simulation procedure would be faster than the other iterative procedures. However, it is not always possible to obtain a prior model of spatial structure for the fine resolution.

Atkinson (2001, 2005) examined the “pixel-swapping” optimization algorithm within a geostatistical framework as an alternative to the HNN algorithm. Like Verhoeve and De Wulf (2002) and Tatem *et al.* (2001, 2002, 2003), Atkinson used the proportions of each land cover within each pixel to map the location of class components within the pixels. These class proportions can be derived from various soft classification methods, which are described above. Unlike Verhoeve and DeWulf (2002) and Boucher and Kyriakidis (2006), the “pixel-swapping” algorithm iteratively allocated sub-pixels to

maximize the contiguity of the landscape. This simple algorithm is similar in character to simulated annealing. Simulated annealing is a family of optimization algorithms based on the principle of stochastic relaxation. An initial image is gradually changed so as to match user-specified constraints (Goovaerts, 1997). However, unlike the basic simulated annealing approach, which randomly selects pairs of sites for swapping, Atkinson's optimization algorithm deterministically selects the two sites most in need of swapping based on an attractiveness index, O_i . Consequently, the pixel-swapping algorithm is relatively fast since convergence occurs in far less iterations. However, several aspects of this algorithm deserve further investigation: the choice of the exponential weighting function is arbitrary, and the value of the non-linear parameter of the exponential model (a) is experimentally derived. Moreover, the algorithm is only applicable for binary images.

Although the linear optimization approach (Verhoeve and De Wulf, 2002) can be applied to multiple class land covers, the non-iterative solution produced the linear artifacts. Both Foody's (1998) and Aplin *et al.*'s (1999) approaches needed additional images to create finer resolution images. The HNN approach was applicable for multiple classes and did not require any additional images. However, it is not particularly accessible to the remote sensing practitioner. A clear need exists for a new algorithm for super resolution mapping that uses only class proportions information that can apply to multiple land covers and is simple enough to implement in computer language.

This study adopts the assumption of spatial dependency both within and among pixels, as do most existing algorithms. Therefore, the various algorithms introduced in

this study work best where the land cover features are larger than the sub-pixels and are spatially autocorrelated.

1.3 Research Objectives

The overall objective of this research is to overcome the mixed pixel problem by developing a method for predicting sub-pixel land cover distribution. The specific objectives of my research are:

1. To improve the Atkinson pixel-swapping algorithm by developing solutions to the two specific limitations: the arbitrary spatial range value, and the arbitrary exponential model for characterizing spatial autocorrelation
2. To develop new algorithms that can be applied to multiple land covers
3. To assess the quality of these methods using remotely sensed image based case studies

The new approach for predicting sub-pixel land cover distribution for multiple land cover classes can benefit researchers who employ remote sensing imagery. In many cases, the satellite sensor that provides large spatial coverage has insufficient spatial detail to identify landscape patterns. In other cases, there are only image archives available with coarse resolution for previous time periods. Application of the super-resolution technique described in this paper could solve these problems by providing detailed land cover predictions from relatively coarse resolution satellite sensor imagery.

The dissertation is outlined as follows:

Chapter 2 (Objective 1) focuses on research to improve the pixel-swapping algorithm proposed by Atkinson for predicting sub-pixel land cover distribution. Two limitations of this method are assessed: the arbitrary spatial range value and the arbitrary exponential model for characterizing spatial autocorrelation. For this assessment, two different simulation models are employed to develop spatially autocorrelated binary class raster maps. These models are then resampled to generate sets of representative medium resolution class maps.

Chapter 3 (Objective 2) describes the development of new algorithms that are applicable to multiple land covers for predicting the land cover distribution at sub-pixel scales. Three methods are examined: sequential, simultaneous, and simulated annealing (SA). An optimum method is selected based on its classification accuracy and computation time.

Chapter 4 (Objective 3) focuses on the application of the optimum technique to satellite imagery in Brazil and China. Landsat ETM+ imageries are classified into six classes for Brazil and seven classes for China. Landsat MSS images with a corresponding Digital Elevation Model in China are used to develop a 12-class scheme. Sub-areas for these classified land cover maps are examined to determine how the characteristics of the landscape affect the ability of the optimum technique. Three types of measurement, Moran's I , mean patch size (MPS), and patch size standard deviation (STDEV), are used to characterize landscape structure.

Chapter 5 contains the conclusions and challenges of this study. It summarizes results and addresses the potential of the sub-pixel models to impact applications in GIS research where observations or measurements are spatially aggregated are also addressed.

Chapter 2

Modeling Binary Landscapes at Sub-pixel Scales

2.1 Current method

Atkinson (2001, 2005) introduced a pixel-swapping algorithm for predicting sub-pixel land cover distribution. This algorithm successfully allocates class distributions within a pixel and is simple enough to code in any computing language. Initially, the algorithm randomly allocates class codes to sub-pixels. The attractiveness of each sub-pixel location is calculated based on the current arrangement of sub-pixel classes. Then the attractiveness metric is used to conduct subsequent cell swapping. In this algorithm, the exponential weighting function is used to calculate the attractiveness:

$$O_i = \sum_{j=1}^n \lambda_{ij} \cdot Z(X_j) \quad (1)$$

n : the number of neighbors

$Z(X_j)$: value of the binary class z at the j th pixel location X_j

λ_{ij} : a weight predicted as:

$$\lambda_{ij} = \exp\left(\frac{-h_{ij}}{a}\right) \quad (2)$$

h_{ij} : the lag between the pixel location for X_i and X_j

a : the range parameter of the exponential model

Once the attractiveness is predicted, the algorithm ranks the scores on a pixel-by-pixel basis. For each pixel, the least attractive (Smallest O_i) location currently allocated to a 1 and the most attractive (Greatest O_i) location currently allocated to a 0 are stored. If Smallest O_i is less than Greatest O_i then the classes are swapped. This procedure is repeated either for a fixed number of iterations or until the optimization algorithm fails to make a change. Thus, the spatial arrangement of sub-pixel values is iteratively changed in order to maximize the correlation between neighboring sub-pixels.

2.2 Research Issues

The pixel-swapping algorithm is demonstrated to produce excellent results for relatively simple images. However, several aspects of this algorithm deserve further investigation: the choice of the exponential weighting function is arbitrary, and the value of the non-linear parameter of the exponential model (a) is experimentally derived. This chapter reports on research to improve the pixel-swapping algorithm by considering alternatives to these two limitations.

In the following section, the method for testing the parameters of this algorithm is described, including a discussion of the derivation of the test data, the measurement of spatial structure, and the weighting schemes employed. This is followed by the presentation of results exploring the relationship between spatial autocorrelation and sub-pixel classification accuracy, as well as a comparison of alternative weighting function performance. This chapter concludes with a discussion of implications for sub-pixel classification and potential lines of subsequent research.

2.3 New Modeling Methods

In this section, I introduce the data sets employed and discuss how various parameters of the Atkinson algorithm were tested. The data sets employed in this research were generated using C code developed by Dr. Shortridge, while the Atkinson algorithm was implemented for this work by the author using custom code in the IDL programming language 6.2 (Interactive Data Language, Research Systems Inc, Boulder Colorado).

2.3.1 Simulation of Autocorrelated Images

This case study employs simulated binary images that have 315 rows and columns with substantial positive autocorrelation. Two methods are used to develop these images. Both methods create binary raster files that are spatially autocorrelated at a level specified by a target Moran's I statistic set by the user (0.7, in both cases). Moran's I is an indicator of spatial autocorrelation for area data (Bailey and Gatrell, 1995). For a spatial proximity matrix (W), spatial correlation in attribute values (y_i) is estimated as:

$$I = \frac{n \sum_{i=1}^n \sum_{j=1}^n w_{ij} (y_i - \bar{y})(y_j - \bar{y})}{\left(\sum_{i=1}^n (y_i - \bar{y})^2 \right) \left(\sum_{i \neq j} w_{ij} \right)} \quad (3)$$

The Moran index is positive when nearby areas tend to be similar in attribute, negative when they tend to be more dissimilar than one might expect, and approximately zero when attribute values are arranged randomly and independently in space (Goodchild, 1986).

Method 1 uses an initially random distribution, while Method 2 uses a fractal model (midpoint displacement) to rapidly initialize a highly autocorrelated surface. Then a cell-swapping algorithm is employed to shift the spatial arrangement to arrive at the specified I . Since these two methods result in notably different images despite the same Moran's I values, this study examines both simulated images (Figure 2-1, 2-2). I call the neutral image that is created by method 1 as Neutral A and that is created by method 2 as Neutral B. Both images contain 33% of 1 (forest) and 67% of 0 (non-forest). In Neutral A, the forest patches are evenly distributed throughout the area. In Neutral B, the forest patches are larger and clumped at some locations, which is a traditional fractal pattern. These images are aggregated to obtain coarser resolution images (Figure 2-3, 2-4), which will be the subject of super-resolution mapping. Therefore, "reference data" is available (Figure 2-1, 2-2) with which to test the results of these experiments on sub-pixel swapping.



Figure 2-1. Neutral A: Cell size 1

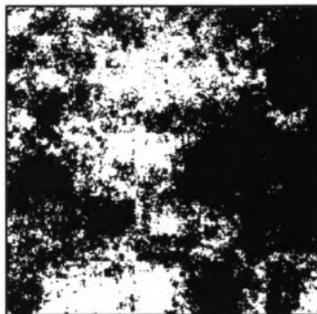


Figure 2-2. Neutral B: Cell size 1

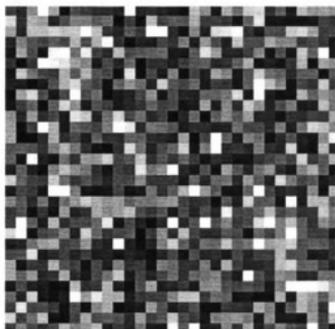


Figure 2-3. Neutral A: Cell size 10



Figure 2-4. Neutral B: Cell size 10

2.3.2 Spatial Resolution

In this study, the attempt is made to derive the optimum weighting function based on the degree of contiguity in the original map. Moran's I is used to characterize the structure of the landscape. However, geographic phenomena generally are scale-dependent, which means the analysis results could differ considerably if different pixel resolutions are used. An important source of uncertainty in remotely sensed data is caused by interactions between the scale of variation within the ground scene and the spatial resolution of the sensor (Friedl *et al.*, 2001). The relationship between the Instantaneous Field of View (IFOV) of the sensor system and the spatial variability in the landscape will influence the types of analyses that may be performed. Thus, Moran's I is a function of spatial resolution, and looking at a value calculated at the original pixel resolution may be misleading. Therefore, I quantify the effect of spatial resolution on the empirically calculated I .

The original Neutral A and B rasters, with cell sizes of 1, are resampled to cell sizes 2, 3, 4, ..., 10. These reduced-resolution version of the images are generated by the summation of the values of the input cells that are encompassed by the extent of the output cell, as demonstrated in Figure 2-5. Note that the resulting cell value indicates the proportion of the cell occupied by class 1, corresponding to the output of soft classification techniques.

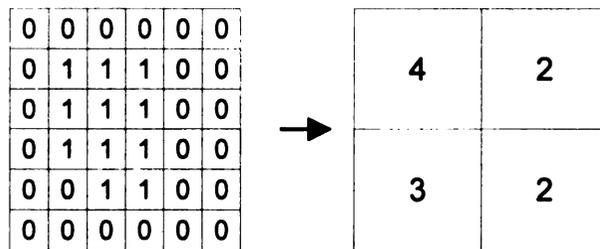


Figure 2-5. Resampled cell size 3

2.3.3 Identifying Neighbors

One important parameter of the pixel-swapping algorithm is the neighborhood definition. There are many ways to measure the proximity (nearness) of observations when dealing with area data. Some are distance-based, while others are neighbor-based. The choice of measure may affect the results. With n zones, the measurement of proximity takes the form of an $n \times n$ matrix W : the spatial proximity matrix. Each element w_{ij} in the matrix measure the spatial proximity of A_i to A_j . For this study, the following binary definition of w_{ij} is used (Bailey and Gatrell, 1995).

$$w_{ij} = \begin{cases} 1 & \text{centroid of } A_j \text{ shares an edge with } A_i \\ 0 & \text{otherwise} \end{cases}$$

A first-order neighbor under this definition would be one that directly borders A_i . A second-order neighbor would be one that does not share a border directly with A_i , but is a first-order neighbor with a zone A_k that does share a border with A_i (Bailey and Gatrell, 1995). In this study, equally weighted first-and second-order neighbors are employed for both simulated images. This corresponds to the 24 closest cells in a raster grid.

2.3.4 Spatial Weighting Functions

The Atkinson weighting function employs the exponential model, which is a standard covariance model. Therefore, I also tested another standard geostatistical covariance model, the Gaussian (Bailey and Gatrell, 1995). Inverse distance weighting (IDW) is a commonly employed interpolation technique that is both familiar and straightforward to implement as a distance-based weighting function.

The Atkinson spatial weighting function can be expressed as an Exponential covariance model:

$$\lambda_{ij} = \exp\left(\frac{-3h_{ij}}{r}\right) \quad (4)$$

h_{ij} : the lag between the pixel location for X_i and X_j

r : practical range of the covariance

The range is the lag distance in cells at which pixels become independent of each other. Earlier works on this function employed a parameter α , which was set to 5 (Atkinson, 2001, 2005); this is equivalent to a range r of 15. In this study, various range values from 1 to 20 for the Exponential model are examined.

The weighting function using a Gaussian model is:

$$\lambda_{ij} = \exp\left(\frac{-3h_{ij}^2}{r^2}\right) \quad (5)$$

The inverse distance weighting function is:

$$\lambda_{ij} = \frac{1}{h_{ij}^k} \quad (6)$$

k : a real number

For the Gaussian model, various range values r from 1 to 20 are examined. For the IDW model, various k values from 0 to 10 are examined.

In addition to testing the various distance weighting functions, a function applying equal weights to all first- and second-order neighbors is examined, which means the total 24 sub-pixels are involved for the computation. Attractiveness O_i can be modeled simply as the sum of the values at the nearest neighbors:

$$O_i = \sum_{j=1}^n Z(X_j) \quad (7)$$

n : the number of neighbors

$Z(X_j)$: value of the binary class z at the j th pixel location X_j

This simplified attractiveness makes the algorithm simpler and probably faster. This model is referred to subsequently as the Nearest Neighbor function.

In this study the resampled coarse spatial resolution images, which have cell sizes of 5 and 7, are subjected to super-resolution mapping using the various functions described to reproduce cell size 1, which is the original fine resolution image. The algorithm employed 9 and 13 iterations for the coarse 5 and 7 resolution images, respectively.

In all cases, Moran's I is used as an index of the spatial contiguity of the landscape, and Percent Correctly Classified (PCC) is used as a classification accuracy assessment. PCC is calculated as the ratio of the sum of correctly classified sub-pixels in all classes to the sum of the total number of sub-pixels (Congalton and Green, 1999). Reported PCC is the average of 20 trials for testing various range or k -values for Gaussian, Exponential and IDW models.

2.4 Results

Figure 2-6 shows the relationship between Moran's I and resampled coarse resolution images for both neutral images. In the case of Neutral A, as the resolution decreases, Moran's I decreases, which means the image is less autocorrelated than the original one. In the case of Neutral B, Moran's I does not change much with the resolution change, and all I values are higher than that for the original image, which means the coarsened image is more autocorrelated than the original one. While the two original neutral images have the same Moran's I value, the pattern of spatial distribution

are quite different; for Neutral A, the patches are distributed evenly throughout the area, while for Neutral B, the patches exhibit a traditional fractal pattern. As the images are coarsened, distribution differences become more apparent. Thus, the Moran's I value for any particular raster is dependent on the underlying process generating the autocorrelated pattern and, therefore, characterizing the relationship between resolution and Moran's I will be challenging without knowing the sub-pixel distribution pattern of the landscape. An underlying assumption of the Atkinson method is that land cover is dependent both within and between pixels. The algorithm attempts to maximize the degree of contiguity. However, as the relationship between Neutral A and B with respect to Moran's I demonstrates, spatial structure is a function of pixel resolution and underlying process. It may not always be desirable to maximize autocorrelation.

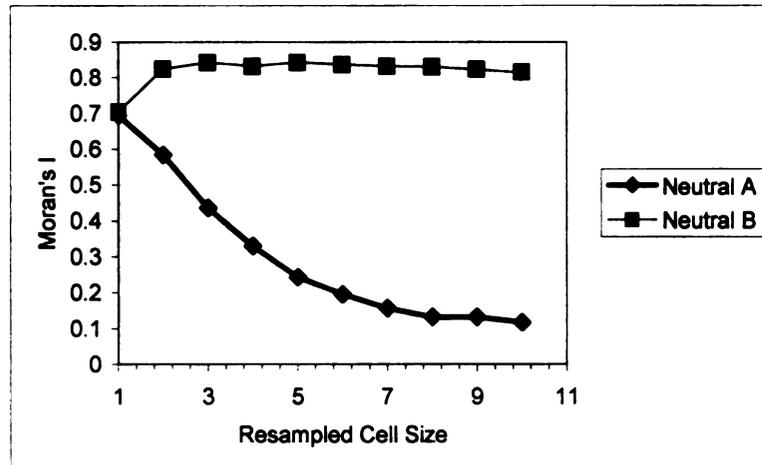


Figure 2-6. Moran's I for various cell sizes

Figure 2-7 demonstrates the effectiveness of the pixel-swapping method through iteration using the Exponential model. Classification accuracy and Moran's I increase until iteration 13 and level off for both neutral images for cell size 7. The results imply that the algorithm successfully increases the degree of spatial autocorrelation, and increases classification accuracy within a pixel. Cell size 5 images show similar results. However, the appropriate iteration number increases as the resampled cell size increases, since there are more sub-pixels to be swapped within a pixel. For the following examination, 13 iterations for cell size 7 and 9 iterations for cell size 5 are used.

For the Exponential and the Gaussian model, range values from 1 to 20 are tested. Moran's I and classification accuracy increase initially and then level off (Figure 2-8, 2-9). Although the plot for the IDW model has an appearance quite different from the other models (Figure 2-10), a similar relationship with accuracy and spatial autocorrelation are implied. In each case, as the weights become more similar, Moran's I and overall accuracy increase. Reported classification accuracies for all three figures are the average of 20 trials. The standard deviations for 20 trials vary with the range values and weighting functions employed; they range approximately from 0.3 to 0.6 percent for Neutral A, and from 0.1 to 0.4 percent for Neutral B. This variability is caused by different initial allocations, since the algorithm randomly creates different initial allocations based on the class proportion.

I tested the Nearest Neighbor function (Figure 2-11, 2-12, 2-13), with equal weights for each neighbor. The figures demonstrate that the Nearest Neighbor function produces results similar to the original Exponential model. Classification accuracy and Moran's I increase until iteration 13 and level off for both neutral images. I performed a

paired two-sample student's t-test using 20 simulated images derived from method 1 and 2 (Table 2-1). The null hypothesis is that the classification accuracies using the Nearest Neighbor function is equal to the one using the other models. Although the probabilities for the empirically-derived t-statistics vary with range and k values, the probabilities tend to exceed 0.05 for larger range values and smaller k values. The classification accuracies using the Nearest Neighbor function could produce equal results to the other models at 90 % confidence level when the weights are similar. When the weights are not similar, the Nearest Neighbor function tends to produce higher accuracy than the other models. Employing a Nearest Neighbor scheme appears preferable to the more complex alternatives since it can produce similar results and may also be computationally more efficient than using a distance-based weighting function. For the Nearest Neighbor function, the time taken for 13 iterations (cell 7 to 1) was less than 1 second on a P4 computer. It took approximately 50 seconds for the Exponential model.

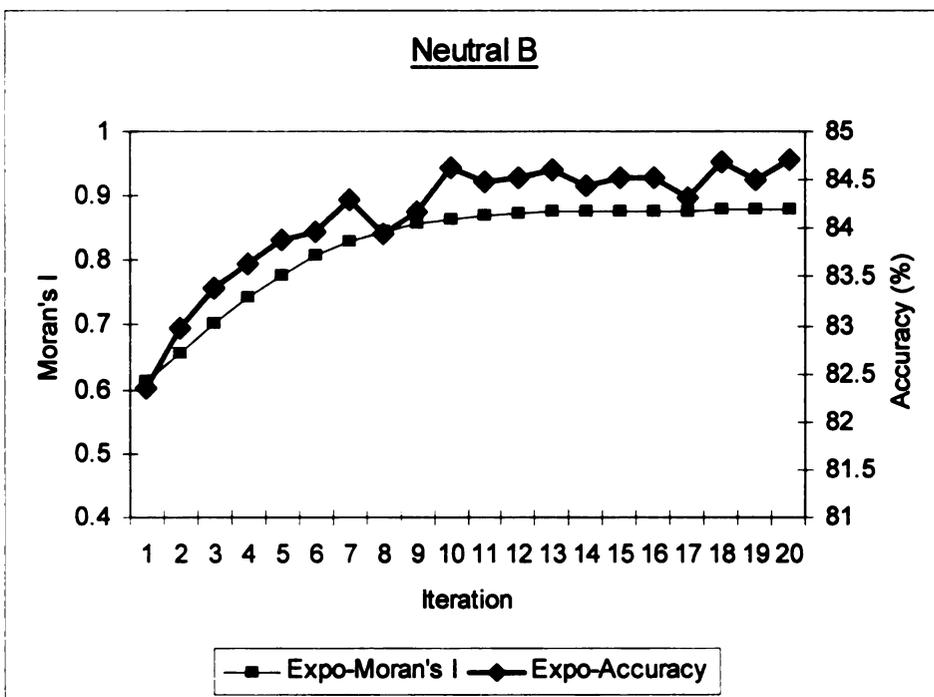
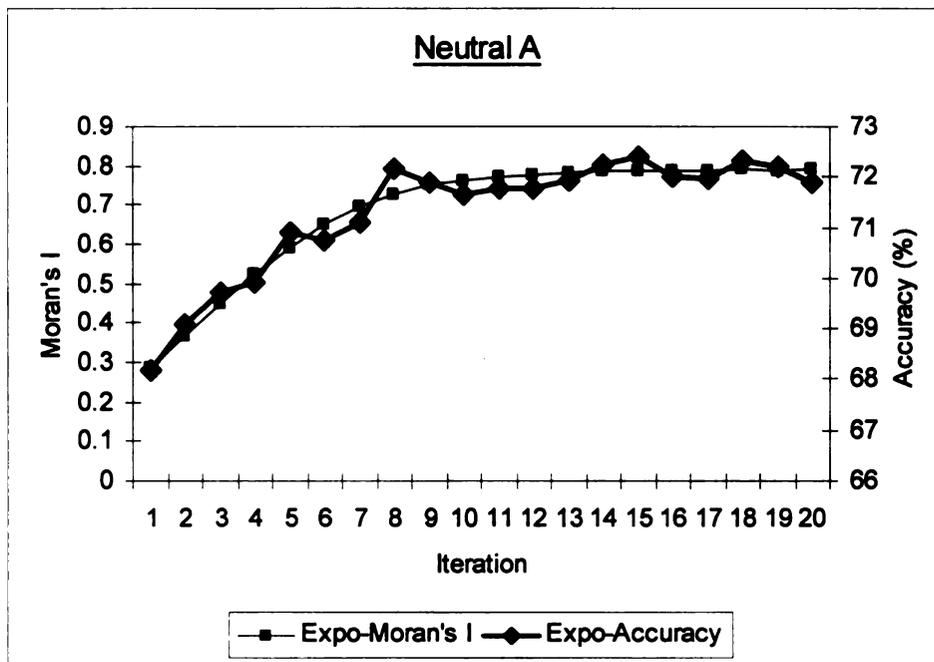


Figure 2-7. Moran's I and accuracy (PCC (%)) for various iterations (Exponential, cell 7 to 1, range=15)

