

**AUTOMATED SEGMENTATION OF URBAN FEATURES
FROM LANDSAT THEMATIC MAPPER IMAGERY
FOR USE IN PSEUDOINVARIANT FEATURE
TEMPORAL IMAGE NORMALIZATION**

by

Carl Salvaggio

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in the Center for Imaging Science in the
College of Graphic Arts and Photography of the
Rochester Institute of Technology

May 1987

Signature of the Author Carl Savaggio

Accepted by Name Illegible
Coordinator, M.S. Degree Program

Center for Imaging Science
College of Graphic Arts and Photography
Rochester Institute of Technology
Rochester, New York

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. degree thesis of Carl Salvaggio
has been examined and approved by the thesis
committee as satisfactory for the thesis requirement
for the Master of Science degree

Dr. John R. Schott, Thesis Advisor

Dr. Roger Easton

Mr. Peter G. Engeldrum

5/26/87
Date

**AUTOMATED SEGMENTATION OF URBAN FEATURES
FROM LANDSAT THEMATIC MAPPER IMAGERY
FOR USE IN PSEUDOINVARIANT FEATURE
TEMPORAL IMAGE NORMALIZATION**

by

Carl Salvaggio

Submitted to the Center for Imaging Science in
partial fulfillment of the requirements for the
Master of Science degree at the
Rochester Institute of Technology

ABSTRACT

An automated segmentation algorithm for the isolation of pseudoinvariant features was developed. This algorithm utilizes rate-of-change information from the thresholding process previously associated with the pseudoinvariant feature normalization technique. This algorithm was combined with the normalization technique and applied to the six reflective bands of the Landsat Thematic Mapper for both urban and rural imagery. The segmentation algorithm and normalization technique were also applied to color infrared high resolution U2 imagery. The accuracy and precision of the normalization results were evaluated. The technique consistently produced normalization results with errors of approximately one or two reflectance units for both the rural and urban Thematic Mapper imagery as well as the visible bands of high resolution airphoto imagery. The segmentation algorithm shows great potential for the removal of human intervention in the pseudoinvariant feature temporal image normalization process.

ACKNOWLEDGEMENTS

The author wishes to extend great appreciation to Dr. John R. Schott for his support and guidance in this effort. His insights proved to be invaluable in this study. The freedom and advice he provided for the author to pursue methods and techniques that proved to be little more than academic exercises provided the author with the confidence that was needed to complete a successful study as well as develop a sound scientific attitude. The author also wishes to thank Steven L. Schultz for the many hours of uncompensated assistance in the coding of the algorithms described in this study. His assistance has proved to be of unparalleled importance in the development of the author's computer programming skills. Finally, the author wishes to thank Dr. Roger Easton for the hours he has spent in review of the form and grammar of this work. His time has provided the author with an entirely new view of scientific writing. Appreciation is also extended to the rest of the members of the Digital Image and Remote Sensing Laboratory (DIRS) for their cooperation and assistance in all aspects of this study as well as to the remaining members of the thesis committee for the time and effort that these scientists have provided.

DEDICATION

This thesis is dedicated to my wife Nanette for the moral and academic support that she has provided. Without her undying dedication and understanding of my goals and dreams, I could not have hoped to have the confidence and drive needed to complete such a study. I would also like to dedicate this thesis to my mother, Eva Helene Batari, for both her moral and financial support throughout my years at this school and before. Her great confidence and unrestraining relationship with me have always proven to be a great gift.

Table of Contents

Table of Contents	i
List of Tables	iii
List of Figures	vi
1.0 Introduction	1
1.1 Historical Background	3
1.1.1 Temporal Image Normalization	3
1.1.2 Pseudoinvariant Feature Normalization	5
1.1.3 Quantification of Normalization Results	7
1.1.4 Spectral signatures and Band Ratioing for Classification and Segmentation	7
1.1.5 Supervised Multivariate Classification	14
1.1.6 Unsupervised Multivariate Classification	20
1.1.7 Preprocessing and Redundancy Reduction	24
1.2 Theoretical Background	27
1.2.1 PIF Technique for Image Normalization	27
1.2.2 Derivation of the PIF Normalization Transforms	32
1.2.3 Automated PIF Segmentation Using Multivariate Techniques	39
1.2.4 Automated PIF Segmentation Using Rate of Change Techniques	45
2.0 Experimental Approach	52
2.1 Selection of Appropriate Imagery	53
2.2 Rate of Change Segmentation Algorithm	56
2.3 PIF Transform Determination	62
2.4 Accuracy of the PIF Transformations	66
2.5 Precision of the Normalization Process	70
2.6 Application of Segmentation Algorithm to High Resolution Imagery	73

Table of Contents (con't)

3.0	Results and Discussion	75
3.1	Segmentation of PIF Features	76
3.2	Development of PIF Transforms	92
3.3	Evaluation of Transforms Using Control Point Analysis	101
3.4	Evaluation of Robustness Using Rural Image Data	116
4.0	Conclusions and Recommendations	125
	References	129
Appendix A	Description of the Principal Components Computer Code	A-1
Appendix B	Description of the K-Means Unsupervised Clustering Code	B-1
Appendix C	Description of the Automated Rate of Change Segmentation Algorithm Code	C-1
Appendix D	Description of the PIF Normalization Code	D-1
Appendix E	Description of the Code Used to Read Data Off of TM Computer Compatible Tape	E-1
Appendix F	Summary of Digital Count Data and Reflectance Conversion Data Used in the Control Point Analysis	F-1
Appendix G	Summary of the Results Obtained Utilizing the Multivariate Segmentation Algorithm	G-1
Appendix H	Description of the Array Processor Based Image Utility Subroutines Used in the Major Programs	H-1
Appendix I	Description Non-Linearity Problems Encountered When Digitizing Photographic Transparencies	I-1
	Vita	

List of Tables

Table 1	Summary of Pertinent TM Image Data Used in This Study	55
Table 2	Summary of Threshold Values Obtained for the Segmentation of Urban Features Using the Automated and Interactive Segmentation Algorithms	77
Table 3	Histogram Statistics Utilized to Determine the Thresholding Ranges Used for the Automated Rate of Change Segmentation Algorithm	83
Table 4	Summary of Temporal PIF Image and Transformation Data for the TM Urban Rochester Imagery	94
Table 5	Summary of Temporal PIF Image and Transformation Data for the TM Urban Buffalo Imagery	95
Table 6	Summary of Temporal PIF Image and Transformation Data for the TM Rural Rochester Imagery	96
Table 7	Summary of Temporal PIF Image and Transformation Data for the High Resolution Airphoto Imagery of Buffalo	97
Table 8	Summary of the Image Coordinates Chosen for the Control Point Analysis of the Urban Rochester Transformations	102
Table 9	Summary of the Image Coordinates Chosen for the Control Point Analysis of the Urban Buffalo Transformations	104
Table 10	Summary of the Image Coordinates Chosen for the Control Point Analysis of the High Resolution Airphoto Imagery of Buffalo	106

List of Tables (con't)

Table 11	Summary of the Errors Determined From the Control Point Analysis for the Urban Rochester Imagery When Compared to the 1984 Untransformed Data	108
Table 12	Summary of the Errors Determined From the Control Point Analysis for the Urban Buffalo Imagery When Compared to the 1984 Untransformed Data	109
Table 13	Summary of the Errors Determined From the Control Point Analysis for the High Resolution Airphoto Imagery of Buffalo When Compared to the 1970 Untransformed Data	110
Table 14	Summary of the Approximate Error Due to Sampling (Residual Error From the Linear Regression of the Transformed Day 1 vs. Day 2 Digital Count Data)	112
Table 15	Summary of the Reflectance Conversion Data for the Digital Count Errors Summarized in the Control Point Analysis	114
Table 16	Comparison of the Accuracy Between the PIF Normalizations Involving the Two Urban TM Images After the Sampling Error Was Removed	115
Table 17	Summary of the R.M.S. Error in Digital Count Between the Linear Transformations Derived for the Rural and Urban Rochester TM Scenes	122
Table F-1	Summary of the Digital Count Values Associated With the Control Points for the Urban Rochester Imagery	F-2
Table F-2	Summary of the Digital Count Values Associated With the Control Points for the Urban Buffalo Imagery	F-3

List of Tables (con't)

Table F-3	Summary of the Digital Count Values Associated With the Control Points for the High Resolution Airphoto Buffalo Imagery	F-4
Table F-4	Summary of the Estimated Reflectance Data for the 1984 Urban Rochester Imagery	F-5
Table F-5	Summary of the Estimated Reflectance Data for the 1984 Urban Buffalo Imagery	F-6
Table F-6	Summary of the Estimated Reflectance Data for the 1972 High Resolution Airphoto Imagery	F-7
Table G-1	Summary of the Principal Components Data Computed For the 1982 and 1984 Urban Rochester Reflective TM Data	G-3
Table G-2	Summary of the Cluster Means Determined From the Unsupervised Multivariate Classifier Run on the First Three Principal Component Images of the 1982 and 1984 Urban Rochester TM Data	G-5

List of Figures

Figure 1	Generalized Spectral Reflectance Envelopes For Deciduous and Coniferous Trees	9
Figure 2(a)	Generalized Spectral Reflectance Curves for Soil, Water and Vegetation	12
Figure 2(b)	Two-Band Signature for Each Class in Figure 1.2(a)	12
Figure 2(c)	Cloud-Like Formations Formed for Real Data	13
Figure 3	Two Dimensional Scattergram of Image Data for Minimum Distance to the Mean Classifier	16
Figure 4	Parallelepiped Classification Strategy Using Rectangular Boundaries	17
Figure 5	Weakness of Parallelepiped Classifier for Highly Correlated Data	18
Figure 6	Probability Density Functions Defined by a Maximum Likelihood Classifier	19
Figure 7	Training Sites for Supervised and Unsupervised Training	21
Figure 8	Clustering by the K-Means Algorithm	23
Figure 9	Illustration of the Logic Used to Segment PIF Features in Landsat TM Six Band Imagery	30
Figure 10	Illustration of the Image Processing to Segment a PIF Mask	31
Figure 11	Example of the Histogram Specification Process	33

List of Figures (con't)

Figure 12	Example of Histogram Specification for PIF Normalization	34
Figure 13	Transformation of Linearly Related Histograms	37
Figure 14	Number of Pixels Remaining On in the Logical Combination of the Individual Thresholded Images	48
Figure 15	Illustration of the Threshold Positions Used in Gradient Calculation	49
Figure 16	The Gradient Surface Derived From Figure 14	50
Figure 17	Flowchart of Thresholding/3-D Surface Building Algorithm	59
Figure 18	Illustration of the Specified Threshold Range	60
Figure 19	Illustration of the Pattern Used in the Thresholding Procedure	61
Figure 20	Error Exhibited Between Two Linear Histogram Transformations	72
Figure 21	Original 1982 and 1984 Urban Rochester Image	79
Figure 22	Original 1982 and 1984 Urban Buffalo Image	80
Figure 23	Original 1982 and 1984 Rural Rochester Image	81
Figure 24	Original 1970 and 1972 High Resolution Airphoto Image	82

List of Figures (con't)

Figure 25	PIF Masks Created for the 1982 and 1984 Urban Rochester Images Using (a) the Automated and (b) the Interactive Segmentation Processes	84
Figure 26	PIF Masks Created for the 1982 and 1984 Urban Buffalo Images Using (a) the Automated and (b) the Interactive Segmentation Processes	85
Figure 27	PIF Masks Created for the 1982 and 1984 Rural Rochester Images Using (a) the Automated and (b) the Interactive Segmentation Processes	86
Figure 28	PIF masks Created for the 1970 and 1972 NHAP Airphoto Images of Buffalo Using (a) the Automated and (b) the Interactive Segmentation Processes	87
Figure 29	Three-Dimensional Surfaces Representing (a) the Number of Pixels as a Function of Threshold Values and (b) the Gradient of the Surface in (a) for the 1982 and 1984 Urban Rochester Images	88
Figure 30	Three-Dimensional Surfaces Representing (a) the Number of Pixels as a Function of Threshold Values and (b) the Gradient of the Surface in (a) for the 1982 and 1984 Urban Buffalo Images	89
Figure 31	Three-Dimensional Surfaces Representing (a) the Number of Pixels as a Function of Threshold Values and (b) the Gradient of the Surface in (a) for the 1982 and 1984 Rural Rochester Images	90

List of Figures (con't)

Figure 32	Three-Dimensional Surfaces Representing (a) the Number of Pixels as a Function of Threshold Values and (b) the Gradient of the Surface in (a) for the 1970 and 1972 NHAP Airphoto Images of Buffalo	91
Figure 33	The PIF Transformations Developed Utilizing the Automated and Interactive Segmentation Results for the Band 1 Urban Rochester TM Image	98
Figure 34	CIR Composite TM Image Representing (a) the Original 1982 Urban Rochester Image, (b) the 1984 Urban Rochester Image and (c) the Transformed 1982 Urban Rochester Image Utilizing the Segmentation Result From the Automated Algorithm	99
Figure 35	CIR NHAP Airphoto Representing (a) the Original 1972 Buffalo Image, (b) the 1970 Buffalo Image and (c) the Transformed 1972 Buffalo Image Utilizing the Segmentation Result From the Automated Algorithm	100
Figure 36	The Control Points Summarized in Table 8 are Denoted by +'s on the 1984 Urban Rochester Image	103
Figure 37	The Control Points Summarized in Table 9 are Denoted by +'s on the 1984 Urban Buffalo Image	105
Figure 38	The Control Points Summarized in Table 10 are Denoted by +'s on the 1970 NHAP Airphoto Image of Buffalo	107
Figure 39	PIF Transformations For the Urban and Rural Rochester Imagery Using (a) the Automated Segmentation and (b) the Interactive Segmentation Algorithm (TM Band 1)	117

List of Figures (con't)

Figure 40	PIF Transformations For the Urban and Rural Rochester Imagery Using (a) the Automated Segmentation and (b) the Interactive Segmentation Algorithm (TM Band 3)	118
Figure 41	PIF Transformations For the Urban and Rural Rochester Imagery Using (a) the Automated Segmentation and (b) the Interactive Segmentation Algorithm (TM Band 4)	119
Figure 42	PIF Transformations For the Urban and Rural Rochester Imagery Using (a) the Automated Segmentation and (b) the Interactive Segmentation Algorithm (TM Band 5)	120
Figure 43	PIF Transformations For the Urban and Rural Rochester Imagery Using (a) the Automated Segmentation and (b) the Interactive Segmentation Algorithm (TM Band 7)	121
Figure G-1	The First Three Principal Component Images Derived From the Six Reflective Landsat TM Bands of the 1984 Urban Rochester Data Set	G-4
Figure G-2	Color Composite Images of the Spectral Clusters Formed by the Unsupervised Multivariate Clustering Algorithm on the First Three Principal Component Images of the 1982 and 1984 Urban Rochester TM Scenes	G-7
Figure I-1	Typical D-Log H Curve / τ vs. H Curve	I-5

1.0 Introduction

As soon as man was able to take to the air for travel, the prospect of collecting information about the Earth became possible. With the advances in aeronautical technology of the past century, concurrent advances in imaging technology have also been made. From hand held photography in hot air balloons to the French SPOT satellite of today, imagery has become more and more complex as well as intellectually exciting.

In whatever form this imagery has been collected, a primary use of aerial images has always been mapping and land use classification. The complexity of these processes range from a photointerpreter sitting down at a light table with a photographic transparency to the statistical processing of a digital image containing up to four million separate picture elements. However the data is processed, this information is then used to track urban development, study the decay or improvement of natural bodies of water, monitor the progress of a farmer's croplands, or even to search the land for natural resources such as petroleum.

A first step in any processing of this remotely sensed imagery is the correction of atmospheric degradations that have disrupted the quality of the signal reaching the sensor. Much time and research has been devoted to this problem with varying degrees of success. Many models have been designed to predict atmospheric effects in an attempt to understand how they could be corrected. In general these models are tedious and involve a great deal of human interaction in their implementation. One of the many algorithms directed at this purpose is referred to as temporal image normalization using pseudoinvariant features. Unlike classical normalization algorithms which correct each image for their individual degradations, this algorithm will force the second image of a temporally separated pair to appear as if it were taken through the identical atmosphere as the first. In this manner, any

remaining differences are truly differences in the actual scene. Pseudoinvariant feature normalization, unlike the others, does not require a large degree of human interaction since it is statistically based. The human user is only required to segment the original images such that only urban features remain.

This study is intended to remove the image analyst from the process of temporal image normalization using pseudoinvariant features. This is accomplished by the development of an automated image segmentation algorithm that isolates pseudoinvariant features from the two temporally separated images. This study involves the theoretical development of the segmentation algorithm, the empirical application of this algorithm to the pseudoinvariant feature normalization of a wide variety of Landsat TM images, and an in-depth analysis of the errors involved with the algorithm and the normalization method. The primary goal of this study is to remove the human interpreter from the pseudoinvariant feature normalization process and to provide the remote sensing community with a completely automated method of scene normalization. The secondary goal is to provide a viable and quantitative measure of scene normalization accuracy which can be used as a comparative tool for the large gamut of normalization techniques that are available.

1.1 Historical Background

The history of remote sensing dates back to the late nineteenth century when man first took to flight. A natural outgrowth of this scientific accomplishment was the acquisition of information about the land, the waters, and other people in a way that was never available before. As techniques for information acquisition have become more complex and sensitive, the informational content of the imagery is degraded by the atmosphere that exists between the sensor and the target of interest. It has become necessary to devote time and effort to the development of methods to remove this degradation, along with those due to the sensor response function, illumination geometry and collection geometry. This effort is part of a branch of remote sensing called temporal image normalization.

1.1.1 Temporal Image Normalization

Temporal image normalization is defined as a process which removes most of the effects of sun angle, look angle, and atmosphere from each of a pair of temporally separated images, therefore causing them to look as if they were taken under similar conditions.¹ The successful application of this type of process augments the accuracy of such processes as temporal change detection, automatic feature extraction, and target identification.² This technique has been the subject of research for high resolution imagery (Piech and Schott, 1974 and Piech *et. al.*, 1981)^{3,4} and has achieved a great deal of success. As a direct extension from these applications, coarser resolution imagery such as that from the Landsat Thematic Mapper (TM) , has been the subject of successful normalization studies.⁵

Gerson and Fehrenbach (1983) have described five different temporal image normalization algorithms,⁶ including normalization procedures for high resolution

black-and-white and color aerial films. Evaluation of these techniques involved a simple subjective comparison of the relative effectiveness of the normalization. Comparison of these techniques have shown that the success of normalization depends on several factors including scene content, the type of imaging process and albedo effects. The best results come from normalizing two black-and-white terrestrial images where several known reflectances can be located. Less successful results were achieved with hydrographic images.

It was decided not to choose any of the normalization techniques described by Gerson and Fehrenbach since all these techniques involved operations on both images of the temporally separated pair. A normalization technique which operates on a single image and causes it to appear like a second image was desired since a reduction in error due to less complex normalization was expected. Such a technique was demonstrated by Volchok and Schott (1985) through the use of pseudoinvariant features.⁷ This technique works quite well on coarser images (e.g. 30 meter spot size), with normalization errors of the order of one reflectance unit. This method is easily implemented and requires very little user intervention, thus making it attractive to an image analyst.

1.1.2 Pseudoinvariant Feature Normalization

The temporal image normalization technique chosen for this study is based on image components known as pseudoinvariant features (PIF's). This is a class of objects whose spectral signatures are relatively constant over time. Such features include concrete, asphalt, stone and rooftops.⁸ The reflectance of these objects over time is not constant in an absolute sense. That is, although concrete and asphalt surfaces may get soiled and hence exhibit a change in reflectance, it is expected that the statistical distribution of the reflectances of these features will remain constant.

The actual normalization is executed in the following manner for Landsat TM imagery (The techniques are described as being applied to TM imagery since this was the subject of the initial research,⁹ but the method was shown to work equally well, if not better, with higher resolution images.¹⁰) Two temporally separated images of the same area must be obtained. Pseudoinvariant features are isolated from these two images and their gray level distributions determined within each band of both images (it is assumed that the digital brightness histogram and the reflectance histogram are linearly related). A set of linear transformations can then be derived to modify the histograms of the second image so that they look like the histograms from the first image. These transformations, derived for the distributions of the pseudoinvariant features, can then be applied to the second digital image. As stated earlier, the results of this type of normalization yielded results with approximately one reflectance unit of error between the two normalized images. The derivation of this technique and the transformations is well documented by Volchok (1985).¹¹

The theory behind image normalization using pseudoinvariant features and the application of this new technology for Landsat TM imagery has proven to be a viable, simple, and much needed technique for temporal image normalization. This

method makes two temporally separated images of the same area "look" the same, both visually and radiometrically. In this manner, an image analyst can perform transformations on only one image of a temporally separated pair and still have confidence in the subsequent radiometric measurements. This is quite an improvement over past methods where each image of the pair had to be individually corrected for their respective atmospheric and radiometric degradations before any analysis could begin.

As a preliminary step toward this study, much work was done to simplify the implementation of the concepts demonstrated in the original investigation.¹² in order to make the implementation of this technique less arduous. The current implementation of this technique takes only minutes rather than the hours once required. This facilitates two goals of all remotely sensed image analysis: (1) maximization of accuracy and speed and, (2) minimization of man's involvement and contribution of error.

An essential first step in the PIF normalization process is the isolation of the pseudoinvariant features from the digital images. To date this has been done in an iterative and interactive fashion as will be described in Section 1.3.1. The removal of human interaction from this technique would enhance this technology further and make available a powerful technique for many temporal studies such as change detection. It is therefore the primary purpose of this study to develop an automated algorithm for segmentation so that the human interaction can be removed from this normalization process.

The question that inevitably arises with any technique used to normalize images is "How good is the technique, i.e. how well are the images normalized?". This question is addressed in the following section.

1.1.3 Quantification of Normalization Results

The quantification of the results obtained in an image normalization procedure has not been considered in the literature in any rigorous form. Gerson and Fehrenbach (1983) used only qualitative analysis in their comparison of different normalization methods.¹³

Normalization error is not a simple concept to evaluate since quantitative information in a scene is not always readily available. Ground truth panels in a scene would undoubtedly be an invaluable tool for the evaluation of the effectiveness of a scene normalization procedure since direct comparison of this ground truth data could be made. Ground truth data is costly to obtain for high resolution imagery since significant manpower is required. It is nearly impossible to obtain with lower resolution imagery such as Landsat TM data since the size of the required ground truth panels would need to be immense (on the order of 60 meters on a side to eliminate mixed pixel effects). It is evident from this that a method of quantitatively evaluating the effectiveness of scene normalization techniques is required. A secondary purpose of this study is to develop a quantitative tool for testing the quality of normalization using in-scene elements. Such a test can be used to evaluate the results from any normalization technique and make possible comparative judgements.

1.1.4 Spectral Signatures and Band Ratioing for Classification and Segmentation

In order to use the pseudoinvariant feature normalization technique, the man-made features that are present in the image need to be segmented for statistical analysis and subsequent development of transforms. To begin the process of isolating pseudoinvariant features from a Landsat TM image, we must first examine

the methods commonly used in the classification of multispectral imagery. It is the purpose of this and the following sections to review these classification techniques.

The analysis of remotely sensed multispectral imagery can be an overwhelming task considering some of the multispectral scanners that are currently being used today. These systems can have as few as three spectral bands while others may have up to twenty-four regions of spectral sensitivity,¹⁴ while sensors being developed at present, such as the airborne imaging spectrometer, have as many as 128 different spectral bands. The amount of data that is collected by these sensors is immense and the amount of computer power necessary to handle this data soon becomes overwhelming. An obvious first-cut solution to this problem is to use some sort of data reduction technique which retains only information relevant to the problem at hand. The field of remote sensing utilizes spectral signatures of objects, band ratios and other statistical transforms, in an attempt to make the data more manageable and relevant while eliminating much of the redundancy that is present.

As defined by Slater (1980), a spectral signature comprises a set of values for the reflectance or radiance of a feature where each value corresponds to the reflectance or radiance of the feature averaged over a different, well-defined, wavelength interval.¹⁵ The signatures are affected by illumination and viewing geometries as well as atmospheric attenuation. More will be said about these signal degradations later. It is important to note that spectral signatures are not unique identifiers for a specified feature. As described by Lillesand and Kiefer (1979), a spectral signature is more like an envelope in which the spectral reflectance curves for a class of objects fall.¹⁶ This is illustrated in Figure 1. It has been shown, however, that although there is variability in the relative magnitude of the spectral reflectance curves for a particular feature class, the relative shape of the curve can be considered to remain nearly constant.¹⁷

Use of the spectral signature alone is hardly sufficient as a means to classify or segment features from a complex scene. To aid in segmenting image features via spectral signature analysis, band ratioing is an effective tool. Normally the spectral ratio of two image bands tells an image analyst more about the imaged objects than do the recorded values in any single band.¹⁸ Examples of the success of band ratioing as a means of

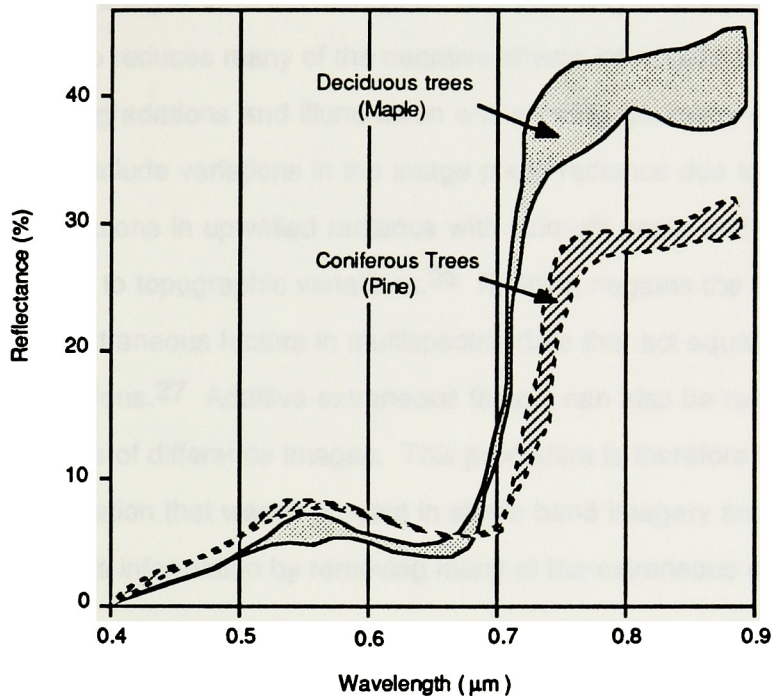


Figure 1 Generalized spectral reflectance envelopes for deciduous and coniferous trees. (Adapted from Lillesand and Kiefer, 1979)

classification are described for water quality by Lillesand and Kiefer (1979),¹⁹ for geology by Chavez *et. al.* (1982)²⁰ and Williams (1983),²¹ and for vegetation by Lillesand and Kiefer (1979).²² As Williams (1983) stated, "the use of band ratioing in a "shotgun" fashion will be discouraging since an exorbitant number of ratio images can be derived. This method should be used in a rational, well thought out manner."²³ It has been shown by Tucker (1973) that the ratio of Landsat

Multispectral Scanner (MSS) band 7 (0.8 to 1.1 μm) to MSS band 5 (0.6 to 0.7 μm) reveals well the amount of vegetation in the scene.²⁴ Biegel and Schott (1984) have shown that the ratio of Landsat TM band 5 (1.57-1.78 μm) to TM band 3 (0.62-0.69 μm) is well-suited for the classification of water, vegetation and urban features.²⁵ Hence, the use of a properly selected band ratio can be an invaluable classification tool for earth features. Band ratioing not only aids in the classification of multispectral images from the standpoint of presenting an enhanced image to the analyst, but also reduces many of the negative effects introduced to the image due to atmospheric degradations and illumination and viewing geometry considerations. These effects include variations in the image plane radiance due to optical vignetting, variations in upwelled radiance with azimuth angle and ground radiance differences due to topographic variations.²⁶ Ratioing negates the effects of any multiplicative extraneous factors in multispectral data that act equally in all wavelength regions.²⁷ Additive extraneous factors can also be removed by computing ratios of difference images. This procedure is therefore beneficial since it provides information that wasn't present in single band imagery and also improves the quality of this information by removing many of the extraneous effects and degradations.

A typical method to utilize the information derived from the spectral signature or band ratioing data is density, or digital count, thresholding. An example of this is shown by Volchok (1985) where pseudoinvariant features (PIF's) were segmented from a Landsat TM scene utilizing a band-4 to band-3 ratio as well as band-7 imagery.²⁸ This type of technique is extremely attractive in that it requires very little computer memory and the results obtained can be adjusted with virtually no effort. The accuracy of such a technique is dependent on the skill of the person performing the thresholding and an *a priori* knowledge of the scene.

As an example of thresholding as a classification technique, consider a general land-use classification including the following land cover types: soil, water,

and vegetation. On examining the spectral signatures of these three general classes in the visible and near-infrared wavelength intervals (Figure 2(a)), it seems feasible that classification can result simply from thresholding (density slicing) the MSS band-7 image. Figure 2(b) is a two-band signature for the three classes which readily illustrates the apparent ease of classification. However, as stated earlier, the spectral signature curves of Figure 2(a) are not defined in such a singular manner but consist of an envelope surrounding the depicted curves. The results of this natural variability are the cloud-like formations of Figure 2(c). Note that the simple thresholding technique will no longer work with a high degree of accuracy.²⁹ Spectral ratioing will improve the quality of thresholding techniques as shown by Volchok (1985), but more sophisticated and accurate classification and segmentation schemes exist.³⁰ Such schemes include multivariate classification, which is a powerful statistical tool that lends itself to automation. Such techniques will be discussed in the following section with the intention of developing an automated multivariate segmentation algorithm for man-made urban features.

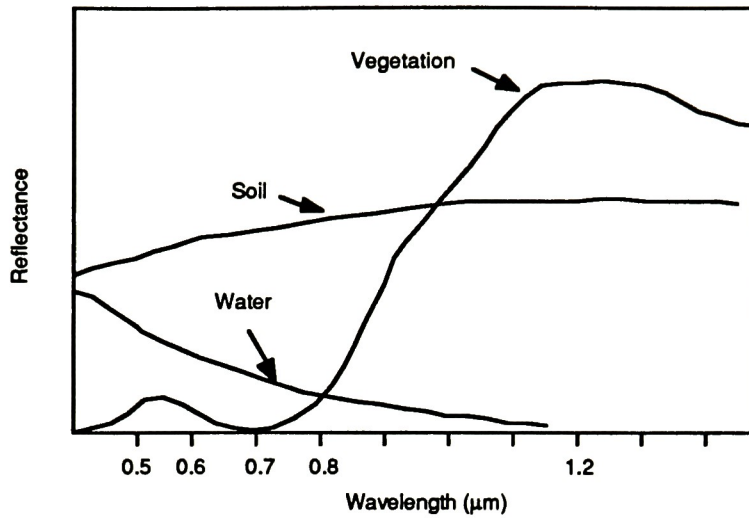


Figure 2(a) Generalized spectral reflectance curves for soil, water and vegetation.

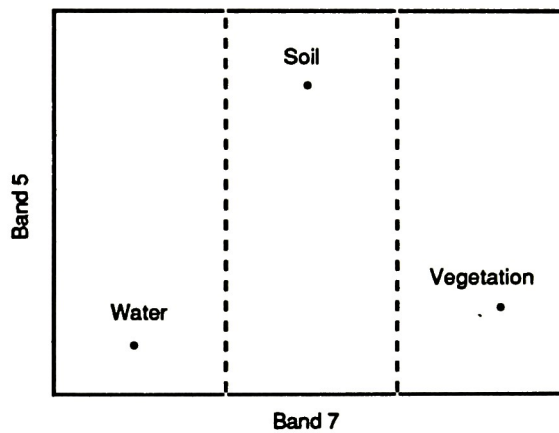


Figure 2(b) Two-band signature for each class in Figure 2(a).

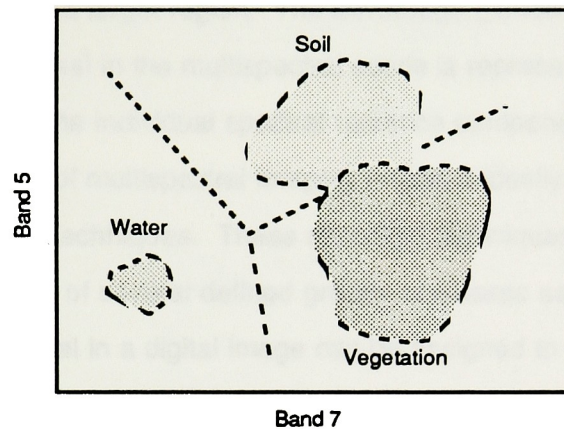


Figure 2(c) Cloud-like formations formed for real data

1.1.5 Supervised Multivariate Classification

Multispectral scanners can be single aperture devices, pushbroom sensors, or focal plane array charged-couple-devices (CCD) which record imagery in several (say k) different spectral bands. That is, they will produce k bands of geometrically registered imagery of a target region. The advantage gained by using this type of device is that each pixel in the multispectral scene is represented by a k -dimensional vector composed of the individual spectral radiance components reaching the sensor. This feature of multispectral imagery makes it ideally suited to many multivariate analysis techniques. These statistical techniques act on individual pixels to determine to which of several defined groups or classes each pixel belongs. In this manner every pixel in a digital image can be assigned to a specific group, i.e. classified.

There are essentially two distinct types of classification schemes. They are supervised and unsupervised classification. In supervised classification, useful information categories are defined by the user and a subsequent analysis performed to determine the spectral separability of these categories. Unsupervised classification, on the other hand, involves first a determination of spectral separability of the raw image data followed by an interpretation of the resulting categories.³¹ By its essence, supervised classification is ruled out as an automated segmentation technique, but it needs to be described in order to bring unsupervised classification into a proper perspective.

Supervised classification consists of three distinct stages as described by Estes (1983).³² The first stage is training, where the analyst compiles an interpretation key or spectral signature set by identifying representative samples of the classes to be identified. In the second, or classification, stage the remaining pixels in the target image are compared to each of the categories chosen in the

training stage. The pixels are assigned to the category to which they most closely correspond. The third stage is output. After the entire scene has been categorized, the results are presented in any of several forms including a color encoded map, tables of areas of specific cover types, or computer-compatible inputs to a grid-based geographical information system. The actual classification of the individual pixels (the second stage of the supervised process), can be accomplished in several ways, each of which has trade-offs. Lillesand and Kiefer (1979) describe three different types of classification algorithms.³³

The first and simplest of these is known as the minimum distance to the mean classifier. During the training stage a mean vector is computed for each of the categories chosen by the analyst. This mean vector is a k-dimensional vector consisting of the mean response levels of each of the spectral bands of the multispectral image. Classification is then performed by determining the multidimensional Euclidian distance of the unknown pixel to each of the mean vectors of the chosen categories. The unknown pixel is then classified as a member of the closest category. This method is computationally fast, but does not work well when natural variability of the means of groups overlap each other. This can be explained by the following example. Suppose one category has a multidimensional mean and a very large variance while another category has a second distinct mean and a very small variance. This phenomenon is illustrated in Figure 3. An unknown pixel can be located within the scattergram of the larger variance group and yet have a shorter Euclidian distance to the mean of the group with the smaller variance. This would result in a misclassification of the pixel. A technique is needed which accounts for this variability. Such a method is the parallelepiped classifier.

The parallelepiped classifier accounts for the variance in the distribution of the category data by setting up ranges for each of the categories. These ranges can be determined by finding the maximum and minimum digital counts in each of the k-categories. For two-dimensional data, this boundary region can be thought of as a

rectangle surrounding all the points included in that category (see Figure 4). An unknown pixel is classified into the category within whose boundary it falls.

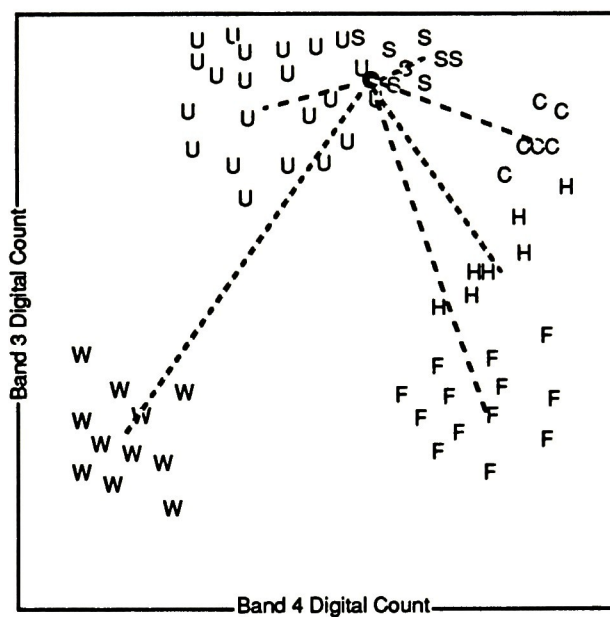


Figure 3 Two Dimensional Scattergram Of Image Data For Minimum Distance To The Mean Classifier

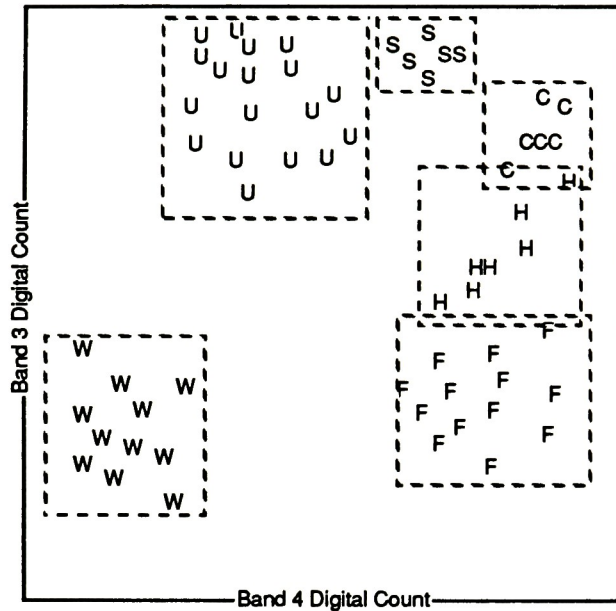


Figure 4 Parallelepiped Classification Strategy Using Rectangular Boundaries

This classifier is extremely fast computationally, but two problems exist. The first is overlapping regions. Two regions in close proximity will likely have their boundary regions covering some common area. This will result in confusion when a classification decision must be made. Typically, these "confused" pixels will arbitrarily be assigned to one group or the other. The second problem is most often the cause of the first. The parallelepiped classifier does not deal well with highly correlated band data, which will tend to cause boundaries to be formed that exhibit lack-of-fit for the category data. If one thinks of this in two-dimensions, data that are highly correlated will appear as a thin ellipsoid sloping either upward or downward from the origin depending on the sign of the correlation. This appearance

will then cause the resulting boundaries to greatly overestimate the size of the classification region for the particular category. This is shown in Figure 5. Remotely sensed data is often highly correlated in this manner, so this problem is a prominent one with this classifier. Lillesand and Kiefer (1979) describe a modification to this classifier that makes it more useful for highly correlated data.³⁴

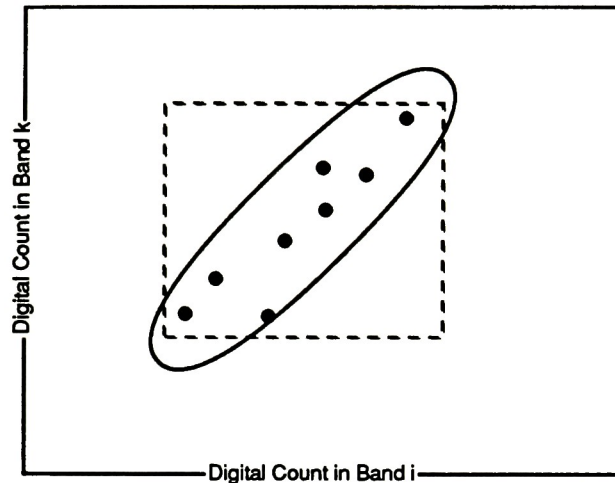


Figure 5 Weakness Of Parallelepiped Classifier For Highly Correlated Data

The two classifiers described above are known as nonparametric classifiers since they make no distributional assumptions on the data. These methods are computationally quick and easy to implement, but often do not exhibit the desired accuracy. The maximum likelihood classifier is parametric, i.e. it makes the assumption that the data in each category takes on a multivariate normal distribution (see Figure 6).

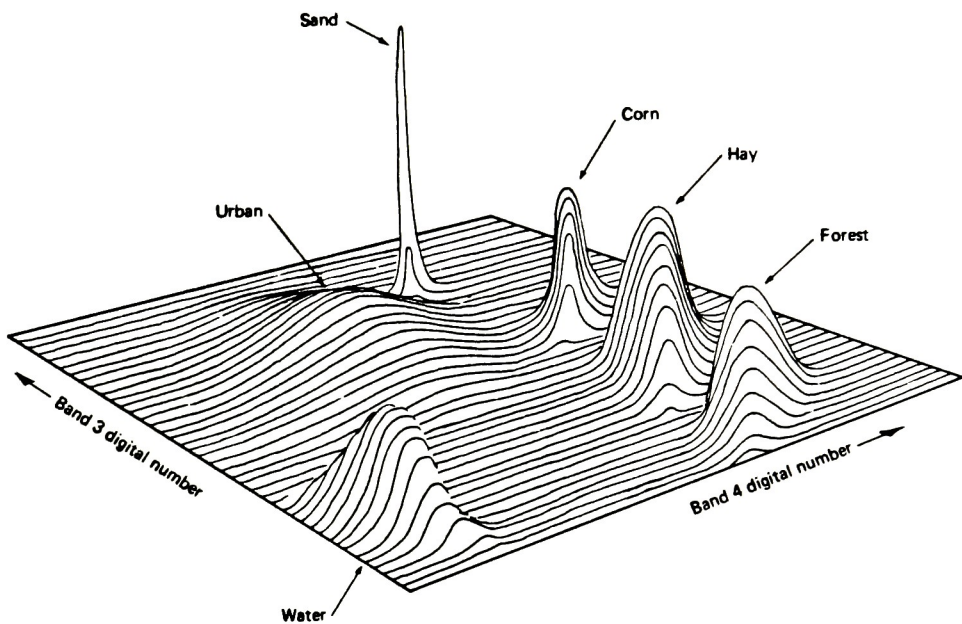


Figure 6 Probability Density Functions Defined By A Maximum Likelihood Classifier (after Lillesand and Kiefer, 1979)

Under the assumption of multivariate normality, the distribution of the data from a particular earth feature is completely described by a mean vector and a covariance matrix. With these quantities defined for each of the classes, the probability that an unknown pixel belongs to each of these classes can be computed and the pixel classified into a category based on the maximum probability determined. Hence, this method accounts for both the variance of the data and the

exhibited correlation. This classifier is computationally more complex to implement and runs much slower than the simple decision-based algorithms of the two previous classifiers, however there is a significant increase in accuracy. Modifications on this method are described in Lillesand and Kiefer (1979)³⁵ and in Schowengerdt (1983).³⁶

1.1.6 Unsupervised Multivariate Classification

While it is the goal of supervised multivariate classification to choose the desired classes based on features of interest, the purpose of unsupervised classification is to determine the classes based on their spectral separability. Unsupervised classification examines a large number of pixels and forms classes based on natural groupings present in the image values.³⁷ The image data are submitted to a clustering algorithm and the resulting clusters in k-dimensional space (representing k spectral bands) are each assumed to represent a class.³⁸ Each class that is formed may not be associated with a distinct land cover type, but the classes are spectrally separable. At this point it is up to the analyst to determine the physical counterparts of these statistical clusters based on information from land cover maps, aerial photographs, and other forms of ground truth data.

In choosing training data for an unsupervised classification algorithm, one must be certain to choose training sites with a heterogeneous mixture of pixels. This ensures that all possible classes and their within-class variabilities are adequately represented. This heterogeneous criterion contrasts the choice of training sites for a supervised classifier where the analyst wishes to choose homogeneous training sites for each individual class.³⁹ These choices of training sites are illustrated in Figure 7. Once the training data is collected, it is submitted to a clustering algorithm of some type. The number of clustering algorithms available are countless and are limited only by the analyst's ingenuity.⁴⁰ The following is a description of one of the

more common algorithms known as the k-means algorithm, or ISODATA as referred to by Kan (1972).⁴¹

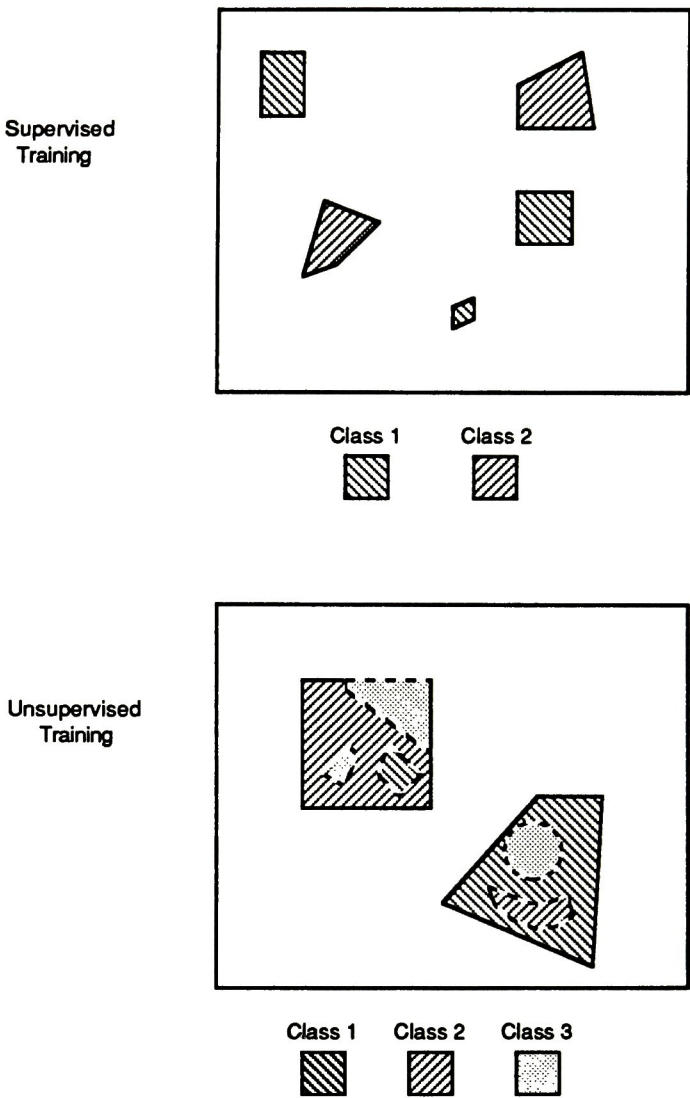


Figure 7 Training Sites for Supervised and Unsupervised Training

The k-means algorithm, as described by Schowengerdt (1983), works in the following way.⁴² Once the training data has been collected based on the criterion mentioned previously, the algorithm arbitrarily chooses k different means by randomly selecting pixel coordinates. The number k depends on the number of clusters the analyst wishes to form. The pixels in the training set are then assigned to one of the k means depending on which particular pixel is closest to in a Euclidean sense. Once all the pixels have been assigned to one of the arbitrary means, new mean values are computed for each of the arbitrarily formed classes. The above is then repeated, that is, each pixel is again assigned to the nearest mean and new mean values calculated. This process is continued until there is no significant change in the location of the k mean vectors. These clusters thus formed are then considered to be k spectrally separable classes. This convergence to the means is illustrated in Figure 8 for two-dimensional image data. This method is relatively insensitive to the original choice of mean vector seeds, however, the number of iterations required to converge to the true cluster means may be large if the original seeds greatly in error. A FORTRAN version of this algorithm is described by Hartigan (1975).⁴³ Subsequent classification of the entire image can then be accomplished by using a minimum distance to the mean classifier. If a more precise classifier is required, an associated covariance matrix may be computed for each of the clusters and a maximum-likelihood classifier could be used. Whichever method is chosen, a successful classification should result if the proper choice of training sites were made originally. As mentioned previously, these sites contain a wide variety of land cover types as well as large within-class variabilities.

The advantage of using unsupervised clustering is that more classes are allowed to be formed; classes that may not have been formed in the case of a supervised classification. In addition, the classes formed are more separable in a spectral sense. An example of this are coniferous trees located

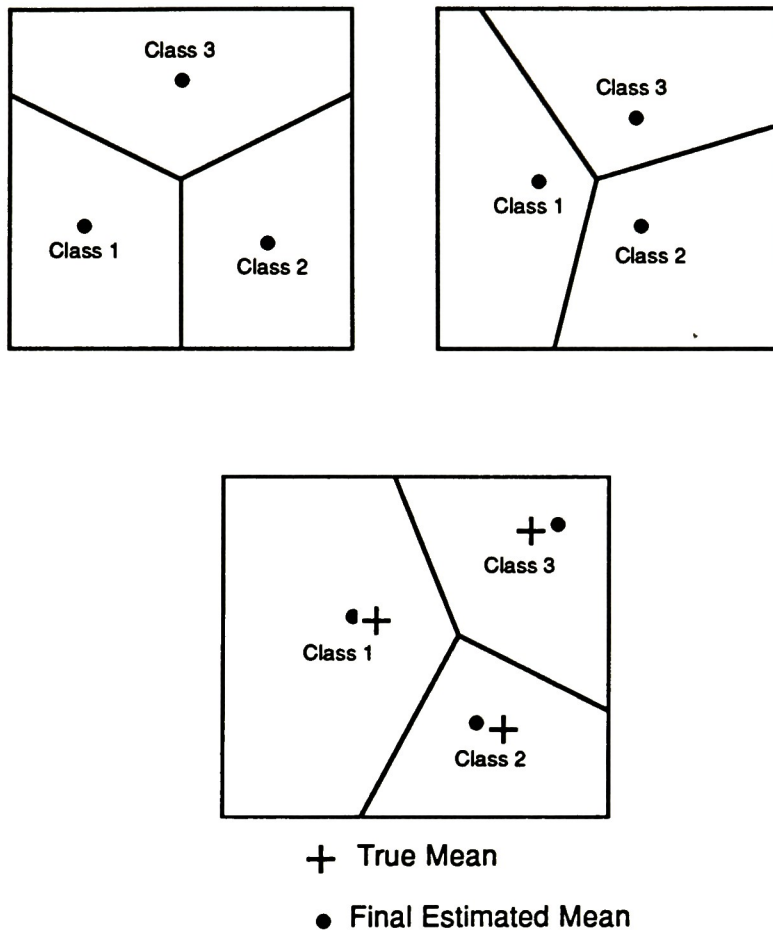


Figure 8 Clustering by the K-Means Algorithm

in shaded and sunlit areas of the same image. An image analyst would most likely place pixels from both these areas in a single training set class. This would result in the classification being confused since these two differently illuminated areas are not spectrally identical, although they are the same land cover type. In fact, they may appear different to a clustering type algorithm. For such reasons, it is often beneficial to first run an unsupervised classifier to determine spectrally separable classes. Clusters that represent features of interest can then be chosen as classes for a supervised training set. Techniques that utilize this combination of methods are referred to by Lillesand and Kiefer (1979) as hybrid techniques.⁴⁴

A problem faced by both supervised and unsupervised classification algorithms is the extensive computation time required due to the immense quantity of data to be analyzed. For example, an image containing a quarter million pixels in each of seven different spectral bands (i.e. a Landsat TM image) requires 2 MB of data storage. The following section is a description of several methods that are useful for reducing the amount of data that is necessary for a successful classification. These methods analyze the original spectral data and derive images that contain the majority of the original information content in a more compact form.

1.1.7 Preprocessing and Redundancy Reduction

It would seem that the accuracy of any of the above classification algorithms would be increased significantly by the addition of more spectral bands of information. According to Estes (1983) this is only true if the added bands contain additional, non-redundant information.⁴⁵ This is often not the case in remotely sensed data since the spectral information of a scene is indeed highly correlated. For Landsat MSS data which contains four separate spectral bands, Kauth and Thomas (1976) have shown that by transforming the spectral data, the redundancy can be reduced to two-dimensions, rather than four, such that approximately 95% of

the original variations is still represented by the transformed data.⁴⁶ Crist and Kauth (1986) have also shown that the six reflective spectral bands of Landsat TM can be transformed to three bands of information that contain at least 95% of the total variation in the original data.⁴⁷ This reduction in redundancy serves two purposes. The first is to increase computational speed since less data needs to be analyzed. Secondly, the quality of this transformed data is superior in that the new coordinate axes take on new physical definitions that aid in classification.⁴⁸

This type of initial transformation of the image data is referred to as preprocessing. Preprocessing used to reduce the amount of redundant information is commonly called principal component or factor analysis.⁴⁹ Another type of preprocessing transformation derive new data sets from the original images. The two primary data derivation methods in use today are spatial clustering and texture analysis.

Spatial clustering is a general name for methods that take an initial look at the image data to determine collections of neighboring pixels that comprise spatially homogeneous units. Kauth *et al.* (1977) have developed a technique known as BLOB which introduces spatial coordinates into the vector description of each pixel to indicate the spatial homogeneity of certain field-like patches in the image.⁵⁰ In this manner, if a subsection of an image is identified to be homogeneous, then the number of subsequent classification calculations can be greatly reduced. Kettig and Landgrebe (1976) have developed a similar method known as ECHO (Extraction and Classification of Homogeneous Objects) which divides an image into small sub-images and tests for homogeneity within these.⁵¹ If a sub-image is homogeneous, it is combined with surrounding homogeneous sub-images and retested. In this manner, homogeneous patches are established throughout the image and, as with BLOB, subsequent classification analyses are enhanced. This type of clustering of the original data not only facilitates an increase in speed of classification, but also puts the result in a more desirable and less noisy form.⁵² Several other techniques

exist in the literature such as ISODATA by Kan (1972),⁵³ CLASS by Fromm (1976)⁵⁴ and others by Bryant (1979).⁵⁵ Bryant (1979) has written an excellent review article on these techniques.⁵⁶

Texture analysis generates another class of techniques that have become prominent in the past decade. As described by Haralick *et al.* (1973), texture is one of the most important characteristics used in identifying objects or regions of interest in an image to the human observer.⁵⁷ Texture is an innate property of all surfaces and can be described as fine, coarse, smooth, rippled, or irregular. This information along with spectral information, provide two of the three primary "clues" to classification of objects for the human observer, the third being contextual information. The addition of this information to a computer-aided classification should also be a significant improvement. Haralick has demonstrated that by using texture features based on the relative frequency distributions of the image gray tones, along with spectral features, he was able to classify seven land use categories in a satellite image with 83% accuracy or better. Success with texture analyses have also been reported by Hsu (1978),⁵⁸ Mitchell and Carlton (1978),⁵⁹ Mitchell *et al.* (1978),⁶⁰ Richards and Landgrebe (1982),⁶¹ and Troy *et al.* (1973).⁶²

These preprocessing algorithms serve to reduce inherent redundancy in multispectral images as well as to introduce extra information to the multivariate classifiers that help to improve the accuracy. These techniques, however, are very susceptible to the influence of signal degradations such as changing atmospheric and illumination conditions. This susceptibility may render such techniques ineffective as segmentation algorithms for temporally separated imagery. It will be a purpose of this study to examine these and other techniques to determine if any useful information can be obtained to aid in the classification and extraction of pseudoinvariant features.

1.2 Theoretical Background

The following section contains a description of the theory behind the PIF normalization process, the derivation of the normalizing transforms, the observed behavior and subsequent development of the automated segmentation algorithm, and the attempted use of classical multivariate techniques for automated segmentation. The theory presented here is similar to that presented by Volchok (1985) in the original proof-of-concept study.⁶³

1.2.1 PIF Technique for Image Normalization

As stated earlier, there are three distinct steps in the PIF normalization process. They are the isolation of pseudoinvariant features from the digital imagery, the computation of the statistical distribution for the pseudoinvariant features in all spectral bands of each of the temporally separated image, and finally the determination of the normalizing transforms and their application to the imagery.

The isolation of pseudoinvariant features from digital imagery proceeds in the following fashion. It has been shown by Biegel and Schott (1984) that an infrared-to-red ratio image is very effective in the classification of water, vegetation, and urban features.⁶⁴ The brightness of vegetation in this ratio image will tend to be very high, while that of urban features and water will be considerably lower. In a digital environment this image can be

derived from the quotient of Landsat TM bands 4 ($0.78 - 0.91\mu\text{m}$) and 3 ($0.62-0.69\mu\text{m}$). To facilitate isolation of PIF's, Landsat TM band 7 ($2.08-2.35\mu\text{m}$) is used since water has nearly zero reflectance in this spectral region. To perform the isolation, the following logic is followed (see Figure 9). The TM band-4 to band-3 ratio image is thresholded from the high digital count values downward. In this manner, the vegetation pixels which have high digital count values, are eliminated from the image, that is, these pixel brightnesses are set to zero. The resulting image contains only water and urban features. The TM band-7 image is then thresholded from the low digital count values upward. This procedure will eliminate water from the band-7 image since water has a low reflectance in this spectral region, and therefore low digital count values. This threshold will also tend to eliminate the wet farmland and vegetation that were missed in the thresholded band-4 to band-3 ratio. The resulting image contains only urban features and dry vegetation-covered areas. If the two thresholded images are transformed to binary images (i.e. all non-zero pixel brightnesses are replaced with a brightness count of 255), and then combined using a logical .AND., the resultant combination will be a binary image that is bright where there were PIF's and dark everywhere else. This image is known as the PIF mask for the current TM data set. Figure 10 is an illustration of the above process. This procedure is then repeated on the second day's imagery.

The next step in the PIF technique is the determination of the statistical distribution of digital brightness values of the PIF's in each of the reflectance bands for the two TM images. This is accomplished using a logical .AND. operator on the band imagery and the PIF mask. The PIF

pixels in the band imagery correspond to pixels in the mask and are therefore left on in the resulting image. This process is suited for implementation in an array processor which enables the isolation to occur in approximately $1/30^{\text{th}}$ of a second. The resulting image represents the true PIF brightness value distribution. The distribution statistics are then simply obtained by taking the histogram of the resulting image, zeroing the zero histogram bin since it has been artificially enlarged, and calculating the subsequent histogram statistics, namely the mean and standard deviation. All that remains in the PIF normalization process is the derivation of the transformations for each image band. Two methods exist for the derivation of these transforms: histogram specification and linear histogram transformation. A discussion of these methods follows.