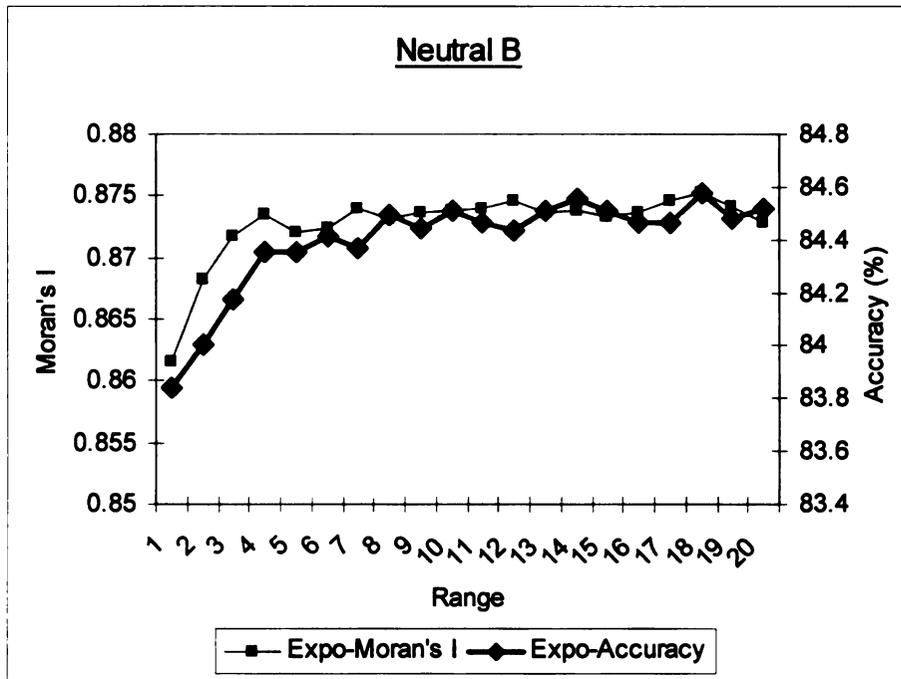
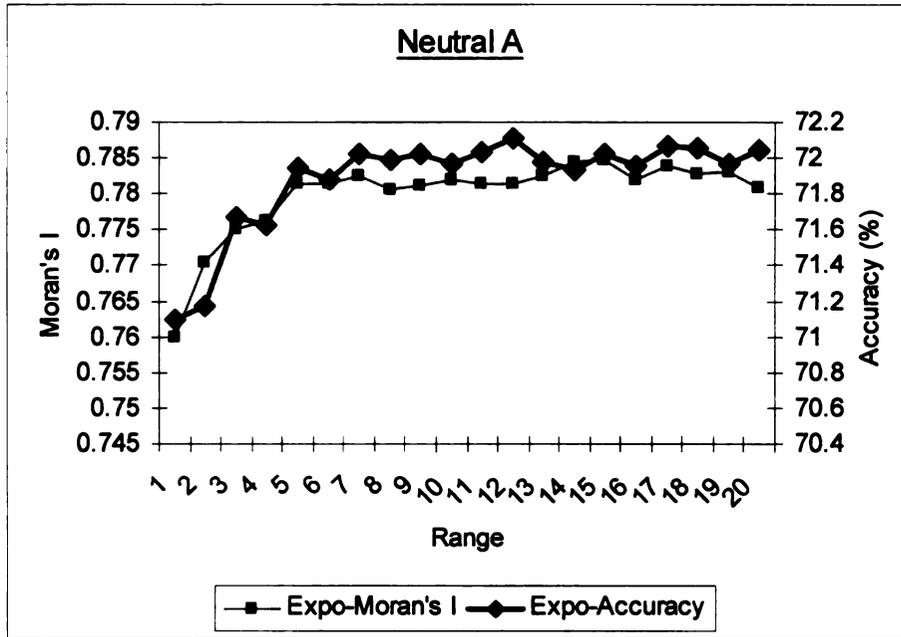


Figure 2-8. Moran's *I* and accuracy (PCC (%)) for various range values (Exponential, cell 7 to 1)

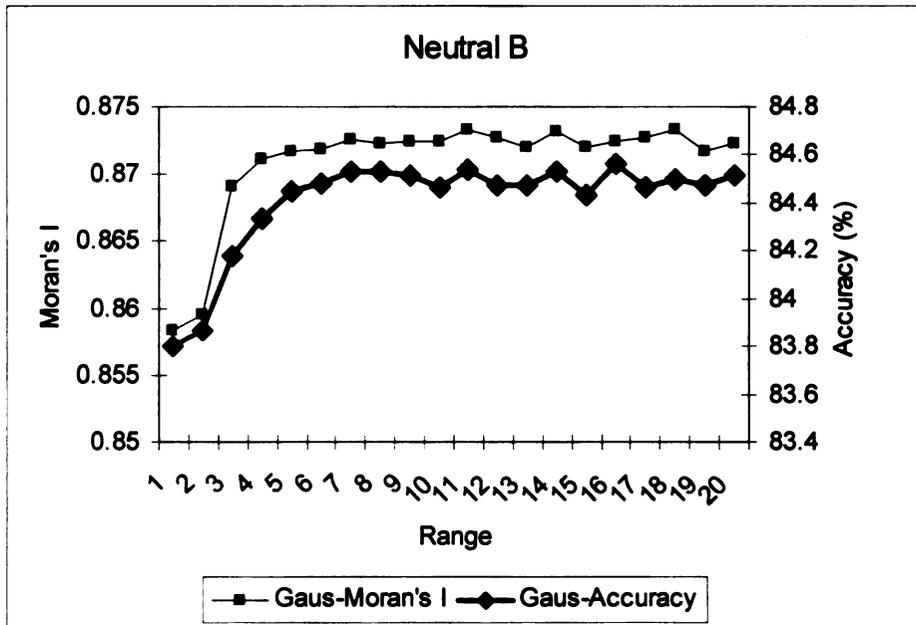
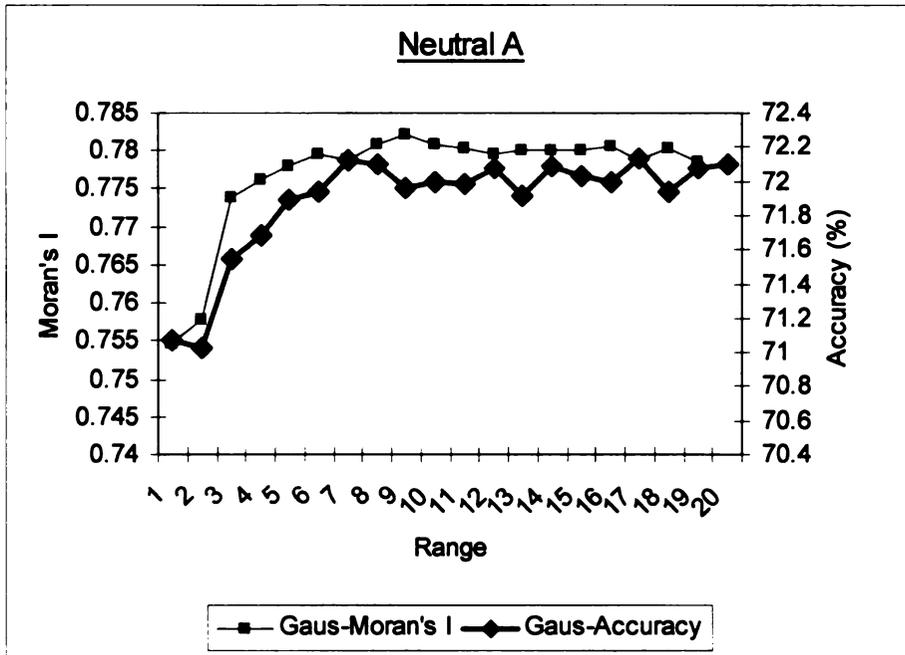


Figure 2-9. Moran's *I* and accuracy (PCC (%)) for various range values (Gaussian, cell 7 to 1)

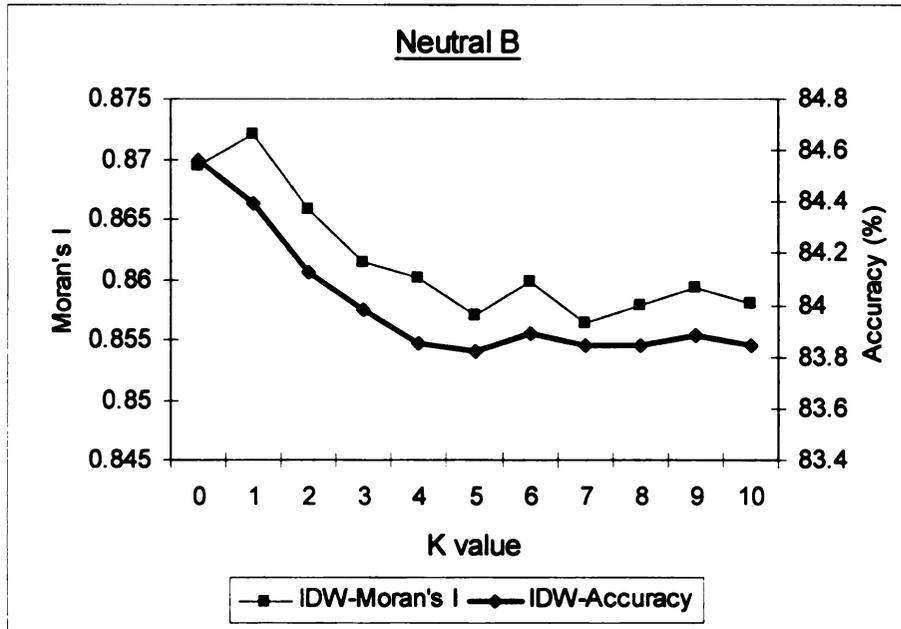
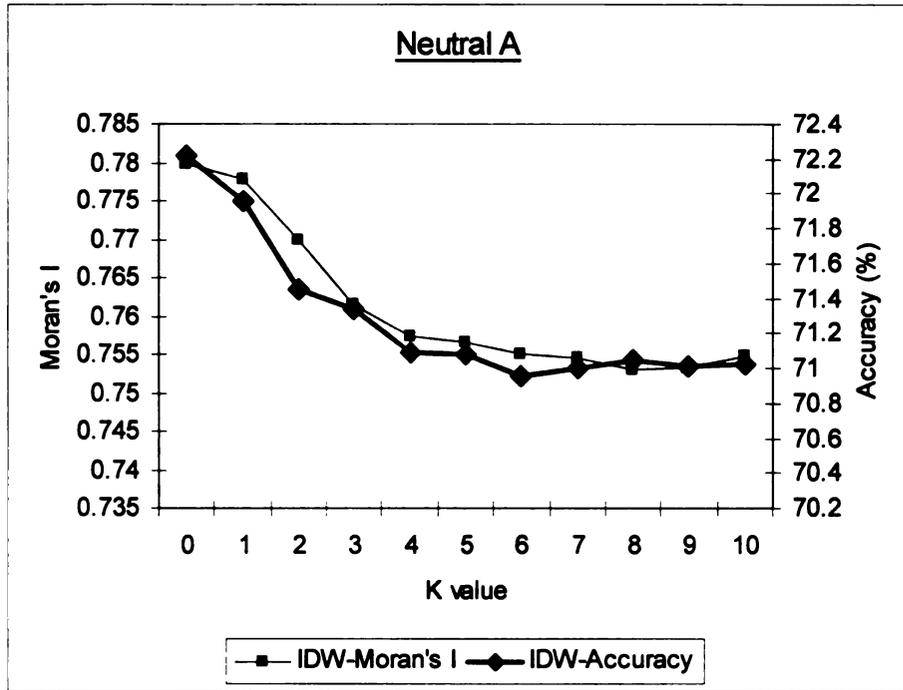


Figure 2-10. Moran's *I* and accuracy (PCC (%)) for various K-values (IDW, cell 7 to 1)

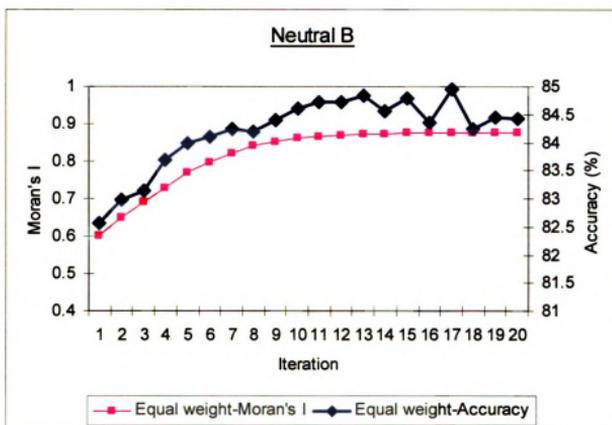
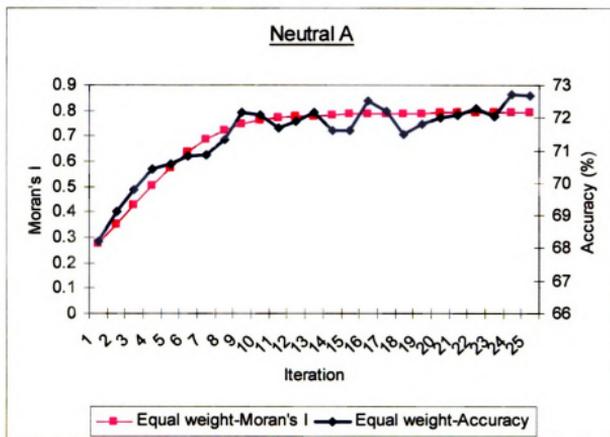


Figure 2-11. Moran's I and accuracy (PCC (%)) for various iterations (Equal weight, cell 7 to 1)

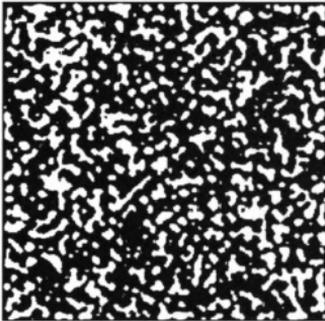


Figure 2-12. Neutral A result (Nearest Neighbors function, cell 7to1, after 20 iterations)

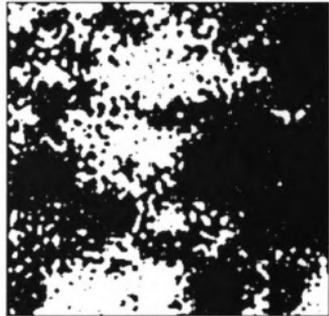


Figure 2-13. Neutral B result (Nearest Neighbors function, cell 7to1, after 20 iterations)

Table 2-1. Mean accuracy and student t-test using 20 simulated neutral images
(Mean accuracy is the average accuracy for 20 images)

		Method 1																			
		68.53																			
Mean Accuracy		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Exponential	Range value	67.9	68	68.2	68.1	68.2	68.4	68.3	68.2	68.3	68.2	68.3	68.4	68.4	68.4	68.4	68.3	68.3	68.4	68.3	68.3
	Mean Accuracy	0.00	0.00	0.02	0.00	0.02	0.05	0.04	0.00	0.02	0.01	0.07	0.17	0.11	0.31	0.23	0.04	0.04	0.10	0.05	0.05
	T-test (probability)	0.00	0.00	0.00	0.00	0.00	0.02	0.05	0.04	0.00	0.02	0.01	0.07	0.17	0.11	0.31	0.23	0.04	0.10	0.05	0.05
Gaussian	Range value	67.9	67.9	68.2	68.3	68.4	68.4	68.4	68.3	68.4	68.4	68.3	68.4	68.3	68.2	68.4	68.3	68.4	68.3	68.4	68.3
	Mean Accuracy	0.00	0.00	0.00	0.05	0.21	0.24	0.11	0.04	0.14	0.23	0.11	0.20	0.01	0.00	0.19	0.02	0.11	0.31	0.05	0.13
	T-test (probability)	0.00	0.00	0.00	0.05	0.21	0.24	0.11	0.04	0.14	0.23	0.11	0.20	0.01	0.00	0.19	0.02	0.11	0.31	0.05	0.13
IDW	Range value	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	2	3	4	5	6	7	8	9	10
	Mean Accuracy	68.3	68.3	68.2	68.4	68.3	68.4	68.3	68.3	68.2	68.3	68.2	68.1	68.1	68.1	67.9	67.8	68	68	67.9	67.8
	T-test (probability)	0.16	0.01	0.01	0.24	0.10	0.18	0.01	0.02	0.00	0.06	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Method 2																			
		83.57																			
Mean Accuracy		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Exponential	Range value	83.0	83.2	83.3	83.4	83.5	83.5	83.6	83.6	83.6	83.6	83.6	83.6	83.6	83.6	83.6	83.7	83.7	83.6	83.6	83.6
	Mean Accuracy	0.00	0.00	0.00	0.01	0.19	0.12	0.50	0.97	0.75	0.40	0.26	0.36	0.49	0.44	0.35	0.05	0.05	0.52	0.94	0.50
	T-test (probability)	0.00	0.00	0.00	0.01	0.19	0.12	0.50	0.97	0.75	0.40	0.26	0.36	0.49	0.44	0.35	0.05	0.05	0.52	0.94	0.50
Gaussian	Range value	82.9	83.0	83.3	83.5	83.6	83.6	83.6	83.6	83.6	83.6	83.7	83.7	83.6	83.6	83.7	83.6	83.7	83.5	83.5	83.6
	Mean Accuracy	0.00	0.00	0.00	0.16	0.98	0.73	0.85	0.39	0.26	0.37	0.14	0.12	0.57	0.66	0.14	0.87	0.08	0.35	0.15	0.73
	T-test (probability)	0.00	0.00	0.00	0.16	0.98	0.73	0.85	0.39	0.26	0.37	0.14	0.12	0.57	0.66	0.14	0.87	0.08	0.35	0.15	0.73
IDW	Range value	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	2	3	4	5	6	7	8	9	10
	Mean Accuracy	83.7	83.6	83.6	83.5	83.6	83.5	83.6	83.5	83.6	83.5	83.5	83.3	83.1	83.0	83.1	83.0	83.0	82.9	82.9	83.0
	T-test (probability)	0.22	0.70	0.25	0.61	0.29	0.50	0.98	0.24	0.88	0.05	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

2.5 Discussions

This study investigates the pixel-swapping optimization algorithm for modeling sub-pixel land cover distribution. I examine the effect of spatial resolution on Moran's I and find that the relationship is highly image dependent: different underlying process models may give rise to images with identical Moran's I values but with very different spatial scaling properties. Two limitations of this method, the arbitrary spatial range value and the arbitrary exponential model of spatial autocorrelation, are assessed. Various weighting functions, as alternatives to the exponential model, are evaluated in order to derive an optimum weighting function. In all tested models, Gaussian, Exponential, and IDW, the pixel swapping method improves classification accuracy compared with the initial random allocation of sub-pixels. However the results suggest that the Nearest Neighbor function could be used to maximize accuracy and Moran's I value instead of using more complex models of spatial structure.

One limitation of the pixel-swapping method is that the algorithm works best for highly contiguous landscapes like these in this study, since the algorithm attempts to maximize the spatial autocorrelation. However, not all landscapes are highly contiguous. Therefore, it is necessary to investigate methods to incorporate the degree of autocorrelation of the landscape (if such information is available) without maximizing the autocorrelation.

Several potential avenues for further research present themselves. In this study, Moran's I is used as an index of the spatial contiguity of the landscape, and Percent

Correctly Classified (PCC) is used as a classification accuracy assessment. There are alternative ways to characterize landscape. Various landscape indices, such as mean patch size, number of patches, and total edge length, can be used. These indices are employed to quantify landscape structure in terms of landscape configuration and landscape composition (Haines-Young and Chopping, 1996). The effect of sub-pixel models on these alternative metrics is unknown; they may show markedly different behavior than the results for Moran's I identified in this work. Alternatively, the employment of different metrics may provide a much richer basis for modeling landscape pattern at the sub-pixel scale.

A second issue involves the complexity of the problem under examination. This research only considered binary class maps (e.g. forest/non-forest). Since our models of the landscape, in general, are composed of a variety of land cover types, it is vital to investigate ways to handle multiple cover classes at the sub-pixel level. Thus, in the next chapter, I will develop new algorithms that are applicable to multiple land covers for predicting land cover distribution at sub-pixel scales.

Chapter 3

Modeling Multiple Class Land Cover at Sub-pixel Scales

3.1 Introduction

In Chapter 2, the pixel-swapping optimization algorithm was examined for predicting sub-pixel land cover distribution. Various alternative weighting functions were evaluated. For all tested weighting functions (Nearest Neighbor, Gaussian, Exponential, and IDW), the pixel swapping method increased classification accuracy compared with the initial random allocation of sub-pixels. I concluded that the Nearest Neighbor function is not only simpler and computationally more tractable but also provides equivalent results as the distance-based weighting functions. One limitation of this pixel-swapping algorithm is that it is applicable to only binary class maps (e.g. forest/non-forest). Since our landscape models, in general, are composed of a variety of land cover types, it is vital to investigate ways to handle multiple cover classes at sub-pixel levels. The objective of this chapter is to develop new algorithms for modeling the spatial distribution of multiple land cover classes at sub-pixel scales. Next, I describe the various algorithms for sub-pixel mapping, and this is followed by the presentation of experimental results and discussion. The chapter concludes with some implications of the findings.

3.2 Methods

In this section, I introduce three methods for sub-pixel mapping: sequential categorical swapping, simultaneous categorical swapping, and simulated annealing. All

three methods employ the notion of attractiveness using a Nearest Neighbor function. These algorithms were implemented by the author using custom code in the IDL programming language. In this study, Moran's I is used as an index of the spatial autocorrelation for the landscape (Bailey and Gatrell 1995). Moran's I of each class is weighted based on the number of pixels of the class. If class A occupies 30 percent of the area and class B occupies 70 percent of area, the weighted Moran's I is sum of Moran's I of A multiply by 0.3 and Moran's I of B multiply by 0.7.

3.2.1 Sequential categorical swapping

Sequential categorical swapping is an extension of the binary pixel-swapping algorithm proposed by Atkinson (2001, 2005). The algorithm allocates each class in turn to maximize its internal spatial autocorrelation. The algorithm considers the landscape as a binary scheme (class 1 and 0 (the other classes)). Once the first class is allocated, the algorithm only uses the remaining cells to allocate the second class, and so on. The final class is allocated to the remaining cells, and therefore, its class proportions are not necessary as input data. Thus, the order of the input classes must be specified for this method. Two input class orders are examined in terms of Moran's I value: descending (high I to low I) and ascending (low I to high I) order.

Sequential categorical swapping benefits from a prior information about the class, such as the Moran's I value. However, in general, Moran's I at the sub-pixel level is not possible to obtain. Moreover, as detailed discussion is provided in Chapter 2, it is not feasible to estimate the Moran's I of each class at sub-pixel scales from coarse resolution imagery, since the relationship between resolution and Moran's I is unpredictable.

3.2.2 Simultaneous categorical swapping

The second method simultaneously examines all pairs of cell-class combinations within a pixel to determine the most appropriate pair of sub-pixels to swap. Initially the algorithm randomly allocates classes to all cells in each pixel based on the class proportion to sub pixels. In Figure 3-1, for example, prior soft classification probabilities indicates that 66 % of the southwestern cell is class 2, while 33% of the southeastern cell is class 2. Therefore 2/3 of the sub pixels in the southwestern cell are assigned to class 2, and 1/3 of the sub pixels in the southeastern cell are assigned to class 2. Based on this initial allocation, binary arrays for each class are created as shown in Figure 3-2.

Then, attractiveness O_i is calculated for each class (Figure 3-3). As mentioned in Chapter 2, attractiveness O_i is simply the sum of the values at the nearest neighbors. The example below uses the first-nearest neighbors. The center pixel of the northeastern cell is occupied by class 3 and there are four sub-pixels of class 3 (includes the center pixel itself) within nearest neighbors. Thus, the attractiveness of the center pixel is 4.

2	1	1	1	1	3
1	3	2	3	3	3
3	2	2	1	1	1
3	2	2	1	3	2
2	3	2	3	2	1
2	1	2	3	1	2

Figure 3-1. Initial random allocation

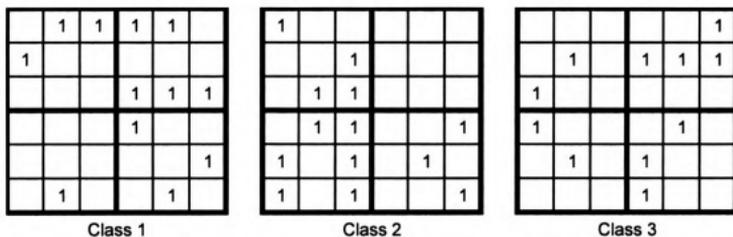


Figure 3-2. Binary matrix

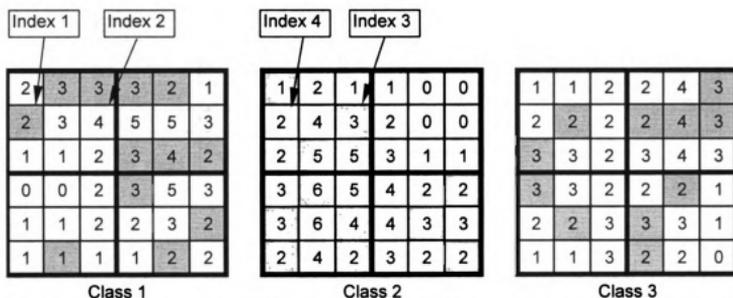


Figure 3-3. Attractiveness O_i (Grey color indicates the pixel is occupied by the class)

Based on the attractiveness O_i value, a decision rule table is created for each pixel (Table 3-1). Index 1 is the minimum O_i value occupied by the class (class a , location x) (Figure 3-3). Index 2 is the maximum O_i value occupied by the other class (class b , location y). Index 3 is the O_i value at location y for class b . Index 4 is the O_i value at location x for class b . The index 5 and 6 are calculated using the results from index 1 to 4. The index 5 (index 2-1) indicates how much class a is more attracted to location y than current location x . The index 6 (index 4-3) also indicates how much class b is more attracted to location x than current location y . The index 7 is the sum of index 5 and 6.

Thus, the larger the value of index 7, the greater the swapping attractiveness of this cell pair. Subsequently, a row that shows maximum value at index 7 is selected. One pair of sub-pixels (class *a* at location *Y* and class *b* at location *Y*) is swapped. This swapping aims to increase the degree of contiguity for both classes. This procedure is repeated for all pixels. Thus, this simultaneous method does not require any prior information about relative class contiguity.

Table 3-1. Decision Rule Table

INDEX->	1	2	3	4	5	6	7
	oi oa_min	oi ua_max	oi ob	oi ub	2-1	4-3	5+6
class 1	2	4	3	2	2	-1	1
class 2	1	4	2	1	3	-1	2
class 3	2	3	5	4	1	-1	0
:							
num_class							

Max

3.2.3 Simulated Annealing

Simulated annealing is a family of optimization algorithms based on the principle of stochastic relaxation. An initial image is gradually perturbed via pixel swapping so as to match constraints such as reproduction of a target histogram and covariance (Goovaerts, 1997). Kasetkasem *et al.* (2005) employs the Simulated Annealing as the optimization algorithm as a part of Markov random field model based approach to generate super-resolution land cover map. However, Simulated Annealing has not been commonly applied to super-resolution mapping techniques. There are different criteria which can be used to decide whether a given perturbation is accepted or rejected during the optimization process. In this study, the Maximum *a Posteriori* (MAP) model is used (Goovaerts, 1997), and Moran's *I* is employed as the objective.

This MAP model only accepts swaps that increase the local Moran's I value.

$$\text{Prob \{Accept } i\text{th swapping}\} = \begin{cases} 1 & \text{if Moran's } I(i) \geq \text{Moran's } I(i-1) \\ 0 & \text{otherwise} \end{cases}$$

The computation time will be exceedingly long if we calculate Moran's I value for each trial, especially for a large study site. Therefore, Moran's I will be recalculated only after a certain number of iterations for decision criteria. The basic steps involved in the algorithm are given below:

1. Let $I_{\text{objective}}$ be the target Moran's I value and N_{max} be the maximum allowable number of iterations
2. Randomly allocate sub-pixels within a pixel based on the class proportions
3. Calculate Moran's I for the current image (I_{current})
4. Repeat the following steps:
 - while ($I_{\text{current}} < I_{\text{objective}}$) AND (number of iterations $< N_{\text{max}}$)
 - i. Calculate attractiveness O_i value for each class (oi_array)
 - ii. Repeat the following steps a and b for (1% of N_{max}) times
 - a. Randomly pick two sub-pixels within a pixel
 - b. Swap or un-swap based on the O_i value
 - iii. Calculate Moran's I value for the current image (I_{current})(go back to Step 4)

At step 4-ii-b, sub-pixel X is currently occupied by class A and sub-pixel Y is occupied by class B. The O_i value at sub-pixel X for class A and the O_i value at sub-pixel Y for class B is added (Non_swap_ O_i). The O_i value at sub-pixel X for class B and the O_i value at sub-pixel Y for class A is added (Swap_ O_i). If Swap_ O_i is larger than Non_swap_ O_i , class A and class B are swapped. One advantage of this algorithm over

the other simultaneous and sequential methods is that users can specify the target Moran's I , which means the algorithm does not always maximize the contiguity of the landscape. However, as previously discussed, knowledge of Moran's I at sub-pixel scales is generally not possible to obtain. Therefore, target I can be specified only if we have a prior information of the landscape. In this study, the target Moran's I is set to 1.0 to maximize the autocorrelation so that results match the other methods.

One distinction of the simulated annealing algorithm relates to the fact that the algorithm randomly selects two sites. In contrast, the sequential and simultaneous algorithm deterministically selects two sites most in need of swapping based on the attractiveness index O_i . Consequently, these two methods are relatively fast since convergence occurs in far fewer iterations. Processing time will be an important aspect of consideration since remotely sensed imagery are generally very large data sets. Thus, three approaches, sequential categorical swapping, simultaneous categorical swapping, and simulated annealing, are assessed in terms of their accuracy performance, and their processing time.

3.3 Case study

A Landsat ETM+ image of East Lansing MI, USA (path21, row30) with a spatial resolution of 30m acquired on 6 June, 2000, is used to evaluate the various algorithms. A sub-image of 490 pixels by 490 pixels (14.7 km by 14.7 km) is extracted from the original image, representing a variety of landscapes from urban to agricultural land. This heterogeneity is ideal to test the "pixel-swapping" algorithm, since it might contain many

mixed pixels. It is more likely to include more than one land cover type within a pixel for a heterogeneous landscape than for a homogeneous landscape. b The sub-image is classified into five classes by unsupervised classification using ERDAS Imagine 8.6 (ERDAS, 2002): Urban and built-up, residential, vegetation, water, and bare soil. Image classification techniques, such as supervised, unsupervised, and hybrid classification, aim to automatically categorize all pixels in an image into land cover classes or themes based on their data file values (Lillesand et al., 2004). These purely spectrally based procedures completely ignore the spatial pattern of the image and often result in a salt-and-pepper appearance due to the inherent spectral variability encountered by a classifier when applied on a pixel-by pixel basis. Post classification smoothing is generally applied to “smooth” the classified output (Lillesand et al., 2004). Mode (majority) filter is one method of classification smoothing. For example, within a sub-pixel classification studies, Atkinson (2005) applied a 7 pixel by 7 pixel mode filter to the classified image, and Verhoeve and DeWulf (2002) applied a mode filter to resulting super-resolution maps to eliminate linear artifacts. As mentioned before, all three techniques examined in this study should only be applicable as long as the basic assumptions about spatial dependence are fulfilled. However, the degree of spatial dependency will be considerably affected by the use of mode filters. Therefore, in addition to the non-filtered original image, various mode filters, 3x3, 5x5 and 7x7, are applied to the original classified image. These images are referred to subsequently as Non-filter, Mode 3, Mode 5, and Mode 7. Figure 3-4 shows the image resulting from the 5x5 mode filter (Mode 5). Urban cells are mainly located in the center of the study site and agricultural (vegetation) area and bare soils are located on the surrounding areas. Residential areas are distributed through the

entire area. The water features are mainly rivers and run through the center. These filtered products and the original image serve as the reference data for the case study.

The image is then spatially degraded by a factor of 7 to a cell resolution of 210 m by 210m, using the AGGREGATE function in ARCGRID of ARC/INFO 9.1 (ESRI, 2005). This function generates a reduced-resolution version of a raster data where each output cell contains the sum of the input cells that are encompassed by the extent of the output cell. This procedure is performed for each class. Therefore, there are five coarser resolution images (Figure 3-5). These pixel-level proportions can be considered the output of soft classification. These class wise fraction images are not easy to interpolate into one classified image, and also they do not provide any indication as to how the classes are spatially distributed within the pixel. Figure 3-6 illustrates a small portion of the study region. The degraded image does not retain its original form of vegetation class, and it is hard to visualize the original shape from the coarse image. The filtered and non-filtered images of 490 pixels by 490 pixels are degraded to 70 pixels by 70 pixels. Thus, four fine images are used as 'reference data' to test the various algorithms.

In order to contrast sub-pixel level classification images and traditional hard classification images, possible results of hard classification are generated using reference data. The image of hard classification is created by using the BLOCKMAJORITY function in ARCGRID of ARC/INFO. This block function is used to control resampling a grid from a finer resolution to a coarser one. In this study, the function finds the majority value (the value that appears most often) for the 7x7 rectangle neighbor blocks, and cell values within a block are changed to the majority value.

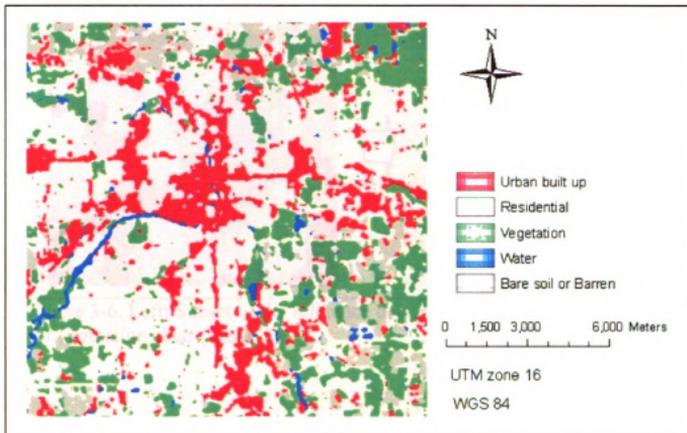


Figure 3-4. Classified image (Mode 5)

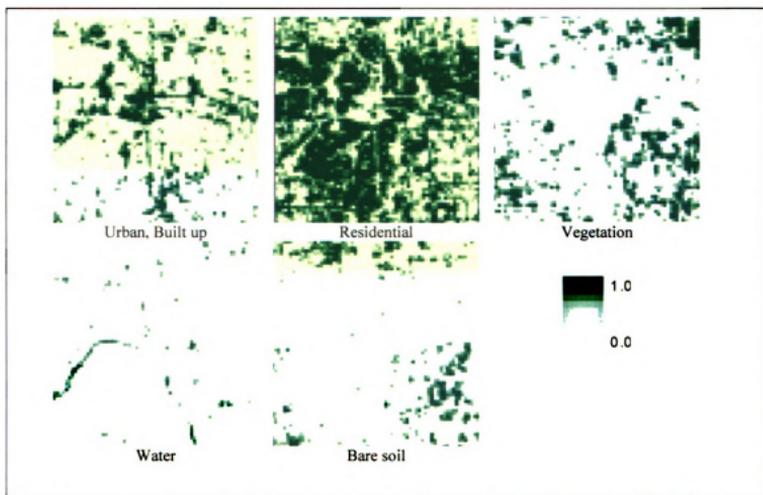


Figure 3-5. Input images for five classes (Mode 5)

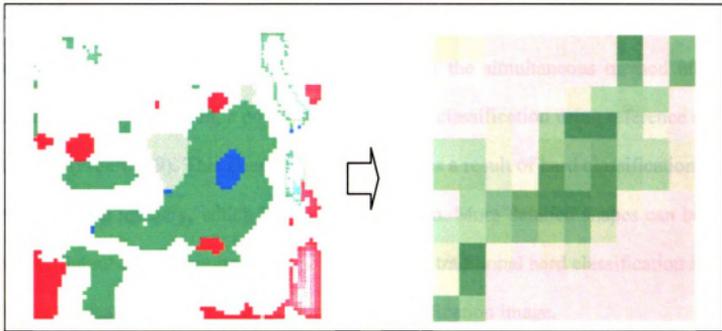


Figure 3-6. Left: Subset of the reference data, Right: Subset of degraded image for vegetation class (Mode 5)

For all three methods, attractiveness O_i is calculated using the equally weighted Nearest Neighbor function. The number of nearest neighbors involved for the computation is 48, which means all first- to third-order neighbors are incorporated. Percent Correctly Classified (PCC) is used to measure classification accuracy in this study. PCC (or overall accuracy) is calculated by the ratio of the sum of correctly classified sub-pixels in all classes to the sum of the total number of sub-pixels (Congalton and Green, 1999). Reported PCC is the average of 20 trials for Simultaneous and Sequential method.

3.4 Results and Discussion

Figures 3-7~3-9 visually demonstrate the effectiveness of sub-pixel level mapping in case of the 5x5 mode filter. Figure 3-7 shows an initial allocation of sub-pixels. Sub-

pixels are randomly allocated based on a class proportions within a pixel. Spatial dependency is minimum in this stage (Moran's $I = 0.39$) and will be increased through iterative cell swapping. Figure 3-8 is the output of the simultaneous method after 30 iterations. As a comparison, a possible result of hard classification using reference data is displayed (Figure 3-9). This image can be regarded as a result of hard classification using remote sensing imagery, which has 210 m resolution. More detailed shapes can be seen from the output of the simultaneous method than the traditional hard classification image. Some small features are missing from the hard classification image.

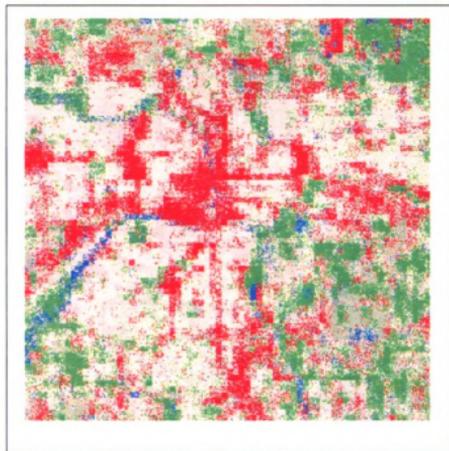


Figure 3-7. Initial random allocation (Mode 5)

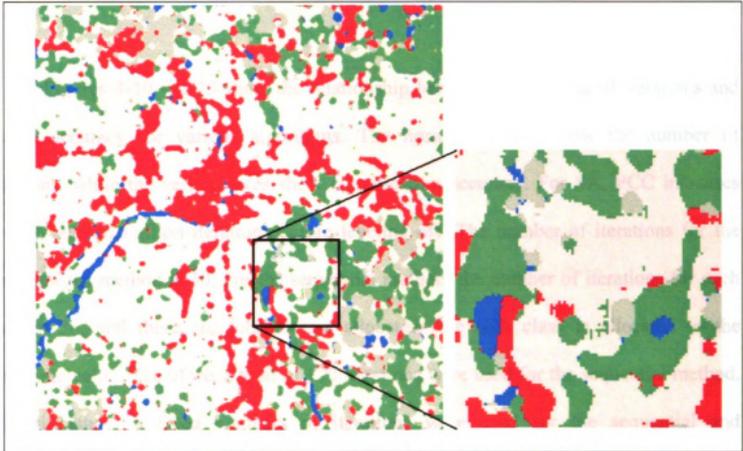


Figure 3-8. Output of Simultaneous method (after 30 iterations) (Mode 5)

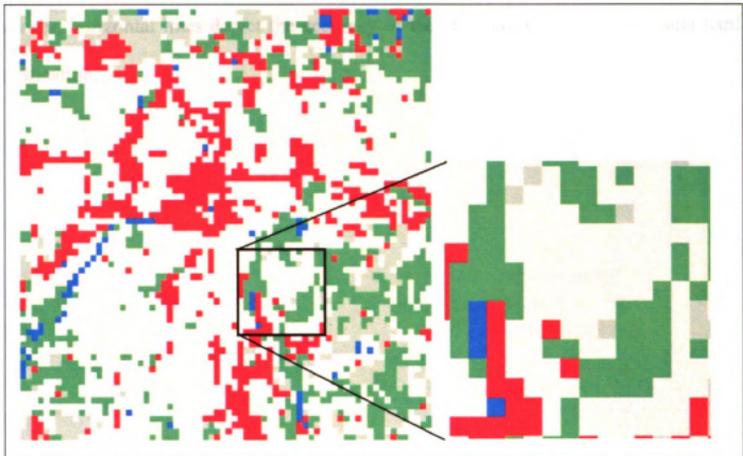


Figure 3-9. Example hard classification output (Mode 5)

Figures 3-10 ~ 3-13 show the relationship between the number of iterations and overall accuracy for various algorithms. The horizontal axes show the number of iterations, while the vertical axes show classification accuracy. For SA, PCC increases through several million iterations before leveling off. The number of iterations for the simultaneous method is 30. For the sequential method, the number of iterations for each class is 30, and there are four classes allocated (the fifth class is allocated to the remaining cells). Therefore, a total of 120 iterations were used for the sequential method. For this study, a fixed numbers of iterations were used for the sequential and simultaneous method. These numbers are empirically derived; however, it is also possible to stop the iteration when the algorithm fails to increase the accuracy. For interpretive purposes, horizontal lines depict the accuracy of the simultaneous, sequential, and hard classification.

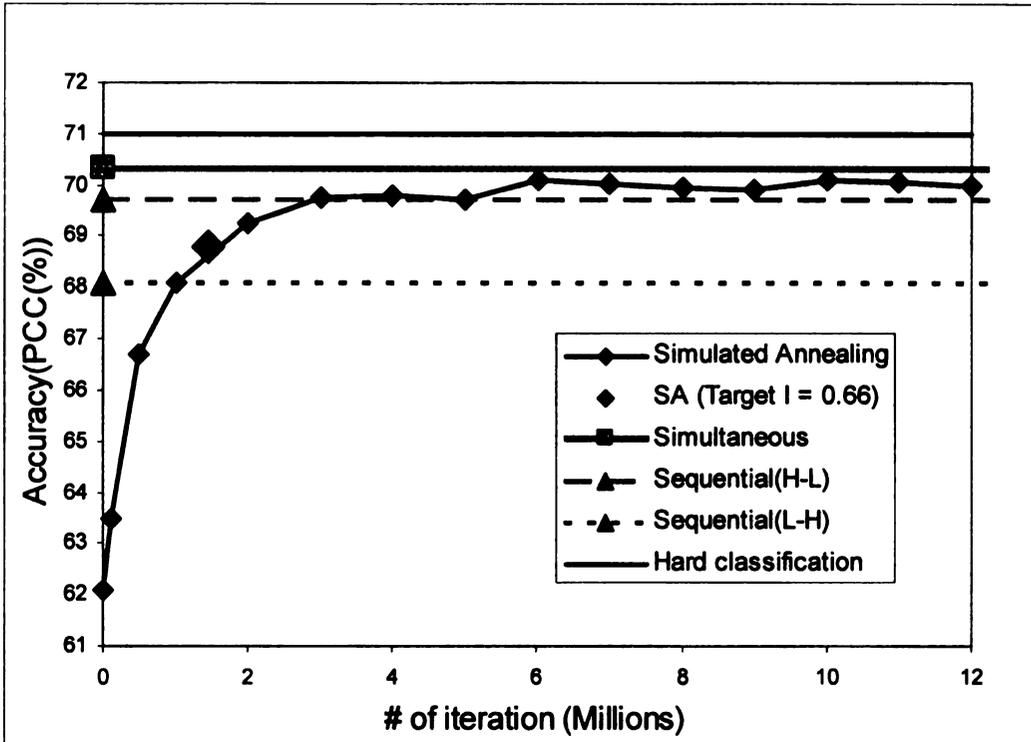


Figure 3-10. Overall accuracy of the methods (Non-filter)

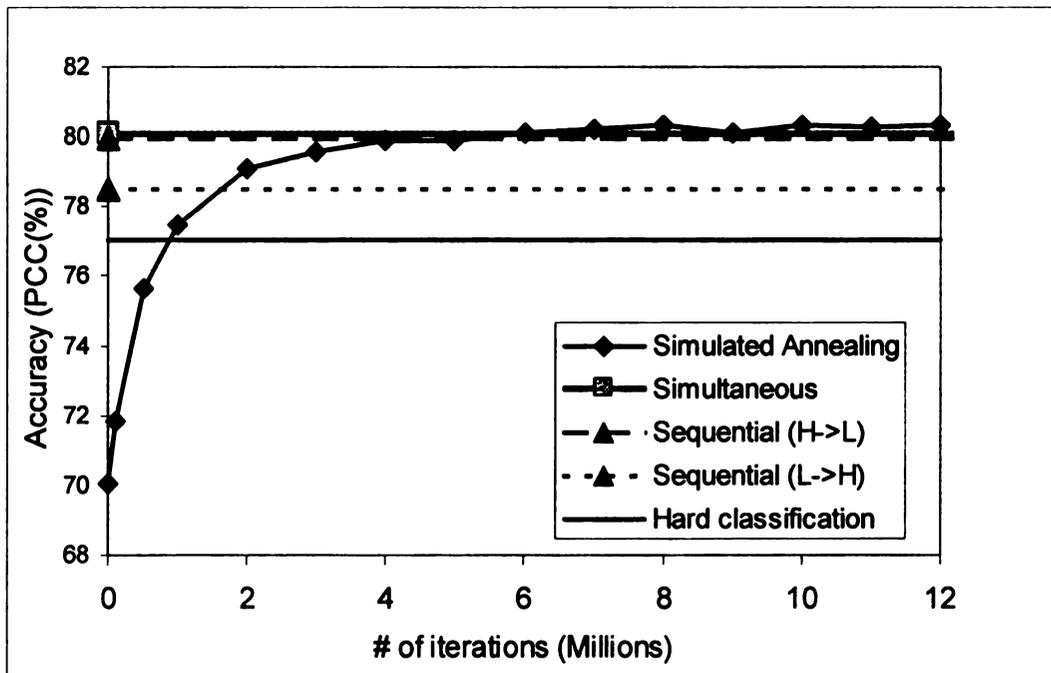


Figure 3-11. Overall accuracy of the methods (Mode 3)

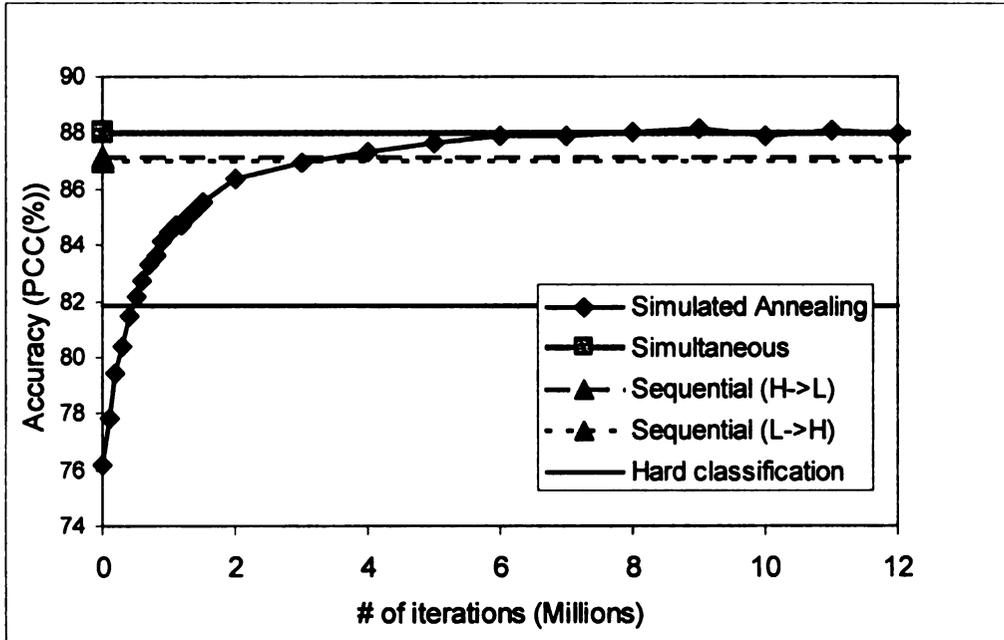


Figure 3-12. Overall accuracy of the methods (Mode 5)

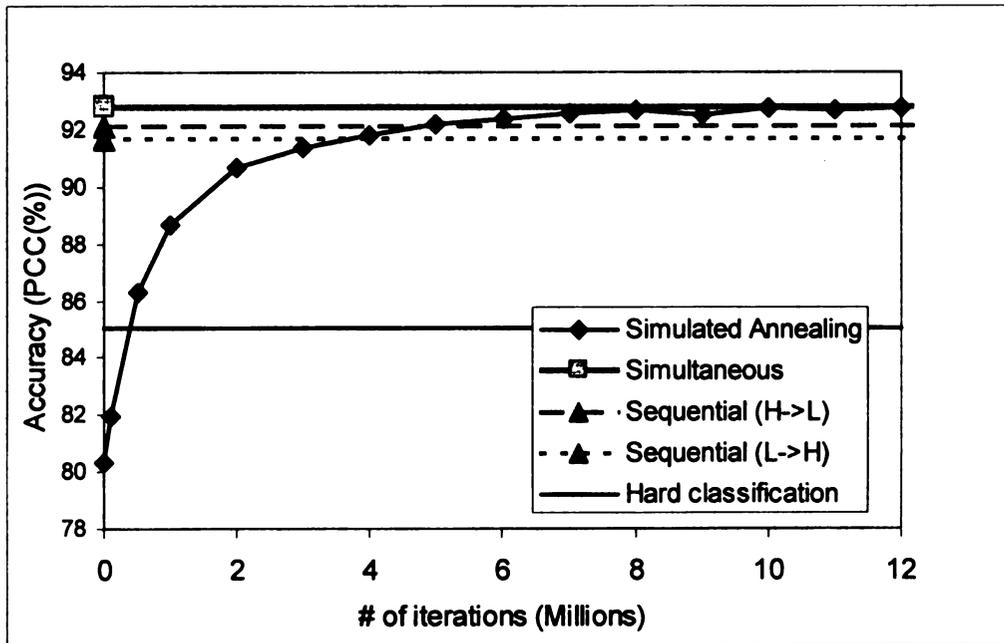


Figure 3-13. Overall accuracy of the methods (Mode 7)

The results imply that all three methods increased classification accuracy over the hard classification for all mode-filtered images. For the non-filtered image, all sub-pixel classification methods fail reach to the same accuracy as the hard classification. As discussed before, the super-resolution techniques used in this study function by increasing the spatial dependency of the image, and thus these techniques are better suited for highly autocorrelated images. Therefore, we can assume that the resulting accuracy will be higher for an image which has high Moran's I . Table 3-2 shows Moran's I value for each image and resulting accuracy using simultaneous method. As Moran's I increases, PCC and its difference between the simultaneous and hard classification also increases. For the non-filtered image, the simultaneous method failed to increase accuracy compared to the hard classification. This could be caused by lower spatial dependency of Non-filter image. Figure 3-14 shows images of the Non-filter, output of simultaneous, and hard classification. The Non-filtered image has a salt-and-pepper appearance due to the use of a non-spatial purely spectrally based classification procedure. Output of the simultaneous method shows circular features, while the hard classification captures the overall shape of the landscape. The small scattered features in the reference data are clumped to larger features in the output image. This result suggests that solitary features smaller than the pixel will deteriorate the effectiveness of the sub-pixel methods and will create circular artifacts. Tatem et al.(2001) indicates that a limitation of the HNN method is that the network will always converge to rounder corners than that of the actual field. These phenomena are inevitable as long as we attempt to maximize the degree of contiguity.