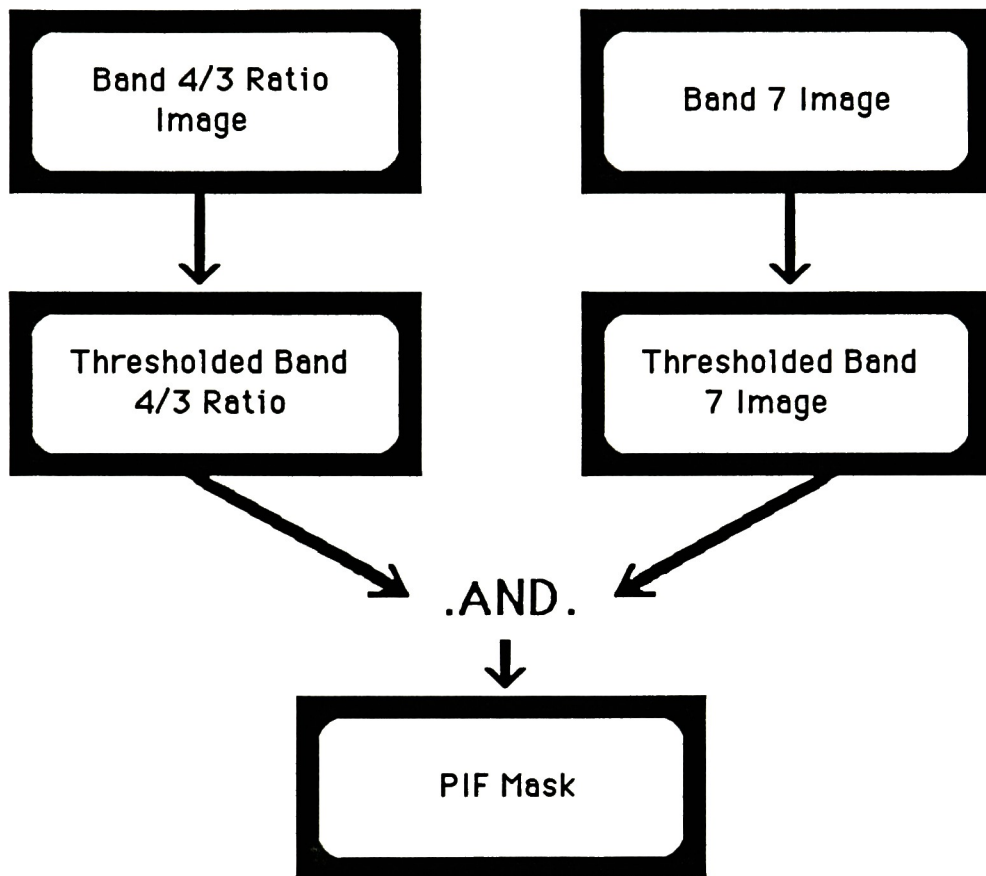


Figure 9 Illustration of the logic used to segment PIF features in Landsat TM six band imagery

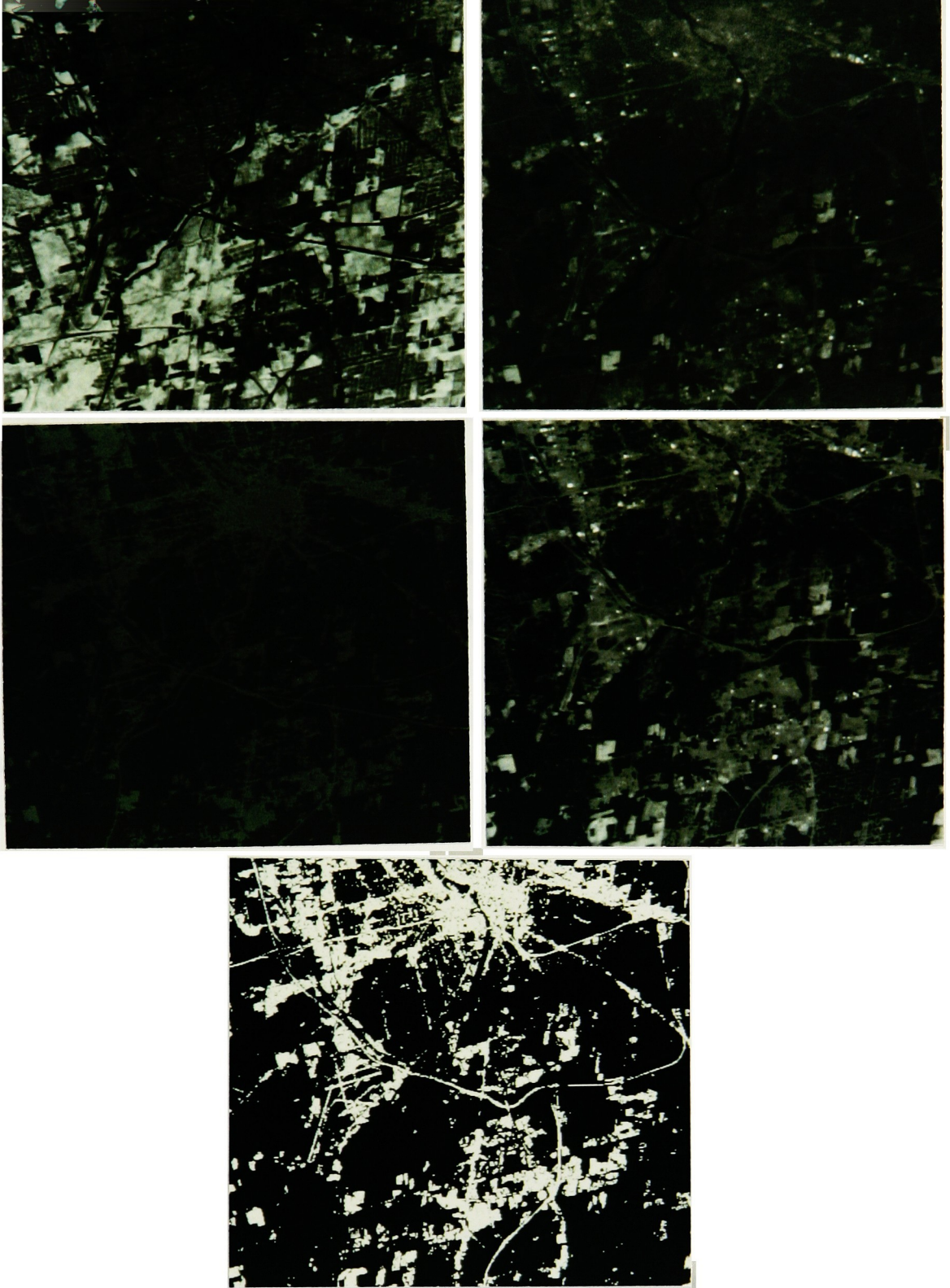


Figure 10 Illustration of the image processing to segment a PIF mask (refer to logic flow in Figure 9)

1.2.2 Derivation of the PIF Normalization Transforms

Histogram specification is a standard image processing technique (Gonzalez and Wintz, 1980).⁶⁵ In short, two histograms are required as input to this technique. The first histogram, referred to as the mapping histogram, is transformed to look like the second, or test, histogram. The theory behind this procedure is taken from histogram equalization. In this procedure, one wants to map the histogram, or the probability distribution function (PDF) for the brightness distribution, through its associated cumulative distribution function (CDF) to obtain a flat normalized output histogram (see Figure 11). This procedure, in and of itself, is not sufficient for our purposes. However, if one applies this technique to two histograms individually, in theory the end result of each equalization will be a flat normalized histogram. From this point on, it is evident that if one took the transformation maps for each of the individual histograms, that is, their associated CDF's, and combined them into a single mapping function (one in a forward fashion and one in a reverse fashion), this mapping function would serve to transform one distribution to look like the other. This is shown in Figure 12.

Linear histogram transformation is another technique which, like histogram specification, transforms one brightness distribution function to another under certain conditions. This procedure was described by Volchok (1985) in a different perspective.⁶⁶ The procedure assumes two conditions:

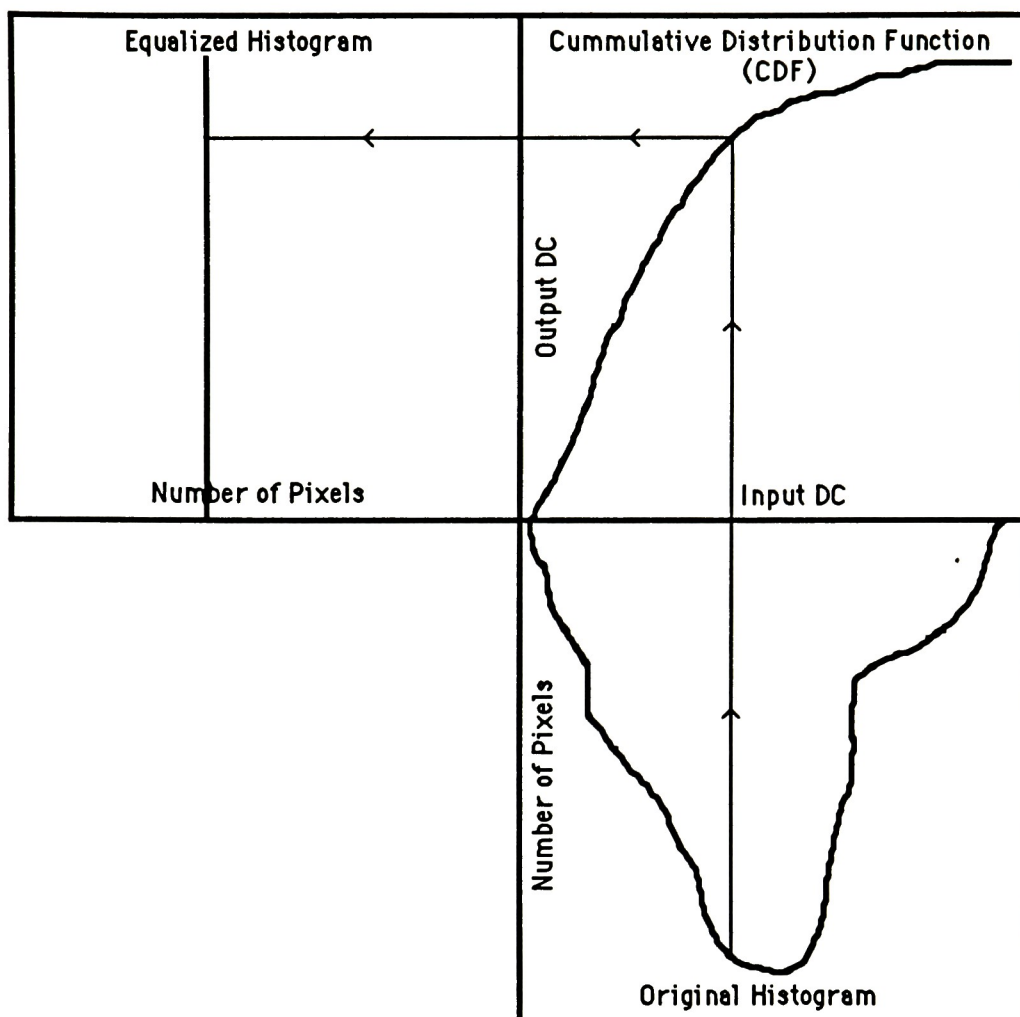


Figure 11 Example of the histogram specification process

the first is that the radiance at the sensor is linearly related to the reflectivity of the scene elements on the ground. The second is that the brightness (i.e. digital count) of the image is linearly related to the radiance reaching the sensor. With these two assumptions in mind, the technique of linear histogram transformation can be developed as follows:

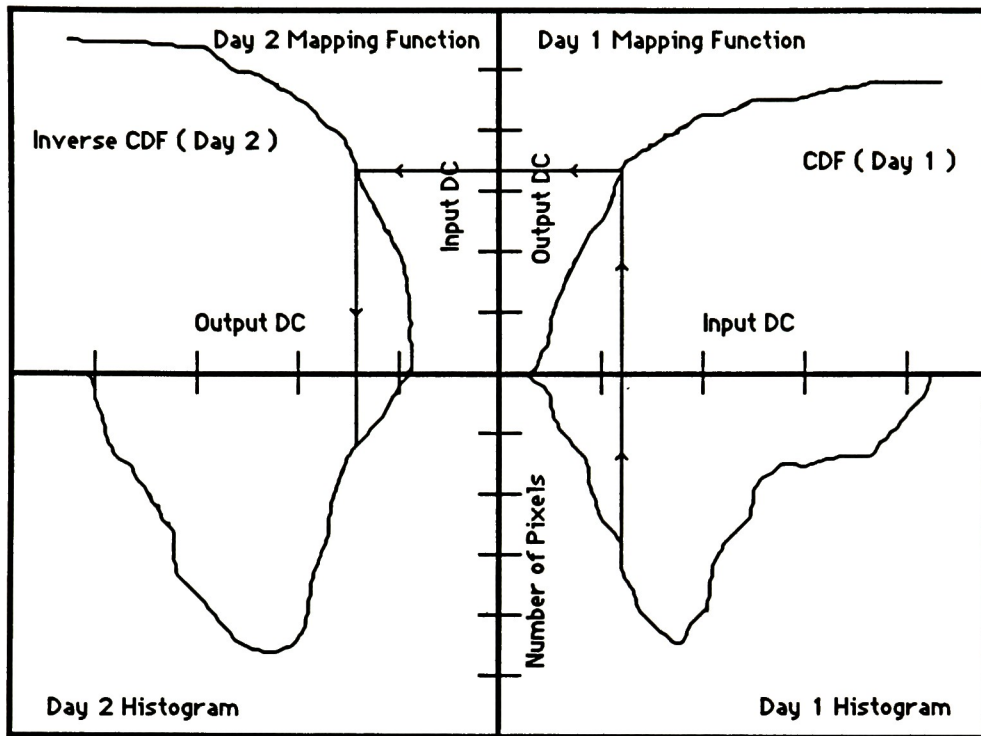


Figure 12 Example of histogram specification for PIF normalization

Let $L_1 = \alpha_1 R_1 + \beta_1$
 and $L_2 = \alpha_2 R_2 + \beta_2$

where L_1 and L_2 are the spectral radiance values of the PIF's in the first and second image, respectively, R_1 and R_2 are the reflectance values of a scene element in the first and second image, the α terms are linear coefficients encompassing the atmospheric transmission and look angle effects, and the β terms are the path radiance terms. In terms of digital count, or brightness,

we have

$$\begin{aligned} DC_1 &= m_1 L_1 + b_1 \\ \text{and } DC_2 &= m_2 L_2 + b_2 \end{aligned}$$

where DC_1 and DC_2 are digital count values for the first and second images, respectively, the m terms are linear coefficients encompassing the sensor response characteristics, and the b terms are the sensor offset values. One can then perform the following simplification, namely that

$$\begin{aligned} DC_1 &= m_1 \alpha_1 R_1 + m_1 \beta_1 + b_1 \\ \text{and } DC_2 &= m_2 \alpha_2 R_2 + m_2 \beta_2 + b_2 \end{aligned}$$

Recalling the basic premise of this procedure, i.e. that the reflectances R_1 and R_2 on average are equal (i.e. $R_1 = R_2$) within PIF's, the following can be written

$$DC_1 = m_t DC_2 + b_t$$

where

$$\begin{aligned} m_t &= \frac{m_1 \alpha_1}{m_2 \alpha_2} \\ b_t &= b_1 + m_1 \beta_1 - \frac{m_1 \alpha_1 b_2}{m_2 \alpha_2} - \frac{m_1 \alpha_1 \beta_2}{\alpha_2} \end{aligned}$$

It can now be seen that DC_1 and DC_2 are linearly related. Because there is a linear transformation between individual digital count values on separate images, this concept can be extended to a distribution-based transformation. The following linear histogram transformation can now be developed. Two linearly related histograms can be transformed to look like each other in the following fashion. The relative width of the histograms are related by the ratio of their standard deviations. The histogram means are then recomputed and the difference between these two means is added to the adjusted mean \bar{x}_2 . The histograms now have equivalent spreads and equal mean values (see Figure 13). The transformations are described as

$$DC_1 = m_t DC_2 + b_t$$

where

$$m_t = \frac{\sigma_2}{\sigma_1}$$

$$b_t = \bar{x}_2 - m_t \bar{x}_1$$

If the assumption stated above (i.e. that digital count is a linear function of reflectivity) is valid, then the linear histogram transformation process will give a better result than histogram specification. Histogram specification makes no assumptions about the nature of the difference between two histograms, but only attempts to map one onto the other. Histogram specification also tends to overcompensate for differences near the extremes of histogram variance and therefore is only accurate within the central portion of the data. Linear histogram transformation is designed to

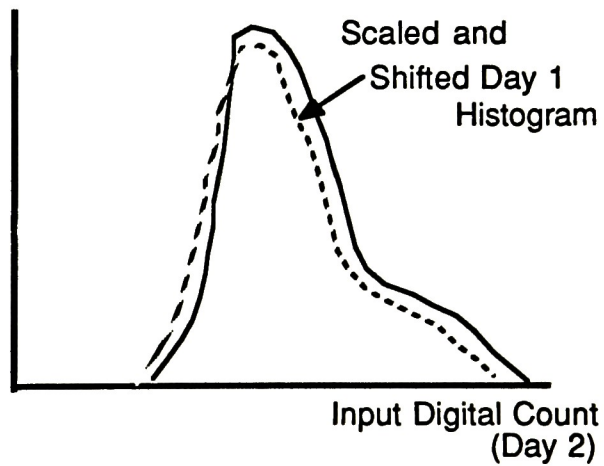
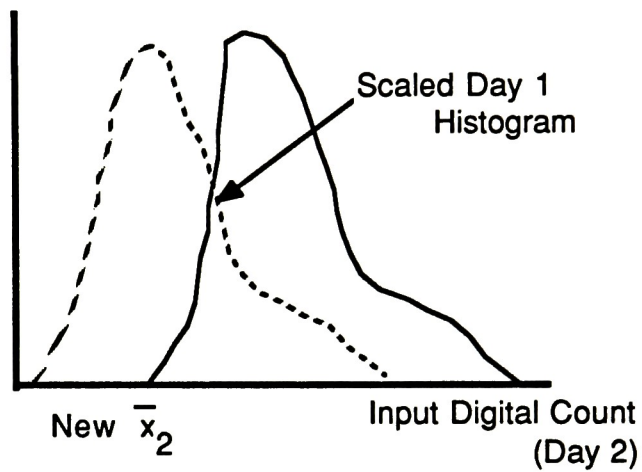
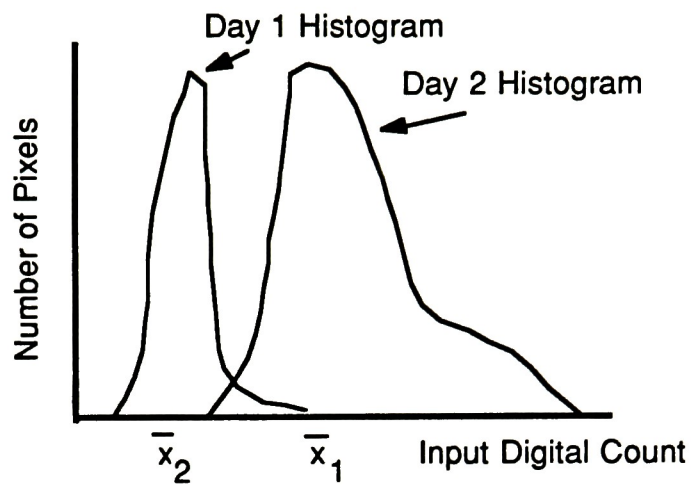


Figure 13 Transformation of linearly related histograms

account for changes in gain and offset introduced by variations in imaging conditions. The conditions described during the development of the PIF transformations have all been of an assumed linear nature and therefore the linear histogram transformation was chosen for use in the rest of this study. Histogram specification will only be used as a quick check on the quality of the original segmentation. If there is a significant difference between the transforms developed using the linear histogram and histogram specification methods in the central data region, it is likely that the original segmentation included some mixed pixels. These mixed pixels will show up in the PIF statistic calculations and will have a significant effect on the developed linear transforms.

The following section will discuss the methodology for the development of an automated segmentation algorithm based on the classical multivariate classification routines discussed in Sections 1.1.5 to 1.1.7.

1.2.3 Automated PIF Segmentation Using Multivariate Techniques

The correction of the effects of a changing atmosphere and of varying illumination and collection geometries on temporally separated images is the primary motivation for this entire study. It is essential to be able to account for and correct these differences if any quantitative processing or change detection is to be carried out on these images. Classically, this has been a problem in the development of multivariate classifiers. Much time and effort is spent on the development of an accurate multivariate classifier. The problem is that the classifier, once it is developed, is accurate only on the particular image being worked with. If the same classifier is applied to another image of this same area acquired on a different day, the classifier would have less success. The success of the classifier may be increased by applying several preprocessors to the image that will reduce the effects of any signal degradation due to the temporal shift. It is the purpose of this section to examine such preprocessors and their effects on the success of multivariate classification as a segmentor of pseudoinvariant features. This should not be viewed as a replacement for the process of image normalization since the following application is developed to classify only a single class of scene elements.

A principal component transformation of multispectral Landsat TM imagery has been shown by Crist and Kauth (1986) to reduce the dimensionality of TM data to three.⁶⁷ According to their analysis, the first principal component can be considered as representing image brightness,

the second as representing greenness and the third as representing wetness. These three principal component vectors characterize 95% of the total variation within the spectral band data (excluding the thermal infrared band). According to Crist and Kauth (1986) the fact that the exact interpretation of the principal components is highly dependent on the sensor being used must be considered. These interpretations mentioned above are for the Landsat TM sensor. These physical interpretations of the first three principal components should prove useful in the identification of urban features. The greenness and wetness component should serve as two ideal vector components to separate urban features from vegetation and water/wetlands.

Texture is one of the most important visual clues to the analyst in the identification of image features.⁶⁸ Texture of image features in Landsat TM data can be represented as the image standard deviation. The effects of varying illumination geometry will tend to cast varying degrees of shade. This will affect the measure of texture for any particular pixel. However, the relative amount of texture at any one pixel should not vary significantly as a function of changing illumination geometries since high texture areas will increase with an increase in shadows, but at the same time the lower texture areas will also increase. Considering this, the relative levels of texture should remain approximately constant. Atmospheric degradation will tend to flatten out the contrast in an image and thus decrease the amount of texture that can be detected, but again this amount of contrast reduction should be constant over the image if the assumption of a homogeneous atmosphere is made at the image plane. For these reasons it is believed that a texture component included in the pixel vector will increase the effectiveness of an

image independent PIF classifier.

A final preprocessing technique which has already been discussed in detail in section 1.2.4 is that of band ratioing. This technique will tend to eliminate the effects of shadows in an image and the variation in upwelled radiance with azimuth.⁶⁹ Due to the increase in the image quality and information due to ratioing, this technique will be useful in feature classification.

It is believed that by using images preprocessed as above in some combination to be empirically determined, an image independent PIF classifier can be developed. The accuracy of the classifier can be adjusted by the use of a *a posteriori* probability screener to allow only those pixels having a certain associated "sureness" to be included in the mask image. In light of the goal of this study, to remove man's intervention from this normalization technique, an unsupervised classifier seems the logical choice as a multivariate technique. With this type of algorithm a number of different image features will be isolated. Identification of the particular pixel collection that represents the pseudoinvariant features can then be accomplished by forming a multidimensional map of the relative positions of the cluster means. If a particular pattern develops in this multidimensional map then the pseudoinvariant feature cluster can be located in an automated fashion by reference to a standard map of typical land cover clusters.

The first attempt to develop an automated multivariate segmentation

algorithm quickly proved to be discouraging at the isolation of pseudoinvariant features. The following is a brief description of the developed algorithm which proved ineffective in the segmentation process but was of great academic interest to the investigator. Once the unsuitability of this technique was recognized, the attempt was abandoned in favor of the rate-of-change algorithm to be described in the following section.

To develop a multivariate segmentation algorithm, the following line of action was pursued. First, the dimensionality of the multispectral data was reduced, primarily to increase the quality of the data used in the algorithm, i.e. to remove the redundancy from the image data. A secondary reason was to increase the computational efficiency of the subsequent segmentation algorithm by reducing the amount of data to be examined since multivariate techniques are computationally very expensive. In the second step, the reduced multispectral data was run through an unsupervised multivariate classification in order to segment the data into spectrally separable classes. The third and final stage of this algorithm was a search of the spectrally separable classes to identify that class or classes that represented pixels that were pseudoinvariant features. Each of the above stages will be discussed and their results presented in Appendix G.

In order to reduce the dimensionality of the multispectral image data, a principal components analysis was chosen based on the material present in the preceding Theoretical Background section. Computer code was written to determine the principal components of six-dimensional image data and to scale the principal component images to fill the full dynamic range of

the image processing display station. The code ran on a DEC VAX 8200 interfaced to a Gould/DeAnza IP8500 Array Processor located at the DIRS laboratory. This code (PrinComp) and a functional description is included in Appendix A. The principal component images were computed for all the scenes described in the previous section (excluding the high resolution CIR airphoto imagery). As had been predicted by other investigators in this area, the first three principal components contained approximately 97% of the variability in the multispectral image data. On this basis it was decided to use only these three principal component images in the unsupervised multivariate classifier. This classifier will separate the pixels into spectrally separated classes.

The unsupervised multivariate classifier was based on the k-means classifier described earlier. The code was designed to run on the equipment mentioned above and this code (named "Cluster") and its functional description is included in Appendix B. The three principal component images for each of the scenes described in the previous section were run through the unsupervised classifier and three-dimensional plots of the cluster means for each scene were made. It was hoped that the investigator could then find a commonality among the distribution of cluster means in each of these plots. After associating each of the spectrally separated clusters with a corresponding land cover type, and after locating the cluster means on the appropriate scattergram, no commonality was found among the distributions. The results leading to this point and a discussion of the reasons for this failure will be discussed in Appendix G.

At this point, it was decided to abandon this approach to the problem of automated segmentation and proceed with the development of the rate-of-change algorithm, which will more closely model the analyst's actions in the previously used interactive segmentation procedure.

1.2.4 Automated PIF Segmentation Using Rate-Of-Change Techniques

To develop an algorithm to automate the first phase of the PIF normalization process, i.e. the segmentation stage, one must understand what is occurring when this segmentation is carried out interactively. If this process can be understood, then there is a good chance that an algorithm can be devised to simulate what an analyst is doing when carrying out a segmentation.

As mentioned in Section 1.2.1, the segmentation procedure of PIF normalization involves the binary thresholding of two separate images (the Landsat TM band-4 to band-3 ratio image and the TM band-7 image). Up to this point, this process was carried out in the following manner. The digital representation of the TM band-4 to band-3 ratio image was displayed on the image processing workstation. Using a joystick, the user could interactively change a threshold value that set all values greater than the current threshold value to zero and left all other values alone. The user observed that there existed a certain range of threshold values where very few pixels would be affected. At some point, this rate of change would increase dramatically and almost all the pixels that represented vegetation would be set to zero as the threshold value changed by a small amount. The rate of change would then decrease dramatically and the falloff of pixels would again be very slow. At this second inflection in the rate of change, i.e. right after the sharp falloff of vegetation pixels, the user would typically stop the thresholding process and thus select the threshold value for the TM band-4

to band-3 ratio image. The TM band-7 image would then be examined on the image processing workstation. The user would then be allowed to change the threshold value on this image, this time setting all digital count values below the current threshold to zero and leaving all others alone. The user typically would observe the following. There would be a sudden drop off of pixels representing water and wet farmland followed by a leveling of this drop off rate. At this inflection point, the user would stop the thresholding process and define the TM band-7 threshold value. The two thresholded images would then be logically combined using an .AND. operator as described previously and the result presented to the user. This is the PIF mask created with the current threshold values. The user would have the option to go back and fine tune the threshold values to obtain the best PIF mask possible.

Several problems exist with the segmentation procedure just described. One such problem is consistency between analysts in choosing threshold values. Every person that would utilize the PIF normalization technique would choose different values of the thresholds for the two images. Some users were conservative in their choice, that is they would go further than the required threshold values to be sure that all the unwanted pixels are eliminated. Other users would be liberal in their choice, that is they would stop thresholding just before the inflection point to be sure that there were enough pixels for the distributional computations. The result of such inconsistency in the choice of thresholds caused different users to obtain transforms that yielded normalizations with different degrees of accuracy. Another problem with the technique described above is that a single analyst may tend to be conservative in the choice of threshold values

on images of areas that are unfamiliar geographically while the same user may be liberal on images of familiar geographical areas. This would cause inconsistency in the choice of threshold values by the same analyst.

With the above observations in mind, a conclusion can be drawn that there may be a certain region of allowable threshold values in each of the thresholding curves. To locate this area, it was decided to observe the change that occurred in the number of pixels as thresholding proceeded. This seemed the most likely choice of criterion to observe since this is what the user is looking at visually when carrying out the interactive segmentation.

Figure 14 is a three dimensional surface plot of the above mentioned phenomenon. The x-axis represents the threshold value chosen for the TM band-4 to band-3 ratio image, the y-axis represents the threshold value chosen for the TM band-7 threshold value and the z-axis represents the number of image pixels that remain on in the image resulting from a logical .AND. of the two binary thresholded images. Just as expected, there is a monotonic decrease in the number of pixels left on as the threshold values eliminate pixels from their respective input images. To locate the area where the threshold values should fall, the points where the rate-of-change of pixels mentioned above levels off must be found.

As mentioned above, Figure 14 represents the number of pixels included in the PIF mask formed as a result of using any combination of threshold values in the two defined regions. In order to find the positions in

these thresholding regions where the rate-of-change in the number of pixels goes through the aforementioned inflection points, the two-dimensional derivative (gradient) is computed for the surface in Figure 14. The computational form of the operator is:

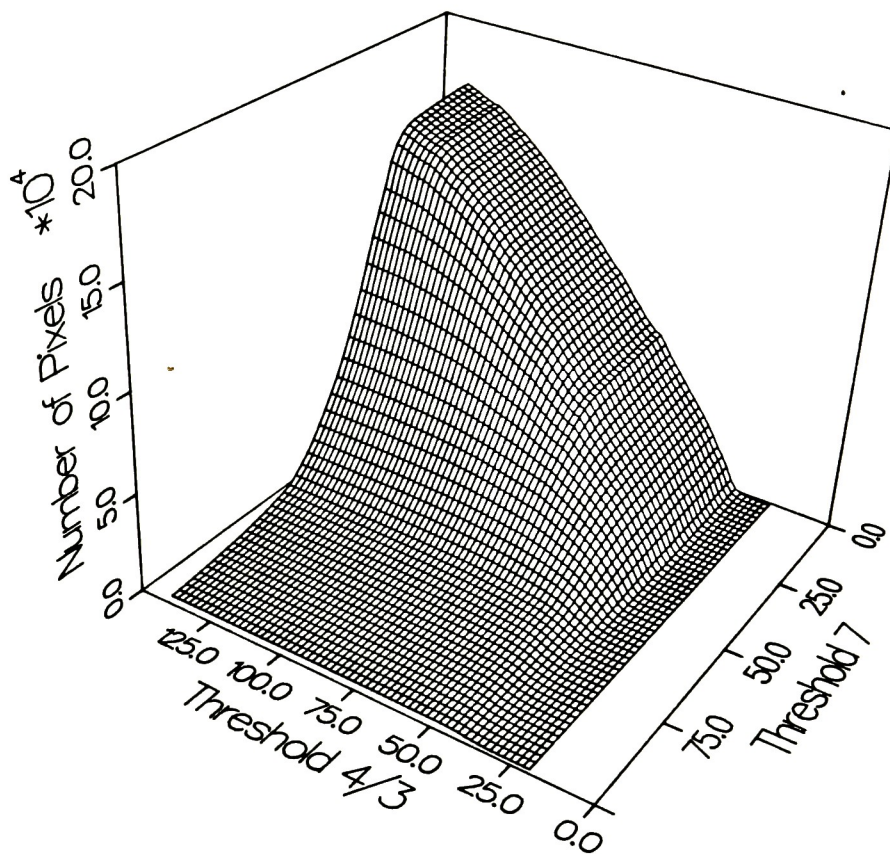


Figure 14 Number of pixels remaining on in the logical combination of the individual thresholded images

$$G_{i,j} = \frac{[(x_{i,j-1} - x_{i,j+1}) + (x_{i-1,j} - x_{i+1,j})]}{2}$$

where $G_{i,j}$ is the gradient value at threshold value position i,j and x is the number of pixels remaining on at the specified threshold positions (see Figure 15). Figure 16 is a three- dimensional representation of the surface derived using the above operator on the surface in Figure 14.