Table 3-2. Accuracy comparison for Simultaneous method

	Accuracy (PCC(%))		Morans' I for fine image
	Simultaneous	Hard classification	
Non-filter	70.33	71.00	-0.67
Mode 3	80.10	77.06	3.04
Mode 5	88.04	81.86	6.18
Mode 7	92.81	85.02	7.78

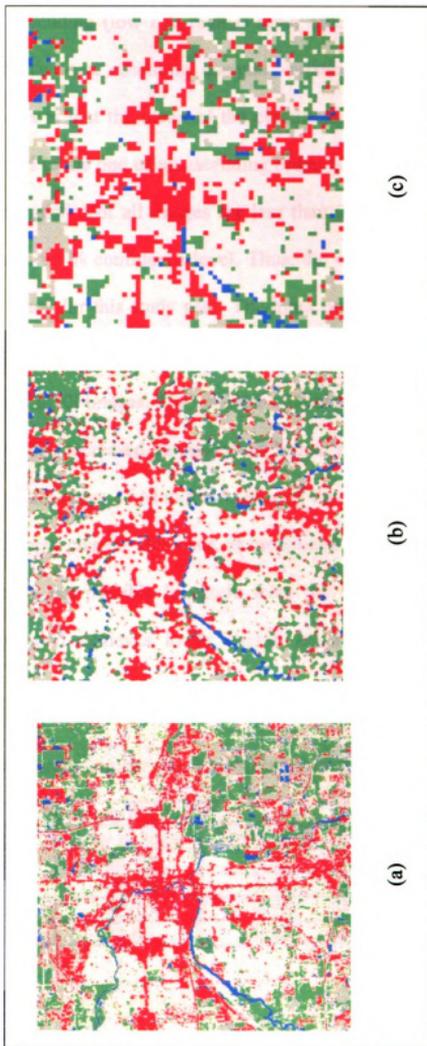


Figure 3-14. Images for Non-filter. (a) reference image (b) output for simultaneous method after 30 iterations (c) hard classification

For the sequential method, descending (high I to low I) input order shows higher classification accuracy than ascending (low I to high I) input order (Table 3-3) for all images. A two-sample student's t-test assuming unequal variance for 20 samples is performed. The null hypothesis is that the mean of the classification accuracies, using descending input order, is equal to the one using ascending order. Since the probabilities for the empirically-derived t-statistics of all images are less than 0.05, I reject the null hypothesis of equal means at a 95 % confidence level. Thus, the order of input classes affects the classification accuracy for this study area. The input order should be started from the class which has high Moran' I to one which has low Moran's I . Table 3-4 lists Moran's I values for each class in all images. The variance of Moran's I decreases as the size of mode filter increases. The increased variability of Moran's I also increases the effect of input order. Thus, if the range of Moran's I value of the classes is large, care must be taken in the order of input classes. However, as mentioned before, the degree of contiguity at sub-pixel level is not possible to obtain from the original coarse map. Moreover, the sequential method achieves lower overall accuracy than any other methods.

Table 3-3. Accuracy comparison for Sequential method

	Accuracy (PCC(%))			T-test probability
	High to Low	Low to High	Difference	
Non-filter	69.72	68.10	1.61	2.09E-28
Mode 3	79.93	78.49	1.43	6.88E-31
Mode 5	87.15	87.00	0.15	1.37E-05
Mode 7	92.12	91.65	0.47	7.08E-17

Table 3-4. Moran's *I* value for each class

	Non-filter	Mode 3	Mode 5	Mode 7
Urban builtup	0.689	0.814	0.858	0.896
Residential	0.631	0.797	0.859	0.892
Vegetation	0.740	0.838	0.885	0.904
Water	0.461	0.716	0.789	0.840
Bare soil	0.680	0.820	0.871	0.891
Variance	0.0116	0.0023	0.0014	0.0006

Simultaneous categorical swapping and simulated annealing show similar maximum accuracy. However, the simultaneous method needs only 30 iterations to reach the highest accuracy, while Simulated Annealing (SA) needs nearly eight million iterations to reach the same accuracy for mode filtered images. For the simultaneous (and sequential) algorithm, the algorithm visits all pixels per iteration. For SA, the algorithm visits only one pixel per iteration. There are 4,900 pixels in the study area, and therefore 30 iterations for the simultaneous method can be considered as 147,000 attempts of pixel-swapping. However, there still is a substantial difference of iterations between two methods. As mentioned before, this is due to the fact that the SA randomly selects two sites. In contrast, the simultaneous algorithm deterministically selects two sites most in need of swapping based on the attractiveness. Thus, there is no unnecessary swapping for the simultaneous method. Consequently, computation time for the SA is much longer than for the simultaneous method. For the simultaneous method, the time taken for 30 iterations (cell 7 to 1) was about 20 seconds on a Pentium 4 computer. It took approximately 10 minutes for the SA method using eight million iterations. As mentioned before, one advantage of this algorithm over the other simultaneous and sequential

methods is that the users can specify the target I instead of maximizing it. In this study, target Moran's I is set to 1.0 for SA in order to compare to the other methods which attempt to maximize contiguity. Since Non-filter image has lowest Moran's I (0.66), it would not be best to maximize Moran's I . Non-filter image is used to examine SA with target Moran's $I = 0.66$. The algorithm stopped at around 1.4 million iterations when current Moran's I value exceeded the target Moran's I value. The average PCC for 20 trials using same target Moran's I is 68.8% (Figure 3- 10), which is still less than the maximum PCC for SA (70%). Thus, although the output of SA has nearly same Moran's I value as the reference data, the accuracy is still low. Output of SA fails to generate the similar spatial distribution as the reference data. This could be caused by the limitation of Moran's I that very different spatial scaling properties can have identical Moran's I values, discussed in Chapter 2.

3.5 Conclusions

Sub-pixel mapping uses the output of soft classification and transforms it into a hard classification at the sub-pixel scale. The results are easier to interpret and more accurate without using any extra data. A key challenge is to identify a plausible spatial distribution of classes within a pixel. Several alternative algorithms have been proposed for allocating classes of sub-pixels. However, many techniques such as Hopfield neural network (HNN), Genetic Algorithms (GA), Markov random field (MRF) model based approach are not easily accessible to the remote sensing practitioners. The algorithm presented here can be coded easily in any scientific computing language, and can be used for modeling the spatial distribution of multiple land cover classes. In this study, three

methods were examined: sequential categorical swapping, simultaneous categorical swapping, and simulated annealing. Method 1 is a modification of a binary pixel-swapping algorithm introduced by Atkinson (2001, 2005) and explored by Makido and Shortridge (2007). The algorithm allocates each class in turn to maximize its internal spatial autocorrelation. Method 2 simultaneously examines all pairs of cell-class combinations within a pixel to determine the most appropriate pairs of sub-pixels to swap. Method 3 employs simulated annealing to swap cells. Simulated annealing is a family of optimization algorithms based on the principle of stochastic relaxation. While convergence is relatively slow, the method offers increased flexibility. It allows users to specify the target Moran's I , which is used as an index of the spatial contiguity of the landscape. Thus, this chapter investigated various methods for modeling the spatial distribution of multiple land cover classes at sub-pixel scales.

The Landsat ETM+ image of East Lansing MI (42.72N / 84.47W) with a spatial resolution of 30m is used to evaluate the various algorithms. A sub-image of 490 pixels by 490 pixels is extracted from the original images. The sub-image is classified into 5 classes using unsupervised classification: Urban and built-up, residential, vegetation, water, and bare soil. The Moran's I for each class is 0.69, 0.63, 0.74, 0.46, and 0.68, respectively. This indicates positive but different amounts of autocorrelation for all classes. All three methods should be applicable as long as the basic assumptions about spatial dependence are fulfilled. However, the degree of spatial dependency will be considerably affected by the use of post classification smoothing used to reduce the salt-and-pepper appearance. Various mode filters: non-filtered, 3x3, 5x5 and 7x7, are applied to the classified image. This study examines how the degree of contiguity of the

landscape affects sub-pixel mapping. The results imply that all three methods increased classification accuracy over the hard classification for all mode-filtered images. For the non-filtered image, however, none of the sub-pixel classification methods reach the same accuracy as hard classification. All results suggest that, as the spatial dependency of the landscape increases, the performance of the three techniques improves. For the sequential method, results indicated the input order affected classification accuracy. In any case, the observed PCC for the sequential method is not as high as that for the other methods. Unlike the sequential method, the simultaneous method and simulated annealing do not require an ordering of the input classes. Although both simultaneous and simulated annealing result in similar accuracy, the number of iterations to reach the maximum accuracy are notably different: 30 iterations for simultaneous method and 8 million iterations for SA. Therefore, for this study area, the simultaneous method can be considered as the optimum method in terms of accuracy performance and computation time.

In this study Moran's I is used as an index of the spatial contiguity of the landscape. An advantage of SA is that it allows the user to specify the target Moran's I value instead of maximizing it. Although the output image has a similar I value to the reference image, specifying target I for SA does not improve the classification accuracy. One possible reason is that the target I is a weighted Moran's I based on the number of pixels of the class. It completely ignores the difference of spatial contiguity between classes. Specifying target I for each class would increase the classification accuracy. Another possible reason is that Moran's I is not sufficient to capture the spatial characteristics of the landscape. As discussed in Chapter 2, Moran's I has the limitation

that very different spatial scaling properties can have identical Moran's I values. In addition to Moran's I , alternative ways to characterize landscape should be tested. Various landscape indices, such as mean patch size and patch size standard deviation, can be used. These indices are used as quantitative measures of spatial pattern in heterogeneous landscapes (Cardille and Turner, 2002). Several researchers have employed the variogram to capture more complex spatial patterns of the landscape. Variograms are used as constraints in sub-pixel mapping algorithms, such as Hopfield Neural Network (Tatem et al, 2002), linear optimization techniques (Verhoeve and DeWulf, 2002), and sequential indicator simulation (Boucher and Kyriakidis, 2006). The employment of landscape indices or variogram models may provide markedly different behavior than the results for Moran's I identified in this work. However, as Boucher and Kyriakidis (2006) point out, per-pixel classification accuracy assessment fails to reveal important aspects of spatial pattern. This means two output maps, which have same classification accuracy, could have strikingly different spatial patterns. Therefore, it depends on the object of study whether to maximize classification accuracy or to capture the spatial structure.

The simultaneous categorical swapping is superior to the other methods in terms of accuracy performance and efficiency. This innovative method can efficiently maximize spatial contiguity without any additional data. More research is necessary to test the technique for various landscapes. The techniques presented here should be applicable to imagery from any remote sensing system as long as the basic assumptions about spatial dependence are fulfilled, and the approach also has potential application in many areas of GIS research where data are spatially aggregated.

Chapter 4

Application to Real Landscapes

4.1 Introduction

Chapter 3 described the development of new algorithms that are applicable to multiple land covers for predicting the land cover distribution at sub-pixel scales. Three methods were examined: sequential categorical swapping, simultaneous categorical swapping, and simulated annealing (SA). The results suggested that the simultaneous method could be considered as the optimum method in terms of accuracy performance and computation time. However, the degree of contiguity of the landscape significantly affects the prediction ability of sub-pixel mapping. Additional research is necessary to test the technique for various landscapes. Thus, Chapter 4 focuses on the application of the simultaneous method to land cover derived from satellite imagery in Brazil and China. These study sites are quite different in terms of climate zone; one is tropical and the other is temperate. Both areas possess varied topography and are mainly covered by vegetations. These areas are used to examine how the characteristics of the landscape affect the ability of the optimum technique. Three types of measurement, Moran's I , mean patch size (MPS), and patch size standard deviation (STDEV), are used to characterize the landscape.

4.2 Study area

4.2.1 Tropical Forests in state, Brazil

A Landsat ETM+ imagery (path226, row68) with a spatial resolution of 30m acquired on 18 June 2000 of State of Mato Grosso, Brazil is used to evaluate the algorithm (Figure 4-1). This area is a major logging center in the Amazon and therefore, the land covers has been significantly changed (Matricardi et al., 2005). There are natural and selectively logged forests as well as clear-cuts. In dense natural forests, tree canopies are the only detectable component. In selectively logged areas, both trees and bare soil are observed (Wang, 2003). The image was taken during dry season from June through September.

The Landsat imagery is classified using an unsupervised image classification model into two major categories (forest and non-forest). Non-forest areas are subsequently masked out of the image. A semi-automated textural algorithm is applied to the forest areas in order to distinguish dense undisturbed forest and logged forest. Finally, the image is classified into six classes: logged forest, undisturbed forest, deforestation, burned forest, old logged forest, and water body (Figure 4-2). A detailed description of the data production process is provided by Marticardi et al. (2005).

A sub-image of 4,725 pixels by 4,725 pixels (141.75km x 141.75km) was extracted from the classified image (Figure 4-2). This sub image was spatially degraded by a factor of 7 to a spatial resolution of 210 m by 210 m. At the coarser spatial resolution of 210 m by 210 m, the contribution of each sub-pixel is summed to obtain a pixel-level proportion for each class. These pixel proportions then formed the input to the

sub-pixel mapping algorithm. One of the advantages of the above approach is the ability to evaluate the sub-pixel mapping exhaustively, since the ground truth is known. The other advantage is the ability to focus on the mapping algorithm rather than the soft classification that would predict the class proportions.

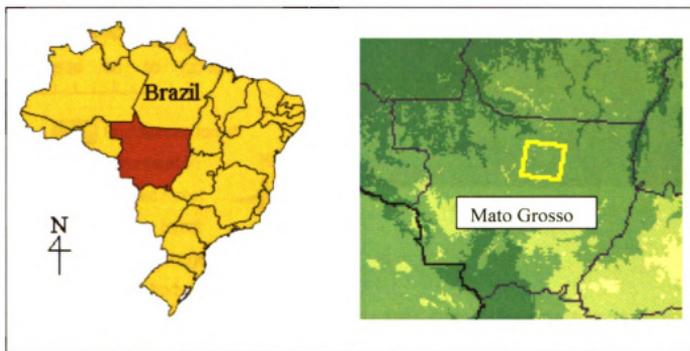


Figure 4-1. Study site location (Brazil: Landsat scene, path 226 and row 068)

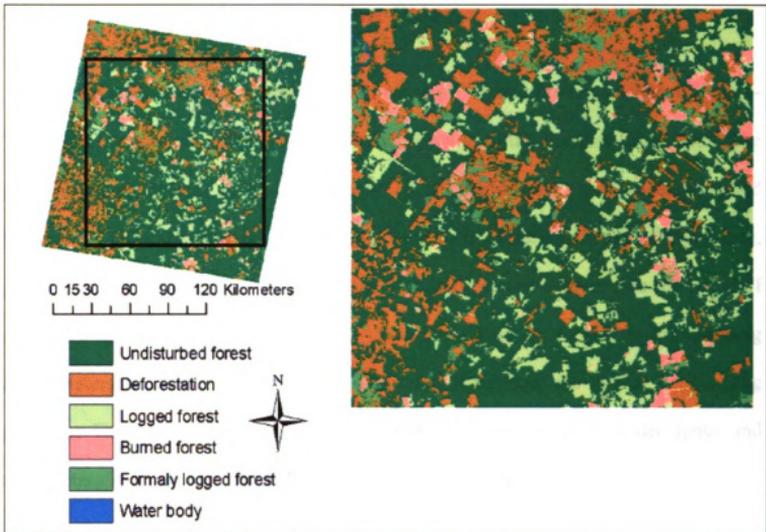


Figure 4-2. Left: Brazil, classified Landsat image, Right: Brazil sub image

4.2.2 Temperate land mosaic in East China

Two study areas are located in south-western Zhejiang Province, China (Figure 4-3). Zhejiang is located in China's southeast coast, south of the Yangtze River Delta. The province covers a total continental area of 101,800 square km, and possesses varied topography. Hills and mountains account for 71 percent of the total area in the province, plains and basins make up 23 percent, while the rest 6 percent is water area composed of rivers and lakes (Zhejiang China, 2003). Both study sites are located in the Qiantang River watershed, where the river passes through Hangzhou; the capital of Zhejiang province. The main land cover types are forests, which include both deciduous and coniferous stands, and agricultural land, mainly paddy fields.

The classified vector data for 1978 is derived from Landsat MSS imagery with a corresponding Digital Elevation Model (DEM) data. A detailed description of the data production process is provided in Zhang et al. (2006). A sub-image of 123.2 km x 117.6 km is extracted from the vector data and converted to raster data with a pixel size of 80 m. The image, China 1978, contains 12 land cover classes: paddy rice, upland crop, forest land, sparse forest, urban, rural habitat, river, lake, reservoir, pond, swamp, and other unused land (Figure 4-4). The other classified imagery is derived from Landsat ETM+ imagery acquired in 2004. The size of the image is 46.2 km x 44.1 km with a pixel size of 30 m. It contains 7 land cover classes: class 1 ~ class 7 (Figure 4-5). This image is referred as China 2004 subsequently. Both China 1978 and 2004 images contain 1540 columns x 1470 rows with different pixel sizes. Both China images are spatially degraded by a zoom factor of 7 to a spatial resolution of 560m by 560 m for China 1976

and 210m by 210m for China 2004. At the coarser spatial resolution imagery, the contribution of each sub-pixel is summed to obtain a pixel-level proportion for each class. These pixel proportions then formed the input to the sub-pixel mapping algorithm.

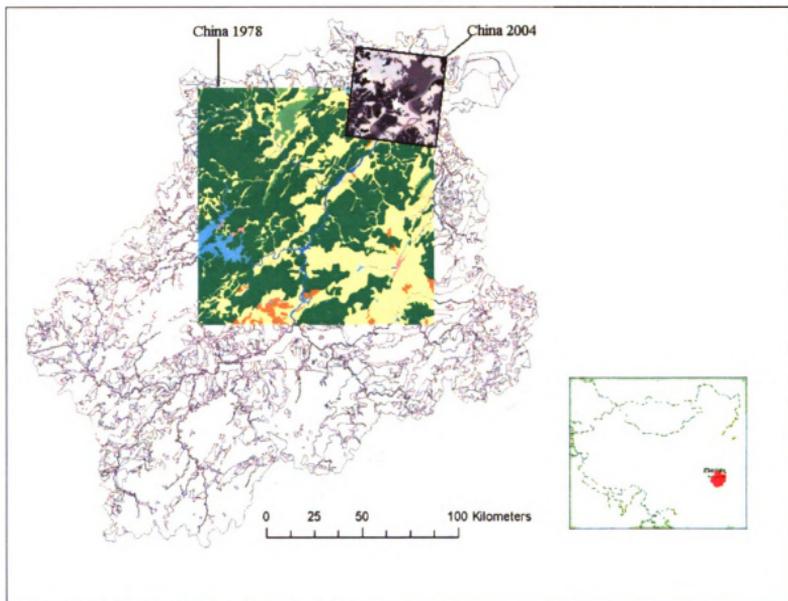


Figure 4-3. Study site location for China

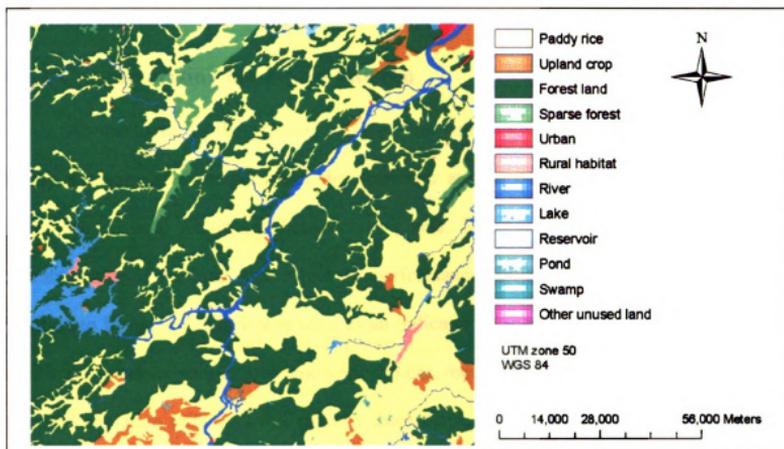


Figure 4-4. China reference image 1978

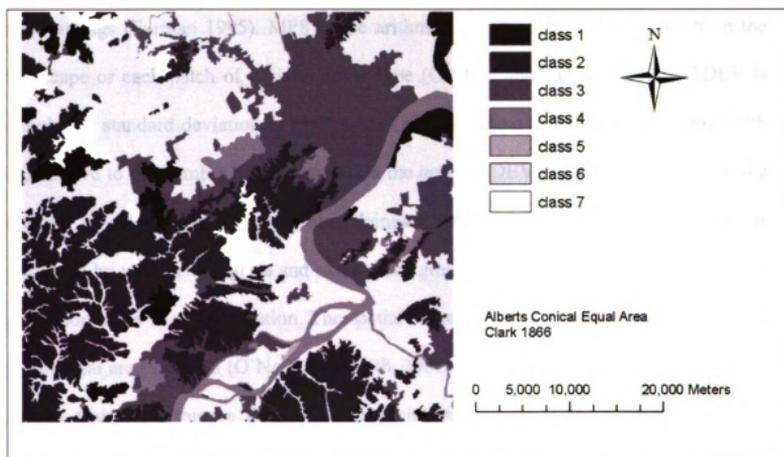


Figure 4-5. China reference image 2004

4.3 Methods

4.3.1 Characterizing landscape pattern

As discussed in the previous chapter, the simultaneous method works best for highly autocorrelated images since the algorithm attempts to maximize the degree of contiguity for both within and between the pixels. Therefore, I also examine how the characteristics of the landscape affect the performance ability of sub-pixel mapping. In addition to Moran's *I*, which was used as an indicator of spatial autocorrelation for area data in the previous section, two landscape indices, mean patch size (MPS) and patch size standard deviation (STDEV), are used to characterize the landscape. Landscape indices are quantitative measures of spatial pattern in heterogeneous landscapes (Cardille and Turner, 2002). A patch is defined as a homogeneous area that differs from its surroundings (Forman 1995). MPS is the arithmetic average size of each patch on the landscape or each patch of a given cover type (Cardille and Turner, 2002). STDEV is simply the standard deviation of patch sizes (Haines-Young and Chopping, 1996). MPS is sensitive to the number of patches within the area. STDEV is sensitive not only to the number of patches but also to scale (Haines-Young and Chopping, 1996). Scale is measured by two factors: grain and extent. The grain is determined by the finest level of resolution made in an observation. The spatial extent of an observation set is established by the total area sampled (O'Neill and Smith, 2002). Both grain and extent are important factors that can influence the results of a pattern analysis (Greenberg et al., 2002). Changing the scale of observation can change the fundamental property of an object (O'Neill and Smith, 2002). In this study, I examine various extent and grain to calculate three types of measurements: Moran's *I*, MPS, and STDEV. For Brazil, the classified

image (4,725 x 4,725 pixels) is subdivided into 15 columns x 15 rows (315 x 315 pixels). For China 1978 and 2004, the images (1,540 x 1,470 pixels) are subdivided into 15 columns x 15 rows (98 x 98 pixels). Since the sub-areas for China are considerably smaller, larger sub-areas (315x315) are also examined. In this case, there are 4 x 4 sub-areas in one image. Table 4-1 shows the various sizes of extent and grain for three images.

Table 4-1. Various extent and grains for entire and sub-areas

		Brazil	China 1978			China 2004
Entire area	Extent (pixel)	4725x4725	1540x1470			1540x1470
	Extent (km)	141.75 x141.75	123.2x117.6			46.2x44.1
	Grain (m)	30	80			30
	Total number of pixels	22,325,625	2,263,800			2,263,800
Sub area	Extent (pixel)	315x315	315x315	98x98	315x315	98x98
	Grain (m)	30	80	80	30	30
	Number of samples	60	8	60	8	60

One problem of using sub-areas as samples is that these images are not independent each other. Since sub-areas are taken from the same imagery, the possibility exists for spatial correlation between them. In order to reduce spatial dependencies due to sampling, 60 samples for 15x15 sub-areas and 8 samples for 4x4 sub-areas are systematically selected as Figure 4-6. These sampled areas are used to calculate Moran's *I*, MPS and STDEV. However, many of landscape indices have been shown to be highly correlated with one another (Riitters et al., 1995). Therefore, I also tested the correlation between these three indices. Figure 4-7 shows examples of 315x315 sub-areas from the China 1978 imagery. Although both sub-areas have similar MPS, the STDEV of the left image is nearly three times larger than the right image. Since STDEV is sensitive not only the number of patches but also the scale, STDEV can capture the variability of the

patches. I exclude samples, which are occupied by only one class for this study. In order to examine the simultaneous method, each sub-area is spatially degraded by a factor of 7 to a spatial resolution of 210 m by 210m for Brazil and China 2004, and 560 m by 560 m for China 1978. Then, the Simultaneous method (Chapter 3) is applied using a pixel-level proportion for each class. This approach is same as the one used in the Chapter 3. 20 iterations along with 48 nearest neighbors were used for the computation.