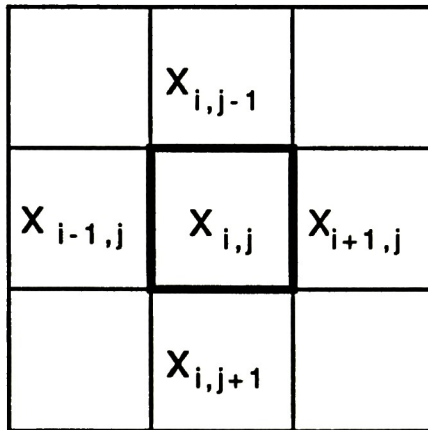


Figure 15 Illustration of the threshold positions used in gradient calculation

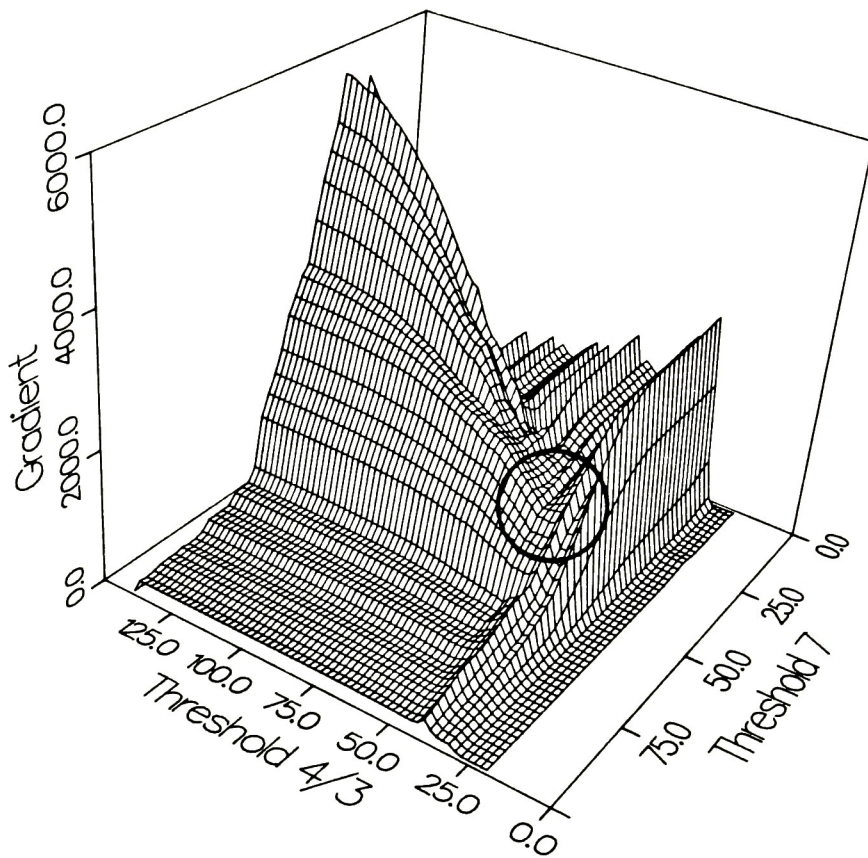


Figure 16 The gradient surface derived from Figure 14

As the rate-of-change of the number of pixels slows down, the value of the two-dimensional surface in Figure 16 will decrease (i.e. the value of the derivative of a function decreases as the rate-of-change of the original function decreases). The gradient surface is then evaluated to locate the point at which the rate-of-change slows down or goes through a minimum in both thresholding directions simultaneously. The circled area in Figure 16 indicates the localized region where the simultaneous minima in rate-of-change occurs. It is in this area where the two threshold values are defined. This observed behavior is the basis for the development of the automated rate-of-change algorithm.

2.0 Experimental Approach

Now that the theory has been established for the automated segmentation of Landsat Thematic Mapper imagery, the following tasks are necessary to validate the theory. First is the development of the automated algorithm for the segmentation of pseudoinvariant features from Landsat TM imagery. This task will be carried out by use of classical multivariate classification techniques as well as the rate-of-change information from the thresholding process that is currently part of the PIF technique. This latter method will be referred to as the interactive technique. Once the algorithmic approach has been established, several questions arise as to its accuracy and precision. The second task will deal with the question "How well does the transformation work?" This task will be explored by examining the accuracy of the resulting transformations, both after applying the automated and the traditional interactive segmentation processes. A final question that arises is "How well can the results of the transformations be repeated?" This is one of the original reasons for this study since the requirement for repeatable, user-independent results is a must for a technique such as pseudoinvariant feature normalization. This question will be addressed by an examination of the precision of the results of this normalization technique using the compilation of data that has been collected since this technique was first proven viable. The results of this study will then be compared to those results obtained using the automated segmentation algorithm. Another consideration that will be explored concerning the precision of the transformation processes is the effectiveness of the technique in the development of transformations for different scenes imaged through

identical atmospheres. Again this will be done in such a manner so that the automated and interactive techniques can be simultaneously compared. The algorithmic development as well as the questions listed above will be addressed in the following section and the outcome of this endeavor discussed separately in the Results section.

2.1 Selection of Appropriate Imagery

The choice of imagery for this study was limited to that data which was available at the Digital Image and Remote Sensing laboratory (DIRS). Three temporally separated Landsat TM scenes were available with acquisition dates 9/13/82 (TM-4), 6/22/84 (TM-5) and 5/24/85 (TM-5). The first two were selected for study for several reasons. The first involved the prevailing atmospheric conditions of the imagery. The 9/13/82 (TM-4) imagery was acquired through a very hazy atmosphere which tended to decrease the overall dynamic range for this scene. The 6/22/84 (TM-5) and 5/24/85 (TM-5) imagery, on the other hand, were taken through a much clearer atmosphere resulting in greater image contrast.. A second consideration was that previous work by Volchok⁷⁰ to validate this normalization technique, also utilized the earlier of the two dates listed above. If the current study were to utilize the database collected by Volchok, as well as take advantage of the difference in atmospheric conditions, this choice of imagery seemed the most suitable.

The imagery above is stored on 1600 bpi computer compatible

magnetic tape at the DIRS laboratory. These tapes were originally obtained from the Earth Resources Observation Satellite Center (EROS) in Sioux Falls, South Dakota. This study utilized full resolution 512 x 512 pixel images extracted from the original 5966 by 6968 pixel Landsat TM scenes. These 512 x 512 pixel subsections were extracted from the magnetic tape utilizing the computer program LT4Read, the code for which is given in Appendix E. This code runs on the Digital Equipment Corporation (DEC) VAX 8200 located at the DIRS laboratory.

Two mid-size urban areas located in upstate New York were extracted from the scenes. These included downtown Rochester and downtown Buffalo. Also extracted was a scene located 512 pixels to the west of the downtown Rochester image. This scene is dominated by rural farmland and contains no large urban development center. This scene was chosen to assure identical atmospheric conditions to its neighboring scene. The pertinent image data for the above scenes are summarized in Table 1.

Also chosen for this study were two high resolution color infrared (CIR) airphotos of downtown Buffalo taken on 7/6/70 and 6/7/72. These photographs were obtained from the National High Altitude Photography Program (NHAP) of the United States Geological Survey (USGS), Denver, Colorado. These CIR transparencies were digitized using an Eikonix 78/99 Digitizer Camera System interfaced to a DEC MicroVAX GKS Workstation. The digitized images were 512 x 512 pixels in dimension with each pixel representing approximately a 10 meter IFOV. These images will be used in a side study to determine how well the automated segmentation routine

works in conjunction with the PIF normalization technique on higher resolution photographic imagery. The 1972 image will be transformed to look like the 1970 image and will then be compared to the results obtained with the previously used interactive method.

Table 1 Summary of pertinent TM image data used for this study

Date	9/13/82	6/22/84
Sensor	TM-4	TM-5
Scene ID	E-40059-15244	E-50113-15260
Approximate		
Acquisition Time	10:30 am (EST)	10:30 am (EST)
Landsat	Path D-17	Path D-17
Scene Coordinates	Row 30	Row 30
Sun Elevation Angle	44°	59°
Sun Azimuth Angle	141°	122°
Downtown Rochester	Row 2401	Row 2350
Subscene	Column 5925	Column 3200
(Segmented coordinates)*		(Quadrant 3)
Rural Rochester	Row 2450	Row 2390
Subscene	Column 5400	Column 2727
(Segmented coordinates)*		(Quadrant 3)
Downtown Buffalo	Row 3900	Row 900
Subscene	Column 2750	Column 2750
(Segmented Coordinates)*		(Quadrant 3)

* Row and Column denote the coordinates of the upper left corner of the 512 x 512 pixel subscene

2.2 Rate-of-Change Segmentation Algorithm

The development of the rate-of-change algorithm for automated segmentation of pseudoinvariant features is a direct application of the theory previously described. The automated segmentation algorithm is described in the following. The actual computer code (BldPIF) and functional description of its operation is included in Appendix C. The results of this algorithm and a comparison of its accuracy and precision are addressed separately in the Result section which follows.

The TM band-4 to band-3 (infrared to red) ratio image is computed and displayed along with the TM band-7 image. The images are thresholded in a systematic manner and combined using a logical .AND. to form a three-dimensional surface similar to that displayed in Figure 14. This surface represents the number of pixels included in the logical combination image as a function of threshold values. The gradient of this three-dimensional surface is computed to form a second surface similar to that of Figure 16. The dependent variable of this surface is the gradient of the previous surface as a function of threshold values. The next stage of this process is the examination of the gradient surface to locate the position of the appropriate threshold values (i.e. the area where the surface has a local minimum in both directions). The final stage of this process is the application of these two threshold values to the original TM band-4 to band-3 ratio and TM band-7 images, and the subsequent logical combination of these thresholded images using an .AND. operator to obtain the PIF mask. The algorithm makes heavy use of the Gould/DeAnza IP8500 Array

Processor and it is not suggested that this algorithm be applied serially in a mainframe computer. A comparison of the computational time difference can be found in Appendix C.

The first and second stages of the rate-of-change algorithm involve the preparation of the TM band-4 to band-3 and TM band-7 images for thresholding. First the TM band-4 and TM band-3 images are loaded into computer memory. The two images are then divided pixel by pixel (band-4 pixel divided by band-3 pixel) in order to form a floating point quotient image in RAM. The maximum quotient value is found and a scaling factor determined by dividing 255 (the maximum digital count in an 8-bit system) by this maximum quotient value. This scaling factor must be found to scale the quotient image so that the full dynamic range of the image display is filled. The quotient image is then multiplied pixel by pixel by this floating point scaling factor to define the final band-4 to band-3 ratio image. This image is then truncated to 8-bit data and displayed in a single image plane. The TM band-7 image is loaded into a second image plane.

The third stage of the algorithm is systematic thresholding of the two images prepared as above. Two image transformation tables (ITT's) are developed to threshold the appropriate images. An ITT is a mapping function which transforms a pixel gray level to a new value. The first ITT, which will be applied to the TM band-4 to band-3 ratio image, is filled with a one-to-one mapping function from 0 up to the current threshold value. After this point the ITT is filled with values of 0 up to its upper limit (i.e. 255). The second ITT, which is applied to the TM band-7 image, is filled with values of

0 up to its current threshold value. After this point it is filled with a one-to-one mapping function. The first ITT will set all digital count values (DC's) above the current threshold to 0 while not affecting all other DC values. The second ITT will do just the opposite, that is, set all DC values below the current threshold value to 0 while leaving all other DC values alone. These ITT's are applied to the appropriate images as follows. The TM band-4 to band-3 threshold is set to its current value and applied to the image. The TM band-7 threshold is then set and applied to its image. The two images are then logically combined using an .AND. operator and the result stored in a third image plane. The histogram is then determined using the Digital Video Processor (DVP) in the IP8500. The value for the zero histogram bin is then subtracted from the value $512^2 = 262144$ to obtain the number of pixels that have a non-zero value in the logical combination image. This value becomes part of the three-dimensional surface. The TM band-7 ratio is stepped through its determined range at increments of 2 DC values before the TM band-4 to band-3 threshold is again changed. This process is then repeated until the TM band-4 to band-3 ratio threshold has been stepped through its determined range. A flowchart of this process is depicted in Figure 17.

The thresholding ranges are determined in the following manner. Past interactive segmentation results demonstrated that the threshold values always fall within the following interval

$$\begin{array}{ll} \bar{X} \geq T_{\text{value}} \geq \bar{X} - 2.0\sigma & \text{for the TM band-4 to band-3} \\ & \text{ratio image} \\ \bar{X} + \sigma \geq T_{\text{value}} \geq \bar{X} - 2.5\sigma & \text{for the TM band-7 image} \end{array}$$

where \bar{x} is the mean and σ the standard deviation of the histogram of the TM band-4 to band-3 ratio or the TM band-7 images and T_{value} is the respective threshold value chosen using the interactive segmentation process. Figure 18 shows the location of

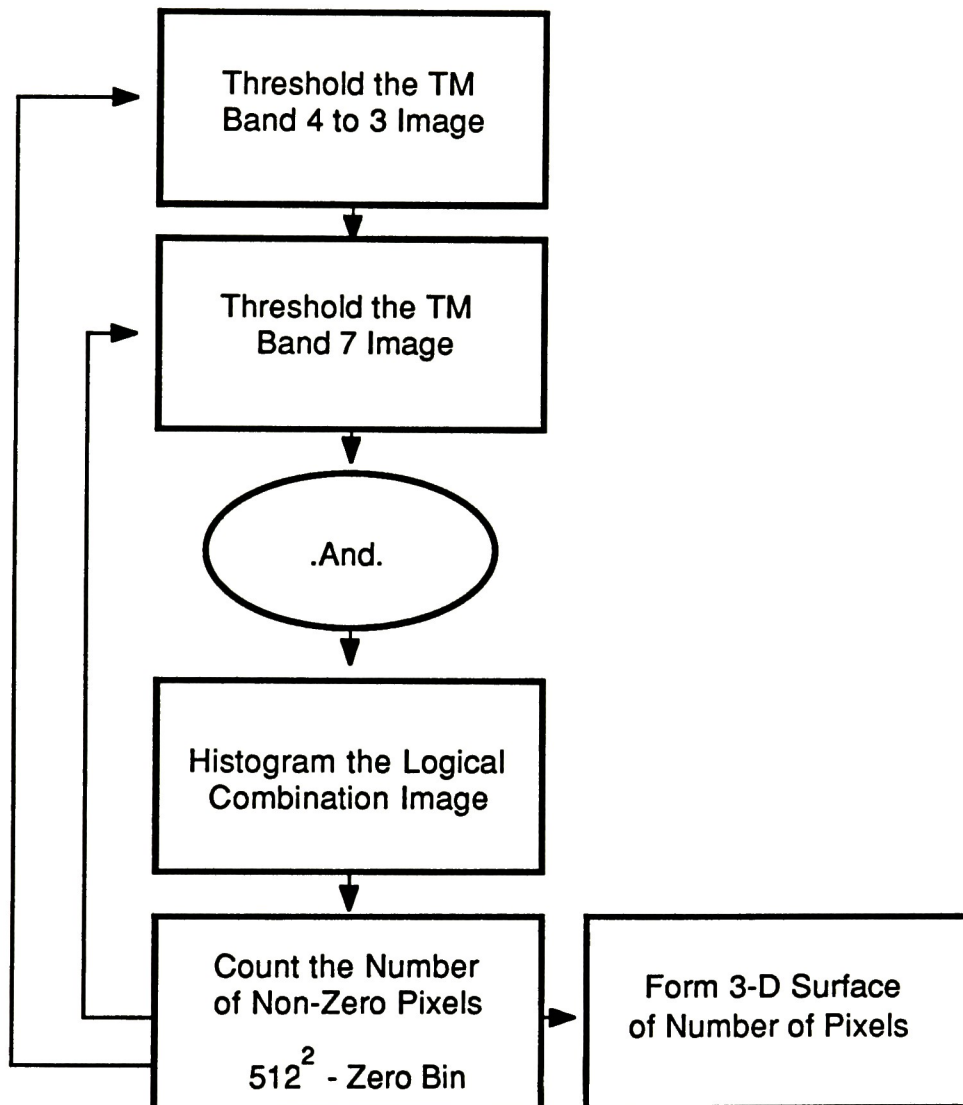


Figure 17 Flowchart of thresholding / 3-D surface building algorithm

these threshold values. Based on this information, the thresholding ranges are determined by taking the histogram of each of the images to be thresholded, determining the histogram statistics, and calculating the ranges according to the above formula.

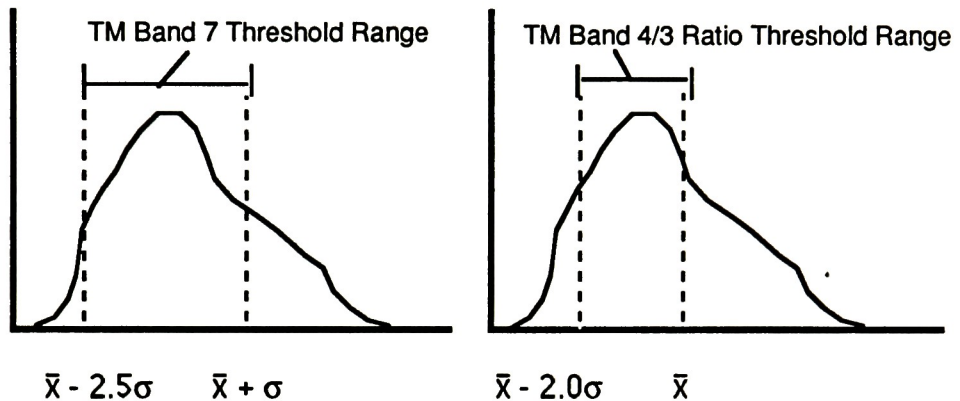


Figure 18 Illustration of the specified thresholding range

The fourth stage of the segmentation algorithm is the computation of the gradient of this surface as discussed in the Theoretical Background section. These calculations are performed serially on the DEC VAX 8200 since these surfaces are typically very small (i.e. on the order of a 30 by 30 matrix).

The fifth stage of the algorithm is the determination of the threshold values locations on the gradient surface. As described in the Theoretical Background section, the circled area in Figure 16 represents the area where the appropriate threshold values fall. This has been determined by the evaluation of previously chosen threshold values using the interactive segmentation technique and then locating their position on their respective gradient plots.

Next, this position is found in an automated fashion. If one looks at the plot in Figure 16, it is apparent that the tall fin-like structure in the surface parallel to the TM band-4 to band-3 ratio threshold axis points directly at this region, which happens to be a localized minimum at the bottom of this structure. This was true for all of the images selected for this study. To locate this spot in an automated fashion, the program searches along the maximum value in the TM band-4 to band-3 threshold range parallel to the TM band-7 threshold axis for a maximum gradient value. The TM band-7 threshold value at which this maximum occurs is the band-7 threshold value, which always occurs at the peak of this fin-like structure. After this value is found, the program then searches parallel to the TM band-4 to band-3 threshold axis from the top of this fin-like structure downward. When the algorithm finds the first local minimum along this scan, it knows the position of the appropriate band-4 to band-3 ratio threshold value. This scanning procedure can be seen in Figure 19.

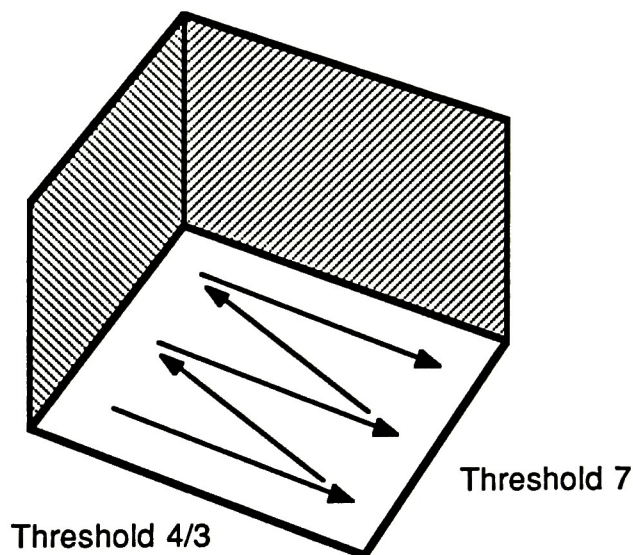


Figure 19 Illustration of the pattern used in the thresholding procedure

The sixth and final stage of the segmentation algorithm is the formation of the PIF mask image. The previously determined threshold values are used to form the two ITT's as described in stage three. These ITT's are applied to the appropriate images, namely the original TM band-4 to band-3 ratio image and the TM band-7 image. The resulting thresholded images are logically combined using an .AND. operator and the resulting image is transformed to a binary image (i.e. all pixels with non-zero values are assigned a DC value of 255 while all pixels with a DC value of 0 are left alone). The result is the PIF mask where all pixels that are high (i.e. DC value of 255) represent pseudoinvariant features, while all other pixels represent non-PIF pixels.

The accuracy and precision of this algorithm is the focus of the following sections. Before these can be examined, it is necessary to complete the PIF normalization process since all subsequent analysis will be based on the results of this normalization procedure.

2.3 PIF Transform Determination

The development of the PIF transformations has previously been described in the Theoretical Background section. Once determined, these transformations are applied to one set of images to make them appear as if taken under identical imaging conditions as another image. For this discussion the image set that is to be transformed will be referred to as the

Day-1 image. The image set that serves as a reference for the transformation will be called the Day-2 image. A TM image set consists of TM bands 1,2,3,4,5 and 7. The thermal infrared band (TM band-6) was excluded since the radiance reaching the sensor in this wavelength region (10.2 to 12.4 μ m) is not a direct function of the reflectivity for an object; an essential caveat for this procedure. It is important to note that not all six of the bands listed above are needed for the transformation process; only an infrared and a red band are actually required. This will be demonstrated when the high resolution CIR airphoto image is PIF-transformed.

The determination of the PIF transform has two distinct phases: the determination of the spectral (digital) distribution of pseudoinvariant features and the derivation of the transforms. First, the PIF mask is loaded into a single image plane of the array processor. A single band of the TM data set is loaded into a second image plane. A third image is created by performing a logical .AND. of the single band image and the PIF mask. The resulting image contains only pixels that represent pseudoinvariant features. These pixels possess the same digital count values that they had in the original single band image. A histogram is taken of this resulting PIF image with the DVP in the array processor. The resulting histogram has a zero bin that is artificially enlarged since everything that is not a PIF in the single band image has been assigned a digital count of 0. This histogram value at zero is reassigned to 0 and the histogram statistics are calculated as follows:

$$\bar{x} = \frac{\sum_{i=1}^n DC_i}{n}$$

and

$$\sigma = \sqrt{\frac{n \sum_{i=1}^n DC_i^2 - \left(\sum_{i=1}^n DC_i \right)^2}{n(n-1)}}$$

where \bar{x} is the mean and σ is the standard deviation of the histogram, n is the number of non-zero pixels in the PIF image, and DC_i is the i^{th} pixel's digital count value. The histogram statistics are calculated for the associated PIF images in each band image for both days. These statistical values determine the PIF transforms for each band of the Day-1 image set.

For each band in both image sets the linear histogram transformations are calculated as follows. The transformation is of the form

$$DC_2 = m_t DC_1 + b_t$$

where, as stated earlier, DC_1 and DC_2 are the corresponding digital count values in the Day 1 and Day 2 band image, respectively, and m_t and b_t are the slope and intercept term of the corresponding linear histogram transformation. These are defined as follows:

$$m_t = \frac{\sigma_2}{\sigma_1} \qquad b_t = \bar{x}_2 - m_t \bar{x}_1$$

The linear histogram transform coefficients are determined for each image band. The transformations are applied to the Day-1 images by forming a linear ITT with the form described above and applying these ITT's to the appropriate images. The computer code that accomplishes the above ("Normalize") is described in Appendix D.

The next major phase of this study is to examine the questions "How good is the accuracy of the resulting PIF transforms?" and "How precise are the transforms that are derived?".

2.4 Accuracy of the PIF Transformations

The accuracy of the PIF transformations developed during the normalization process will be evaluated by choosing several identical targets in the image sets of the two days and evaluating how closely the transformed Day-1 targets match the corresponding Day-2 targets. This evaluation, will be referred to as a control point analysis, and is conducted in the following manner. Several sets of control points are collected from each of the images according to several criteria. The analyst will look for man-made objects that ideally occupy several pixels in area, i.e. large parking lots, warehouse rooftops, etc. These objects should span as much of the dynamic range of the image as possible such that the error determined is representative of the normalization process over the full dynamic range of PIF pixels. Digital count values are then obtained for these identical objects in each band of the Day-2 and transformed Day-1 images. The raw error associated with the normalization can be expressed as

$$\epsilon_k = \left[\frac{\sum_{i=1}^n (DC_{ik} - DC_{ik}')^2}{n} \right]^{1/2}$$

where ϵ_k is the raw r.m.s. error in the normalization associated with the k^{th} band, DC_{ik} is the digital count value of the i^{th} point in the k^{th} band of the Day 2 image, DC_{ik}' is the transformed digital count value of the i^{th} point in the k^{th} band of the Day 1 image, and n is the total number of control point pairs chosen from the two images.

Choosing in-scene targets as a means for a control point analysis leads to several problems in the evaluation of the error in the normalization process. First, the reflectance of the element could change during the temporal separation between the two images. Second, the analyst may not choose exactly the same portion of the target on the two different dates. Thirdly, the targets chosen may be too small and may thus cause mixed pixel effects, as happens when the ground spot size of the sensor encompasses more than the target of interest. The brightness value then becomes a function of the brightness of surrounding objects as well as the target brightness. These effects will be referred to as sampling errors and the magnitude will be assessed to estimate their magnitude. Using the same control points involved in the determination of the raw error above, a linear regression is performed between the digital counts of the targets in the Day-1 image and the digital counts of the corresponding targets in the Day-2 image. This regression is of the form:

$$DC_{ik} = m_k DC_{ik}' + b_k$$

where DC_{ik} and DC_{ik}' are as described above, and m_k and b_k are the slope and intercept terms from the linear regression for the k^{th} band. Under the assumption that true PIF pixels in the two day's images are linearly related, the residual error in this regression analysis is the sample error described above. The distribution of this residual error should be random about the regression line and is of the form:

$$\epsilon_{\text{samp}} = \left[\frac{\sum_{i=1}^n (DC_{ik} - \hat{DC}_{ik})^2}{n} \right]^{1/2}$$

where DC_{ik} is the digital count for the i^{th} target on the Day-2 image, \hat{DC}_{ik} is the predicted digital count value for the Day-2 image, and n is as before the number of targets studied. Considering this assessment of the magnitude of the error due to sampling and knowing the estimate of the raw error determined earlier, the error due exclusively to the PIF normalization can be defined:

$$\epsilon_{\text{PIF}} = (\epsilon^2 - \epsilon_{\text{samp}}^2)^{1/2}$$

where ϵ_{PIF} is the error in digital count after the PIF normalization, ϵ is the raw error from the control point analysis immediately after the transformation and ϵ_{samp} is the error due to the sampling errors discussed previously.

All the errors are expressed in units of digital count values. This is only a relative means of evaluating the error resulting from a normalization process. In order to establish an absolute scale on which to express error, the above digital count values must be transformed to some other unit which is constant between images. Such a unit is the reflectance of the ground objects. In order to transform an error expressed in digital count values to reflectance units, the analyst must make an estimate as to the reflectivity of the control point targets as they are chosen. The digital counts of the Day-2

image are then regressed against their corresponding estimated reflectance values (best-guess estimates of reflectance are used based on standard reflectance curves). The slope of this regression line is an expression of digital counts per reflectance unit. The errors listed above are transformed to errors in reflectance units by the following expression:

$$\epsilon_r = \frac{\epsilon_{DC}}{\alpha_k}$$

where ϵ_r is the error expressed in units of reflectivity, ϵ_{DC} is the error expressed in units of digital counts, and α_k is the slope of the regression line of digital count as a function of reflectance.

The methods described above were used to describe the errors associated with the PIF normalizations of the Rochester and Buffalo Landsat TM images. This analysis was not used on the rural Rochester scene since too few large PIF features could be located to perform a justifiable study, but since this scene was taken under "identical" imaging conditions as the urban Rochester scene, direct comparison of the two sets of transformations will indicate the relative accuracy of these rural transformations. The digital count error was computed for all image bands that were transformed, however, the reflectance unit conversion was performed only on TM bands 1 through 4. The conversion was performed only on these bands since these were the only bands where the reflectance of the targets could be accurately estimated. Errors typically encountered using this method for the normalized images were of the order of one to two reflectance units in the visible and

two to three percent reflectance in the near IR wavelength regions.

This section has described a method for estimating the effectiveness of a normalization procedure that is not unique to the PIF normalization process but which can be used with any normalization technique. The results of this analysis on the current study for the normalizations utilizing the automated and interactive segmentation processes appear in the Results section. The following section will describe the methods used to establish the precision of the PIF normalization process utilizing both the automated and interactive segmentation processes.

2.5 Precision of the Normalization Process

A problem that has plagued the PIF normalization technique prior to this study has been the lack of uniformity in accuracy achieved by different investigators. As was earlier alluded, each user of this technique tended to form a different PIF mask. This is the only point at which difference in the developed transforms could be introduced. This validates the requirement of a consistent segmentation technique. It is clear that the use of the automated segmentation algorithm will yield normalization results with the same accuracy each time it is run on an image set. The phenomenon that will be examined here is if the degree of accuracy achieved on one set of images is consistent with the degree of accuracy on other image sets.

There are two directions from which this question will be approached.

The first is to take the results of the control point analysis on the three heavy urban areas and compared their resulting accuracies in each image band. No rigorous statistical analysis can be conducted on this data set due to the small sample size, however, the apparent consistency, or lack of it, will be discussed. The second approach will test two aspects concerning the precision of the PIF technique by comparing the results of the rural Rochester normalization with the results obtained from the adjacent heavily urban scene. Since these images are assumed to be taken through the same atmosphere, the resulting transforms should be identical. The results of this comparison will further substantiate the precision (or lack of it) for the PIF normalization technique as well as test the robustness of the technique to the degree of urbanization in the images to be normalized.

The identity of two transforms can be tested using the following criterion. In Figure 20, let the transform labeled T_1 represent the transform for the urban Rochester imagery and let T_2 be the transform for the rural Rochester imagery. The error at any point between these two transforms ϵ_i can be determined as $|T_{1i} - T_{2i}|$ where T_{1i} and T_{2i} are the values of the respective transforms for the i^{th} digital count value. The overall error between the two transforms can be computed as

$$\epsilon = \left[\frac{\sum_{i=1}^{255} \epsilon_i^2 n_i}{n} \right]^{1/2}$$

where ϵ is the overall error between the two transforms, ϵ_i are the individual

errors weighted by the number of pixels n_i in the i^{th} histogram bin of the Day-1 image, and n is the total number of pixels in the histogram. Using the above approach, the overall error associated with applying one transform or another to an entire scene can be computed on a band-by-band basis.