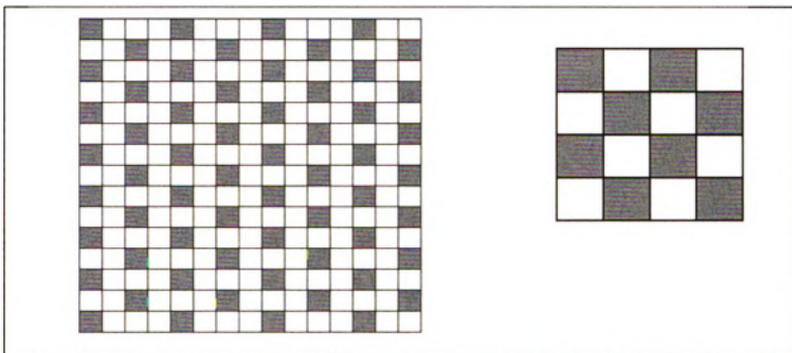


Figure 4-6. Systematic samples of sub-areas: 60 samples (Left), 8 samples (Right)

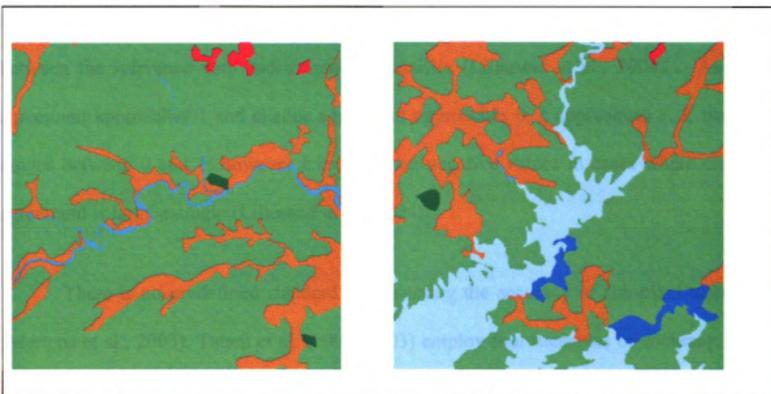


Figure 4-7. China 1978 sub-areas (315x315) (Left: STDEV=11,482, MPS = 2,177, Right: STDEV = 3,748, MPS =2,000)

4.3.2 Accuracy Assessment

Although many methods of accuracy assessment have been discussed in the remote sensing literature, the confusion matrix (sometimes called a error matrix or a contingency table) is most widely used (Foody, 2002). The matrix compares the relationship between known reference data (ground truth) and the corresponding results of a classification on a category-by-category basis (Lillesand et al., 2004). The confusion matrix can provide the basis on which to both describe classification accuracy and characterize errors (Foody, 2002). One of most popular measures that is derived from a confusion matrix is the overall accuracy or percent correctly classified (PCC) (Foody, 2002). PCC is calculated by the ratio of the sum of correctly classified sub-pixels in all classes to the sum of the total number of sub-pixels (Lillesand et al., 2004). The Kappa analysis is now a standard component of most every accuracy assessment (Congalton and Green, 1999). The Kappa statistic is a measure of difference between the actual

agreement between reference data and an automated classifier and the chance agreement between the reference data and a random classifier (Lillesand et al., 2004). As true agreement approaches 1 and chance agreement approaches 0, k approaches 1. K usually ranges between 0 and 1; however k can take on negative values in cases where change agreement is large enough (Lillesand et al., 2004).

There is no predefined standard for assessing the accuracy of sub-pixel mapping (Mertens et al., 2003). Tatem et al. (2001, 2003) employ four measures of accuracy, area error proportion, correlation coefficient, closeness and root mean square error, to assess the difference between the prediction and the validation images. Verhoeve and De Wulf (2002) created a confusion matrix and assessed the accuracy by calculating the overall accuracy and Kappa statistics. Mertens et al. (2003) employs an adjusted Kappa coefficient (k') in addition to Kappa coefficient (k). The adjusted Kappa coefficient is identical to Kappa except that it is calculated only for mixed pixels, which means it ignores the sub-pixels that have a pure pixel as parent. These sub-pixels that all belong to the same class will raise the Kappa coefficient regardless the prediction abilities of the algorithm (Mertens et al. 2003). Thus, adjusted Kappa statistics provide an indication of the ability of the algorithm to produce an accurate sub-pixel mapping while Kappa statistics evaluates the result of the mapping (Mertens et al. 2003). Figure 4-8 indicates the mixed pixels, which are used for calculate k' for Brazil imagery. For the Brazil imagery, approximately 5 million random points are used to estimate k and 1 million random points are used to estimate k' . For China 1978, 550 thousand random points are used for k and 150 thousand random points are used for k' . For China 2004, 550 thousand random points are used for k and 120 thousand random points are used for k' .

In this study, PCC, Kappa coefficient and adjusted Kappa coefficient are used to evaluate the super-resolution technique for each entire study area. For each sub-area, the improvement in PCC is calculated as:

$$\frac{(\text{PCC from simultaneous method} - \text{PCC from random allocation})}{\text{PCC from random allocation}} \quad (8)$$

Thus, in addition to PCC, the improvement from the random allocation is also calculated for sub-areas.

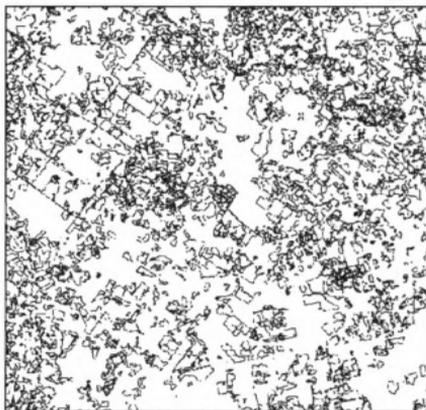


Figure 4-8. Mixed pixels used for adjusted Kappa statistics for Brazil

4.4 Results

4.4.1 Brazil

The following results describe the case where every pixel consists of 49 sub-pixels. The output of the simultaneous algorithm was visually (figure 4-9, 4-10) more accurately capture the shape of the boundaries than the hard classification. Figure 4-9 shows the output of the simultaneous method after 20 iterations. Figure 4-10 displays subsets of figure 4-9 by comparing to the reference data, simultaneous method, and hard classification. The algorithm successfully characterizes the curved features (river) and the circled features (patios). Notice that the diameters of logged forest around patios are about 500m and the width of the river is 400m. Thus, these features are larger than the one pixel size (210m) of coarse resolution imagery.

Figure 4-11 shows the relationship between the number of iterations and the classification accuracy. The result implies that the algorithm successfully increases the classification accuracy. The maximum accuracy (98.59%) of the simultaneous method is higher than that of the hard classification (95.53%). The optimum number of iterations for this data set would be 20, since the accuracy does not increase after that. For the Brazil image, the time taken for 20 iterations (cell 7 to 1) was approximately 20 minutes on a P4 computer.

Table 4-2 shows the confusion matrix and Table 4-3 shows the accuracy of degraded real imagery per class. Both user's and producer's accuracy of water is the lowest among the classes, and water class is mainly confused with deforestation class.

Table 4-4 shows the results of Kappa (k) and Adjusted Kappa statistics (k'). Although k' is lower than k , both values are higher than 0.9. Figure 4-12 demonstrated the relationship between PCC and Moran's I , MPS, STDEV for 60 sub-areas. PCC are positively related to all three measurements of landscape patterns. Table 4-5 shows the correlation coefficient. PCC is most correlated to Moran's I .

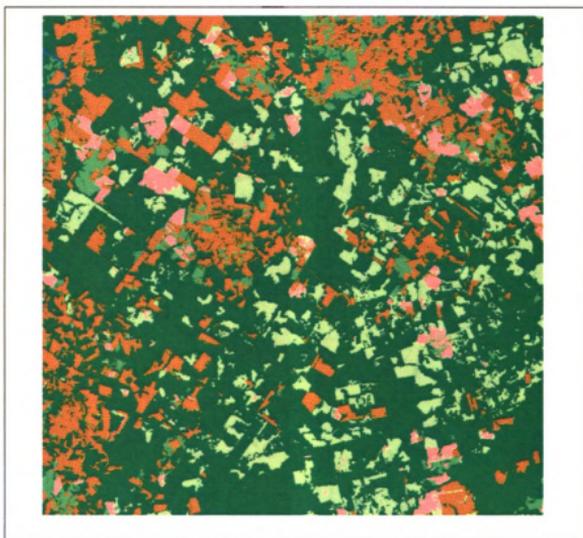


Figure 4-9. Output of simultaneous method (after 20 iterations)

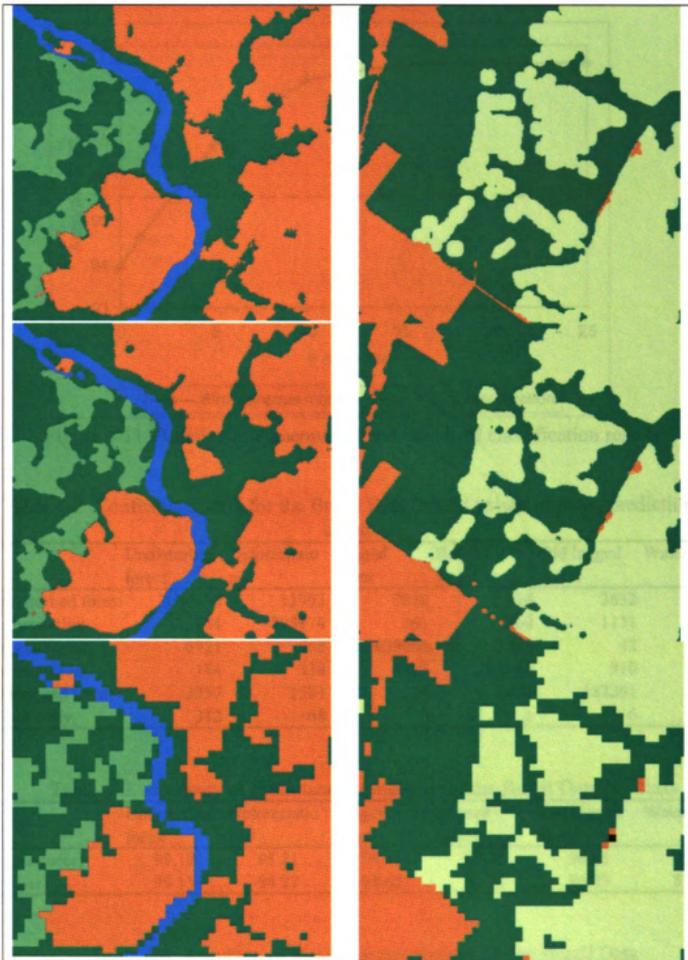


Figure 4-10. Brazil sub set (2 different areas)

(Upper: Reference data, Middle: Simultaneous method after 20 iterations,
Lower: Hard classification)

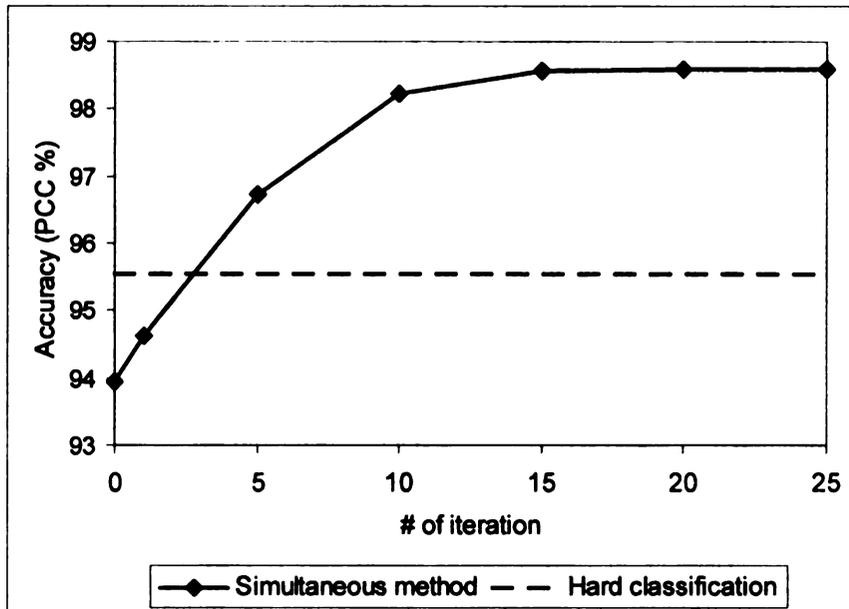


Figure 4-11. Brazil: simultaneous method and hard classification results

Table 4-2. Confusion matrix for the Brazil Data (actual values in rows, predictions in columns)

	Undisturbed forest	Deforestation	Logged forest	Burned forest	Old logged forest	Water body
Undisturbed forest	3034680	13903	7830	214	3632	356
Deforestation	13954	898974	361	260	1171	76
Logged forest	6921	468	639466	1545	42	2
Burned forest	184	336	1495	181846	910	5
Old logged forest	3357	1581	20	1002	182261	12
Water body	312	68	4	2	6	2671

Table 4-3. Accuracy of Simultaneous method for the Brazil Data per class

	Undisturbed forest	Deforestation	Logged forest	Burned forest	Old logged forest	Water body
% (producer)	99.19	98.21	98.50	98.36	96.93	85.55
% (user)	99.15	98.27	98.62	98.41	96.83	87.20

Table 4-4. Accuracy of Simultaneous method for the Brazil Data

	Entire area	Mixed pixels
Number of pixels	22,325,625	4,268,341
PCC	98.30%	92.90%
k (left), k' (right)	0.976	0.908

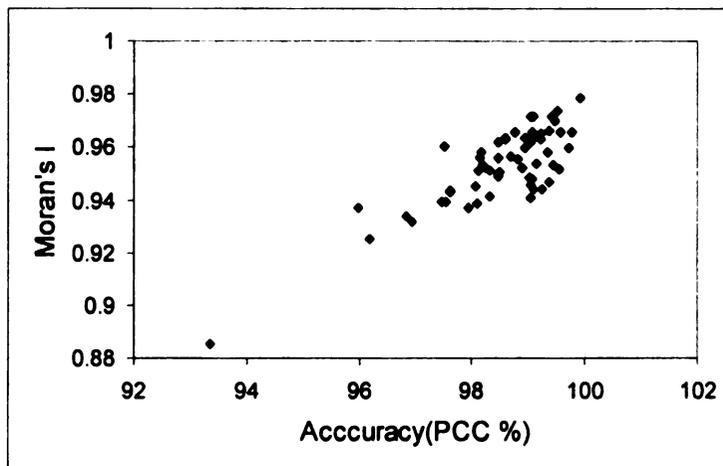
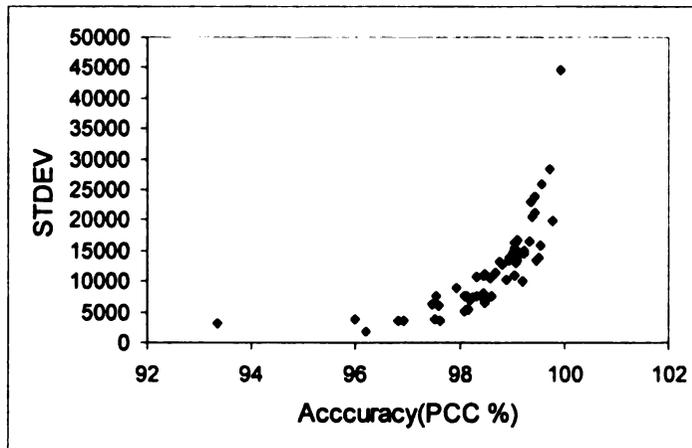
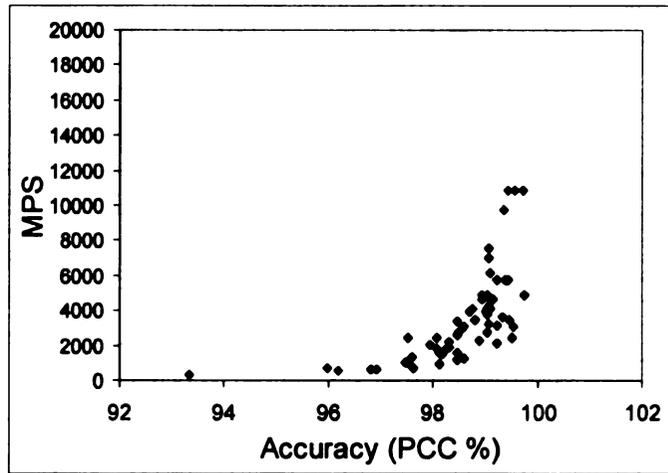


Figure 4-12. Simultaneous method: top (MPS at PCC), middle (STDEV at PCC), bottom (Moran's *I* at PCC)

Table 4-5. Correlation coefficients between PCC and MPS, STDEV and Moran's *I* using 60 sub-areas (315 x 315 pixels)

	MPS	STDEV	Moran's I
correlation coefficient	0.541	0.699	0.819

4.4.2 China

Figure 4-13 shows the relationship between the number of iterations and the classification accuracy. The results imply that the algorithm successfully increases the classification accuracy. For both China 1978 and 2004, the maximum accuracies from simultaneous method (97.97% (1978), 98.58% (2004)) are higher than that from the hard classification (93.53% (1978), 95.04% (2004)). At any iteration, the accuracy for China 2004 is higher than China 1978. The optimum iteration for both data sets is 20, since the accuracy does not increase after that iteration. The time taken for 20 iterations was approximately 3 minutes for China 1978, and 2 minutes for China 2004 on a P4 computer.

Table 4-6~4-9 shows the confusion matrix and the accuracy of degraded real imagery per class for China 1978 and 2004. Both user's and producer's accuracy of pond and river are the two lowest values among the classes. Ponds are mainly confused with Paddy rice, and Rivers are mainly confused with Paddy rice and Forest Land. Table 4-10 show the results of Kappa (k) and Adjusted Kappa (k') measures for China 1978 and 2004. Although k' is lower than k , both values are still higher than 0.9. Table 4-11 shows the correlation coefficient between PCC and Moran's *I*, MPS, STDEV for two types of

sub-areas (98x98 and 315x315) for China 1978 and 2004. All indices are positively correlated to PCC at any case. The strongest correlation (0.884) can be seen between Moran's *I* and PCC for sub-area size 315x315 in China 1978. The striking result is very low correlation coefficient (0.094) between PCC and Moran's *I* for sub-area size 98x98 in China 1978. This relationship is demonstrated in Figure 4-14. One sub-area shows particularly low Moran's *I*. Figure 4-15 shows the image of the sub-area. There are only two classes and the proportion of one class is very small; the other class occupies most of the area. For this pathological case, MPS also does not convey much information. In this study, MPS for multiple classes is simply calculated as total patch areas divided by total number of patches including all classes. In the case of the figure 4-15, there are only two classes; one class has two small patches (8, 32 pixels) and the other has a very large patch (9,176 pixels). The MPS of this sub-area is the average of these three patches (3072 pixels). Figure 4-16 shows the example of histogram of patch size within sub-area size 315x315. The patch size distribution is positively skewed, and there are many small patches and few large patches. MPS will not be informative when the patch size distribution is not normal. For example, MPS will be same where two medium sized patches exist and where one large patch and one small patch exist. STDEV is more informative, since it can capture the variability of the patches.

Table 4-12 shows the correlation coefficient between increase rate of PCC from random allocation and three indices. There are negative correlations at any case except between PCC and Moran's *I* for China 1978 with extent 98x98. Table 4-13 shows the correlation coefficient between three indices. MPS and STDEV are positively correlated in some degree at all cases. STDEV and Moran's *I* do not show any strong correlations.

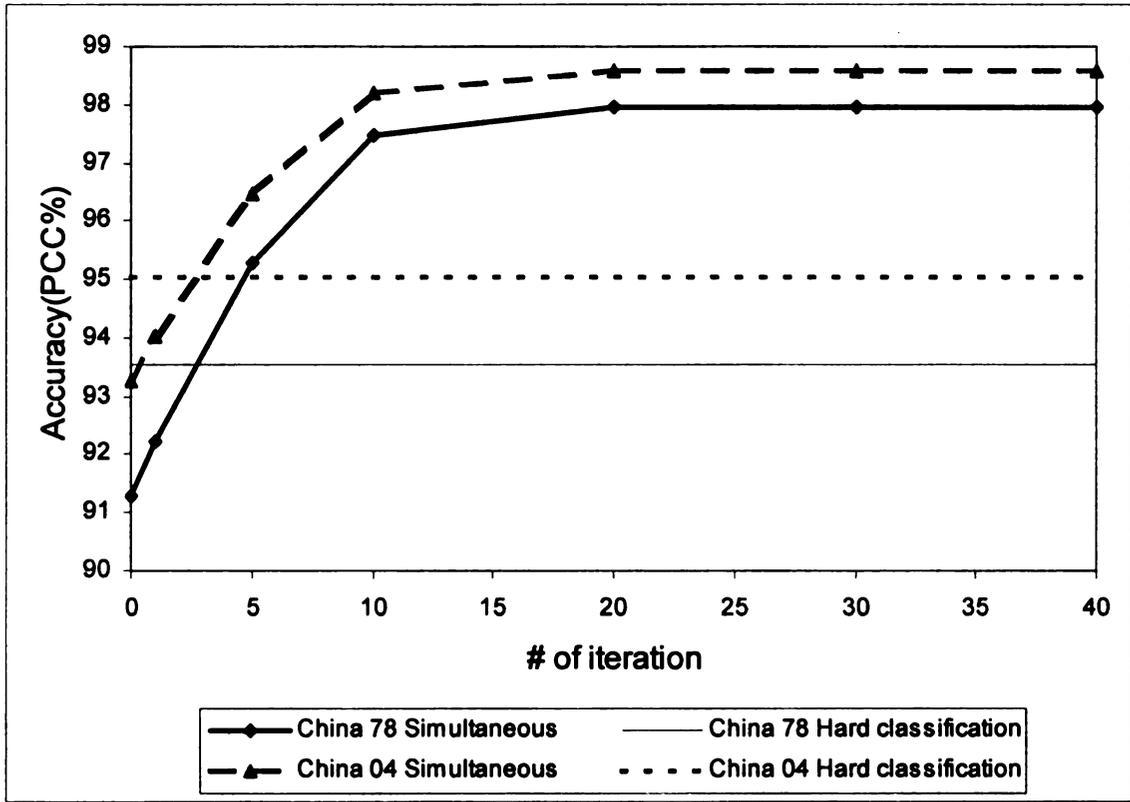


Figure 4-13. China 1978, 2004: simultaneous method and hard classification results

Table 4-6. Confusion matrix for China 1978 (actual values in rows, predictions in columns)

	Paddy rice	Upland crop	Forest land	Sparse forest	Urban	Rural habitat	River	Lake	Reservoir	Pond	Swamp	Other unused land
Paddy rice	192279	150	2630	69	4	11	1131	35	5	55	10	0
Upland crop	124	12077	37	0	0	0	30	34	0	7	0	1
Forest land	2732	33	299221	65	8	29	316	413	8	1	0	0
Sparse forest	73	0	53	11899	0	0	5	0	0	0	0	0
Urban	3	0	5	0	643	0	2	0	0	0	0	1
Rural habitat	17	0	22	0	0	1432	8	22	0	0	0	0
River	1008	25	288	1	1	5	10720	1	0	0	0	1
Lake	42	33	392	0	0	10	2	10576	0	0	0	0
Reservoir	17	0	6	0	0	0	0	0	265	0	0	0
Pond	42	5	10	0	0	0	0	0	0	330	0	0
Swamp	12	0	0	0	0	0	0	0	0	0	317	0
Other unused land	0	1	0	0	0	0	0	0	0	0	0	167

Table 4-7. Accuracy of the Simultaneous Method for China 1978 per class

	Paddy rice	Upland crop	Forest land	Sparse forest	Urban	Rural habitat	River	Lake	Reservoir	Pond	Swamp	Other unused land
% (producer)	97.92	98.00	98.86	98.88	98.02	96.30	87.77	95.44	95.32	83.97	96.65	98.24
% (user)	97.91	98.11	98.81	98.91	98.32	95.40	88.96	95.67	92.01	85.27	96.06	98.82

Table 4-8. Confusion matrix for China 2004 (actual values in rows, predictions in columns)

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
Class 1	26279	64	54	2	0	164	20
Class 2	56	141654	578	49	0	1492	130
Class 3	39	532	102633	102	1	514	13
Class 4	2	32	95	31219	1	180	0
Class 5	0	0	0	2	359	1	0
Class 6	149	1483	534	183	1	227215	40
Class 7	12	117	9	0	0	57	13915

Table 4-9. Accuracy of Simultaneous Method for China 2004 per class

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
% (producer)	99.03	98.45	98.78	98.93	99.17	98.95	98.55
% (user)	98.86	98.39	98.84	99.02	99.17	98.96	98.61

Table 4-10. Accuracy of Simultaneous Method

	China 1978		China 2004	
	Entire area	Mixed pixels	Entire area	Mixed pixels
Number of pixels	2,263,800	624,995	2,263,800	498,428
PCC	96.78%	88.89%	98.78%	93.81%
k, k'	0.982	0.928	0.983	0.917

Table 4-11. Correlation coefficient between PCC and MPS, STDEV and Moran's I

	China 1978		China 2004	
	98x98	315x315	98x98	315x315
Size of sub area	98x98	315x315	98x98	315x315
Number of samples	60	8	60	8
PCC and MPS	0.642	0.668	0.596	0.704
PCC and STDEV	0.645	0.247	0.323	0.556
PCC and Moran's I	0.094	0.884	0.483	0.607