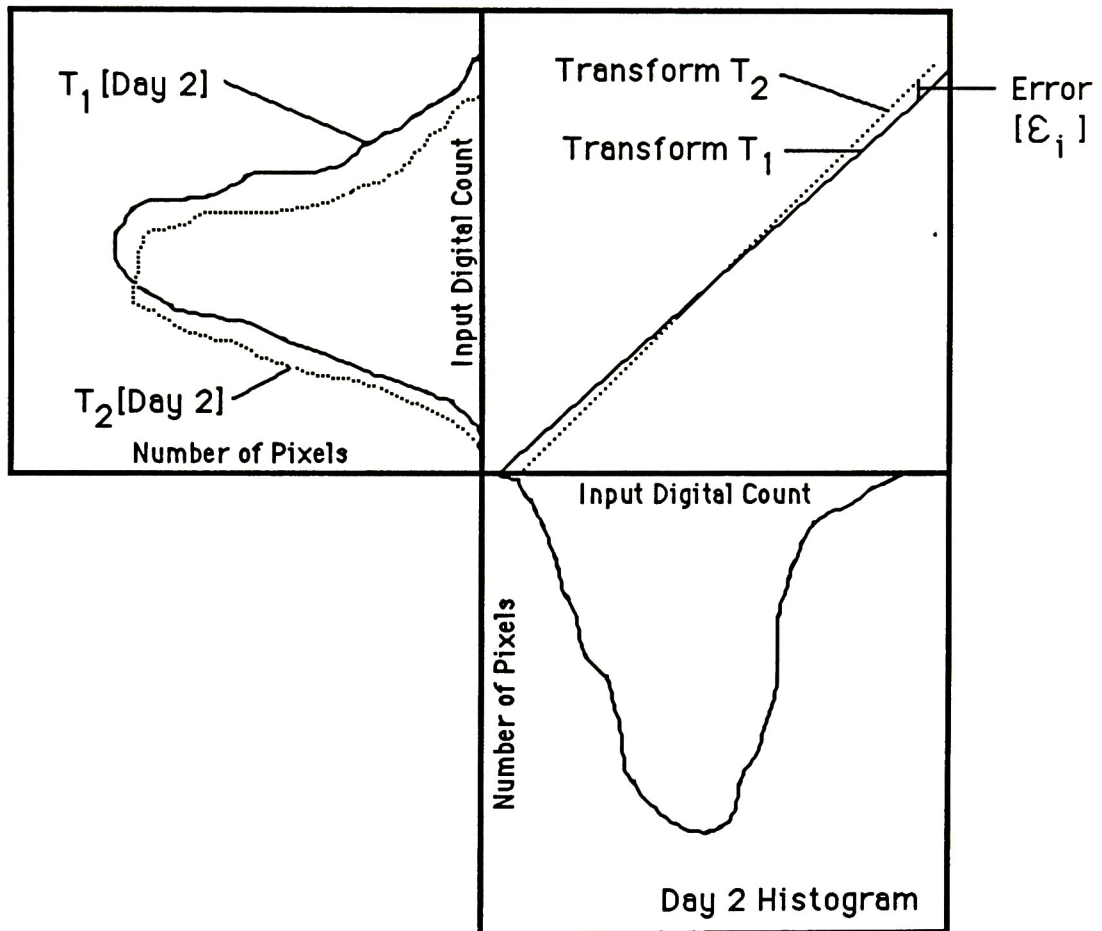


Figure 20 Error exhibited between two linear histogram transformations

All the analyses described above were conducted on the results of normalizations carried out using the automated as well as the interactive segmentation procedures. This is a limited study of the precision of the normalizations that result from this technique, but it should give the reader an idea of what can be expected when this technique is further utilized. For this study, errors of less than 10% of the total dynamic range of PIF digital counts were typical.

2.6 Application of Segmentation Algorithm to High Resolution Imagery

The above analysis has been conducted using Landsat TM imagery as data sets for the segmentation and normalization algorithms. In this section, a further test of the robustness of both algorithms will be considered by testing their effectiveness on high-resolution airphoto imagery. The imagery used in this study are the two NHAP CIR airphotos described previously (refer to Appendix I for possible problems encountered when digitizing photographic imagery). The segmentation process will be carried out identically to that described for the TM imagery. The investigator will be required to replace the TM band-4 to band-3 ratio image with the infrared to red ratio obtained by dividing the red and green image bands of the CIR transparency. The TM band-7 image will be replaced by the infrared information contained in the red image band. The normalization process are carried out in an identical fashion to that described above. The results can

then be evaluated by conducting a control point analysis choosing the data set from the CIR airphoto imagery. The results obtain from this analysis will be compared to those results from the above control point analyses.

3.0 Results

The results obtained from the pseudoinvariant feature normalization technique are largely dependent on the quality of the initial segmentation of PIF's from the original imagery. To date, this process has been carried out in a user interactive fashion. Immediate problems arising from such a process are user-to-user inconsistency and familiarity effects. The familiarity of the user with the study area aids in the segmentation process greatly. Conversely, if the area under study is foreign to the analyst, the resulting segmentation may be less reliable. This study aimed to eliminate these inconsistencies from the segmentation process. The method developed will automate this procedure, thus eliminating user-to-user inconsistency as well as removing the effect of scene familiarity.

The success in segmenting the imagery in this study is a difficult subject to quantify out of the context of the normalization procedure. Therefore, the quality of the segmentation will be evaluated in a comparative fashion. Segmentation using the developed automated algorithm will be carried out in parallel with segmentation utilizing the interactive thresholding process. The complete normalization procedure will be performed using both the above segmentation results. Side-by-side comparisons can then easily be made.

3.1 Segmentation of PIF Features

Segmentation of pseudoinvariant features from the imagery in this study was performed using the interactive thresholding process described in Section 1.3.1 as well as the automated segmentation algorithm defined in Section 2.3. The aim of any automation process is to mimic the results of the manual process previously carried out. Table 2 is a summary of the threshold values obtained using both the interactive and automated segmentation algorithms.

The first notable observation from the data in Table 2 is that the automated segmentation algorithm consistently produced threshold values that were more conservative than those chosen by the analyst when using the interactive process. By conservative, it is meant that the thresholds chosen for the TM band-4 to band-3 ratio were consistently lower than those chosen using the interactive process (i.e. more high brightness count pixels were eliminated using the automated process) and conversely the threshold values chosen for the TM band-7 images were consistently higher than those from the interactive process (i.e. more low brightness count pixels were set to zero). This characteristic of the automated segmentation algorithm causes the resulting PIF mask to contain fewer pixels, thus indicating that the number of mixed pixels composed of PIF and non-PIF scene elements will be lower than the number remaining after the interactive process. This conservative aspect also tends to eliminate useful data from the PIF mask. Due to the decrease in the number of pixels remaining in the PIF mask, urban features that may have been kept when using the

Table 2

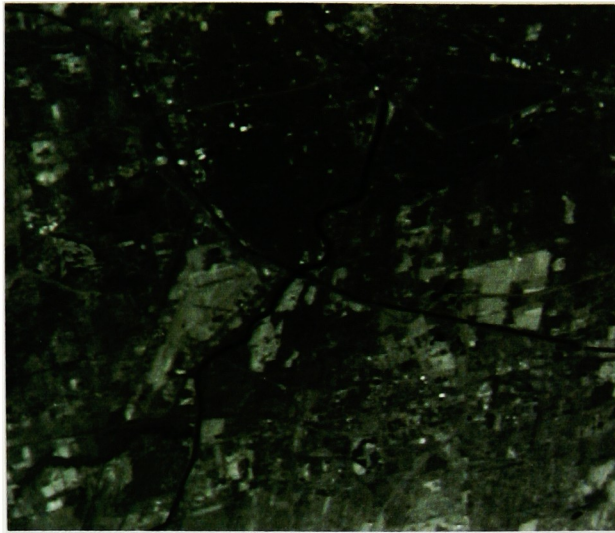
Summary of threshold values obtained for the segmentation of urban features
using the automated and interactive segmentation algorithms

Scene	TM 4/3 Ratio Threshold			TM Band 7 Threshold		
	Automated 1982	Automated 1984	Interactive 1982	Interactive 1984	Automated 1982	Interactive 1982
Urban Rochester	69	39	81	62	24	18
Urban Buffalo	55	43	59	47	27	23
Rural Rochester	80	33	81	62	23	18
			IR/R Ratio Image		IR Band Image	
			1970	1972	1970	1972
High Res. Buffalo	80	72	97	82	124	58

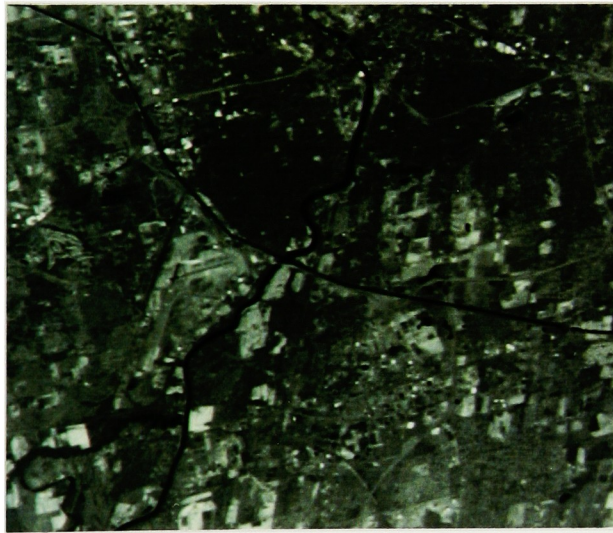
interactive process are eliminated. This can hurt the resulting normalization since the dynamic range of PIF brightness values is decreased, thus leaving more chance for error to develop in the resulting transforms.

A comparison of the PIF masks developed using the interactive and automated segmentation algorithms is shown in Figures 25 through 28 for the Landsat TM imagery of urban Rochester, urban Buffalo and rural Rochester and the NHAP high-resolution airphoto imagery of Buffalo. A monochrome representation of the original images on which each mask is based on are shown in Figures 21 through 24. Figures 29 through 32 depict the three-dimensional surfaces representing the number of pixels as a function of threshold values as well as the corresponding gradient surfaces used to determine the appropriate threshold values for each of the PIF masks shown. A summary of the histogram statistics used to determine the thresholding ranges as described in Section 2.2 for these plots is contained in Table 3 for the TM and the high-resolution airphoto imagery.

To compare the quality of segmentation obtained from the automated algorithm compared to the interactive method, the respective PIF masks were used to complete the PIF transformation process. The resulting accuracy and precision of these transformations were then used as a metric for the quality of the segmentation process. As considered here, accuracy is a measure of how well the normalization worked, i.e. the magnitude of the control point errors. Precision makes reference to the robustness of the technique, i.e. whether or not the technique maintains the same level of accuracy for all image types.

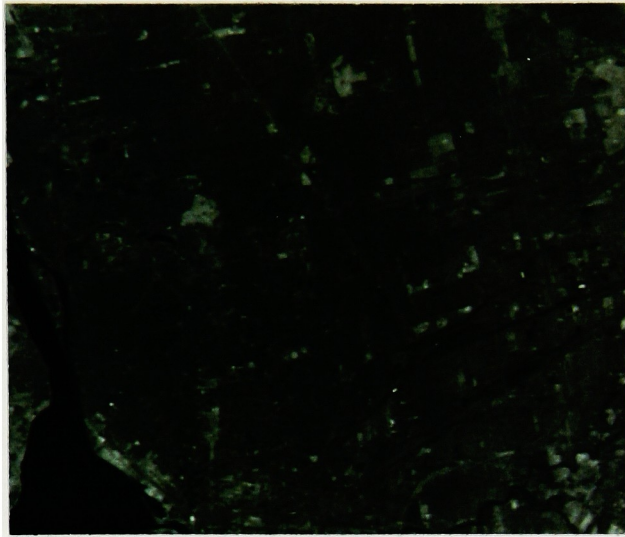


1982 Image



1984 Image

Figure 21 Original 1982 and 1984 urban Rochester Image

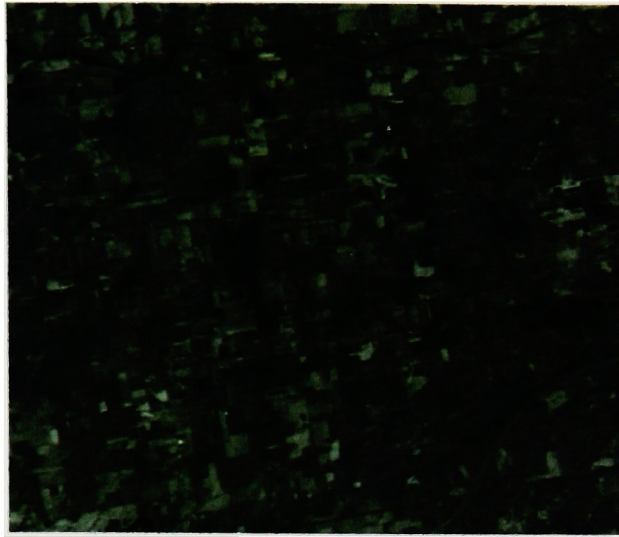


1982 Image

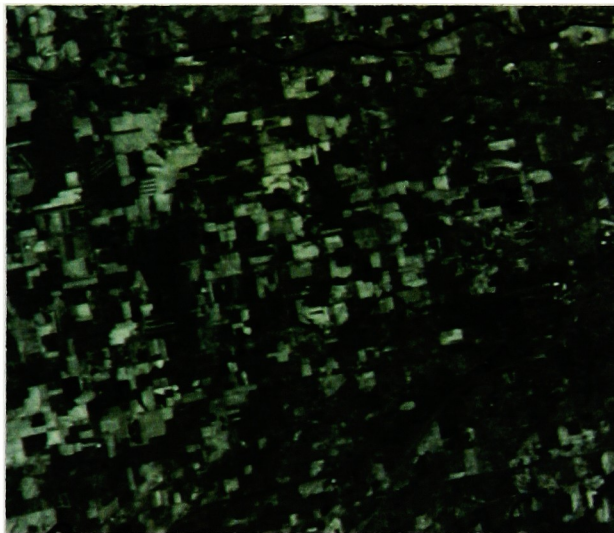


1984 Image

Figure 22 Original 1982 and 1984 urban Buffalo Image



1982 Image

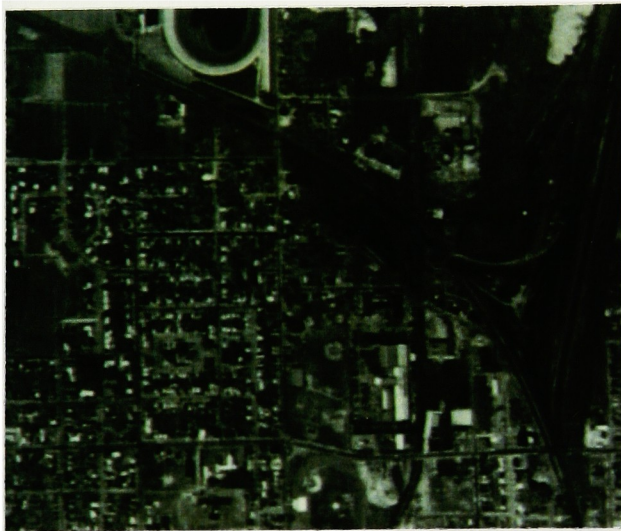


1984 Image

Figure 23 Original 1982 and 1984 Rural Rochester Image



1970 Image

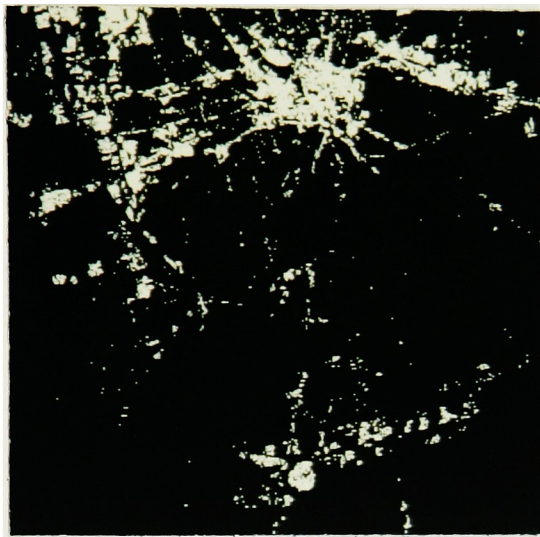


1972 Image

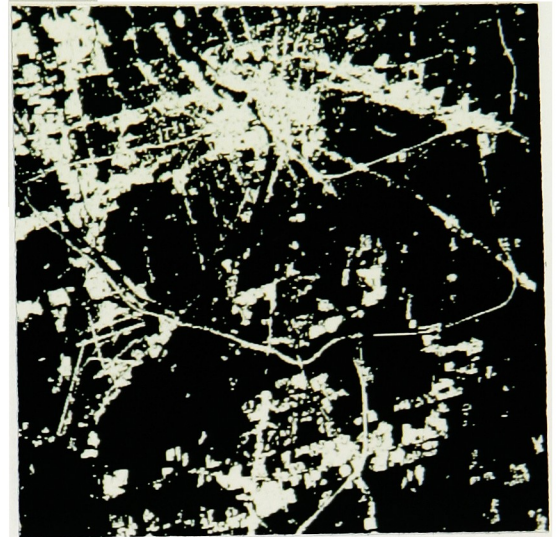
Figure 24 Original 1970 and 1972 high-resolution airphoto image

Histogram statistics utilized to determine the thresholding ranges used for the automated rate of change segmentation algorithm

83



1982



1984

(a)



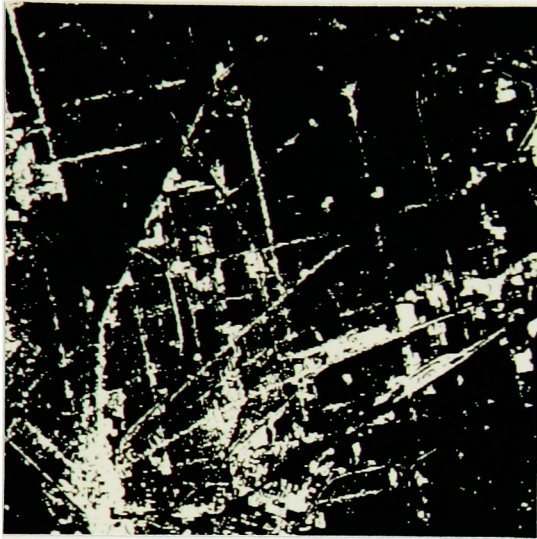
1982



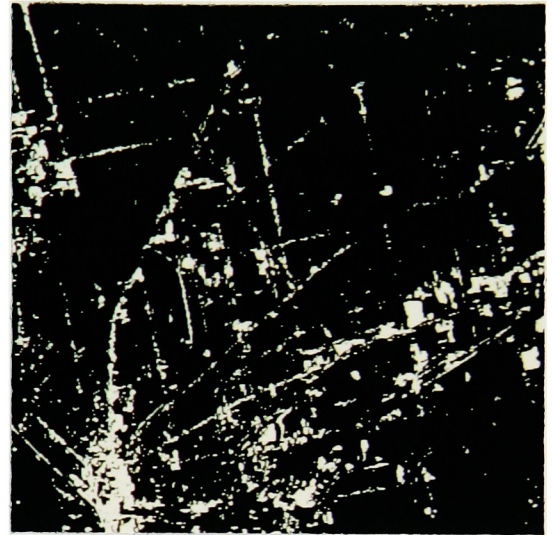
1984

(b)

Figure 25 PIF masks created for the 1982 and 1984 urban Rochester images using (a) the automated and (b) the interactive segmentation processes

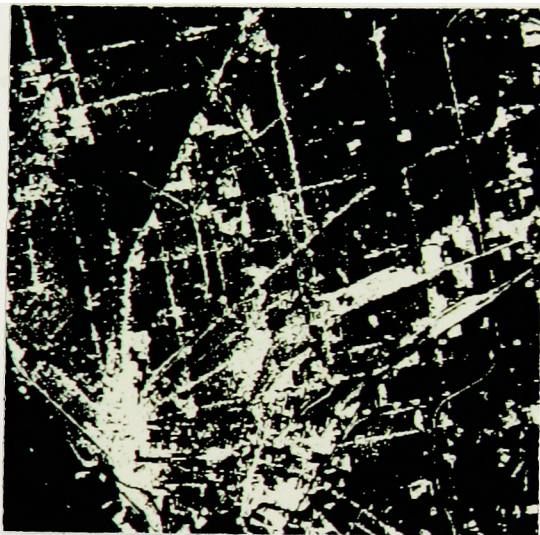


1982



1984

(a)



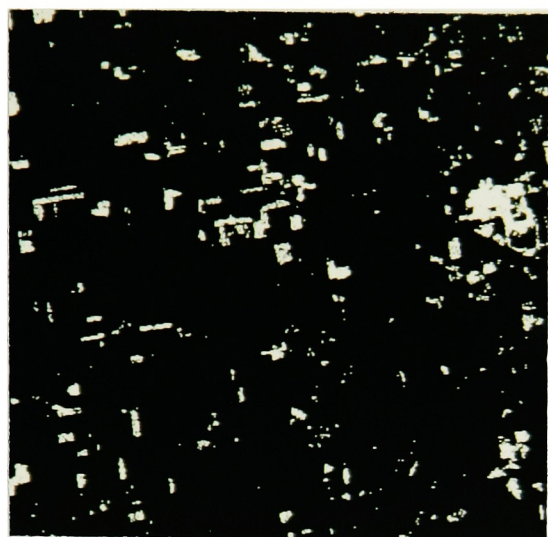
1982



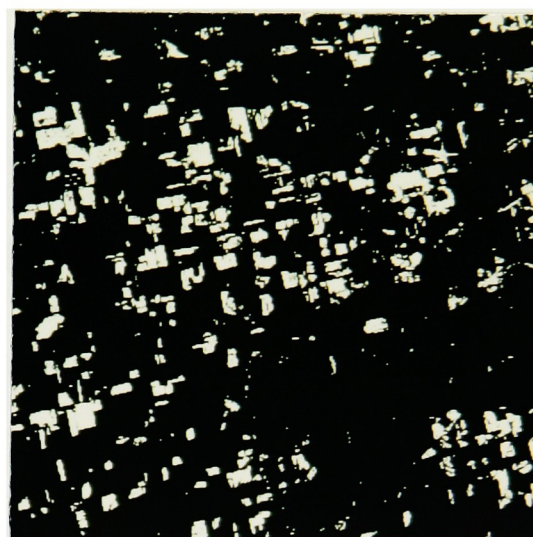
1984

(b)

Figure 26 PIF masks created for the 1982 and 1984 urban Buffalo images using (a) the automated and (b) the interactive segmentation processes

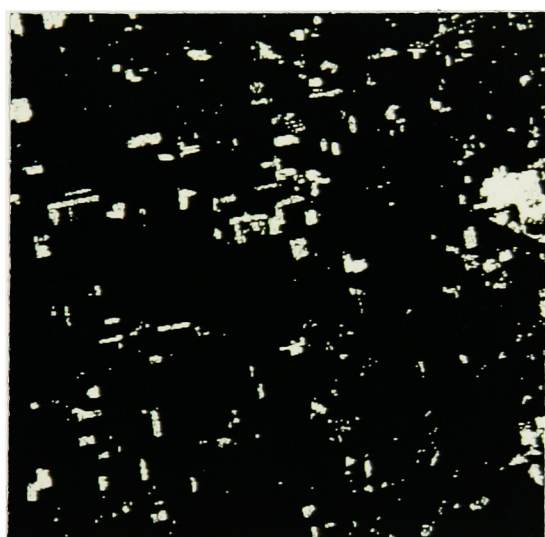


1982

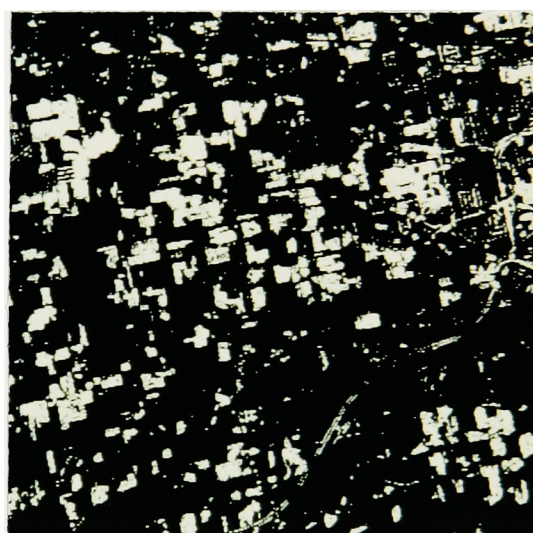


1984

(a)



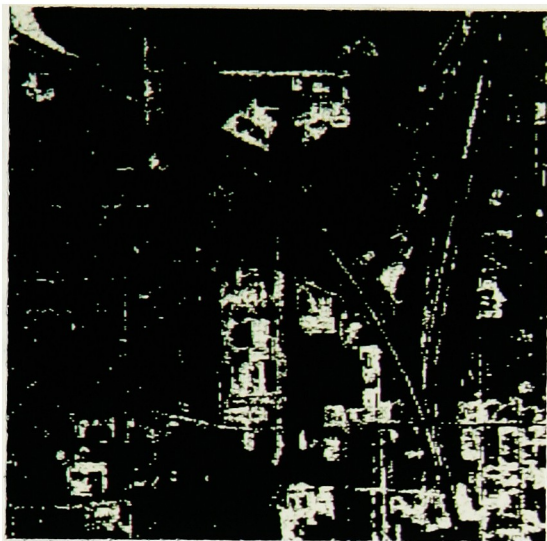
1982



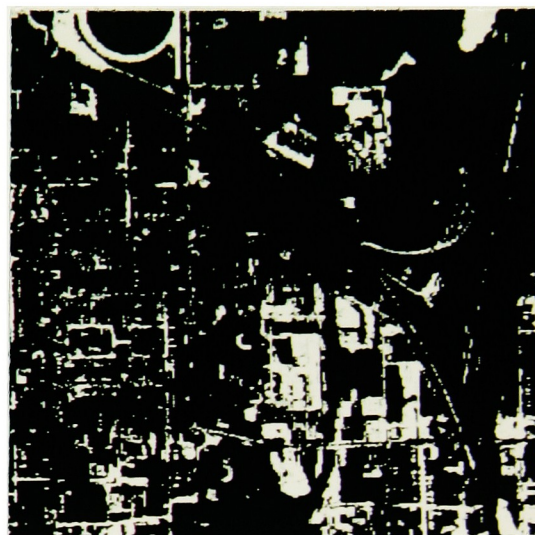
1984

(b)

Figure 27 PIF masks created for the 1982 and 1984 rural Rochester images using (a) the automated and (b) the interactive segmentation processes

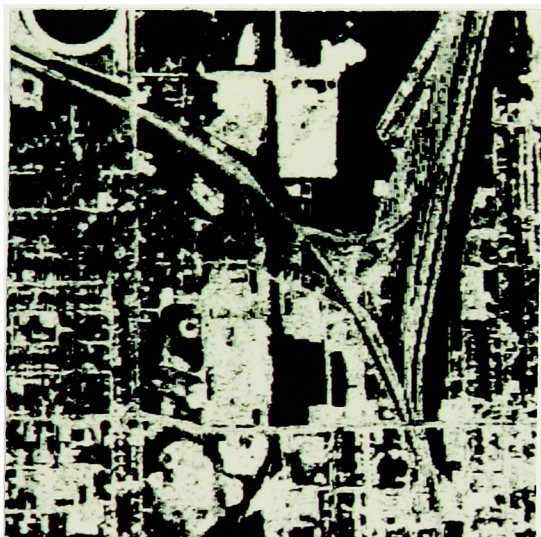


1970



1972

(a)



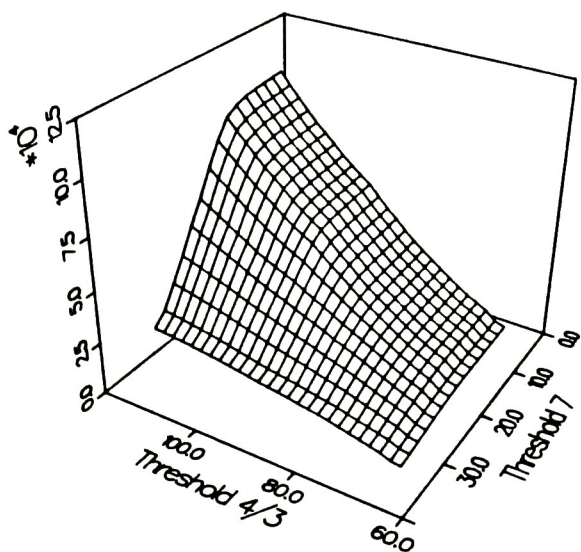
1970



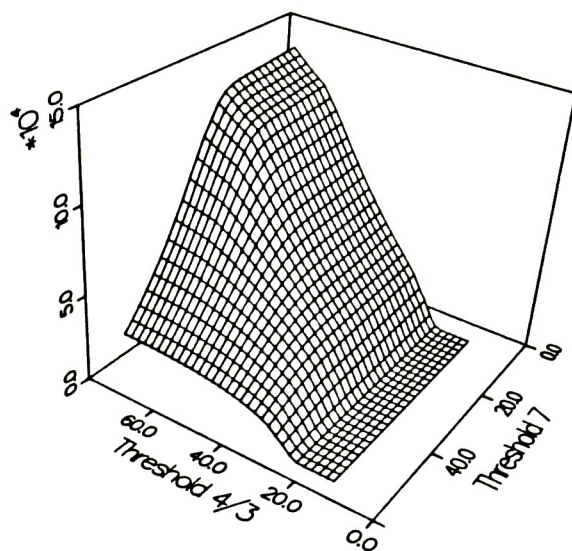
1972

(b)

Figure 28 PIF masks created for the 1970 and 1972 NHAP airphoto images of Buffalo using (a) the automated and (b) the interactive segmentation processes