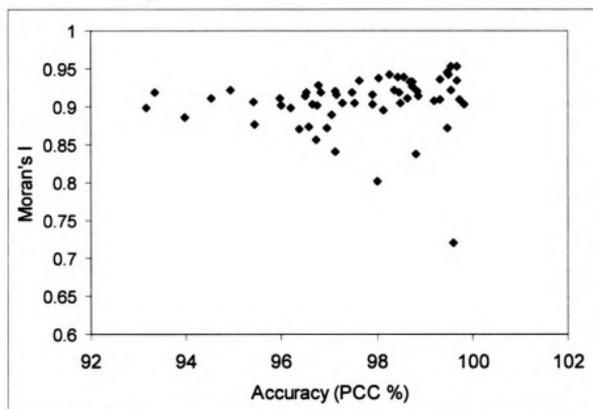


Figure 4-14. China 1978 (sub-area=98x98): Moran's I at PCC

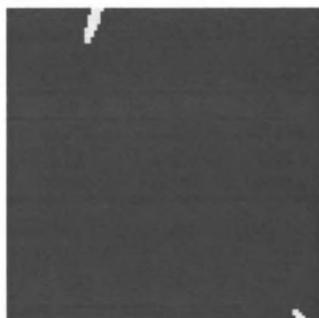


Figure 4-15. Sub-area (98x98) for China 1978: Moran's I = 0.72, PCC = 99.6%, MPS = 3072, STDEV = 5286

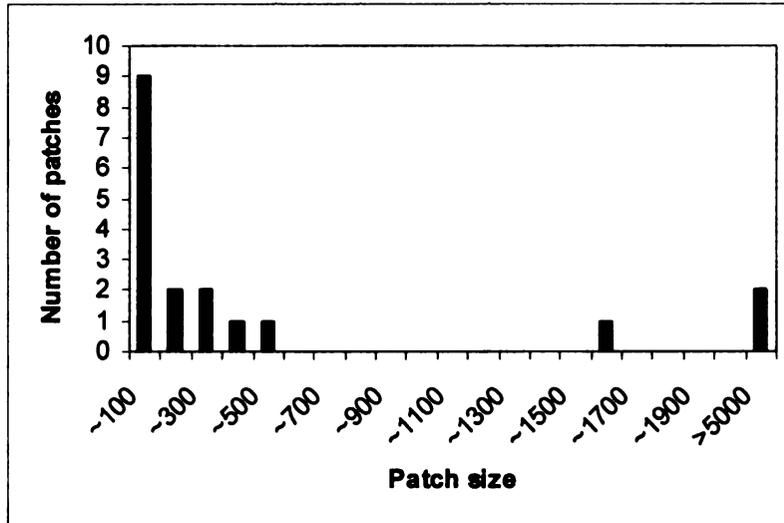


Figure 4-16. Patch size distribution at sub-area (315x315) at China 1978

Table 4-12. Correlation coefficient between Increase rate of PCC from random allocation and MPS, STDEV and Moran's *I*

	China 1978		China 2004	
	98x98	315x315	98x98	315x315
Size of sub area				
Number of samples	60	8	60	8
Increase rate and MPS	-0.728	-0.619	-0.589	-0.834
Increase rate and STDEV	-0.827	-0.874	-0.666	-0.683
Increase rate and Moran's <i>I</i>	0.209	-0.273	-0.096	-0.584

Table 4-13. Correlation coefficient between three indices (MPS, STDEV and Moran's *I*)

	China 1978		China 2004		Brazil
	98x98	315x315	98x98	315x315	315x315
Size of sub area					
Number of samples	60	8	60	8	60
MPS and STDEV	0.899	0.615	0.298	0.916	0.931
MPS and Moran's <i>I</i>	-0.059	0.713	0.403	0.130	0.509
STDEV and Moran's <i>I</i>	-0.272	0.074	-0.254	0.093	0.523

4.5 Discussion and Conclusions

The use of the simultaneous allocation method in sub-pixel mapping proved a valuable alternative compared with already existing techniques. All results suggested that this technique could increase the classification accuracy when compared with traditional hard classification. Adjusted Kappa statistics (k'), which only evaluate the mixed pixels, also support the results. As expected, k' value is lower than k value, since k' is estimated by excluding pure pixels. However, k' is still above 0.9 for all Brazil and China data set. Thus, the simultaneous method could accurately capture the landscape patterns at sub-pixel scales.

At any iteration, the accuracy for China 2004 is higher than one for China 1978. Although two images have different pixel sizes, both images contain the same number of pixels. Since two sites are in close proximity, their landscape pattern will be similar and the degree of contiguity for both landscapes is very close. Moran's I for China 1978 is 0.96 and China 2004 is 0.97. The reason for different accuracy at two study sites would be related to the number of classes. There are 12 classes for China 1978 and 7 classes for China 2004. It will cause less error to allocate few classes within a pixel than many classes. Hence, the accuracy for China 2004 is higher than one for China 1978 at the same number of iteration. There are 6 classes for Brazil, and the maximum accuracy for Brazil is similar to one for China 2004. Thus, the number of classes of in the image will be one important factor that affects the ability of sub-pixel mapping technique.

For Brazil, both user's and producer's accuracy of water is lowest among the classes. For China 1978, both user's and producer's accuracy of pond and river are the

first and second lowest among the classes. In the Brazil study area, most of patches in water class are small and their shapes are long and narrow. In the China 1978 study area, ponds are mostly small, with a MPS of 76 pixels and more than half of patches less than 30 pixels. Since the degraded image contains 49 fine pixels within a pixel, many ponds are smaller than one coarse pixel. Many rivers in the study sites are narrow and less than 3 pixels wide. These narrow features converge to circular features. As I mentioned in Chapter 3, these phenomena would be inevitable as long as we attempt to maximize the degree of contiguity. The simultaneous allocation method would not be appropriate to apply where the land cover patches are smaller or narrower than a pixel, since it focuses on land cover features larger than the scale of a pixel by using the information contained in surrounding pixels.

In this study, I examined how the characteristics of the landscape affect the ability of sub-pixel mapping. One initial assumption was that PCC was high where Moran's *I* was high, since the simultaneous method attempts to maximize the autocorrelation. Although the results from the Brazil and China Data set using relatively large sub-areas (315x315) supports this assumption, it is not the case for small sub-areas (98x98). At these smaller sub-areas, some areas contain only a few classes with small proportions. Moran's *I* tend to show relatively low values where the proportions between the classes are large. However, it would be unusual to apply sub-pixel mapping to an area which contains only a few classes with very different proportions such as figure 4-15. For this case, the improvement in PCC is very low. As Table 4-12 shows, STDEV is strongly negatively correlated to the increased PCC rate. This indicates that where the large patches dominate the area, the increase rate of PCC is low since the landscape may not

contain many mixed pixels. The application of the simultaneous method to a relatively heterogeneous landscape can increase classification accuracy compared to the random allocation. The results show that there was no particular index always most correlated to PCC. However, Moran's *I* does have the strongest positive relationship to PCC in most cases. The simultaneous method would not be the best method for all types of landscapes, since it attempts to maximize the degree of contiguity. The landscape metrics provide valuable information about landscape characteristics. Moran's *I* could be used as an indicator for the overall accuracy and STDEV could be used as an indicator for the increased rate of accuracy for testing the applicability of the simultaneous method.

In this study, Moran's *I*, STDEV and MPS are used as indices to characterize landscape. However, the results of these indices varied by extents and areas. It will be more effective to apply indices which are scale independent such as Deviation from Neutral (DfN). DfN measures the true distance between the pattern metric values of the sample landscape and those of the pattern metric values of the neutral landscape (Messina et al. 2006a). DfN can be used as a measurement base from which it becomes possible to compare different landscapes with various grain sizes and extents, independent of place (Messina et al. 2006b). DfN will be used to examine the applicability of the simultaneous method.

Chapter 5

Conclusions and Future Research

This dissertation is devoted to explore and develop new algorithms for predicting land cover distribution using imagery at the sub-pixel levels. I investigated the “pixel-swapping” optimization algorithm, which was introduced by Atkinson (2001, 2005) for predicting sub-pixel land cover distribution. Two limitations of this method, the arbitrary spatial range value and the arbitrary exponential model of spatial autocorrelation are assessed. Various weighting functions, as alternatives to the exponential model, are evaluated in order to derive the optimum weighting function. Two different simulation models are employed to develop spatially autocorrelated binary class maps. In all tested models, Gaussian, Exponential, and IDW, the pixel swapping method improve classification accuracy compared with the initial random allocation of sub pixels. However, the results suggest that equal weight, which give equal weight to its nearest neighbors, could be used to maximize accuracy and sub-pixel spatial autocorrelation instead of using these more complex models of spatial structure.

I develop and evaluate three distinct methods for modeling the spatial distribution of multiple land cover classes at sub-pixel scales. Three methods are examined: sequential categorical swapping, simultaneous categorical swapping, and simulated annealing. These three methods area applied to classified Landsat ETM+ data resampled to 210 meters. The result suggested that the simultaneous method was the optimum method in terms of accuracy performance and computation time.

The case study employs remote sensing imagery at the following sites: tropical forests in Brazil and temperate multiple land mosaic in East China. Sub-areas of both sites are used to examine how the characteristics of the landscape affect the ability of the simultaneous method. Three types of measurement, Moran's I , mean patch size (MPS), and patch size standard deviation (STDEV), are used to characterize the landscape. The landscape metrics provide valuable information about landscape characteristics. Moran's I could be used as an indicator for the overall accuracy and STDEV could be used as an indicator for the increase rate of accuracy for testing the applicability of the simultaneous method. All results suggested that the simultaneous categorical swapping technique could increase classification accuracy in comparison with traditional hard classification for different types of landscapes.

The methods developed in this study can benefit researchers who employ coarse remote sensing imagery but are interested in detailed landscape information. In many cases, the satellite sensor that provides large spatial coverage has insufficient spatial detail to identify landscape patterns. Therefore, application of the super-resolution technique described in this dissertation could potentially solve this problem by providing detailed land cover predictions from the coarse resolution satellite sensor imagery. In this study, one pixel is subdivided into 7 by 7 sub-pixels to obtain a finer resolution image. The algorithms developed in this study allow users to specify the number of subdivision. However, the accuracy will decrease as the number of subdivision increase (Makido and Qi, 2005). For example, if 77 percent accuracy is accepted, 1km resolution MODIS images can be used to derive 30 meter resolution images (Makido and Qi, 2005). Thus, the method developed here can be used to provide a sufficient spatial and temporal

coverage with finer spatial resolution than the currently exist. It also can be applied to image archives to increase the spatial resolution for revealing greater detail of land cover information. Such information will be valuable for many researches such as monitoring land cover land use change.

The techniques developed here should be applicable to imagery from any remote sensing system as long as the basic assumptions about spatial dependence are fulfilled. In addition, it should not be considered applicable exclusively in the field of land cover mapping. The technique has potential in any area of GIS research where data are spatially aggregated (Tatem et al., 2003). The method could be used dis-aggregate spatially aggregated sociological data, such as health or crime statistics within enumeration districts.

Additional research is necessary to extend the current algorithms to handle land cover features that are smaller than a pixel. This study adopts the assumption of spatial dependency both within and among pixels, as do most existing algorithms. Therefore, the various algorithms introduced in this study work best where land cover features are larger than the sub-pixels and are spatially autocorrelated (e.g., agricultural fields), since these techniques employ the information contained in surrounding pixels. However, this source of information is unavailable when examining imagery of land cover features that are smaller than a pixel (e.g., trees). While these features can be detected within a pixel by soft classification techniques, surrounding pixels do not hold any information for inference of spatial relationships to aid their mapping (Tatem et al., 2002). One potential scheme would be the employment of prior information on the spatial arrangement of land

cover, such as a variogram for the target resolution. Instead of using Moran's I as the objective, reproduction of the variogram model could be used as the objective of the Simulated Annealing technique. The technique could reproduce the prior structural information given as variogram models. The spatial pattern of application-specific land cover classes would be valuable information for managing and understanding the environment.

APPENDICES

**Appendix A (IDL program for pixel-swapping algorithm using exponential weighting
function)**

pro expo

```
-----  
; Sub-pixel Mapping for binary imagery using exponential weighting function  
; Written by Yasuyo Makido  
; Created July 2004 (Modified October 2006)  
;  
; This function employs pixel-swapping method to maximize the autocorrelation  
; between sub-pixels  
;  
; Reference:  
; Atkinson, P.M. (2005). Sub-pixel Target Mapping from Soft-classified,  
; Remotely Sensed Imagery. Photogrammetric Engineering & Remote Sensing:  
; 71(7), 839-846  
-----
```

```
time = systime(1)
```

```
;;;;;  
;; Input parameters  
  
;;;;;  
;original coarse proportion array which contain the class proportion  
  
coarse_prop = read_tiff('d:\Yasuyo\paper\p_pers\natural\c7_315_2h.tif')  
fine = read_tiff('d:\Yasuyo\paper\p_pers\natural\neut3152h.tif')  
;reference image  
  
;;;;;  
;the zoom factor (sub-pixels within 1 pixel)  
zoom = 7  
  
;;;;;  
;arbitrary a value for calculating  $O_i$  , Atkinson use  $a\_val = 5$   
a_val= 5  
  
;;;;;  
; dx is the neighboring distance in sub-pixels (1:N=8, 2:N=24)  
dx = 2  
dy = 2  
  
;;;;;  
; The number of iteration  
iteration = 12  
  
;;;;;
```

```

;;;;; STEP 1 ;;;;;;;;;;;;;;
;;;;; Zoom up the number of pixels
;;;;;

```

```

n_col = N_ELEMENTS(coarse_prop[* ,1]) ;# of column for prop
n_row = N_ELEMENTS(coarse_prop[1,*]) ;# of row for prop

```

```

sub_col = n_col * zoom
sub_row = n_row * zoom

```

```

arr_bi = MAKE_ARRAY(sub_col, sub_row, /INTEGER, VALUE = 0)

```

```

;;;;; STEP 2 ;;;;;;;;;;;;;;
;;;;; Randomly allocate the class proportion (0/1) within a pixel
;;;;;

```

```

FOR x=0, n_col-1 DO BEGIN
FOR y=0, n_row-1 DO BEGIN

```

```

prop = coarse_prop[x,y] ;prop is the class proportion
R = RANDOMU(seed, zoom, zoom) ;make the random zoom*zoom matric

```

```

IF prop EQ 0 THEN BEGIN
R[* ,*] = 0

```

```

ENDIF ELSE BEGIN

```

```

order = SORT(R)
index_1 = order[0: prop-1]
R[index_1] = 1

```

```

index_0 = where(R NE 1, count)
IF count NE 0 THEN R[index_0] = 0 ;To avoid the case of "where"

```

```

returns -1

```

```

ENDELSE

```

```

; Insert the alley R to the alley arr_bi

```

```

arr_bi[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] = R

```

```

ENDFOR
ENDFOR

```

```

.....
,,,,, STEP 3 ,,,,,,
,,,,, Calculate attractiveness Oi at each sub-pixel
,,,,, (ITERATE STEP 3 and STEP 4 "iteration" times)
.....

```

```

;;Exponential weighting function is used

```

```

FOR i=0, iteration-1 DO BEGIN

```

```

  print, 'iteration -----', i+1

```

```

  arr_oi = MAKE_ARRAY(n_col*zoom, n_row*zoom, /FLOAT, VALUE = 0.0)

```

```

  FOR xcol = 0, (sub_col-1) DO BEGIN

```

```

    FOR yrow = 0, (sub_row-1) DO BEGIN

```

```

      oi = 0

```

```

      sum_oi = 0

```

```

        FOR xcell=(0-dx), dx DO BEGIN

```

```

          FOR ycell=(0-dy), dy DO BEGIN

```

```

            IF ((xcol+xcell GE 0) AND (xcol+xcell LT sub_col)) AND ((yrow+ycell
            GE 0)AND(yrow+ycell LT sub_row)) THEN BEGIN

```

```

              dist = 0

```

```

              dist = SQRT(xcell*xcell+ycell*ycell)

```

```

              IF (dist EQ 0) THEN oi = 0 ELSE oi = EXP((0-
              dist)/a_val)*arr_bi[xcol+xcell, yrow+ycell]

```

```

              sum_oi = sum_oi +oi

```

```

            ENDIF

```

```

          ENDFOR

```

```

        ENDFOR

```

```

      arr_oi[xcol,yrow] = sum_oi ; Put the oi value to the subpixel

```

```

    ENDFOR

```

```

  ENDFOR

```

```

.....
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,, STEP 4 ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,, Find the Min Oi where arr_bi=1,
,,,,, Find the Max Oi where arr_bi=0
,,,,, AND swap the value if Min Oi < Max Oi
.....
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

```

```

FOR x=0, n_col-1 DO BEGIN
FOR y=0, n_row-1 DO BEGIN

```

```

sub_bi = arr_bi[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] ;sub pixel of
arr_bi
sub_oi = arr_oi[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] ;sub pixel of
arr_oi

```

```

IF (MAX(sub_bi) NE MIN(sub_bi)) THEN BEGIN

```

```

    bi_1 = where(sub_bi EQ 1)

```

```

    min_oi = MIN(sub_oi[bi_1])
    min_bi_1 = where((sub_oi EQ min_oi) AND (sub_bi EQ 1))

```

```

    bi_0 = where(sub_bi EQ 0)

```

```

    max_oi = MAX(sub_oi[bi_0])
    max_bi_0 = where((sub_oi EQ max_oi) AND (sub_bi EQ 0))

```

```

; If Min Oi < Max Oi, min_bi_1 = 0 and max_bi_0 = 1

```

```

IF (min_oi LT max_oi) THEN BEGIN

```

```

; If more than two index exist, use the smaller/larger one

```

```

    sub_bi[MIN(min_bi_1)] = 0
    sub_bi[MAX(max_bi_0)] = 1

```

```

ENDIF

```

```

ENDIF

```

```

; Insert the sub_bi to the alley arr_bi

```

```
arr_bi[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] = sub_bi
```

```
ENDFOR  
ENDFOR
```

```
ENDFOR ;End of iterations
```

```
WRITE_TIFF,'d:\Yasuyo\paper\p_pers\nneutral\test.tif',arr_bi ;out put file
```

```
.....  
.....  
.....;ACCURACY ASSESSMENT (PCC);.....  
.....
```

```
pcc = (fine+1) - arr_bi ; add1 to avoid (0-1=-1 ->255)
```

```
correct = N_ELEMENTS(where(pcc EQ 1))
```

```
pcc = float(correct)/(sub_col*sub_row)*100
```

```
print, 'PCC: ', pcc, ' :% for iteration', iteration
```

```
print, 'time =', (systemtime(1) - time)/60, ' Minutes'
```

**Appendix B (IDL program for pixel-swapping algorithm using exponential weighting
function)**

pro nn

```
-----  
;  
;  
; Sub-pixel Mapping for binary imagery using Nearest Neighbor function  
; Written by Yasuyo Makido  
; Created July 2004 (Modified March 2005, November 2006)  
;  
; This function employs pixel-swapping method to maximize the autocorrelation  
; between sub-pixels  
;  
; Reference:  
;   Atkinson, P.M. (2005). Sub-pixel Target Mapping from Soft-classified,  
;   Remotely Sensed Imagery. Photogrammetric Engineering & Remote Sensing:  
;   71(7), 839-846  
;  
-----
```

```
time = systime(1)
```

```
.....  
; INPUT IMAGE ;  
.....
```

```
.....  
; INPUT FILE (FINE RESOLUTION BINARY);
```

```
filepatha ='d:\yasuyo\nneutral\output\aneut'  
filepathb ='d:\yasuyo\nneutral\output\bneut'
```

```
;  
;   filepatha ='/home/yasuyo/Neutral/aneut' ; for linux  
;   filepathb ='/home/yasuyo/Neutral/bneut' ; for linux
```

```
filepath3 = '.tif'
```

```
FOR file_num = 1, 2 DO BEGIN ;2
```

```
IF file_num EQ 1 THEN filepath1 = filepatha  
IF file_num EQ 2 THEN filepath1 = filepathb
```

```
FOR image_num = 1,20 DO BEGIN ; 20
```

```
filepath2 = string(image_num)  
filepath2 = strtrim(filepath2, 2)
```

```

filepath = filepath1 + filepath2 + filepath3

fine = read_tiff(filepath)

truth = fine

.....
,,,,, INPUT PARAMETER,,,,,
.....

.....
;the zoom factor (sub-pixels within 1 pixel)
zoom = 7

.....
; dx is the neighboring distance in sub-pixels (1:N=8, 2:N=24)
dx = 2
dy = 2

width = dx*2+1 ;3 for dx/dy=1, 5 for dx/dy=2

.....
; The number of iteration

; FOR iteration = 1,20 DO BEGIN ; 20

iteration = 20 ;20

.....
;;Check the size of the array

sub_col = N_ELEMENTS(fine[* ,1]) ;# of column for the fine image
sub_row = N_ELEMENTS(fine[1,*]) ;# of row for fine image

n_col = sub_col/zoom
n_row = sub_row/zoom

.....
,,,,, Create a new coarse image (coarse_prop)
.....

coarse_prop = MAKE_ARRAY(n_col, n_row, /byte, VALUE = 0)

FOR x=0, n_col-1 DO BEGIN
FOR y=0, n_row-1 DO BEGIN

```

```

sub_arr = fine[(x*zoom):(x*zoom+zoom-1),
(y*zoom):(y*zoom+zoom-1)]
coarse_prop[x,y] = TOTAL(sub_arr)

```

```

ENDFOR
ENDFOR

```

```

.....
,,,,, STEP 1 ,,
,,,,, Zoom up the number of pixels
,,,,,
.....

```

```

arr_bi = MAKE_ARRAY(sub_col, sub_row, /INTEGER, VALUE = 0)

```

```

.....
,,,,, STEP 2 ,,
,,,,, Randomly allocate the class proportion (0/1) within a pixel
,,,,,
.....

```

```

FOR x=0, n_col-1 DO BEGIN
FOR y=0, n_row-1 DO BEGIN

```

```

; Create zoom x zoom array R, which randomly allocate 1 by the class proportion

```

```

prop = coarse_prop[x,y] ;prop is the class proportion
R = RANDOMU(seed, zoom, zoom) ;make the random zoom*zoom matrix

```

```

;; IF prop=0, r=0, otherwise ....

```

```

IF prop EQ 0 THEN BEGIN
R[*,*] = 0

```

```

ENDIF ELSE BEGIN

```

```

order = SORT(R)
index_1 = order[0: prop-1]
R[index_1] = 1

```



```
.....  
; Iteration END
```

```
ENDFOR ; end for iteration
```

```
.....  
; ACCURACY ;  
.....
```

```
; print, '-----'  
  
pcc = truth - arr_bi  
correct = N_ELEMENTS(where(pcc EQ 0))  
percent = float(correct)/(sub_col*sub_row)*100  
  
print, 'PCC (%): ', percent, ' : iteration :', iteration
```

```
;ENDFOR ; end for various iteration
```

```
ENDFOR ; end for image_num
```

```
ENDFOR ; end for file_num
```

```
print, 'time =', (systemtime(1) - time)/60, ' Minutes'
```

```
end
```

Appendix C (IDL program for Sequential Categorical Swapping)

pro seque

```
-----  
;  
;  
; Sequential Categorical Swapping (Sub-pixel Mapping for multiple classes imagery)  
; Written by Yasuyo Makido  
; Created May 2005 (Modified August 2006)  
;  
; This function employs pixel-swapping method to maximize the autocorrelation  
; between sub-pixels  
;  
; Attractiveness Oi : Equal Weight function  
; Number of class = 5  
;  
; Reference:  
; Atkinson, P.M. (2005). Sub-pixel Target Mapping from Soft-classified,  
; Remotely Sensed Imagery. Photogrammetric Engineering & Remote Sensing:  
; 71(7), 839-846  
;  
;  
-----
```

time = systime(1)

```
x_times = 20 ;20  
pcc_arr = fltarr(x_times)
```

```
FOR test=1, x_times DO BEGIN
```

```
.....  
;# of class  
num_class = 5
```

```
.....  
;the zoom factor (sub-pixels within 1 pixel)  
zoom = 7
```

```
.....  
; dx is the neighboring distance in sub-pixels (1:N=8, 2:N=24)  
dx = 3  
dy = 3
```

```
width = dx*2+1 ;3 for dx/dy=1, 5 for dx/dy=2
```

```
.....  
; The number of iteration  
iteration = 30 ;For class1 ;30
```



```

original_prop = MAKE_ARRAY(n_col, n_row, num_class, /byte, VALUE = 0)

FOR class = 0, num_class-1 DO BEGIN

FOR x=0, n_col-1 DO BEGIN
FOR y=0, n_row-1 DO BEGIN

        sub_arr = fine[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-
1),class]
        original_prop[x,y,class] = TOTAL(sub_arr)

ENDFOR
ENDFOR

ENDFOR

```

```

;print, '-----CLASS 1-----'

```

```

coarse_prop = original_prop(*,*,0)

```

```

.....
;;;; STEP 1-1 ;;;;;
;;;; Zoom up the number of pixels
.....

```

```

sub_col = n_col * zoom
sub_row = n_row * zoom

```

```

arr_bi = MAKE_ARRAY(sub_col, sub_row, /INTEGER, VALUE = 0)

```

```

.....
;;;; STEP 1-2 ;;;;;
;;;; Randomly allocate the class proportion (0/1) within a pixel
.....