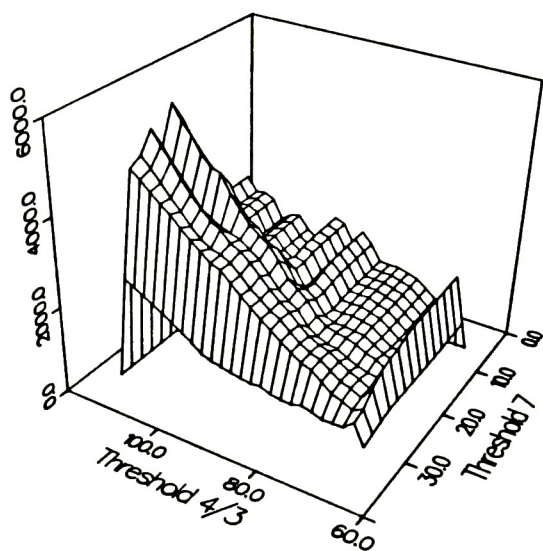


1982

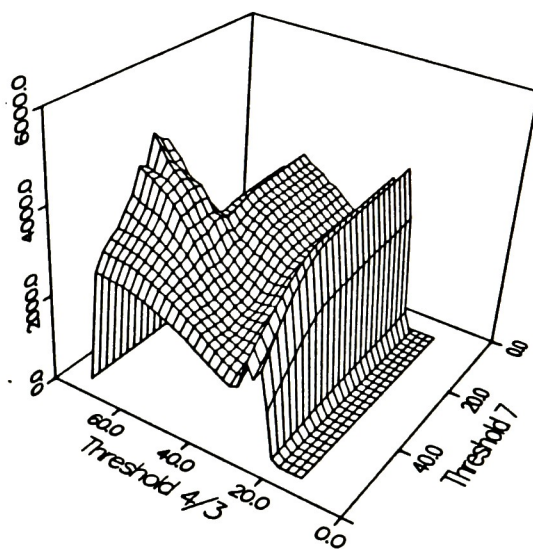


1984

(a)



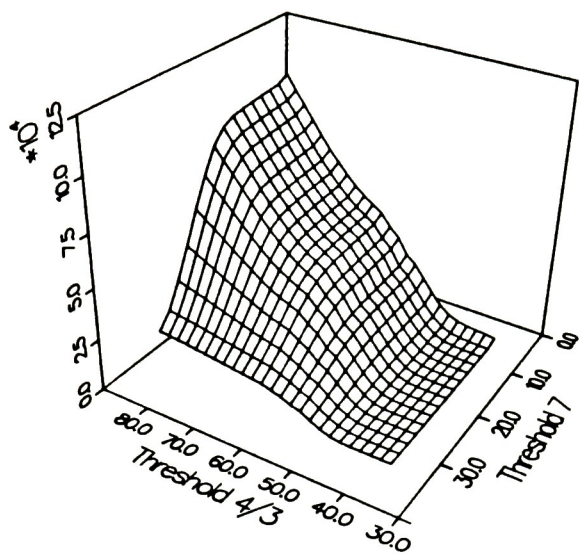
1982



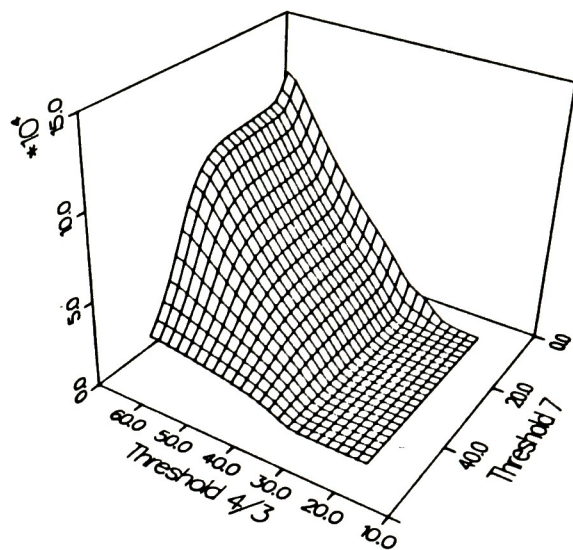
1984

(b)

Figure 29 Three-dimensional surfaces representing (a) the number of pixels as a function of threshold values and (b) the gradient of the surface in (a) for the 1982 and 1984 urban Rochester images

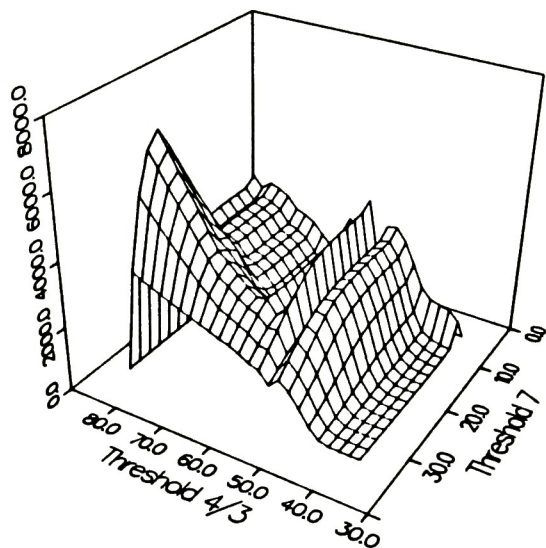


1982

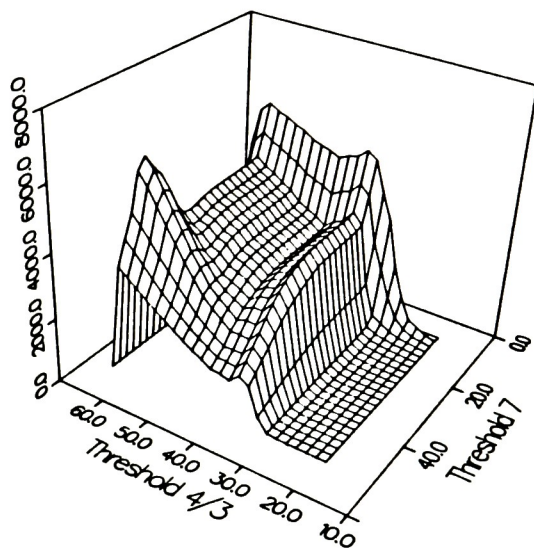


1984

(a)



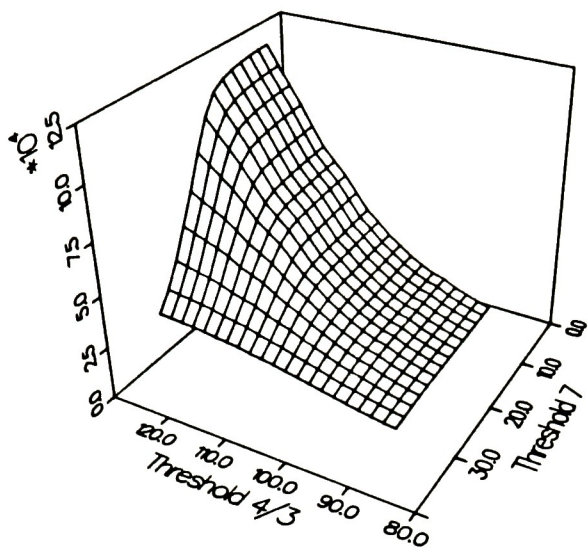
1982



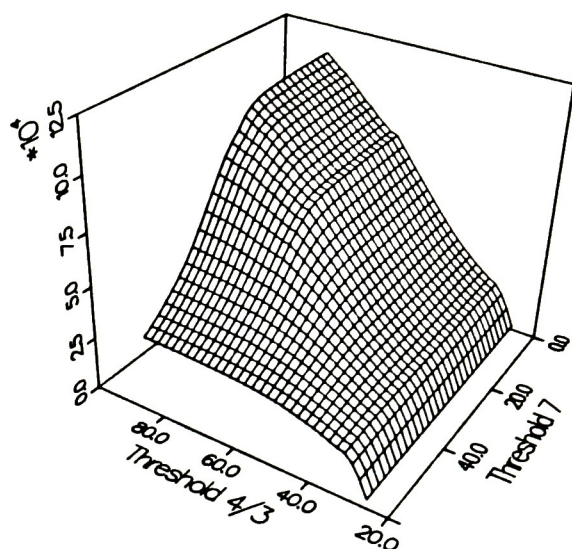
1984

(b)

Figure 30 Three-dimensional surfaces representing (a) the number of pixels as a function of threshold values and (b) the gradient of the surface in (a) for the 1982 and 1984 urban Buffalo images

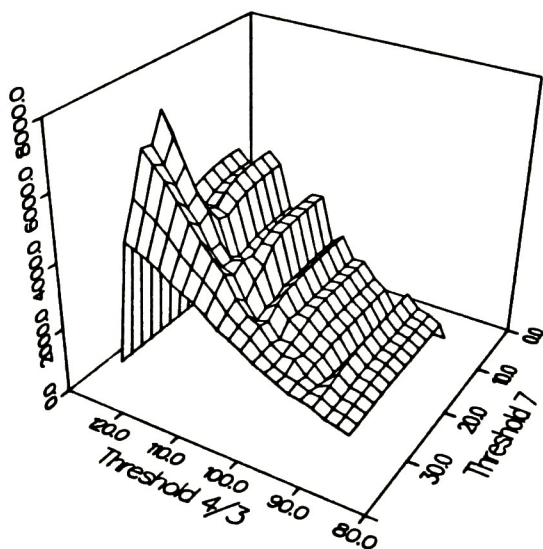


1982

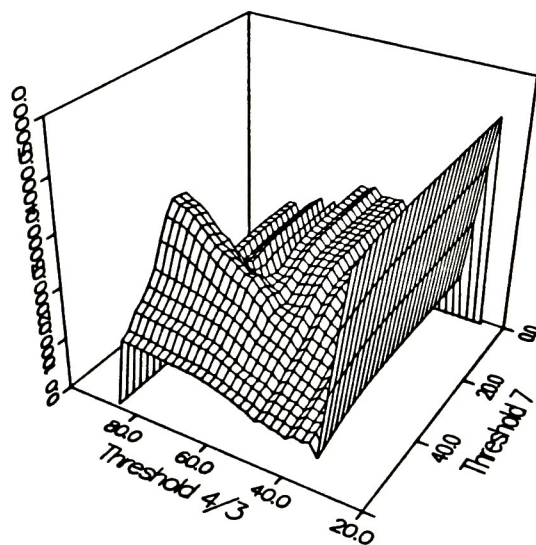


1984

(a)



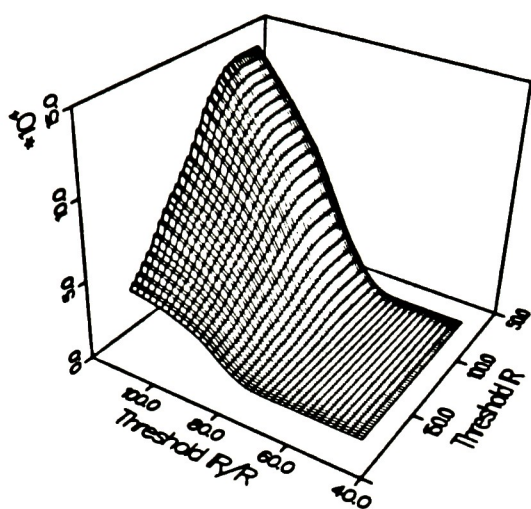
1982



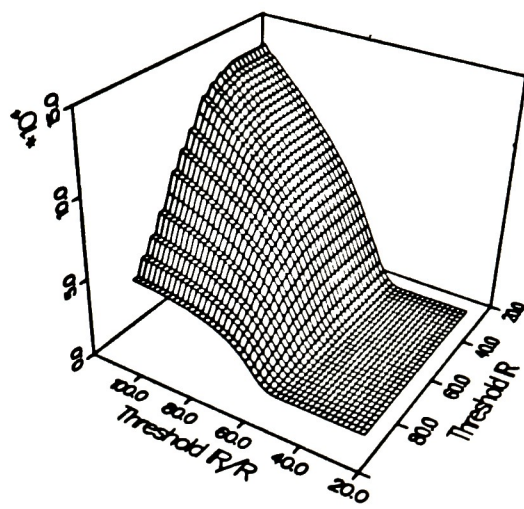
1984

(b)

Figure 31 Three-dimensional surfaces representing (a) the number of pixels as a function of threshold values and (b) the gradient of the surface in (a) for the 1982 and 1984 rural Rochester images

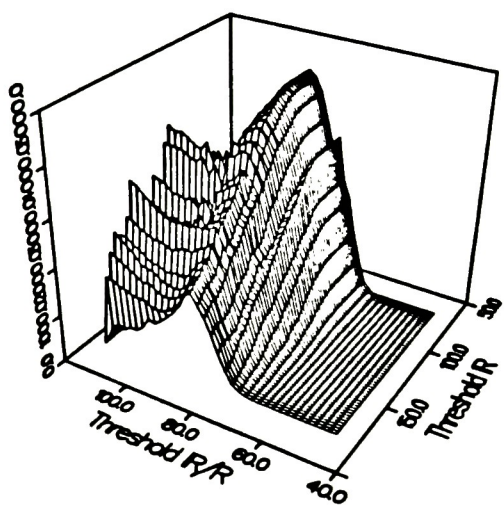


1970

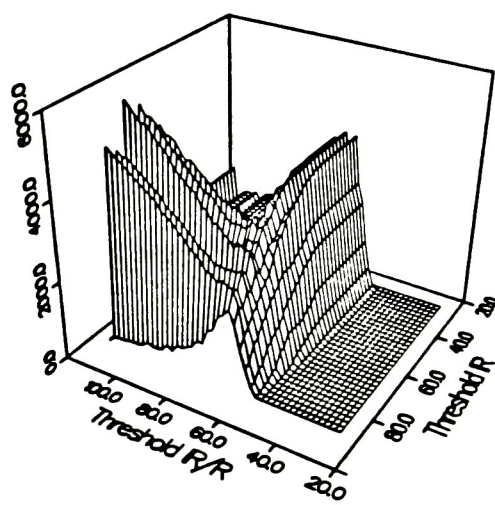


1972

(a)



1970



1972

(b)

Figure 32 Three-dimensional surfaces representing (a) the number of pixels as a function of threshold values and (b) the gradient of the surface in (a) for the 1970 and 1972 NHAP airphoto images of Buffalo

3.2 Development of PIF Transformations

The PIF masks developed in the previous section using the automated and interactive segmentation procedures were used to calculate statistics of the PIF histogram and subsequent PIF transformations for each of the band images of the urban Rochester, urban Buffalo, and rural Rochester TM data sets as well as the NHAP airphoto data of Buffalo. The PIF histogram statistics and the developed linear histogram transformations are summarized in Tables 4 through 7 using both of the segmentation procedures. Figure 33 compares the linear transforms developed using the automated and interactive segmentation techniques for the urban Rochester TM imagery. As can be seen from these plots and the data summarized in Tables 4 through 7, the transformations are nearly identical over the range of digital count values defined by the dynamic range of the PIF pixels. It should be noted that the errors between the transforms developed using these two methods tend to be slightly higher at the longer wavelengths (i.e. in the far infrared band (TM band-5 and band-7)). The reason for this increase in error is the effect of mixed pixels. Mixed pixels, especially those containing PIF and vegetation scene elements, tend to increase the error between these transforms more in longer wavelength regions since the reflectance of vegetation is so high there. Since there is a mix of high and low reflectance objects, the contribution of these pixels will erroneously affect the PIF histogram statistics more in these wavelengths than in the shorter bandpass regions where the reflectances of these scene elements are more closely matched. Aside from this observation, there is high equivalence between the developed transforms using the individual

segmentation approaches.

Figures 34 and 35 show the urban Rochester data set and the NHAP data set before and after the normalization procedure. The TM data of Rochester is a color infrared composite with the infrared image information displayed in the red display channel, the red information in the green channel, and the green image information in the blue display channel. The NHAP airphoto imagery is also a CIR image digitized from an original CIR transparency.

Table 4

Summary of temporal PIF image and transformation data
for the TM urban Rochester imagery

TM Spectral Channel	PIF Histogram Statistics										Linear Transformations*			
	Automated					Interactive					1982 to 1984		Automated	
	1982		1984		1982	1982		1984		1984	1982 to 1984		Automated	
	\bar{x}	σ	\bar{x}	σ		\bar{x}	σ	\bar{x}	σ		m	b	m	b
1	107.7	8.4	127.3	17.7	103.9	6.9	121.4	15.2	2.12	-100.6	2.20	-107.4	2.12	-100.6
2	44.6	5.2	57.8	11.2	42.7	4.4	54.4	9.7	2.16	-38.5	2.22	-40.1	2.16	-38.5
3	46.9	7.5	67.2	15.8	43.7	6.5	61.9	14.3	2.09	-30.9	2.18	-33.5	2.09	-30.9
4	44.1	7.8	69.0	15.4	48.4	7.8	70.6	15.3	1.99	-18.6	1.98	-24.9	1.99	-18.6
5	61.0	16.5	100.7	26.1	60.6	15.6	95.4	25.3	1.58	4.1	1.62	-2.5	1.58	4.1
7	35.3	10.7	60.3	18.8	31.8	10.2	54.4	18.1	1.76	-1.8	1.78	-2.3	1.76	-1.8

* m and b are the slope and intercept of the linear transformation function

Table 5

Summary of temporal PIF image and transformation data
for the TM urban Buffalo imagery

TM Spectral Channel	PIF Histogram Statistics								Linear Transformations*			
	Automated				Interactive				Automated		Interactive	
	1982		1984		1982		1984		m	b	m	b
	\bar{x}	σ	\bar{x}	σ	\bar{x}	σ	\bar{x}	σ				
1	102.3	9.1	137.9	16.1	100.2	8.3	134.3	14.7	1.78	-43.7	1.77	-42.5
2**	-	-	-	-	-	-	-	-	-	-	-	-
3	45.4	7.5	68.1	12.8	43.4	7.2	64.9	11.7	1.70	-9.2	1.62	-5.3
4	40.8	7.1	64.3	11.9	40.3	7.2	64.2	11.7	1.67	-4.0	1.63	-1.5
5	59.3	12.1	88.5	17.0	56.5	13.0	84.8	16.7	1.40	5.5	1.28	12.5
7	34.5	7.5	53.5	11.3	32.1	8.1	49.9	11.0	1.51	1.5	1.36	6.5

* m and b are the slope and intercept of the linear transformation functions

** the band 2 images were unable to be read from the magnetic tape

Table 6

Summary of temporal PIF image and transformation data
for the TM rural Rochester imagery

TM Spectral Channel	PIF Histogram Statistics								Linear Transformations*			
	Automated				Interactive				1982 to 1984		Interactive	
	1982		1984		1982		1984		Automated		m	
	\bar{x}	σ	\bar{x}	σ	\bar{x}	σ	\bar{x}	σ	m	b	m	b
1	102.8	5.0	120.1	11.4	102.9	5.2	116.9	11.1	2.29	-115.3	2.14	-103.1
2**	-	-	-	-	-	-	-	-	-	-	-	-
3	48.0	5.2	74.6	10.7	48.1	5.5	69.4	12.5	2.09	-25.6	2.26	-39.4
4	58.7	6.6	78.9	11.1	58.1	7.4	82.9	13.8	1.79	-20.7	1.86	-25.2
5	83.7	15.9	128.4	19.7	83.2	17.4	124.1	22.7	1.26	22.4	1.30	16.1
7	47.2	11.5	81.5	15.6	47.4	12.3	74.4	19.5	1.38	16.1	1.58	-0.3

* m and b are the slope and intercept of the linear transformation functions

** the band 2 images were unable to be read from the magnetic tape

Table 7

Summary of temporal PIF image and transformation data
for the high resolution airphoto imagery of Buffalo

CIR Spectral Region	PIF Histogram Statistics										Linear Transformations*			
	Automated					Interactive					1972 to 1970		Automated	
	1970		1972			1970			1972					
	\bar{x}	σ	\bar{x}	σ	\bar{x}	\bar{x}	σ	\bar{x}	σ	\bar{x}	m	b	m	b
IR	151.3	20.9	96.3	14.2	162.1	35.0	84.5	18.3	1.48	9.1	1.91	0.6		
R	143.9	18.8	94.6	12.1	137.2	28.1	80.0	17.2	1.55	-3.0	1.63	6.6		
G	150.9	11.3	87.6	5.5	144.6	18.3	80.4	8.8	2.07	-29.9	2.07	-22.1		

* m and b are the slope and intercept of the linear transformation functions

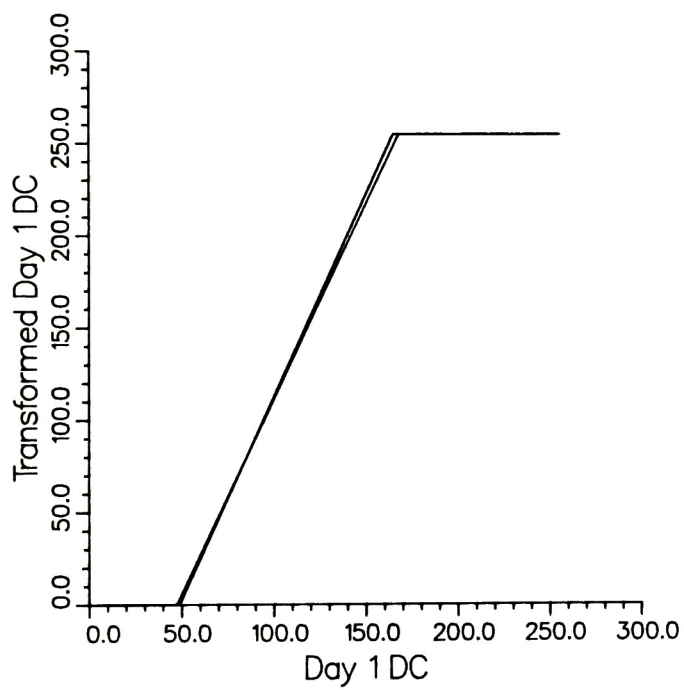
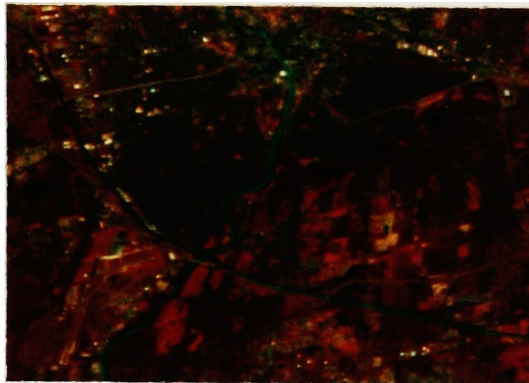


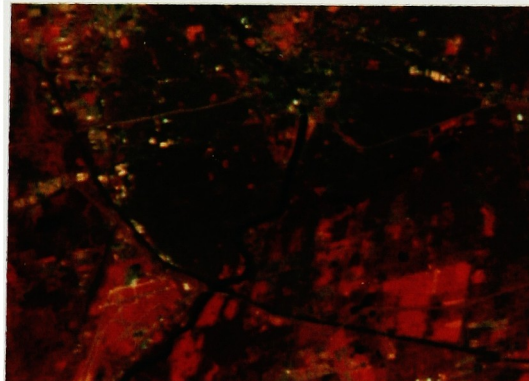
Figure 33 The PIF transformations developed utilizing the automated and interactive segmentation results



(a)

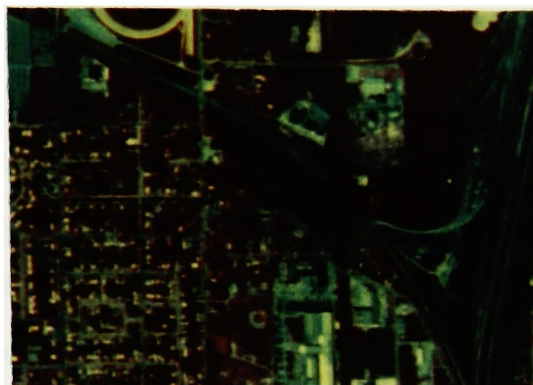


(b)

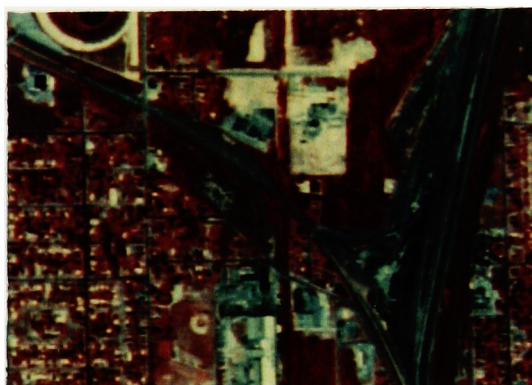


(c)

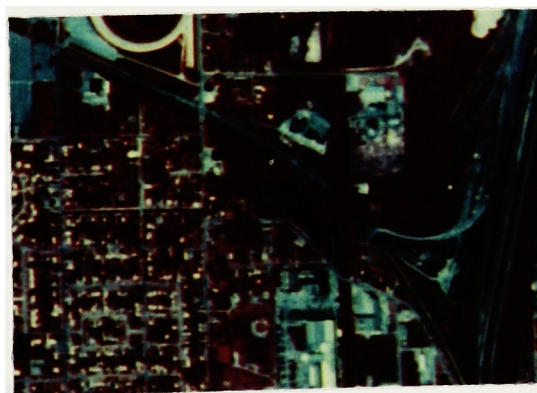
Figure 34 CIR composite TM image representing (a) the original 1982 urban Rochester image, (b) the 1984 urban Rochester image and (c) the transformed 1982 urban Rochester image utilizing the segmentation result from the automated algorithm



(a)



(b)



(c)

Figure 35 CIR NHAP airphoto representing (a) the original 1972 Buffalo image, (b) the 1970 Buffalo image and (c) the transformed 1972 Buffalo image utilizing the segmentation result from the automated algorithm

3.3 Evaluation of Transforms Using Control Point Analysis

To evaluate the effectiveness of the resulting PIF transformation, identical targets were chosen in both the transformed Day-1 imagery and the original Day-2 imagery. A perfect normalization process would cause these scene elements to have identical digital count values provided no physical change in the reflectance of the chosen targets had occurred between image acquisition dates.

The control points were chosen on the basis of the criteria described in Section 2.4. Targets chosen were large man-made objects occupying as wide a dynamic reflectance range as possible. The coordinates of the chosen targets are summarized in Tables 8 through 10 and are depicted in the accompanying Figures 36 through 38.

The digital count values were collected from the Day-1 imagery, the Day-2 imagery, and the transformed Day-2 imagery using both the automated and interactive segmentation algorithms. The digital count values are summarized in Tables F-1 through F-3 in Appendix F for the control point analysis of the two urban Landsat TM scenes as well as the high-resolution NHAP airphoto data.

Tables 11 through 13 summarize the errors in the PIF transformations that

Table 8

Summary of the image coordinates chosen for the control point analysis
of the urban Rochester transformations

Control Point No.	1982 Image Coordinates	1984 Image Coordinates
1	(407, 95)	(396, 114)
2	(303, 65)	(293, 83)
3	(183, 220)	(171, 239)
4	(135, 338)	(121, 365)
5	(97, 416)	(88, 433)
6	(45, 367)	(441, 390)
7	(123, 424)	(112, 437)
8	(92, 445)	(81, 462)

Note: The coordinate (0,0) corresponds to the lower left corner of the digital image format

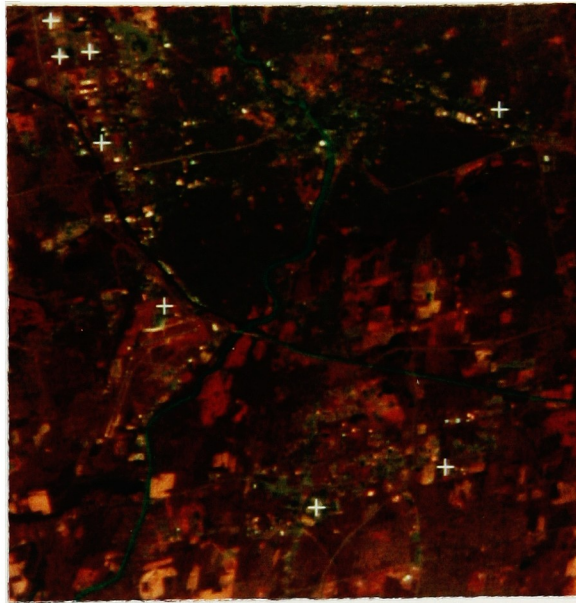


Figure 36 The control points summarized in Table 8 are denoted by +^s on the 1984 urban Rochester image

Table 9

Summary of the image coordinates chosen for the control point analysis
of the urban Buffalo transformations

Control Point No.	1982 Image Coordinates	1984 Image Coordinates
1	(80,356)	(91,403)
2	(269,388)	(281,435)
3	(197,132)	(208,180)
4	(378,319)	(389,365)
5	(476,32)	(487,78)
6	(338,312)	(349,360)
7	(43,453)	(33,405)

Note: The coordinate (0,0) corresponds to the lower left corner of the digital image format

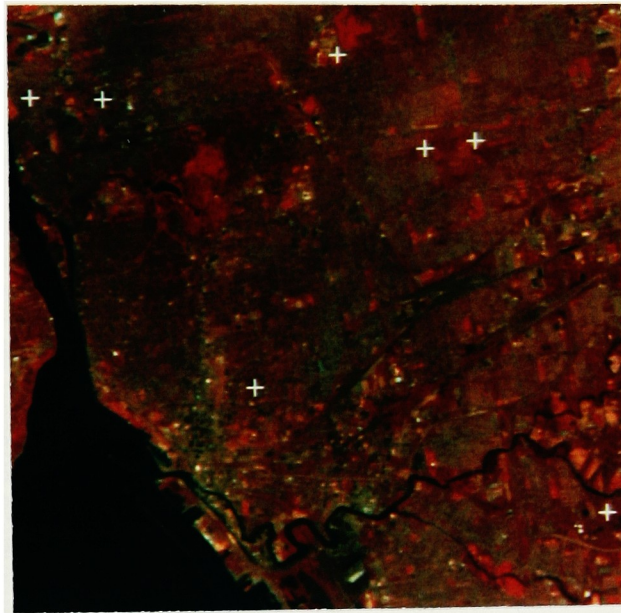


Figure 37 The control points summarized in Table 9 are denoted by +^s on the 1984 urban Buffalo image

Table 10

Summary of the image coordinates chosen for the control point analysis
of the high resolution airphoto imagery of Buffalo

Control Point No.	1970 Image Coordinates	1972 Image Coordinates
1	(95,404)	(137,400)
2	(33,487)	(74,484)
3	(282,170)	(320,160)
4	(37,341)	(78,339)
5	(200,104)	(237,94)
6	(82,156)	(120,151)
7	(312,320)	(354,314)
8	(8,56)	(43,51)

Note: The coordinate (0,0) corresponds to the lower left corner of the digital image format

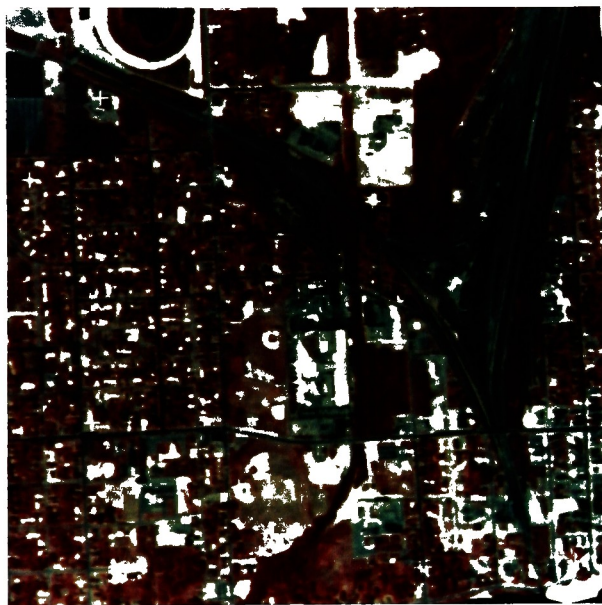


Figure 38 The control points summarized in Table 10 are denoted by +'s on the 1970 NHAP airphoto image of Buffalo

Table 11

Summary of the errors determined from the control point analysis for the urban Rochester imagery when compared to the 1984 untransformed data

TM Spectral Channel	Untransformed Data (DC)	After Raw Transformation		After Corrected for Sampling			
		(DC)		(DC)		(%R)	
		Automated	Interactive	Automated	Interactive	Automated	Interactive
1	33.6	9.3	7.7	6.4	3.6	1.94	1.09
2	20.4	3.2	2.3	2.5	1.1	1.19	0.52
3	29.8	5.0	4.2	3.7	2.5	1.28	0.86
4	22.4	10.8	6.8	9.5	4.4	2.79	1.29
5	37.5	12.8	10.5	11.8	9.3	-	-
7	27.3	7.8	8.2	6.4	6.9	-	-

DC refers to the error expressed in digital count values
%R refers to the error expressed in units of reflectivity

Table 12

Summary of the errors determined from the control point analysis for the urban Buffalo imagery when compared to the 1984 untransformed data

TM Spectral Channel	Untransformed Data (DC)	After Raw Transformation		After Corrected for Sampling			
		(DC)		(DC)		(%R)	
		Automated	Interactive	Automated	Interactive	Automated	Interactive
1	31.2	7.5	7.6	3.7	3.5	1.48	1.40
2**	-	-	-	-	-	-	-
3	19.7	3.7	4.2	0.5	2.4	0.24	1.20
4	17.6	5.4	5.7	4.4	4.8	2.75	3.00
5	20.7	6.4	7.6	4.9	6.5	-	-
7	12.9	4.3	4.8	3.5	4.2	-	-

DC refers to the error expressed in digital count values
%R refers to the error expressed in units of reflectivity

** the band 2 images were unable to be read from the magnetic tape

Table 13

Summary of the errors determined from the control point analysis for the high resolution airphoto imagery of Buffalo when compared to the 1970 untransformed data

CIR Spectral Region	Untransformed Data (DC)	After Raw Transformation		After Corrected for Sampling			
		Automated	Interactive (DC)	Automated	Interactive (DC)	Automated	Interactive (%R)
IR	91.8	25.9	15.6	22.8	11.1	9.12	4.44
R	60.5	4.9	16.2	2.4	15.7	0.82	5.41
G	68.9	4.5	8.1	2.1	6.9	0.77	2.55

DC refers to the error expressed in digital count values
%R refers to the error expressed in units of reflectivity

were determined from these control point analyses. The raw r.m.s. error was computed as described in Section 2.5 for the control points chosen before and after the raw transformation. In each scene studied, it is seen that this raw transformation resulted in a dramatic decrease in the difference between the control point digital count values. As described earlier, this error involved not only the error due to the PIF transformation, but also any error due to incorrect sampling methods by the analyst and the nature of the imagery. In order to remove this sampling error, described in Section 2.5, the transformed Day-1 digital count values were regressed against the Day-2 digital count values for the control points chosen. Under the assumptions for PIF transformation validity, these digital count values should be related by a direct linear function. Any residual error in this regression analysis would be a result of non-PIF pixel influence on this control point data. Table 14 is a summary of this sampling error for the urban Rochester, urban Buffalo TM scenes as well as the NHAP scene of Buffalo. The errors were computed for both the automated and interactive segmentation results. This sampling error should not be a function of the segmentation procedure used and the data in Table 14 show this to be the case. The differences in these sampling errors are very small in all cases shown, with none exceeding a difference greater than 5.5%. These errors were removed from the error after the raw transformation and are summarized in Tables 11 through 13.

As stated earlier, the errors have been expressed in units of digital counts. These errors have only relative meaning in the context of the analysis of an individual scene and must be expressed on a common scale independent of the image. Reflectance is such a measure. The digital count values

Table 14

Summary of the approximate error due to sampling
(residual error from the linear regression of the transformed Day 1 vs. Day 2 digital count values)

Scene	TM Spectral Channel	Sampling Error (ϵ_{samp}) in DC				
		1	2*	3	4	5
Urban Rochester	(Interactive)	6.79	2.01	3.40	5.17	4.93
	(Automated)	6.76	1.98	3.32	5.13	4.40
Urban Buffalo	(Interactive)	6.73	-	3.47	3.15	4.00
	(Automated)	6.51	-	3.67	3.10	4.04
High Resolution Airphoto of Buffalo	CIR Spectral Region	IR	R	G		
		(Interactive)	10.93	4.02	4.09	
		(Automated)	12.29	4.24	3.98	

* the band 2 images were unable to
be read from the magnetic tape

could be expressed in terms of reflectance units of scene elements in a ground-based reflectance space. To do this, reflectance values were estimated for several scene elements and the respective digital counts recorded for each of the Day-2 images in each of the scenes studied. Targets of varying brightness were chosen so that a reflectance range covering as much of the dynamic range of the scene as possible could be obtained. Tables F-4 through F-6 in Appendix F summarize the reflectance and digital count data selected from the urban Rochester and urban Buffalo TM image sets as well as from the high-resolution airphoto data of Buffalo. Also included in these tables are the results when the reflectivity was regressed as a function of digital count value. This analysis was carried out only on the visible and near-infrared spectral regions of the data since these are the only regions where the reflectance values could be estimated with any degree of accuracy. Along with the slope and intercept values for these regressions, the r^2 value (i.e. the coefficient of correlation) was also computed. In all cases this value exceeded 0.92 which indicates that almost all of the variability was accounted for in the data. The slope terms, α_k , are summarized in Table 15 for each band of each of the scenes described. These values were used to convert the errors in terms of digital count values for the visible and near-infrared bands of the data described in Tables 11 through 13.

Finally, in looking at the data summarized in Tables 11 through 13, the error in digital count after the sampling error was removed is less than 5% of the total dynamic range in all cases of the Day-2 imagery in the visible and near-infrared spectral regions (except for the IR airphoto data), and less than 7.5% of this range in the far infrared regions. As described earlier, this increase in error is expected to occur at longer wavelengths. Table 16 summarizes the error due

Table 15

Summary of the reflectance conversion data for the digital count errors summarized in the control point analyses

Scene	TM Spectral Channel	Slope (α_k) [*]				
		1	2 ^{**}	3	4	5
Urban Rochester	3.3		2.1	2.9	3.4	-
Urban Buffalo	2.5		-	2.0	1.6	-
CIR						
High Resolution Airphoto of Buffalo	Spectral Region	IR	R	G		
					2.5	2.7

* α_k is the change in digital count per reflectance unit

** the band 2 images were unable to be read from the magnetic tape

Table 16

Comparison of the accuracy between the PIF normalizations involving the two urban TM images after the sampling error was removed (errors expressed in percent reflectance %R)

Error in PIF Transformation After Sampling Error was Removed (ϵ_{PIF})

TM Spectral Channel	Urban Rochester		Urban Buffalo	
	Automated	Interactive	Automated	Interactive
1	1.94	1.09	1.48	1.40
2	1.19	0.52	-	*
3	1.28	0.86	0.24	1.20
4	2.79	1.29	2.75	3.00

* the band 2 images were unable to be read from the magnetic tape

to the PIF transformation expressed in reflectance units for the two urban TM scenes studied. As seen from this table, the error is less than 2% reflectance in the visible wavelength regions and less than 3% in the near-infrared. It should also be noted that the errors derived from the results of the automated segmentation technique versus the interactive technique are approximately equal. There is no clear-cut difference that can be established from this limited study.

3.4 Evaluation of Robustness Using Rural Image Data

To test the dependency of the automated segmentation technique on the number of urban features located in a scene, a TM scene of rural Rochester was chosen for PIF normalization. The scene was located 512 pixels to the west of the urban Rochester scene to ensure that the imaging conditions, i.e. the atmospheric homogeneity, the sun angle and viewing geometry, matched as identically as possible. With this assumption, the transforms developed separately for these two scenes should also be identical.

The transforms described in Tables 4 and 6 are shown graphically in Figures 39 through 43 for both the automated and interactive segmentation normalization results. The r.m.s. errors between these transforms weighted by the Day-1 histograms are summarized in Table 17.

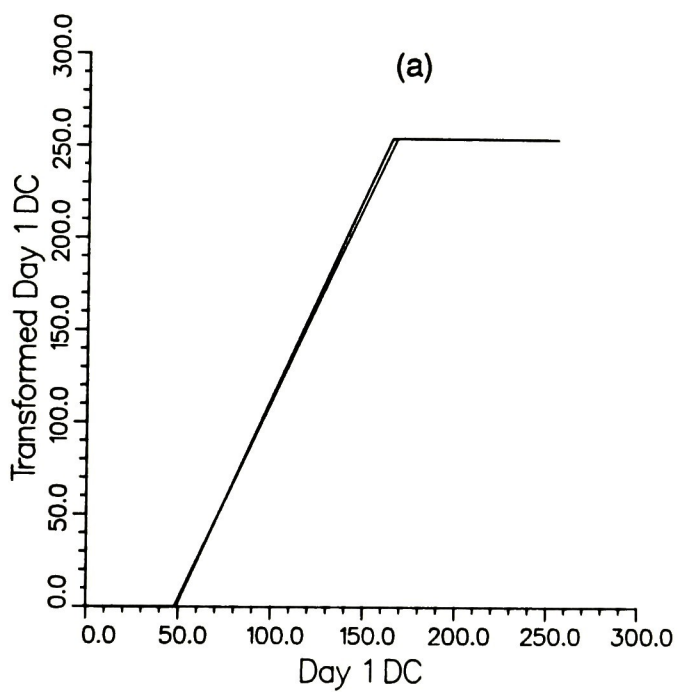
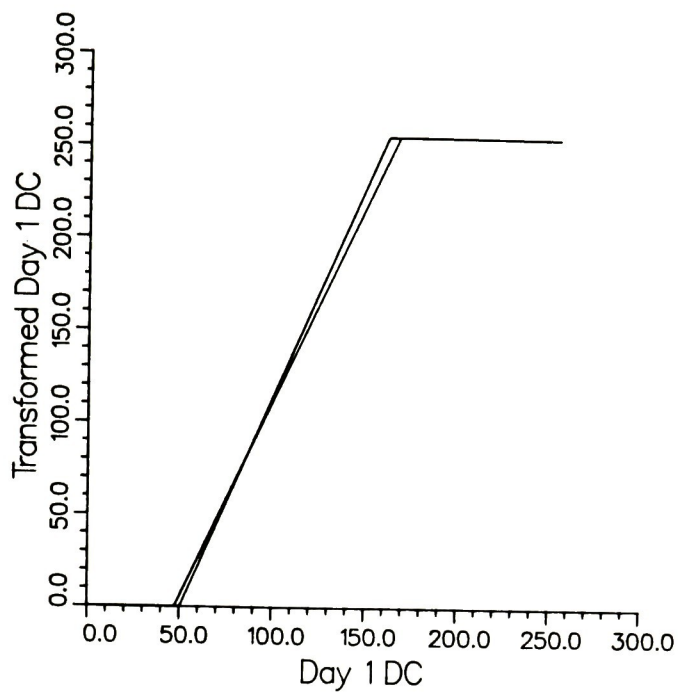
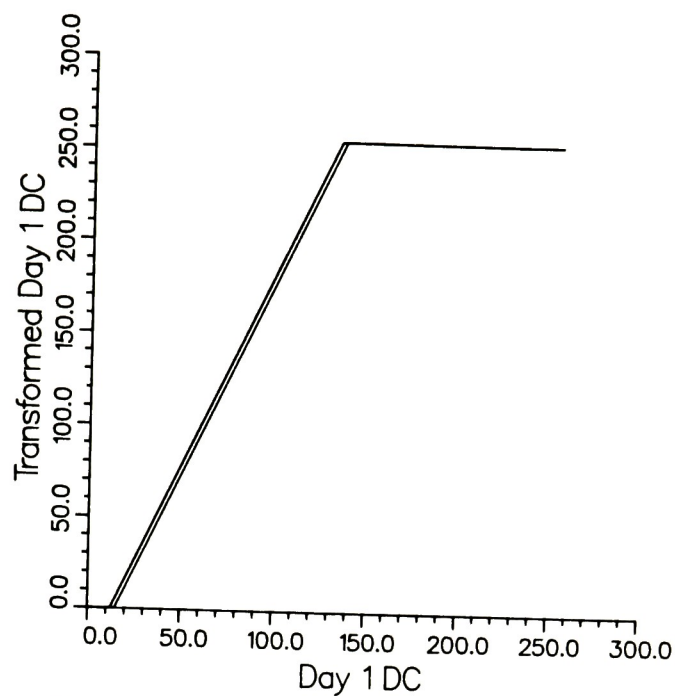
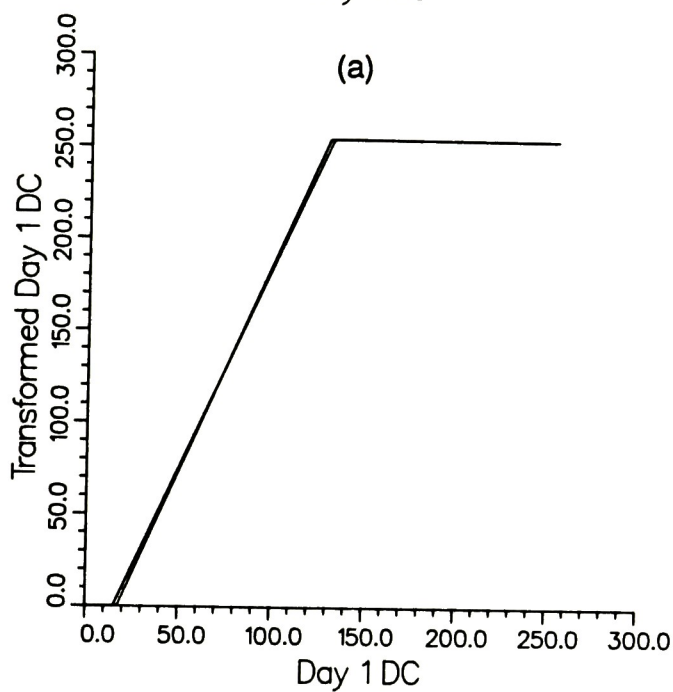


Figure 39 PIF transformations for the urban and rural Rochester imagery using (a) the automated segmentation and (b) the interactive segmentation algorithm (TM Band-1)

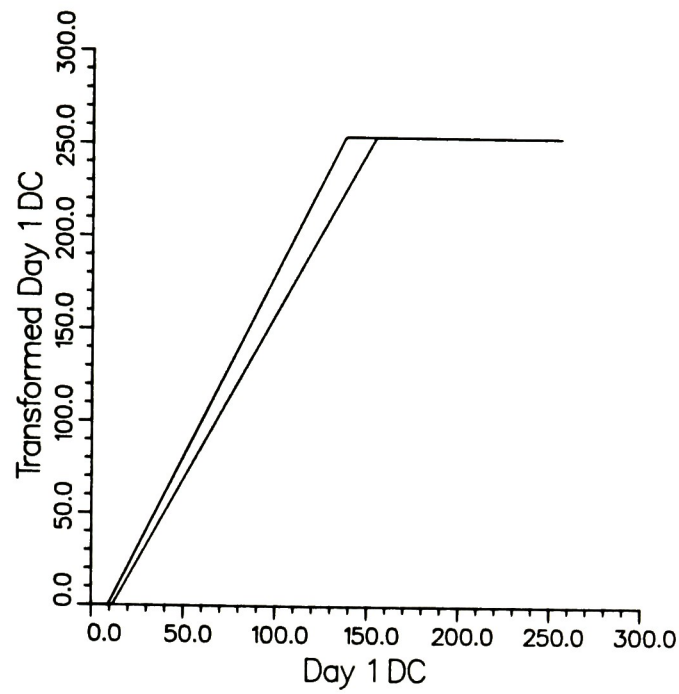


(a)

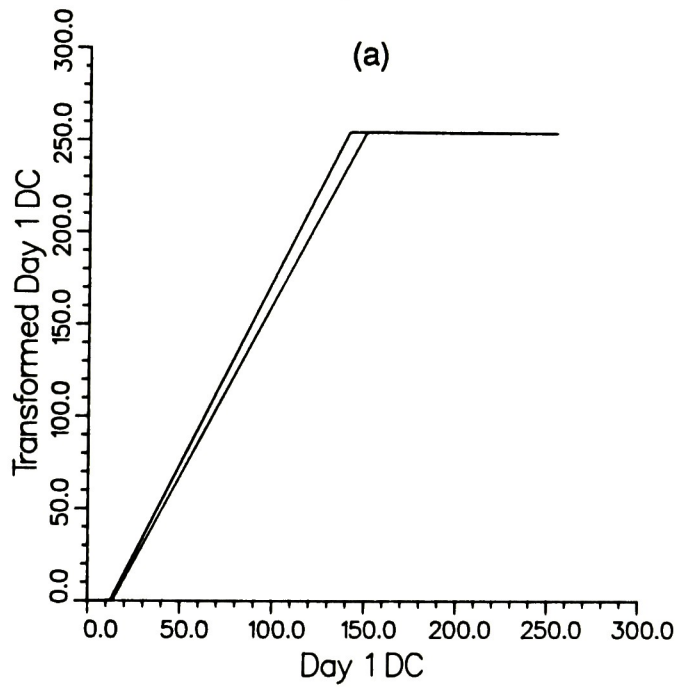


(b)

Figure 40 PIF transformations for the urban and rural Rochester imagery using (a) the automated segmentation and (b) the interactive segmentation algorithm (TM Band-3)

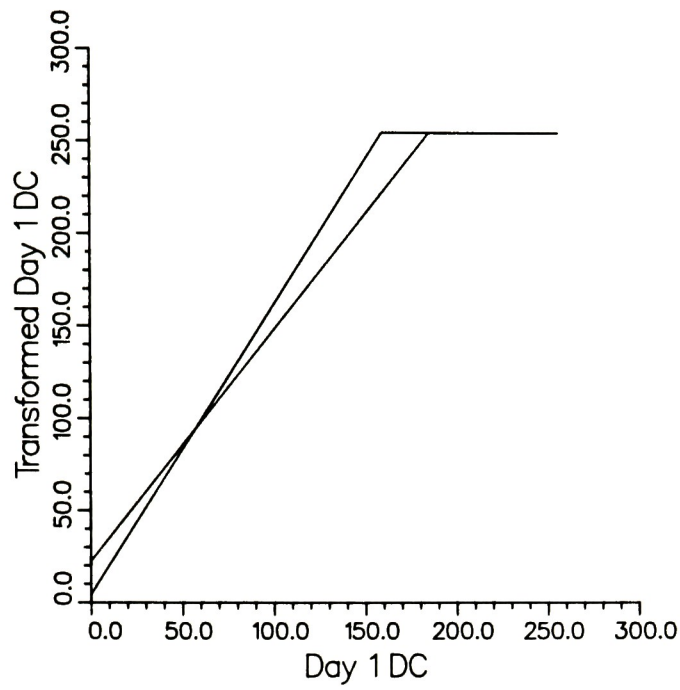


(a)

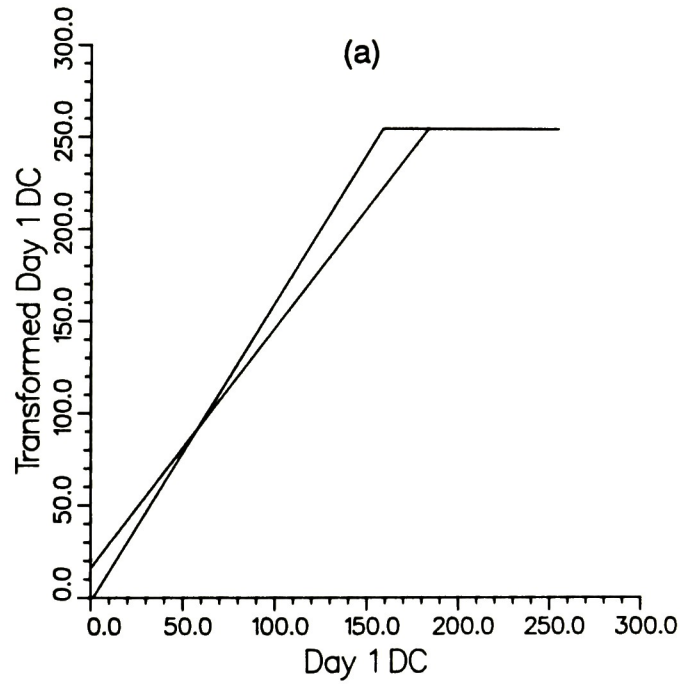


(b)

Figure 41 PIF transformations for the urban and rural Rochester imagery using (a) the automated segmentation and (b) the interactive segmentation algorithm (TM Band-4)



(a)



(b)

Figure 42 PIF transformations for the urban and rural Rochester imagery using (a) the automated segmentation and (b) the interactive segmentation algorithm (TM Band-5)

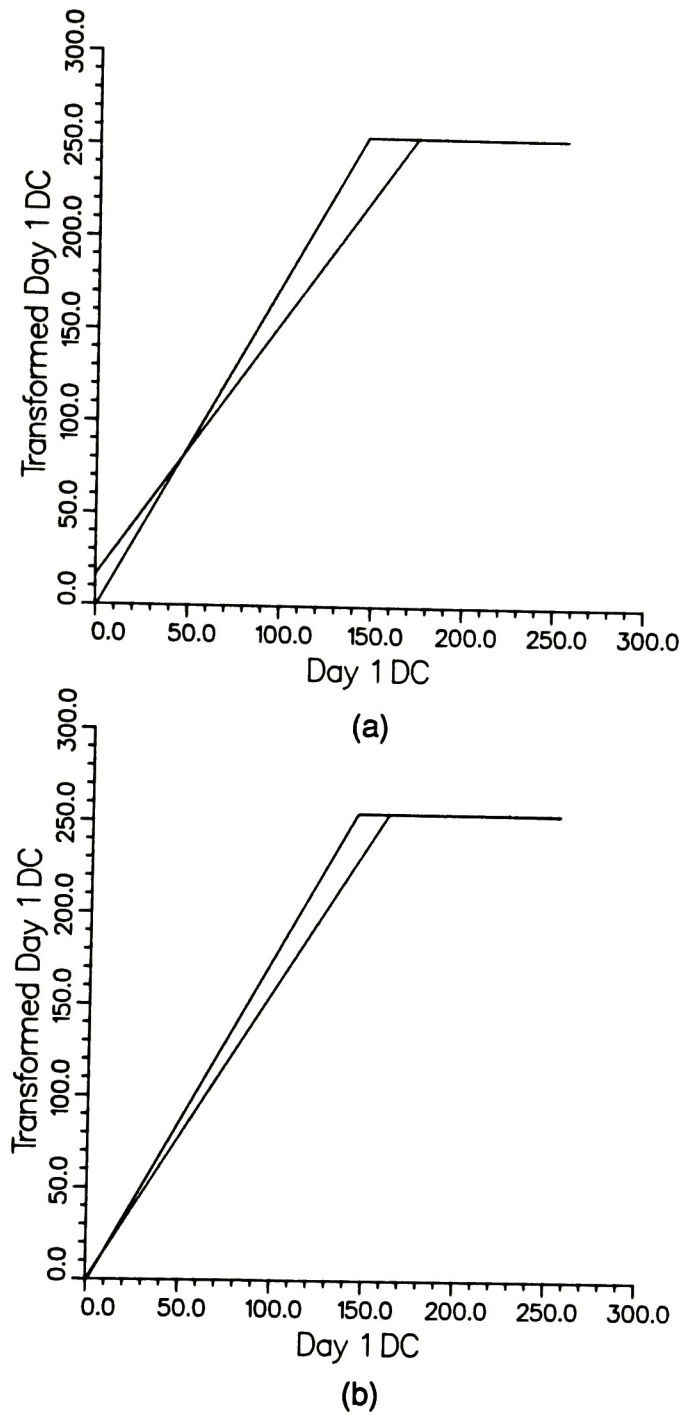


Figure 43 PIF transformations for the urban and rural Rochester imagery using (a) the automated segmentation and (b) the interactive segmentation algorithm (TM Band-7)

Table 17

Summary of the r.m.s. error in digital count between the linear histogram transformations derived for the rural and urban Rochester TM scenes

TM Spectral Channel	Error Between Transformations (in DC's)	
	Automated	Interactive
1	3.09	2.43
2	-	- *
3	5.34	2.02
4	14.09	7.05
5	10.17	9.63
7	4.38	8.19
Number of Pixels =	8325	7686

* the band 2 images were unable to be read from the magnetic tape

As seen from the data in Table 17, the errors between the urban and rural transforms are typically higher when the automated algorithm was used for segmentation than when the interactive technique was utilized. The reason for this occurrence is that the analyst serves as an invaluable tool deciding what is and what is not an urban feature in a rurally dominated scene. As seen in the summary of the numbers of pixels that were classified as PIF pixels in these images using the two segmentation techniques, the numbers are indeed small when compared to the number contained in a 512 x 512 pixel digital image (approximately 3%). When the number of PIF pixels becomes so small, any error due to mixed pixel effects will start to have a more dramatic effect, especially in a rural area. It is interesting to note that the number of pixels classified as PIF's is larger for the automated segmentation algorithm than for the interactive technique. This had not been the case with any of the previous analyses where the automated technique had always proven more conservative.

In reviewing the data in Table 17, it should be noted that the errors between the two transforms in the interactive segmentation results are less than 8.5% of the Day-2 PIF brightness range, with the highest error occurring in the near-infrared wavelength region. The errors between the two transforms in the automated segmentation technique results are less than 16%, with the highest error again in the near-infrared region for this same brightness range. The magnitude in percentage of these errors are compared to the dynamic range of the PIF pixels only, and will be lower when applied to the entire scene.

These errors arise from primarily two sources. The first is the breakdown of the assumption of atmospheric homogeneity of the urban and rural Rochester images. If the atmospheres of these two images are not exactly the same, then the transforms cannot be expected to be identical. The second source of error between these two transforms is the effectiveness of the automated segmentation algorithm in a primarily rural scene. As stated earlier, the automated algorithm is slightly more liberal than the interactive technique in the case of these images. This liberal thresholding causes too many mixed pixels to remain in the PIF mask, thus the quality of the resulting normalization will decrease, especially in the longer wavelength regions.

The results obtained in this study indicate that the overall accuracy of the PIF transformations is unaffected by the choice of the automated or interactive segmentation algorithms. The r.m.s. errors encountered during the control point analysis were approximately one or two reflectance units regardless of which segmentation method was used. The robustness of the automated algorithm held up well for urban imagery but suffered some loss of accuracy when applied to rural imagery.

The technique failed when applied to imagery with significant cloud cover. The presence of clouds changes the shape of the three-dimensional surface since the rate-of-change of the number of pixels present in the PIF mask did not decline enough at any point to show a plateau region in the plot. The technique developed here was not derived to yield the optimal segmentation results but rather to take advantage of the convenient empirical observations made from the original interactive method.

4.0 Conclusions and Recommendations

A technique has been developed to allow automated segmentation of pseudoinvariant features from Landsat TM imagery and NHAP high-resolution color infrared airphoto imagery. The technique has been shown to work on imagery that is primarily urban as well as imagery that is dominated by rural scene elements.

The automated technique was tested side by side with the previously developed interactive segmentation algorithm by comparing the results of the normalizations conducted using each of the segmented PIF masks as input to the normalization scheme. Comparison of the digital count values of selected in-scene control points have shown the automated and interactive segmentation techniques to be equivalent, producing normalization errors of less than 7.5% of the dynamic range of PIF brightnesses in the image. Both techniques also produced results with errors in units of reflectivity of less than 2% in the visible wavelength regions and less than 3% in the near infrared.

The techniques developed in this study are expected to work equally well on any imagery of a multispectral nature and of higher geometric resolution than the Landsat Thematic Mapper sensor. The segmentation technique was shown to work well on the NHAP high-resolution airphoto imagery with a ground resolution of approximately 10 meters. The resulting normalization exhibited very high error in the infrared wavelength region (on the order of 9%) while producing excellent results in the visible bands (less

than 1% reflectance error). This weakness in the normalization is most likely the result of a breakdown in the assumption of linearity between reflectance and brightness count for digitized photographic data. More work needs to be done to correct this non-linearity of the response function of the photographic material.

This automated segmentation algorithm provides a non-interactive temporal scene-to-scene radiometric normalization technique that is an invaluable tool for such remote-sensing applications as temporal change detection, development of time-independent land-use classification algorithms, and any number of time-dependent ecological studies.

This study also provided a technique-independent method for quantitatively evaluating the success of any normalization process. The control-point analysis described with the conversion to an estimated reflectance space provides a metric on which to measure the success of any image normalization technique on any image medium.

A primary weakness encountered in this study was the failure of both the interactive and the automated segmentation algorithms to work on imagery containing significant cloud cover. The spectral distribution of these image features causes confusion to occur when trying to isolate pseudoinvariant features. Techniques need to be developed that can either distinguish between clouds and urban features by spectral characteristics or to preprocess the imagery to remove the cloud cover.