```

```

FOR x=0, n_col-1 DO BEGIN
FOR y=0, n_row-1 DO BEGIN

```

; Creat zoom x zoom alley R, which randomly allocate 1 by the class proportion

```

prop = coarse_prop[x,y] ;prop is the class proportion
R = RANDOMU(seed, zoom, zoom) ;make the random zoom*zoom matric

IF prop EQ 0 THEN BEGIN
  R[*,*] = 0

ENDIF ELSE BEGIN

  order = SORT(R)
  index_1 = order[0: prop-1]
  R[index_1] = 1

  index_0 = where(R NE 1, count)
  IF count NE 0 THEN R[index_0] = 0 ;To avoid the case of "where"
returns -1

ENDELSE

; Insert the alley R to the alley arr_bi

arr_bi[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] = R

ENDFOR
ENDFOR

.....
;;; STEP 1-3 .....
;;; Calculate attractiveness Oi at each sub-pixel
;;; (ITERATE STEP 3 and STEP 4 "iteration" times)
;;; EQUAL WEIGHT FUNCTION .....
.....

;
FOR i=0, iteration-1 DO BEGIN

.....
;; make 0 buffer around the boundary

big_arr_oi = MAKE_ARRAY(sub_col+dx+dx, sub_row+dy+dy)

big_arr_oi[dx:dx+sub_col-1, dy:dy+sub_row-1] = arr_bi

```

```

.....
; Calculate Oi(the sum the neighboring pixel value) using moving window

```

```

big_arr_oi = SMOOTH(big_arr_oi, width)* width*width

```

```

arr_oi = big_arr_oi[dx:dx+sub_col-1, dy:dy+sub_row-1]

```

```

.....
;;; STEP 1-4 ;;;;
;;; Find the Min Oi where arr_bi=1,
;;; Find the Max Oi where arr_bi=0
;;; AND swap the value if Min Oi < Max Oi
.....

```

```

FOR x=0, n_col-1 DO BEGIN
FOR y=0, n_row-1 DO BEGIN

```

```

sub_bi = arr_bi[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] ;sub pixel of
arr_bi
sub_oi = arr_oi[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] ;sub pixel of
arr_oi

```

```

IF (MAX(sub_bi) NE MIN(sub_bi)) THEN BEGIN ;Do only where both class 1&0 exist

```

```

    bi_1 = where(sub_bi EQ 1)

```

```

    min_oi = MIN(sub_oi[bi_1])
    min_bi_1 = where((sub_oi EQ min_oi) AND (sub_bi EQ 1))

```

```

    bi_0 = where(sub_bi EQ 0)

```

```

    max_oi = MAX(sub_oi[bi_0])
    max_bi_0 = where((sub_oi EQ max_oi) AND (sub_bi EQ 0))

```

```

; If Min Oi < Max Oi, min_bi_1 = 0 and max_bi_0 = 1

```

```

IF (min_oi LT max_oi) THEN BEGIN

```

```

; If more than two index exist, use the smaller/larger one

```

```

    sub_bi[MIN(min_bi_1)] = 0
    sub_bi[MAX(max_bi_0)] = 1

```

```

ENDIF

```

```
ENDIF
```

```
; Insert the sub_bi to the alley arr_bi
```

```
arr_bi[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] = sub_bi
```

```
ENDFOR
```

```
ENDFOR
```

```
.....  
; Iteration END
```

```
ENDFOR
```

```
.....  
; CLASS 2 START !!!!!  
.....
```

```
;print, '-----CLASS 2-----'
```

```
coarse_prop2 = original_prop(*,*,1)
```

```
arr_bi2 = arr_bi
```

```
.....  
; STEP 2-2 ;  
; Randomly allocate the class proportion (0/2) within a pixel  
; Except where arr_bi = 1  
.....
```

```
;print, 'step 2-2, random allocation'
```

```
FOR x=0, n_col-1 DO BEGIN  
FOR y=0, n_row-1 DO BEGIN
```

```
    ; Create zoom x zoom alley R, which randomly allocate 1 by the class proportion
```

```
    prop = coarse_prop2[x,y] ;prop is the class proportion  
    R = RANDOMU(seed, zoom, zoom) ;make the random zoom*zoom matrix
```

```
    R = R + arr_bi2[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)]  
        ;it will exceed 1.0 where class 1 is already allocated
```

```

IF prop EQ 0 THEN BEGIN
    R[*,*] = 0

ENDIF ELSE BEGIN

    order = SORT(R)
    index_1 = order[0: prop-1] ;select small number with proportion
    R[index_1] = 2 ;Allocate class 2

    index_0 = where(R NE 2, count) ;Allocate class 0 where class NE 2
    IF count NE 0 THEN R[index_0] = 0 ;To avoid the case of "where"
returns -1

ENDELSE

; Insert the alley R to the alley arr_bi2

arr_bi2[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] = R

ENDFOR
ENDFOR

.....
;;; STEP 2-3 .....
;;; Calculate attractiveness Oi at each sub-pixel
;;; (ITERATE STEP 3 and STEP 4 "iteration" times)
.....

;print,'step 2-3 calculate oi'
;print,'-----Iteration Start-----'

FOR i2=0, iteration2-1 DO BEGIN

.....
;; make 0 buffer around the boundary

big_arr_oi = MAKE_ARRAY(sub_col+dx+dx, sub_row+dy+dy,/int)
big_arr_oi[dx:dx+sub_col-1, dy:dy+sub_row-1] = arr_bi2

.....
;; Calculate Oi(the sum the neighboring pixel value) using moving window

```

```

big_arr_oi = SMOOTH(big_arr_oi/2.0, width)* width*width

arr_oi2 = big_arr_oi[dx:dx+sub_col-1, dy:dy+sub_row-1]

```

```

.....
,,,,, STEP 2-4:.....
,,,,, Find the Min Oi where arr_bi=2,
,,,,, Find the Max Oi where arr_bi=0
,,,,, AND swap the value if Min Oi < Max Oi
.....

```

```

arr_bi_12= arr_bi2+arr_bi

```

```

FOR x=0, n_col-1 DO BEGIN
FOR y=0, n_row-1 DO BEGIN

```

```

sub_bi = arr_bi_12[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] ;sub
pixel of arr_bi
sub_oi = arr_oi2[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] ;sub pixel
of arr_oi

```

```

IF ((MAX(sub_bi) EQ 2) AND (MIN(sub_bi) EQ 0)) THEN BEGIN ;DO Only where
both class 0&2 exist

```

```

bi_2 = where(sub_bi EQ 2) ;Find the place where class=2
min_oi = MIN(sub_oi[bi_2]) ;Find the minimum Oi value within class=2
min_bi_2 = where((sub_oi EQ min_oi) AND (sub_bi EQ 2)) ;Find the place
where oi=minimum within class=2

```

```

bi_0 = where(sub_bi EQ 0)
max_oi = MAX(sub_oi[bi_0])
max_bi_0 = where((sub_oi EQ max_oi) AND (sub_bi EQ 0))

```

```

IF (min_oi LT max_oi) THEN BEGIN

```

```

sub_bi[MIN(min_bi_2)] = 0
sub_bi[MAX(max_bi_0)] = 2

```

```

ENDIF

```

```
ENDIF
```

```
; Insert the sub_bi to the alley arr_bi_12
```

```
arr_bi_12[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] = sub_bi
```

```
ENDFOR
```

```
ENDFOR
```

```
index_bi_01 = where(arr_bi_12 NE 2) ;create new arr_bi2
```

```
arr_bi2 = arr_bi_12
```

```
arr_bi2[index_bi_01] = 0
```

```
ENDFOR
```

```
.....  
.....  
..... CLASS 3 START !!!!! .....  
.....  
.....
```

```
;print, '-----CLASS 3-----'
```

```
coarse_prop3 = original_prop(*,*,2)
```

```
arr_bi3 = arr_bi_12
```

```
.....  
..... STEP 3-2 .....  
..... Randomly allocate the class proportion (0/3) within a pixel  
..... Except where class 1&2 allocate  
.....
```

```
;  
;print, 'step 3-2, random allocation'
```

```
FOR x=0, n_col-1 DO BEGIN
```

```
FOR y=0, n_row-1 DO BEGIN
```

```
; Creat zoom x zoom alley R, which randomly allocate 1 by the class proportion
```

```
prop = coarse_prop3[x,y] ;prop is the class proportion
```

```
R = RANDOMU(seed, zoom, zoom) ;make the random zoom*zoom matrix
```

```
R = R + arr_bi3[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)]  
;it will exceed 1.0 where class 1&2 is already allocated
```

```

IF prop EQ 0 THEN BEGIN
    R[*,*] = 0

ENDIF ELSE BEGIN

    order = SORT(R)
    index_1 = order[0: prop-1] ;select small number with proportion
    R[index_1] = 3 ;Allocate class 3

    index_0 = where(R NE 3, count) ;Allocate class 0 where class 1&2 s
allocated
    IF count NE 0 THEN R[index_0] = 0 ;To avoid the case of "where"
returns -1

ENDELSE

; Insert the alley R to the alley arr_bi3

arr_bi3[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] = R

ENDFOR
ENDFOR

.....
;;; STEP 3-3 .....
;;; Calculate attractiveness Oi at each sub-pixel
;;; (ITERATE STEP 3 and STEP 4 "iteration" times)
.....

;print,'step 3-3 calculate oi'
;print,'-----Iteration Start-----'

FOR i3=0, iteration3-1 DO BEGIN

.....
;; make 0 buffer around the boundary

    big_arr_oi = MAKE_ARRAY(sub_col+dx+dx, sub_row+dy+dy,/int)
    big_arr_oi[dx:dx+sub_col-1, dy:dy+sub_row-1] = arr_bi3

.....
;; Calculate Oi(the sum the neighboring pixel value) using moving window

```

big_arr_oi = SMOOTH(big_arr_oi/2.0, width)* width*width

arr_oi2 = big_arr_oi[dx:dx+sub_col-1, dy:dy+sub_row-1]

```
.....  
.....  
;;;; STEP 3-4;.....  
;;;; Find the Min Oi where arr_bi=3,  
;;;; Find the Max Oi where arr_bi=0  
;;;; AND swap the value if Min Oi < Max Oi  
.....  
.....
```

arr_bi_123= arr_bi_12 +arr_bi3

FOR x=0, n_col-1 DO BEGIN
FOR y=0, n_row-1 DO BEGIN

sub_bi = arr_bi_123[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] ;sub
pixel of arr_bi
sub_oi = arr_oi2[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] ;sub pixel
of arr_oi

IF ((MAX(sub_bi) EQ 3) AND (MIN(sub_bi) EQ 0)) THEN BEGIN ;DO Only where
both class 0&3 exist

bi_3 = where(sub_bi EQ 3) ;Find the place where class=3
min_oi = MIN(sub_oi[bi_3]) ;Find the minimum Oi value within class=3
min_bi_3 = where((sub_oi EQ min_oi) AND (sub_bi EQ 3)) ;Find the place
where oi=minimum within class=3

bi_0 = where(sub_bi EQ 0)
max_oi = MAX(sub_oi[bi_0])
max_bi_0 = where((sub_oi EQ max_oi) AND (sub_bi EQ 0))

IF (min_oi LT max_oi) THEN BEGIN

sub_bi[MIN(min_bi_3)] = 0
sub_bi[MAX(max_bi_0)] = 3

ENDIF

ENDIF

```
; Insert the sub_bi to the alley arr_bi_123
```

```
arr_bi_123[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] = sub_bi
```

```
ENDFOR
```

```
ENDFOR
```

```
index_bi_01 = where(arr_bi_123 NE 3);create new arr_bi3
```

```
arr_bi3 = arr_bi_123
```

```
arr_bi3[index_bi_01] = 0
```

```
ENDFOR
```

```
.....  
.....  
..... CLASS 4 START !!!!! .....  
.....  
.....
```

```
;print, '-----CLASS 4-----'
```

```
coarse_prop4 = original_prop(*,*,3)
```

```
arr_bi4 = arr_bi_123
```

```
.....  
.....  
..... STEP 4-2 .....  
..... Randomly allocate the class proportion (0/4) within a pixel  
..... Except where class 1&2&3 allocate  
.....
```

```
.....  
.....  
.....
```

```
;print, 'step 4-2, random allocation'
```

```
FOR x=0, n_col-1 DO BEGIN
```

```
FOR y=0, n_row-1 DO BEGIN
```

```
; Create zoom x zoom alley R, which randomly allocate 1 by the class proportion
```

```
prop = coarse_prop4[x,y] ;prop is the class proportion
```

```
R = RANDOMU(seed, zoom, zoom) ;make the random zoom*zoom matrix
```

```
R = R + arr_bi4[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)]
```

```
;it will exceed 1.0 where class 1&2&3 is already allocated
```

```

IF prop EQ 0 THEN BEGIN
    R[*,*] = 0

ENDIF ELSE BEGIN

    order = SORT(R)
    index_1 = order[0: prop-1] ;select small number with proportion
    R[index_1] = 4 ;Allocate class 4

    index_0 = where(R NE 4, count) ;Allocate class 0 where class 1&2&3 s
allocated
    IF count NE 0 THEN R[index_0] = 0 ;To avoid the case of "where"
returns -1

ENDELSE

; Insert the alley R to the alley arr_bi4

arr_bi4[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] = R

ENDFOR
ENDFOR

.....
;;; STEP 4-3 .....
;;; Calculate attractiveness Oi at each sub-pixel
;;; (ITERATE STEP 3 and STEP 4 "ieration" times)
.....

;print,'step 4-3 calculate oi'
;print,'-----Iteration Start-----'

FOR i4=0, iteration4-1 DO BEGIN

.....
;; make 0 buffer around the boundary

    big_arr_oi = MAKE_ARRAY(sub_col+dx+dx, sub_row+dy+dy,/int)
    big_arr_oi[dx:dx+sub_col-1, dy:dy+sub_row-1] = arr_bi4

.....
;; Calculate Oi(the sum the neighboring pixel value) using moving window

```

big_arr_oi = SMOOTH(big_arr_oi/2.0, width)* width*width

arr_oi2 = big_arr_oi[dx:dx+sub_col-1, dy:dy+sub_row-1]

```
.....  
.....  
..... STEP 4-4.....  
..... Find the Min Oi where arr_bi=4,  
..... Find the Max Oi where arr_bi=0  
..... AND swap the value if Min Oi < Max Oi  
.....  
.....
```

arr_bi_1234= arr_bi_123 + arr_bi4

FOR x=0, n_col-1 DO BEGIN
FOR y=0, n_row-1 DO BEGIN

sub_bi = arr_bi_1234[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] ;sub
pixel of arr_bi
sub_oi = arr_oi2[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] ;sub pixel
of arr_oi

IF ((MAX(sub_bi) EQ 4) AND (MIN(sub_bi) EQ 0)) THEN BEGIN ;DO Only where
both class 0&4 exist

bi_4 = where(sub_bi EQ 4) ;Find the place where class=4
min_oi = MIN(sub_oi[bi_4]) ;Find the minimum Oi value within class=4
min_bi_4 = where((sub_oi EQ min_oi) AND (sub_bi EQ 4)) ;Find the place
where oi=minimum within class=4

bi_0 = where(sub_bi EQ 0)
max_oi = MAX(sub_oi[bi_0])
max_bi_0 = where((sub_oi EQ max_oi) AND (sub_bi EQ 0))

IF (min_oi LT max_oi) THEN BEGIN

sub_bi[MIN(min_bi_4)] = 0
sub_bi[MAX(max_bi_0)] = 4

ENDIF

ENDIF

; Insert the sub_bi to the alley arr_bi_1234

```
arr_bi_1234[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] =  
sub_bi
```

```
ENDFOR  
ENDFOR
```

```
index_bi_01 = where(arr_bi_1234 NE 4);create new arr_bi4  
arr_bi4= arr_bi_1234  
arr_bi4[index_bi_01] = 0
```

```
ENDFOR
```

```
.....  
.....;ACCURACY ASSESSMENT (PCC);.....  
.....
```

```
pcc_map = truth - arr_bi_1234
```

```
correct = N_ELEMENTS(where(pcc_map EQ 0))
```

```
pcc = float(correct)/(sub_col*sub_row)*100
```

```
pcc_arr[test-1] = pcc
```

```
ENDFOR
```

```
var_result = VARIANCE(pcc_arr)
```

```
mean_result = MEAN(pcc_arr)
```

```
print, transpose(pcc_arr)
```

```
print, "variance:", var_result, ": mean:", mean_result
```

```
print, 'time :', (systemtime(1) - time)/60, ': Minutes'
```

```
end
```

Appendix D (IDL program for Simultaneous Categorical Swapping)

```

pro simulat
;-----
;
;
; Simultaneous Categorical Swapping (Sub-pixel Mapping for multiple classes imagery)
; Written by Yasuyo Makido
; Created Sept 2005 (Modified April 2006)
;
;
; Attractiveness Oi : Equal Weight function
;
;
; Reference:
;   Atkinson, P.M. (2005). Sub-pixel Target Mapping from Soft-classified,
;   Remotely Sensed Imagery. Photogrammetric Engineering & Remote Sensing:
;   71(7), 839-846
;-----

```

```

time = systime(1)

```

```

.....
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,, INPUT PARAMETER ;,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
.....

,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
;# of class
num_class = 12

,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
;the zoom factor (sub-pixels within 1 pixel)
zoom = 7

,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
; dx is the neighboring distance in sub-pixels (1:N=8, 2:N=24, 3:NN=48)
dx = 3
dy = 3

width = dx*2+1 ;3 for dx&dy=1, 5 for dx&dy=2

,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
; The number of iteration

iteration = 20

;
;   ite_arr = [1,5,10,20,30,40] ;For testing various iteration
; FOR ite_num=0, n_elements(ite_arr)-1 DO BEGIN
;

```

```

;           iteration = ite_arr[ite_num]
;           print, '-----'
;
;
;
;
;           FOR t=1, 10 DO BEGIN ;number of test

.....
;INPUT IMAGE ;
.....

;-----IMAGE (China classified)-----
;
;INPUT FILE (FINE RESOLUTION BINARY)
;
    fine1 = read_tiff('d:\Yasuyo\china\lc78\binary\china78_1.tif')
    fine2 = read_tiff('d:\Yasuyo\china\lc78\binary\china78_2.tif')
    fine3 = read_tiff('d:\Yasuyo\china\lc78\binary\china78_3.tif')
    fine4 = read_tiff('d:\Yasuyo\china\lc78\binary\china78_4.tif')
    fine5 = read_tiff('d:\Yasuyo\china\lc78\binary\china78_5.tif')
    fine6 = read_tiff('d:\Yasuyo\china\lc78\binary\china78_6.tif')
    fine7 = read_tiff('d:\Yasuyo\china\lc78\binary\china78_7.tif')
    fine8 = read_tiff('d:\Yasuyo\china\lc78\binary\china78_8.tif')
    fine9 = read_tiff('d:\Yasuyo\china\lc78\binary\china78_9.tif')
    fine10 = read_tiff('d:\Yasuyo\china\lc78\binary\china78_10.tif')
    fine11 = read_tiff('d:\Yasuyo\china\lc78\binary\china78_11.tif')
    fine12 = read_tiff('d:\Yasuyo\china\lc78\binary\china78_12.tif')

    truth = read_tiff('d:\Yasuyo\china\lc78\china78_truth.tif');groud truth file
;
;
;Check the size of the array

    f_col = N_ELEMENTS(fine1[*,1]) ;# of column for the fine image
    f_row = N_ELEMENTS(fine1[1,*]) ;# of row for fine image

    n_col = f_col/zoom
    n_row = f_row/zoom

;Stack up the original binay file

    fine = Make_array(f_col, f_row, num_class, /byte, VALUE = 0)

```

```

fine(*,*,0)=fine1
fine(*,*,1)=fine2
fine(*,*,2)=fine3
fine(*,*,3)=fine4
fine(*,*,4)=fine5
fine(*,*,5)=fine6
fine(*,*,6)=fine7
fine(*,*,7)=fine8
fine(*,*,8)=fine9
fine(*,*,9)=fine10
fine(*,*,10)=fine11
fine(*,*,11)=fine12

```

```

.....
;;;;;;;; Create a new coarse image (original_prop)
.....

```

```

original_prop = MAKE_ARRAY(n_col, n_row, num_class, /byte, VALUE = 0)

```

```

FOR class = 0, num_class-1 DO BEGIN

```

```

  FOR x=0, n_col-1 DO BEGIN
  FOR y=0, n_row-1 DO BEGIN

```

```

    sub_arr = fine[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1),class]
    original_prop[x,y,class] = TOTAL(sub_arr)

```

```

  ENDFOR
ENDFOR

```

```

ENDFOR

```

```

.....
;;;;;;;; Eliminate unnecessary arrays
.....

```

```

  fine1=0
  fine2=0
  fine3=0
  fine4=0
  fine5=0

```

```

fine6=0
fine7=0
fine8=0
fine9=0
fine10=0
fine11=0
fine12=0

```

```

fine =0

```

```

.....
;;; STEP 0 .....
;;; TEST THE coarse image (check the sum of the proportion = zoom*zoom)
.....

```

```

test = MAKE_ARRAY(n_col, n_row, /BYTE , VALUE = 0)

```

```

FOR z=0, num_class-1 DO BEGIN

```

```

    test = test + original_prop(*,*,z)
    proportion = total(original_prop(*,*,z))

```

```

ENDFOR

```

```

IF max(test) NE min(test) THEN print, "WARNING!!! CLASS PROPORTION MAY
WRONG!!"

```

```

.....
;;; STEP 1 .....
;;; Zoom up the number of pixels
.....

```

```

sub_col = n_col * zoom ; # of column for fine image
sub_row = n_row * zoom ; # of row for fine image

```

```

arr_multi = MAKE_ARRAY(sub_col, sub_row, /BYTE , VALUE = 0)

```

```

;print, '----- Random allocation -----'

```

```

.....
;;; STEP 2 .....
;;; Randomly allocate the class proportion within a pixel

```

```
.....  
.....
```

```
dummy = arr_multi
```

```
FOR z=0, num_class-2 DO BEGIN
```

```
FOR x=0, n_col-1 DO BEGIN
```

```
FOR y=0, n_row-1 DO BEGIN
```

```
    ; Create zoom x zoom alley R, which randomly allocate class number by the class  
    proportion
```

```
    prop = original_prop[x,y,z]
```

```
    R = RANDOMU(seed, zoom, zoom) ;make the random zoom*zoom matrix  
    R = R + dummy[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)]
```

```
    IF prop EQ 0 THEN BEGIN
```

```
        R[*,*] = 0
```

```
    ENDIF ELSE BEGIN
```

```
        order = SORT(R) ;index # at ascending order
```

```
        index_c = order[0: prop-1] ;get the index # for the number of prop
```

```
        R[index_c] = z+1
```

```
        index_0 = where(R NE z+1, count)
```

```
        IF count NE 0 THEN R[index_0] = 0 ;To avoid the case of "where"
```

```
returns -1
```

```
    ENDELSE
```

```
    ; Insert the alley R to the alley arr_bi
```

```
    dummy[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] = R
```

```
ENDFOR
```

```
ENDFOR
```

```
    ; print, '-----'  
    arr_multi = arr_multi + dummy
```



```

ENDIF

big_bi_arr[dx:dx+sub_col-1, dy:dy+sub_row-1,class-1] = temp
;make 0 buffer around the boundary

temp[*,*] = 0

ENDFOR

.....
,,,,, Eliminate unnecessary arrays
.....

temp = 0

.....
STEP 4 .....
,,,,, Calculate attractiveness Oi at each class
,,,,, EQUAL WEIGHT FUNCTION .....
.....

.....
;;; Calculate Oi(the sum of the neighboring pixel value) using moving window
;;; SMOOTH function returns a copy of Array smoothed with a boxcar average of the
specified width
;;; DOES not work well for multiple layers -> DO separately

FOR class = 0, num_class-1 DO BEGIN

big_bi_arr[*,* ,class] = byte(SMOOTH(float(big_bi_arr[*,* ,class]),width)* width*width)

ENDFOR

oi_arr = big_bi_arr[dx:dx+sub_col-1, dy:dy+sub_row-1, *]

.....
,,,,, Eliminate unnecessary arrays
.....

big_bi_arr = 0

.....
STEP 5 .....
,,,,, Start swapping between the class which is in need most
.....

```



```
FOR class=0, num_class-1 DO BEGIN
```

```
result[7,class] = class
```

```
temp_o = sub_oi_o[*,* ,class] ;create a temporary array
```

```
index_n0 = where(temp_o NE 0, count)
```

```
IF count NE 0 THEN BEGIN
```

```
oi_oa_min = Min(temp_o[index_n0]) ;oi_oa_min is minimum Oi value
```

```
locx = where(temp_o EQ oi_oa_min)
```

```
.....  
;If there're more than one number, pick one randomly!!  
;; Pick one from vector locx  
;;  
;;
```

```
num=N_ELEMENTS(locx)
```

```
random_num= FLOOR(RANDOMU(seed, 1)*num) ;create a random
```

```
integer(0~num-1)
```

```
locx= locx(random_num)
```

```
.....  
;.....
```

```
result[0,class] = oi_oa_min
```

```
result[8,class] = locx
```

```
;; 2. Find the MAXIMUM Oi value at class A, Unoccupied by class A (LOCATION Y)
```

```
temp_u= sub_oi_u[*,* ,class] ;create a temporary array
```

```
oi_ua_max = Max(temp_u) ;oi_ua_max is maximum Oi value
```

```
locY = where(temp_u EQ oi_ua_max)
```

```
.....  
;If there're more than one number, pick one randomly!!  
;; Pick one from vector locY
```

```
num=N_ELEMENTS(locY)
```

```

        random_num= FLOOR(RANDOMU(seed, 1)*num) ;create a random
integer(0~num-1)
        locY= locY(random_num)

```

```

.....
;

```

```

        result[1,class] = oi_ua_max
        result[10,class] = locY

```

```

;; 2' Find the CLASS (B) which currently occupied location Y (check the random_arr)

```

```

        class_b = sub_arr[locY]-1
        result[9,class] = class_b

```

```

;; 3. Find the Oi value at class B, occupied by class B at location Y

```

```

        temp_o = sub_oi_o[*,* , class_b]
        oi_ob = temp_o[locY]

        result[2,class] = oi_ob

```

```

;; 4. Find the Oi value at class B, Unoccupied by class B at location X

```

```

        temp_u = sub_oi_u[*,* , class_b]
        oi_ub = temp_u[locx]

        result[3,class] = oi_ub

```

```

;; .....Complete the result array;.....

```

```

        result[4,class] = result[1,class] - result[2,class]
        result[5,class] = result[3,class] - result[0,class]
        result[6,class] = result[4,class] + result[5,class]

```

```

        ENDIF

```

```

    ENDFOR ;END of the class

```

```

.....
;.....DECISION RULE OF SWAPPING START ;.....
.....

```

```

        largest6 = max(result[6,*]) ;Largest value of index 6

```

```

IF largest6 GT 0 THEN BEGIN
    candidate1 = where(result[6, *] EQ largest6)

    .....
    ;;;;METHOD A: Pick one cadidate Randomly
    ;;;

        num=N_ELEMENTS(candidate1)
        random_num= FLOOR(RANDOMU(seed, 1)*num) ;create a
random integer(0~num-1)
        swap_class = candidate1(random_num)

;           print, 'num=', num
;           print, 'random_num=', random_num
;           print, 'randomly picked candidate1 =', swap_class

    .....
    ;;;;METHOD B: Pick one using more restricted selection
    ;;;;      (More Logical, but not produce better result)
    ;;;;
    ;;;;

;           IF count EQ 1 THEN BEGIN
;               swapp_class = candidate1
;               print, ' come to here1'
;
;           ENDIF ELSE BEGIN
;
;               largest4 = max(result[4,candidate1]) ;get the largest4
VALUE within a candidate
;               candidate2 = where(result[4, *] EQ largest4) ;get the
largest4 INDEX within a candidate
;
;               swap_class = min(candidate2)
;
;               print, 'largest4 (value of largest 4 value)', largest4
;               print, ' candidate2 (index of the largset 4 index within
candidate)', candidate2
;               print, 'swap_class is', swap_class
;
;           ENDELSE

    .....

class_a= result[7, swap_class]+1
locX = result[8, swap_class]

```

```

class_b= result[9, swap_class]+1
locY = result[10, swap_class]

sub_arr[locX] = class_b
sub_arr[locY] = class_a

changeable_arr[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-
1)] = sub_arr

ENDIF

ENDFOR
ENDFOR

.....
,,,,, Eliminate unnecessary arrays
.....

bi_arr = 0
oi_arr = 0

ENDFOR ; END OF iteration

.....
,,,,, ACCURACY .....
.....

; print, '-----'
pcc = truth - changeable_arr
correct = N_ELEMENTS(where(pcc EQ 0))
print, 'PCC (%): ', float(correct)/(sub_col*sub_row)*100, ': iteration :', iteration

; ENDFOR ;number of test
; ENDFOR ;various iteration test

,,,,, Write tiff resulting file,,,,,

; WRITE_TIFF, 'd:\yasuyo\china\lc78\output\simult_78.tif, changeable_arr

print, 'time =', (systemtime(1) - time)/60, ' Minutes'

end

```

Appendix E (IDL program for Simulated Annealing)

pro sa

```
-----  
;  
;  
; Simulated annealing (Sub-pixel Mapping for multiple classes imagery)  
; Written by Yasuyo Makido  
; Created October 2005 (Modified January 2006)  
;  
; Attractiveness : Equal Weight function  
;  
; Reference:  
;   Atkinson, P.M. (2005). Sub-pixel Target Mapping from Soft-classified,  
;   Remotely Sensed Imagery. Photogrammetric Engineering & Remote Sensing:  
;   71(7), 839-846  
;  
;   Goodchild, M.A. (1986). Spatial Autocorrelation, CATMOG 47, Norwich, UK:  
;   Geo Books.  
;  
;   Goovaerts, P. (1997). Geostatistics for Natural Resources Evaluation. NY: Oxford  
;   University Press.  
;  
-----
```

```
time = systime(1)
```

```
FOR trial=0, 14 DO BEGIN
```

```
.....  
;# of class
```

```
    num_class = 5
```

```
.....  
;the zoom factor (sub-pixels within 1 pixel)
```

```
    zoom = 7
```

```
.....  
; dx is the neighboring distance in sub-pixels (1:N=8, 2:N=24, 3:NN=48)
```

```
    dx = 3
```

```
    dy = 3
```

```
    width = dx*2+1 ;3 for dx&dy=1, 5 for dx&dy=2
```

```
.....  
; SELECT EITHER NEIGHBORS for Moran's I
```

```

w_matrix = [[0,1,0],[1,0,1],[0,1,0]] ;4 neighbors (faster)
;w_matrix = [[0.7,1,0.7],[1,0,1],[0.7,1,0.7]] ;8 neighbors

; THE Max number of repetition

rep_max = 8000000

; rep_arr = [7000000, 8000000, 9000000, 10000000, 11000000, 12000000]
;
; FOR ite_num=0, n_elements(rep_arr)-1 DO BEGIN
;
; rep_max = rep_arr[ite_num]

; THE number of trial for swapping

num_try = ULONG(0.01* rep_max)
; The target Moran's I

target_i = 0.661 ;original(0.661).m3(0.808),m5(0.866), m7(0.894)

;-----IMAGE C-----
;:INPUT FILE (FINE RESOLUTION BINARY)

fine1 = read_tiff('d:\Yasuyo\Atkinson\msu\original\binary\msu_1.tif');fine image for
class1
fine2 = read_tiff('d:\Yasuyo\Atkinson\msu\original\binary\msu_2.tif');fine image for
class2
fine3 = read_tiff('d:\Yasuyo\Atkinson\msu\original\binary\msu_3.tif');fine image for
class3
fine4 = read_tiff('d:\Yasuyo\Atkinson\msu\original\binary\msu_4.tif');fine image for
class4
fine5 = read_tiff('d:\Yasuyo\Atkinson\msu\original\binary\msu_5.tif');fine image for
class5
truth = read_tiff('d:\Yasuyo\Atkinson\msu\original\msu_truth.tif');groud truth msu origi

;:Check the size of the array

f_col = N_ELEMENTS(fine1[*],1) ;# of column for the fine image
f_row = N_ELEMENTS(fine1[1,*]) ;# of row for fine image

```

```

n_col = f_col/zoom
n_row = f_row/zoom

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;Stack up the original binay file

fine = Make_array(f_col, f_row, num_class, /byte, VALUE = 0)

fine(*,*,0)=fine1
fine(*,*,1)=fine2
fine(*,*,2)=fine3
fine(*,*,3)=fine4
fine(*,*,4)=fine5

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;; Create a new coarse image (original_prop)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

original_prop = MAKE_ARRAY(n_col, n_row, num_class, /byte, VALUE = 0)

FOR class = 0, num_class-1 DO BEGIN

FOR x=0, n_col-1 DO BEGIN
FOR y=0, n_row-1 DO BEGIN

sub_arr = fine[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-
1),class]
original_prop[x,y,class] = TOTAL(sub_arr)

ENDFOR
ENDFOR

ENDFOR

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;; STEP 0
;;;;; TEST THE coarse image (check the sum of the proportion = zoom*zoom)
;;;;; Create table which contain the number of the classes(for Moran's I)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

test = MAKE_ARRAY(n_col, n_row, /BYTE , VALUE = 0)

```

```
Moran_arr = MAKE_ARRAY(num_class,2, value = 0.0)
```

```
FOR z=0, num_class-1 DO BEGIN
```

```
    test = test + original_prop(*,*,z)
    proportion = total(original_prop(*,*,z))
```

```
    Moran_arr[z, 0] = proportion
```

```
ENDFOR
```

```
.....
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; STEP 1 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; Zoom up the number of pixels
.....
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
sub_col = n_col * zoom ; # of column for fine image
sub_row = n_row * zoom ; # of row for fine image
```

```
big_col = sub_col+zoom+zoom ;Use for MoranI
big_row = sub_row+zoom+zoom ;Use for MoranI
```

```
arr_multi = MAKE_ARRAY(sub_col, sub_row, /BYTE , VALUE = 0)
```

```
;print, '----- Random allocation -----'
```

```
.....
;;; STEP 2 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; Randomly allocate the class proportion within a pixel
.....
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
dummy = arr_multi
```

```
FOR z=0, num_class-2 DO BEGIN
```

```
FOR x=0, n_col-1 DO BEGIN
```

```

FOR y=0, n_row-1 DO BEGIN

    ; Create zoom x zoom alley R, which randomly allocate class number by the class
    proportion

    prop = original_prop[x,y,z]

    R = RANDOMU(seed, zoom, zoom) ;make the random zoom*zoom matrix
    R = R + dummy[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)]

    IF prop EQ 0 THEN BEGIN
        R[*,*] = 0

    ENDF ELSE BEGIN

        order = SORT(R) ;index # at ascending order
        index_c = order[0: prop-1] ;get the index #. which has smallest

        R[index_c] = z+1

        index_0 = where(R NE z+1, count)
        IF count NE 0 THEN R[index_0] = 0 ;To avoid the case of "where"
returns -1

    ENDELSE

    ; Insert the alley R to the alley arr_bi

    dummy[(x*zoom):(x*zoom+zoom-1), (y*zoom):(y*zoom+zoom-1)] = R

ENDFOR
ENDFOR

;    print, '-----'
    arr_multi = arr_multi + dummy

    dummy = arr_multi * 10 ;Multiply by 10 where the class is already allocated

ENDFOR

index_0 = where(arr_multi EQ 0, count)
IF count NE 0 THEN arr_multi[index_0] = num_class
;

```



```

.....
;; make 0 buffer around the boundary

big_bi_arr = MAKE_ARRAY(sub_col+dx+dx, sub_row+dy+dy, num_class,
/BYTE, VALUE =0)
big_bi_arr[dx:dx+sub_col-1, dy:dy+sub_row-1,*] = bi_arr

.....
;;; Calculate Oi(the sum of the neighboring pixel value) using moving window
;;; SMOOTH function returns a copy of Array smoothed with a boxcar average
of the specified width

FOR class = 0, num_class-1 DO BEGIN

    big_bi_arr[*,*,class] =
byte(SMOOTH(float(big_bi_arr[*,*,class]),width)* width*width)

ENDFOR

oi_arr = big_bi_arr[dx:dx+sub_col-1, dy:dy+sub_row-1, *]

count_swap = 0U ;The # of swapping within num_try trial

.....
;;ITERATION Start !!
    ;repeat num_try(1% of max iteration) times without recalcurating the oi
value

=====
FOR unchange = 1UL, num_try DO BEGIN

.....
;;; B Find the pair (within a pixel).....
.....

;Pick two random number for x coordinate & y coordinate

rand = RANDOMU(seed, 4)

loc_xi = FLOOR(rand[0] * sub_col) ; index xi at entire image
loc_yi = FLOOR(rand[1] * sub_row) ; index yi at entire image

```

;Find the location of the sub pixel

sub_x = loc_xi/zoom ; location of the sub x area
sub_y = loc_yi/zoom ; location of the sub y area

rand_x = FLOOR(rand[2] * zoom)
rand_y = FLOOR(rand[3] * zoom)

loc_xj = sub_x * zoom + rand_x ; index xj at entire image
loc_yj = sub_y * zoom + rand_y ; index xj at entire image

.....
.....
;;; C. Swap based on the table ;;;
.....
.....

class_a = changeable_arr[loc_xi,loc_yi] ;class # at loc i
class_b = changeable_arr[loc_xj,loc_yj] ;class # at loc j

oi_a = oi_arr[loc_xi, loc_yi, class_a-1] ;oi value at array oi_a at loc i
oi_b = oi_arr[loc_xj, loc_yj, class_a-1] ;oi value at array oi_a at loc j

oj_a = oi_arr[loc_xi, loc_yi, class_b-1] ;oi value at array oi_b at loc i
oj_b = oi_arr[loc_xj, loc_yj, class_b-1] ;oi value at array oi_b at loc j

.....
.....
;;;DECISION MAKING TIME; ;;;
.....
.....

comp = (fix(oi_b) - fix(oi_a)) + (fix(oj_a) - fix(oj_b))
; the larger (fix(oi_b) - fix(oi_a)) value, the better for swapping (to
increase contiguity)
; the larger (fix(oj_a) - fix(oj_b)) value, the better for swapping (to
increase contiguity)

IF (comp GT 0) THEN BEGIN

changeable_arr[loc_xi, loc_yi] = class_b
changeable_arr[loc_xj, loc_yj] = class_a

count_swap = count_swap + 1

ENDIF

```

ENDFOR ;END OF The num_try
;=====

num_repeat = num_repeat + num_try

;ENDWHILE ;CHOICE A: Use this when maximizing the I(=Do max_repeat)
;-----

;

.....
;;;; D. after repeat B&C, Calculate Moran's I,
.....

;;;;; Check the size of the array;;;;;

inside_pixel = (sub_col-2)*(sub_row-2)

;;;;; Change mutil class array to the binary multilayer array ;;;;;;

bi_arr = Make_array(sub_col, sub_row, num_class, /BYTE , VALUE = 0)

FOR class = 1, num_class DO BEGIN

temp = Make_array(sub_col, sub_row, /BYTE , VALUE = 0) ;creat empty
array

index_1 = where(changeable_arr EQ class, count) ;if that class -> asisgn 1
IF count NE 0 THEN BEGIN
temp[index_1]= 1
bi_arr[* , *,class-1] = temp

ENDIF

ENDFOR

.....
;;;;;AVOID the bordering pixels;;;;;

```

```
;;;;;;Iteration (num_class) start
```

```
FOR class = 0, num_class-1 DO BEGIN
```

```
    test = bi_arr[*,*,class]
```

```
    inside_arr = test[1:sub_col-2, 1:sub_row-2]
```

```
    ave_z = MEAN(inside_arr) ;ave_z = mean attribute inside the boundary
```

```
;;;;;; Calculate C and W value ;;;;;;;;;;;;;;;;;;
```

```
;;;;;; ---Inside Boundary-- ;;;;;;;;;;;;;;;;;;
```

```
    cw_sum = 0.0
```

```
    w_sum = 0.0
```

```
    FOR r=1, sub_row-2 DO BEGIN ;avoide border
```

```
    FOR c=1, sub_col-2 DO BEGIN ;avoide border
```

```
        cr_ave = test[c,r] - ave_z
```

```
        ; clip the 3x3 pixels
```

```
        clip = test[c-1:c+1, r-1:r+1]
```

```
        cw = cr_ave * (clip-ave_z)
```

```
        cww = cw * w_matrix
```

```
        cww_total = TOTAL(cww)
```

```
        w_total = TOTAL(w_matrix)
```

```
        cw_sum = cw_sum + cww_total
```

```
        w_sum = w_sum + w_total
```

```
    ENDFOR
```

```
ENDFOR
```



```
;ENDFOR ; for rep_max
```

```
    print, 'time =', (systemtime(1) - time)/60, ' Minutes'
```

```
end
```

REFERENCES

- Aplin, P., Atkinson, P.M., and Curran, P.J. (1999). Fine spatial resolution simulated satellite imagery for land cover mapping in the United Kingdom. Remote Sensing of Environment: 68, 206-216.
- Atkinson, P.M., and Curran, P.J. (1995). Defining an Optimal Size of Support for Remote Sensing Investigations. IEEE Transactions on Geoscience and Remote Sensing: 33(3), 768-776.
- Atkinson, P.M. (1997). Mapping sub-pixel boundaries from remote sensed images. In Innovations in GIS 4 (Z.Kemp, Eds.), London: Taylor & Francis, 166-180.
- Atkinson, P.M., Cutler, M.E.J., and Lewis, H. (1997). Mapping sub-pixel proportional land cover with AVHRR imagery. International Journal of Remote Sensing: 18(4), 917-935
- Atkinson, P.M., and Tate, N.J. (2000). Spatial scale problems and geostatistical solutions: a review. Professional Geographer: 52, 607-623.
- Atkinson, P.M. (2001). Super-resolution target mapping from soft-classified remotely sensed imagery. Online Proceedings of the 6th International Conference on GeoComputation, 24-26 September 2001, Brisbane, Australia, URL: <http://www.geocomputation.org/2001/papers/atkinson.pdf> (last date accessed: 23 October 2006).
- Atkinson, P.M. (2005). Sub-pixel Target Mapping from Soft-classified, Remotely Sensed Imagery. Photogrammetric Engineering & Remote Sensing: 71(7), 839-846
- Bailey, T.C., and Gatrell, A.C. (1995). Interactive spatial data analysis. Harlow Essex, England: Longman Scientific & Technical.
- Boucher, A. and Kyriakidis, P.C. (2006). Super-Resolution Land Cover Mapping with Indicator Geostatistics. Remote Sensing of Environment: 104(3), 264-282.
- Cao, C. and Lam, N.S. (1997). Understanding the Scale and Resolution Effects in Remote Sensing and GIS, In Scale in Remote Sensing and GIS (D.A.Quattrochi and M.F.Goodchild, Eds), NY: Lewis publishers, 57-72.
- Cardille, J.A. and Turner, M.G. (2002). Understanding Landscape Metrics I. In Learning Landscape Ecology: A practical guide to concepts and techniques, (S.E. Gergel and M.G. Turner, Eds), NY: Springer, 85-100.

- Curran, P.J., and Atkinson, P.M.(1998). Geostatistics and remote sensing. Progress in Physical Geography: 22, 61-78.
- Congalton, R.G., and Green, K. (1999). Assessing the Accuracy of Remote Sensed Data: Principles and Practices. NY: Lewis Publishers.
- ERDAS (2002). Erdas Field Guide. 6th ed. Erdas Inc.
- ESRI (2005). Environmental System Research Institute, Inc, ARC/INFO user's guide 9.1. ESRI Press.
- Fisher, P. (1997). The pixel: a snare and a delusion. International Journal of Remote Sensing:18(3), 679-685.
- Foody, G.M., and Cox, D.P. (1994). Sub-pixel land cover composition estimation using a linear mixture model and fuzzy membership functions. International Journal of Remote Sensing :15(3), 619-631.
- Foody, G.M. (1998). Sharpening fuzzy classification output to refine the representation of sub-pixel land cover distribution. International Journal of Remote Sensing: 19(13), 2593-2599.
- Foody, G.M. (2002). Status of land cover classification accuracy assessment. Remote Sensing of Environment: 80,185-201.
- Forman, R.T.T. and Godron, M. (1986). Landscape Ecology. NY: John Wiley and Sons, Inc.
- Friedl, M.A., McGwire, K.C., and McIver, D.K. (2001). An Overview of Uncertainty in Optical Remotely Sensed Data for Ecological Applications. In Spatial uncertainty in ecology : implications for remote sensing and GIS application (C.T. Hunsaker, M.F. Goodchild, M.A. Friedl and T.J.Case, Eds), NY: Springer, 258-283.
- Gavin, J. and Jennison, C. (1997). A subpixel image restoration algorithm. Journal of Computational and Graphical Statistics: 6, 182-201.
- Goodchild, M.A. (1986). Spatial Autocorrelation, CATMOG 47, Norwich, UK: Geo Books.
- Goovaerts, P. (1997). Geostatistics for Natural Resources Evaluation. NY: Oxford University Press.
- Greenberg, J.D., Gergel, S.E, and Turner, M.G. (2002). Understanding Landscape Metrics II, In Learning Landscape Ecology: A practical guide to concepts and techniques, (S.E. Gergel and M.G. Turner, Eds), NY: Springer. 101-111.

- Haines-Young, R., and Chopping, M. (1996). Quantifying landscape structure: a review of landscape indices and their application to forested landscapes. Progress in Physical Geography: 20(4), 418-445.
- Hlavka, C.A., and Livingston, G.P. (1997). Statistical models of fragmented land cover and the effect of coarse spatial resolution on the estimation of area with satellite sensor imagery. International Journal of Remote Sensing: 18(10), 2253-2259
- Justice, C.O., and Townshend, J.R.G. (1981). Integrating ground data with remote sensing, Terrain Analysis and Remote Sensing (J.R.G.Townshend, Ed), London: Allen and Unwin, 38-101.
- Kanellopoulos, I., A. Varfis, G. G. Wilkinson, and J. Megier. (1992). Land cover discrimination in SPOT imagery by artificial neural network-a twenty class experiment. International Journal of Remote Sensing: 13(5), 917-924.
- Kasetkasem, T., Arora, M.K., and Varshney, P.K. (2005). Super-resolution land cover mapping using a Markov random field based approach. Remote Sensing of Environment: 96, 302-314.
- Kerdiles, H., and Grondona, M. O. (1995). NOAA- AVHRR NDVI decomposition and subpixel classification using linear mixing in the Argentinean Pampa. International Journal of Remote Sensing: 16(7),1303-1325.
- Lillesand, T.M., and Kiefer, R.W., and Chipman, J.W. (2004). Remote sensing and image interpretation, 5th edition. NY: John Wiley & Sons.
- Makido, Y and Qi, J. (2005) Land Cover Mapping at Sub-Pixel Scales: Using Remote Sensing Imagery. International Geoscience and Remote Sensing Symposium (IGARSS) 2005. Seoul, Korea, July 25-29.
- Makido, Y and Shortridge, A.M. (2007) Weighting function alternatives for a sub-pixel allocation model. Photogrammetric Engineering and Remote Sensing: In Press.
- Matricardi, E.A.T., Skole, D.L., Cochrane, M.A., Qi, J., and Chomentowski, W.(2005). Monitoring Selective Logging in Tropical Evergreen Forests Using Landsat: Multitemporal Regional Analysis in Mato Grosso, Brazil. Earth Interactions: 9(24), 1-24.
- McKelvey, K.S., and Noon, B.R. (2001). Incorporating Uncertainties in Animal Location and Map Classification into Habitat Relationships Modeling. Spatial uncertainty in ecology : implications for remote sensing and GIS applications, (C.T. Hunsaker, M.F. Goodchild, M.A. Friedl and T.J.Case, Eds), NY: Springer, 72-90.

- Messina, J.P., Walsh, S.J., Mena, C.F., and Dlamater, P.L. (2006a). Land tenure and deforestation patterns in the Ecuadorian Amazon: Conflicts in land conservation in frontier settings. Applied Geography: 26, 113-128.
- Messina, J.P., Delamater, P.L., Hupy C.M., and Makido, Y. (2006b). (under review) Neutral Models and the Deviation from Neutral Pattern Metric. Ecological Informatics.
- Mertens, K.C., Verbeke, L.P.C., Ducheyne, E.I., and De Wulf, R.R. (2003). Using generic algorithms in sub-pixel mapping. International Journal of Remote Sensing: 14(21), 4241-4247.
- O'Neill, R.V., and Smith, M.A. (2002). Scale and Hierarchy Theory, In Learning Landscape Ecology: A practical guide to concepts and techniques, (S.E. Gergel and M.G. Turner, Eds), NY: Springer, 3-8
- Pelkey, N.W., Stoner, C.J., and Caro, T.M. (2003). Assessing habitat protection regimes in Tanzania using AVHRR NDVI composites: comparisons at different spatial and temporal scales. International Journal of Remote Sensing: 24(12), 2533-2558.
- Riitters, K.H., O'Neill, R.V., Hunsaker, C.T., Wickham, J.D., Yankee, D.H., Timmins, S.P., Jones, K.B., Jackson, B.L., (1995). A factor analysis of landscape pattern and structure metrics. Landscape Ecology:10, 23-39
- Tatem, A.J., Lewis, H.G., Atkinson, P.M., and Nixon, M.S. (2001). Super-resolution target identification from remotely sensed images using a Hopfield neural network. IEEE Transactions on Geoscience and Remote Sensing: 39, 781-796.
- Tatem, A.J., Lewis, H.G., Atkinson, P.M., and Nixon, M.S. (2002). Super-resolution land cover pattern prediction using a Hopfield neural network. Remote Sensing of Environment: 79, 1-14.
- Tatem, A.J., Lewis, H.G., Atkinson, P.M., and Nixon, M.S. (2003). Increasing the spatial resolution of agricultural land cover maps using a Hopfield neural network, International Journal of Geographical Information Science: 17(7), 647-672.
- Tobler, W.R. (1988). Resolution, resampling and all that, Building Databases for Global Science (H. Mounsey, Ed), NY: Taylor & Francis, 129.
- Verhoeve, J., and De Wulf, R. (2002). Land cover mapping at sub-pixel scales using linear optimization techniques. Remote Sensing of Environment: 79, 96-104.
- Wang, C., Qi, J., and Cochrane, M. (2005). Assessment of Tropical Forest Degradation with Canopy Fractional Cover from Landsat ETM+ and IKONOS imagery. Earth Interactions: 9(22), 1-18.

Woodcock, C.E., and Strahler, A.H. (1987). The factor of scale in remote sensing.
Remote Sensing of Environment, 21, 311-322.

Zhang, Y., Wu, J., and Qi. J. (2006). (under review) A classification method based on spectral angle mapping to MSS remote sensing image for mapping land cover.
Remote Sensing of Environment.

Zhejiang china. (2003). www.zhejiang.gov.cn

MICHIGAN STATE UNIVERSITY LIBRARIES



3 1293 02845 6808