The automated segmentation and PIF normalization techniques have been tested exclusively on imagery of upstate New York. A further test of the robustness of the segmentation algorithm as well as the normalization technique needs to be conducted by applying these techniques to different geographic locations. Urban features in other geographic locations will tend to have different spectral signatures due to the different building materials. It needs to be established whether or not this has a significant effect on the segmentation algorithm.

The algorithm developed was not intended to determine the optimal PIF segmentation. It was designed to take advantage of the convenient segmentation technique developed for the original proof-of-concept study. Future study should be conducted to examine the quality of the segmentation resulting from the automated routine. A quality metric needs to be established as a measure of "goodness/poorness of fit" of segmentation results. As mentioned earlier, the histogram equalization approach to the development of the PIF transforms can be used as a quick check for the quality of the original segmentation when compared to the linear histogram transforms. Future work with this quality criterion may lead to a more rigorous segmentation quality metric to fine-tune the segmentation.

Due to the conservative number of pixels contained in the PIF masks formed using the automated segmentation technique, a question arises concerning the quality of the transformations derived. There is not only a different number of pixels in the PIF masks for the two images, but these

pixels often represent different ground features. The next step in the development of a segmentation technique would be to use information from both images to form a single PIF mask that contains the same number and location of PIF pixels which can be used to segment the two day's registered imagery. The resulting normalization would then be based on identical gray-level distributions from the two images.

References

1. D.J. Gerson and L.K. Fehrenbach, "Temporal Image Normalization", Defense Mapping Agency, Final Report, January 1983.
2. J.R. Schott, "Atmospheric Characterization Using Pseudo Invariant Features", A White Paper Submitted to the Department of the Air Force, Rome Air Development Center, April 1983.
3. K.R. Piech and J.R. Schott, "Atmospheric Corrections for Satellite Water Quality Studies", Proceeding of S.P.I.E., **51**, (1974), pp. 84-89.
4. K.R. Piech, D.W. Gaucher, E.W. Schimminger and C.W. Rogers, "Photometric Calibration of Black and White Imagery", Rome Air Development Center Technical Report RADC-TR-78-53, 1978.
5. W.J. Volchok, "A Study of Multispectral Scene Normalization Using Pseudo-Invariant Features Applied to Landsat TM Imagery", M.S. Thesis, Center for Imaging Science, Rochester Institute of Technology, Rochester, New York, October 1985, p. 36.
6. Reference 1.
7. W.J. Volchok and J.R. Schott, "Scene to Scene Radiometric Normalization of the Reflected Bands of the Landsat Thematic Mapper", Proceedings of the S.P.I.E. Symposium, "Optical and Optoelectronic Applied Sciences and Engineering", Innsbruck, Austria, April 1986.
8. Reference 3.
9. Reference 5.
10. J.R. Schott and C.Salvaggio, "Radiometric Scene Normalization Using Pseudo Invariant Features", RIT/DIRS Report 86/87-51-113, Center for Imaging Science, Rochester Institute of Technology, Rochester, New York, January 1987.
11. Reference 5.

12. Reference 5.
13. Reference 1.
14. T.M. Lillesand and R.W. Kiefer, Remote Sensing and Image Interpretation, John Wiley & Sons, Inc., New York, 1979, p. 445.
15. P.N. Slater, "Photographic Systems For Remote Sensing", pp. 231-291 in Manual of Remote Sensing, Volume I, Second Edition, D.S. Simonett and F.T. Ulaby, eds., American Society of Photogrammetry, Virginia, 1983, p. 253.
16. Reference 14, p. 14.
17. P.H. Swain and S.M. Davis, eds., Remote Sensing: The Quantitative Approach, McGraw-Hill, New York, 1978, pp. 156-157.
18. Reference 14, p. 376.
19. Reference 14, p. 376.
20. P.S. Chavez Jr., G.L. Berlin and L.B. Sowers, "Statistical Methods for Selecting LANDSAT MSS Ratios", Journal of Applied Photographic Engineering, **8**, 1 (1982), pp. 22-30.
21. R.S. Williams, "Geological Applications", pp. 1667-1916 in Manual of Remote Sensing, Volume II, Second Edition, J.E. Estes and G.A. Thorley, eds., American Society of Photogrammetry, Virginia, 1983, p. 1749.
22. Reference 14, p. 16.
23. Reference 21, p. 1749.
24. C.J. Tucker, "Measurement of the Combined Effect of Biomass Chlorophyll and Leaf Water on Canopy Spectroreflectance on the Short Grass Prairie", Second Annual Remote Sensing of Earth Resources Conference, (1973), p. 601.

25. J.D. Biegel and J.R. Schott, "Radiometric Calibration and Image Processing of Landsat TM Data to Improve Assessment of Thermal Signatures", Proceedings of S.P.I.E., Infrared Technology, **IX**, 430 (1984).
26. P.N. Slater, Remote Sensing: Optics and Optical Systems, Addison-Wesley Publishing Co., Reading, Massachusetts, 1980, p. 292.
27. Reference 14, p. 378.
28. Reference 5.
29. R.A. Schowengerdt, Techniques for Image Processing and Classification in Remote Sensing, Academic Press, Inc., New York, 1983, pp. 132-137.
30. Reference 5.
31. Reference 14, p. 478.
32. J.E. Estes, E.J. Hajic and L.R. Tinney, "Fundamentals of Image Analysis: Analysis of Visible and Thermal Infrared Data", pp. 987-1118 in Manual of Remote Sensing, Volume I, Second Edition, D.S. Simmonett and F.T. Ulaby, eds., American Society of Photogrammetry, Virginia, 1983, pp. 1042-1043.
33. Reference 14, pp. 463-469.
34. Reference 14, pp. 464-465.
35. Reference 14, p. 467.
36. Reference 29, pp. 51-54.
37. Reference 14, p. 477.
38. Reference 14, p. 478.
39. Reference 29, p. 145.
40. Reference 29, p. 147.

41. E.P.F. Kan, "ISODATA: Thresholds for Splitting Clusters", Technical Report 640-TR-058, Lockheed Electronics Co., Inc., HASD, Houston, Texas (1972).
42. Reference 29, pp. 145-147.
43. J.A. Hartigan, Clustering Algorithms, John Wiley & Sons, New York, 1975, pp. 108-112.
44. Reference 14, p. 480.
45. Reference 32, p. 1044.
46. R.J. Kauth and G.S. Thomas, "The Tasslecap: A Graphic Description of the Spectral-Temporal Development of Agricultural Crops as Seen by LANDSAT", Proceedings of the Symposium of Machine Processing of Remotely Sensed Data, 6/29/76 to 7/2/76, L.A.R.S., Purdue, I.E.E.E. Cat. No. 76, Ch. 1103-1 MPRSD, pp. 41-51.
47. E.P. Crist and R.J. Kauth, "The Tassled Cap De-Mystified", Photogrammetric Engineering and Remote Sensing, **52**, 1 (1986), pp. 81-86.
48. Reference 46.
49. R.A. Johnson and D.W. Wichern, Applied Multivariate Statistical Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1982, pp. 361-362.
50. R.J. Kauth, A.P. Pentland, and G.S. Thomas, "BLOB, an Unsupervised Clustering Approach to Spatial Preprocessing of MSS Imagery", Proceedings of the Eleventh Symposium on Remote Sensing of Environment, **2**, (1977), pp. 1309-1317.
51. R.L. Kettig and D.A. Landgrebe, "Classification of Multispectral Image Data by Extraction and Classification of Homogeneous Objects", IEEE Transactions on Geoscience Electronics, **GE-4**, 1 (1976), pp. 19-26.
52. Reference 43.

53. Reference 41.
54. F.R. Fromm and R.A. Northouse, "CLASS: A Nonparametric Clustering Algorithm", Pattern Recognition, **8**, (1976), pp. 107-114.
55. J. Bryant, "On the Clustering of Multidimensional Pictorial Data", Pattern Recognition, **11**, (1979), pp. 115-125.
56. Reference 55, p. 115-125.
57. R.M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification", IEEE Transactions on Systems, Man, and Cybernetics, **SMC-3**, 6 (1973), pp. 610-621.
58. S. Hsu, "Texture-Tone Analysis for Automated Land-Use Mapping", Photogrammetric Engineering and Remote Sensing, **44**, 11 (1978), pp. 1393-1404.
59. O.R. Mitchell and S.G. Carlton, "Image Segmentation Using a Local Extrema Texture Measure", Pattern Recognition, **10**, 2 (1978), pp. 205-210.
60. O.R. Mitchell, C.R. Meyers and W. Boyne, "A Max-Min Measure for Image Texture Analysis", IEEE Transactions on Computers, **C-26**, 4 (1977), pp. 409-414.
61. J.A. Richards, D.A. Landgrebe and P.H. Swain, "A Means for Utilizing Ancillary Information in Multispectral Classification", Remote Sensing of Environment, **12**, (1982), pp. 463-477.
62. E.B. Troy, E.S. Deutsch and A. Rosenfeld, "Gray Level Manipulation Experiments for Texture Analysis", IEEE Transactions on Systems, Man, and Cybernetics, **SMC-3**, 1 (1973), pp. 91-98.
63. Reference 5.
64. Reference 25.
65. R.C. Gonzalez and P. Wintz, Digital Image Processing, Addison-Wesley,

Reading, Massachusetts, 1980, pp. 119-135.

66. Reference 5.

67. Reference 47.

68. Reference 57.

69. Reference 26, p. 293.

70. Reference 5.

Appendix A

Description of the Principal Components Computer Code

PrinComp
ImgMean
Variance

```

.....
*
*   PRINCOMP      -  WILL FIND THE PRINCIPAL COMPONENT TRANSFORMS
*                   FOR ANY N-BAND IMAGE.  THE IMAGES ARE EXPECTED
*                   TO BE 512 X 512 IN SIZE AND 8 BITS DEEP.
*                   THE PC'S ARE SCALED BY THE SCALAR DETERMINED
*                   TO MAXIMIZE THE DYNAMIC RANGE OF PC 1.
*
*   REQUIRED SUBROUTINES:  PIXIN
*                           PIXOUT
*                           IMGMEAN
*                           VARIANCE
*
*   REQUIRED IMSL SUBROUTINES:  EIGRS
*
*   WRITTEN BY CARL SALVAGGIO   AUGUST 1986
*
.....

CHARACTER*1      BELL
INTEGER*4        N, M, MAXBND, MINBND, SMPINC, DIM, Q

PARAMETER ( N = 512, M = 512, SMPINC = 5 )
PARAMETER ( MAXBND = 10, MINBND = 2 )
PARAMETER ( BELL = CHAR(7) )
PARAMETER ( DIM = ( MAXBND * ( MAXBND + 1 ) ) / 2 )

INTEGER*2        IMAGE( N, M ), IMAGEA( N, M ), IMAGEB( N, M )
INTEGER*4        JOBN, IERR, IZ
REAL*4           MEAN( MAXBND ), VAR( MAXBND, MAXBND )
REAL*4           VALUES( MAXBND ), VECTORS( MAXBND, MAXBND )
REAL*4           A( DIM ), D( MAXBND ), Z( MAXBND, MAXBND )
REAL*4           TOTVAR, SCALE
CHARACTER*80     FILNAM( MAXBND ), PCNAM( MAXBND )
CHARACTER*80     INNAME, OUTNAME, DEFAULT
CHARACTER*1      ANS
INTEGER*4        CONTEXT, STATUS, LIB$FIND_FILE

OPEN ( 9, STATUS = 'NEW' )

*
* .....      OBTAIN IMAGE INFORMATION
*
CALL LIB$ERASE_PAGE(1,1)
WRITE (6,*)
1  WRITE (6,2)
2  FORMAT ( '$', 'HOW MANY IMAGE BANDS DO YOU HAVE ? ' )
READ (5,*,END=900) NUMBND

IF ( NUMBND .GT. MAXBND .OR. NUMBND .LT. MINBND ) THEN
    WRITE (6,*) BELL
    WRITE (6,800)
    GOTO 1
ELSE
    GOTO 3
END IF

*
* .....      READ IN BINARY IMAGE FROM DISK

```

```

.
3  WRITE (6,*)
   DEFAULT = 'PICS:[CNS9024.LANDSAT]*.PIX'
   DO 6 K = 1, NUMBND, 1
4     WRITE (6,5) K
5     FORMAT ('$', 'ENTER FILENAME FOR BAND', I3, ': ')
     READ (5, '(A)', END=900) FILNAM( K )
     INNAME = FILNAM( K )
     STATUS = LIB$FIND_FILE( INNAME, OUTNAME,
                           CONTEXT, DEFAULT, OUTNAME )
     CALL LIB$FIND_FILE_END( CONTEXT )
     IF ( STATUS .EQ. 65537 ) THEN
       FILNAM( K ) = OUTNAME
     ELSE
       WRITE (6,*) '*** ERROR *** FILE NOT FOUND'
       GOTO 4
     END IF
6   CONTINUE

   WRITE (6,*)
   WRITE (6,*) 'PLEASE STAND BY ...'
   DO 30 K = 1, NUMBND, 1
     CALL PIXIN( IMAGE, FILNAM( K ) )      ! READ IN IMAGE

.
.....    CALCULATE THE MEAN AND VARIANCE FOR THE INDIVIDUAL BANDS

     CALL IMGMEAN( IMAGE, SMPINC, MEAN( K ) )
     CALL VARIANCE( IMAGE, MEAN(K), IMAGE, MEAN(K),
                   SMPINC, VAR(K,K) )
30  CONTINUE

.
.....    FILL IN THE REST OF THE COVARIANCE MATRIX

   DO 70 L = 1, NUMBND-1, 1
     CALL PIXIN( IMAGEA, FILNAM( L ) )
     DO 60 K = L+1, NUMBND, 1
       CALL PIXIN( IMAGEB, FILNAM( K ) )
       CALL VARIANCE( IMAGEA, MEAN(L), IMAGEB, MEAN(K),
                     SMPINC, VAR( L, K ) )
60  CONTINUE
70  CONTINUE

   DO 90 I = 1, NUMBND, 1
     DO 80 J = 1, NUMBND, 1
       VAR( J,I ) = VAR( I,J )
80  CONTINUE
90  CONTINUE

.
.....    DISPLAY THE IMAGE STATISTICS

   WRITE (9,*)
   WRITE (9,*) 'IMAGE NAMES'
   WRITE (9,*)

```

```

      Do 100 I = 1, NUMBND, 1
      WRITE (9,*) FILNAM( I )
100  CONTINUE

      WRITE (9,*)
      WRITE (9,*) 'MEAN VECTOR'
      WRITE (9,*)
      WRITE (9,*) ( MEAN( I ), I = 1, NUMBND, 1 )

      WRITE (9,*)
      WRITE (9,*) 'VARIANCE-COVARIANCE MATRIX'
      WRITE (9,*)
      Do 130 I = 1, NUMBND, 1
      WRITE (9,127) ( VAR( I,J ), J = 1, NUMBND, 1 )
127  FORMAT ( <NUMBND>F10.3 )
130  CONTINUE

*
*.....      PUT COVARIANCE MATRIX IN A SYSTEMATIC FORMAT FOR EIGRS
*

      ISUB = 0
      Do 150 I = 1, NUMBND, 1
      Do 140 J = 1, I, 1
      ISUB = ISUB + 1
      A( ISUB ) = VAR( I,J )
140  CONTINUE
150  CONTINUE

*
*.....      CALCULATE THE EIGENVALUES AND EIGENVECTORS OF COVARIANCE MATRIX
*

      JOBN = 1      ! JOBN = 0  COMPUTES ONLY EIGENVALUES
                  ! JOBN = 1  COMPUTES EIGENVALUES AND EIGENVECTORS
      IZ = MAXBND
      WK = MAXBND + 1

      CALL EIGRS( A, NUMBND, JOBN, D, Z, IZ, WK, IERR )

      Do 151 I = 1, NUMBND, 1
      VALUES( I ) = D( I )
151  CONTINUE

      Do 153 I = 1, NUMBND, 1
      Do 152 J = 1, NUMBND, 1
      VECTORS( I,J ) = Z( I,J )
152  CONTINUE
153  CONTINUE

*
*.....      FLIP THE VALUES AND VECTORS TO DESCENDING ORDER
*

      Do 155 I = 1, NUMBND/2, 1
      T = VALUES( I )
      VALUES( I ) = VALUES( NUMBND+1-I )      ! FLIP THE VALUES
      VALUES( NUMBND+1-I ) = T
155  CONTINUE

```

```

Do 165 I = 1, NUMBND, 1
  Do 160 J = 1, NUMBND/2, 1
    T = VECTORS( I,J )
    VECTORS( I,J ) = VECTORS( I, NUMBND+1-J ) ! FLIP THE VECTORS
    VECTORS( I, NUMBND+1-J ) = T
  CONTINUE
160 CONTINUE
165 CONTINUE

*
***** CORRECT THE EIGENVECTOR MATRIX TO A POSITIVE QUADRANT FORMAT
*

Do 180 I = 1, NUMBND, 1
  Do 170 J = 1, NUMBND, 1
    VECTORS( I,J ) = (-1)**(J) * VECTORS( I,J )
  CONTINUE
170 CONTINUE
180 CONTINUE

*
***** COMPUTE THE TOTAL VARIANCE OF THE TRANSFORMED DATA
*

TotVar = 0.0
Do 190 I = 1, NUMBND, 1
  TotVar = TotVar + VALUES( I )
190 CONTINUE

*
***** PRINT OUT EIGENVALUES AND EIGENVECTORS
*

CALL LIB$ERASE_PAGE(1,1)
WRITE (6,*)
WRITE (9,*)
WRITE (6,*) 'EIGENVALUES          % VARIABILITY'
WRITE (9,*) 'EIGENVALUES          % VARIABILITY'
WRITE (6,*)
WRITE (9,*)
Do 210 I = 1, NUMBND, 1
  WRITE (6,200) VALUES( I ), ( VALUES( I ) / TotVar ) * 100.0
  WRITE (9,200) VALUES( I ), ( VALUES( I ) / TotVar ) * 100.0
  FORMAT ( F12.5, 9X, F7.3 )
200 CONTINUE
  WRITE (6,*)
  WRITE (9,*)
  WRITE (6,215) TotVar
  WRITE (9,215) TotVar
215 FORMAT ( '$', 'TOTAL VARIANCE = ', F13.5 )

  WRITE (6,*)
  WRITE (9,*)
  WRITE (6,*) 'EIGENVECTORS'
  WRITE (9,*) 'EIGENVECTORS'
  WRITE (6,*)
  WRITE (9,*)
  Do 220 I = 1, NUMBND, 1
    WRITE (6,217) ( VECTORS( I,J ), J = 1, NUMBND, 1 )
    WRITE (9,217) ( VECTORS( I,J ), J = 1, NUMBND, 1 )
    FORMAT ( <NUMBND>F10.5 )
  CONTINUE
217 CONTINUE
220

```

```

WRITE (9,*)
WRITE (9,*)

*
*..... DETERMINE THE NUMBER OF PC'S ( EIGENVECTORS ) TO KEEP
*

WRITE (6,*)
WRITE (6,*)
290 WRITE (6,300)
300 FORMAT ('$', 'CALCULATE AND STORE THE FIRST [Q] P.C.'s: Q: ')
READ (5,*) Q

IF ( Q .GT. NUMBND ) THEN
WRITE (6,310) NUMBND
310 FORMAT ('$', '*** ERROR *** ONLY', I3, ' PC's ARE POSSIBLE')
GOTO 290
ELSE IF ( Q .LE. 0 ) THEN
GOTO 900
END IF
WRITE (6,*)

DO 313 I = 1, Q, 1
WRITE (6,312) I
312 FORMAT ('$', 'ENTER OUTPUT FILENAME FOR PC(', I3, ') : ')
READ (5, '(A)', END=900) PCNAM( I )
313 CONTINUE

*
*..... CALCULATE THE SCALE FOR THE PC IMAGES
*

SCALE = 0.0
DO 320 J = 1, NUMBND, 1
SCALE = SCALE + ( 255.0 * VECTORS( J,1 ) )
320 CONTINUE
SCALE = SCALE / 255.0

*
*..... SAVE THE PC's
*

DO 400 I = 1, Q, 1
WRITE (6,*)
WRITE (6,314) I
314 FORMAT ('$', 'COMPUTING PC(', I3, ') ... ')

*
*..... CLEAR THE PAST PC IMAGE
*

DO 316 K = 1, N, 1
DO 315 L = 1, M, 1
IMAGE( K,L ) = 0.0
315 CONTINUE
316 CONTINUE

*
*..... CALCULATE THE CURRENT PC IMAGE

```

```

      .
      Do 390 J = 1, NUMBND, 1
      CALL PIXIN( IMAGEA, FILNAM( J ) )
      Do 380 K = 1, N, 1
      Do 370 L = 1, M, 1
      IMAGE( K,L ) = FLOAT( IMAGE( K,L ) ) +
      :                               FLOAT( IMAGEA( K,L ) ) *
      :                               VECTORS( J,I ) / SCALE
      IF ( IMAGE( K,L ) .LT. -255 ) IMAGE( K,L ) = -255
      IF ( IMAGE( K,L ) .GT. 255 ) IMAGE( K,L ) = 255
370      CONTINUE
380      CONTINUE
390      CONTINUE

      .
      .....      BE SURE THE IMAGE IS POSITIVE IN SIGN
      .

      Do 394 J = 1, N, 1
      Do 393 K = 1, M, 1
      IMAGE( J,K ) = ABS( IMAGE( J,K ) )
393      CONTINUE
394      CONTINUE

      .
      .....      SAVE THE CURRENT PC IMAGE
      .

395      WRITE (6,*) 'SAVING PC IMAGE IN:'
      WRITE (6,*) PCNAM( I )
      CALL PIXOUT( IMAGE, PCNAM( I ) )

400      CONTINUE

      .
      .....      ERROR MESSAGES
      .

800      FORMAT (//, '*** ERROR -- INVALID NUMBER OF BANDS ***',//)

      .
      .....      TERMINATION BLOCK
      .

900      CLOSE ( 9 )
      WRITE (6,*)
      STOP ' PRINCIPAL COMPONENTS COMPLETED.'
      END

```

```

.....SUBROUTINE IMGMEAN( IMAGE, SMPINC, MEAN )
.....
*      IMGMEAN      -  WILL FIND THE MEAN VALUE OF AN IMAGE.
*                     THE IMAGES ARE EXPECTED TO BE 512 x 512
*                     IN SIZE AND 8 BITS DEEP.
*
*      WRITTEN BY CARL SALVAGGIO      OCTOBER 1986
*
.....

      INTEGER*4      N, M, SMPINC
      PARAMETER ( N = 512, M = 512 )

      INTEGER*2      IMAGE( N, M )
      REAL*4         MEAN
      REAL*8         SUM, TOTNUM

*
*.....      CALCULATE THE IMAGE MEAN
*
      TOTNUM = INT( N / SMPINC )**2
      SUM = 0.0
      DO 20 J = 1, M, SMPINC
        DO 10 I = 1, N, SMPINC
          SUM = SUM + FLOAT( IMAGE( I, J ) )
10        CONTINUE
20      CONTINUE

      MEAN = SUM / TOTNUM

*
*.....      RETURN CONTROL TO CALLING PROGRAM
*
      RETURN
      END

```

```

.....SUBROUTINE VARIANCE( IMAGEA, MEANA, IMAGEB, MEANB, SMPINC, VAR )
.....
VARIANCE      -  WILL FIND THE VARIANCE-COVARIANCE VALUE
                  BETWEEN TWO IMAGES.  THE IMAGES ARE EXPECTED
                  TO BE 512 X 512 IN SIZE AND 8 BITS DEEP.

WRITTEN BY CARL SALVAGGIO      OCTOBER 1986
.....

INTEGER*4      N, M, SMPINC
PARAMETER ( N = 512, M = 512 )

INTEGER*2      IMAGEA( N, M ), IMAGEB( N, M )
REAL*4         MEANA, MEANB, VAR
REAL*8         SUM, TOTNUM

.....          CALCULATE THE COVARIANCE (VARIANCE) BETWEEN TWO IMAGES

TOTNUM = INT( N / SMPINC )**2
SUM = 0.0
DO 20 J = 1, M, SMPINC
  DO 10 I = 1, N, SMPINC
    SUM = SUM + ( FLOAT( IMAGEA( I, J ) ) - MEANA ) *
                ( FLOAT( IMAGEB( I, J ) ) - MEANB )
  10 CONTINUE
20 CONTINUE

VAR = ( 1.0 / ( TOTNUM - 1 ) ) * SUM

.....          RETURN CONTROL TO CALLING PROGRAM
.....
RETURN
END

```

Appendix B

Description of the K-Means Unsupervised Clustering Code

Cluster
Display
MinDst
Merge
Elim

.....

CLUSTER - THIS PROGRAM WILL LOOK AT AN N-BAND IMAGE,
 CHOOSE K RANDOMLY LOCATED CENTROIDS, AND
 EXECUTE A K-MEANS CLUSTERING ALGORITHM. IMAGE
 PIXEL WILL BE ASSIGN TO THE NEAREST CENTROID
 (IN A EUCLIDEAN SENSE). ONCE ALL THE PIXEL
 HAVE BEEN ASSIGNED TO A CENTROID, THE CENTROIDS
 ARE RECOMPUTED, AND THE ASSIGNMENT PROCESS
 BEGINS AGAIN. THIS CONTINUES UNTIL THERE IS NO
 SIGNIFICANT CHANGE IN THE CLUSTER CENTROIDS.
 THIS ALGORITHM WAS DESIGNED AFTER FORGY'S
 METHOD, DESCRIBED IN:

M.R. ANDERBERG, CLUSTER ANALYSIS FOR APPLICATIONS, ACADEMIC
 PRESS, NEW YORK, 1973, PP. 159-162.

.....

VARIABLE DECLARATION

.....

CNTROD(,) - THE MATRIX OF CURRENT CENTROID VECTORS
 CVTB,CVTI - THE BYTE TO INTEGER TEMPORARY CONVERSION VALUES
 DIST() - THE ITH DISTANCE FROM 'POINT' TO 'CNTROD(,)'
 IRND,JRND - THE PIXEL COORDINATES OF THE RANDOM SEEDS
 ISEED - THE NUMBER OF SECONDS SINCE MIDNIGHT, USED TO
 MAXBND - MAXIMUM NUMBER OF IMAGE BANDS ALLOWED
 MAXCLS - MAXIMUM NUMBER OF CLUSTERS ALLOWED
 MINBND - MINIMUM NUMBER OF IMAGE BANDS ALLOWED
 MINPNT - THE NUMBER OF THE CLUSTER CLOSEST TO 'POINT()'
 NUMBND - THE NUMBER OF IMAGE BANDS
 NUMCLS - THE DESIRED NUMBER OF CLUSTERS
 OLD CNT(,) - THE CENTROID VALUE FROM THE PREVIOUS ITERATION
 PIXEL() - THE CURRENT PIXEL VECTOR WITH 'MAXBND' ELEMENTS
 ROWBUF() - TEMPORARY BYTE BUFFER TO STORE IMAGE ROW DATA
 TOL - THE MINIMUM TOLERANCE LEVEL FOR MERGING CLUSTERS
 SEED THE RANDOM NUMBER GENERATOR
 TOLNCE - THE MINIMUM TOLERANCE IN DC'S BETWEEN CLUSTER
 CENTROIDS TO TERMINATE THE ITERATIONS
 TOTAL - THE TOTAL NUMBER OF SAMPLED PIXELS

.....

REQUIRED SUBROUTINES: MINDST
 MERGE
 ELIM

.....

AUTHOR: CARL SALVAGGIO NOVEMBER 1, 1986
 ROCHESTER INSTITUTE OF TECHNOLOGY
 CENTER FOR IMAGING SCIENCE

.....

BYTE CVTB
 CHARACTER*1 ANS, ANS1, ANS2, BELL
 CHARACTER*80 INNAME, OUTNAME, DEFAULT
 CHARACTER*31 CMD
 CHARACTER*1 UNIT

```

      INTEGER*2      NUMBND, NUMCLS, CVTI, LUNSFT, MINPNT, TOL, NUM
      INTEGER*2      NUMMRG, NUMELM
      INTEGER*4      N, M, MaxBND, MINBND, SMPINC, MaxCLS, TOTAL
      INTEGER*4      CONTEXT, STATUS, LIB$FIND_FILE, ISEED
      REAL*4         IRND, JRND, TOLNCE

      C      PARAMETER      ( N = 512, M = 512 )
      PARAMETER      ( SMPINC = 1 )
      PARAMETER      ( MaxBND = 10, MINBND = 2 )
      PARAMETER      ( MaxCLS = 25 )
      PARAMETER      ( BELL = CHAR(7) )
      PARAMETER      ( LUNSFT = 9 )
      C      PARAMETER      ( TOLNCE = 1.0 )      !!!! COVERAGE TOLERANCE
      PARAMETER      ( TOL = 20 )      !!!! MERGING TOLERANCE

      BYTE           ROWBUF(M)
      CHARACTER*80    FILNAM(MaxBND)
      INTEGER*2       PIXEL(MaxBND)
      INTEGER*4       BIN(MaxCLS), SUM(MaxCLS,MaxBND)
      REAL*4          CNTRD(MaxCLS, MaxBND), OLD CNT(MaxCLS, MaxBND)
      REAL*4          DIST(MaxCLS), DST(MaxCLS,MaxCLS)

      EQUIVALENCE      ( CVTI, CVTB )

      *
      .....          OBTAIN THE NUMBER OF BANDS AND CLUSTERS

      1      CALL LIB$ERASE_PAGE(1,1)
      2      WRITE (6,2)
      FORMAT ('$', 'HOW MANY IMAGE BANDS DO YOU HAVE ? ')
      READ (5,*,END=900) NUMBND
      IF (( NUMBND .LT. MINBND ) .OR. ( NUMBND .GT. MaxBND )) THEN
         WRITE (6,*) BELL
         WRITE (6,*) '**** ERROR **** ILLEGAL NUMBER OF BANDS'
         WRITE (6,*)
         GOTO 1
      END IF
      3      WRITE (6,4)
      4      FORMAT ('$', 'HOW MANY CLUSTERS DO YOU WISH TO FORM ? ')
      READ (5,*,END=900) NUMCLS
      IF (( NUMCLS .LT. 1 ) .OR. ( NUMCLS .GT. MaxCLS )) THEN
         WRITE (6,*) BELL
         WRITE (6,*) '**** ERROR **** ILLEGAL NUMBER OF CLUSTERS'
         WRITE (6,*)
         GOTO 3
      END IF
      5      WRITE (6,5)
      FORMAT ('$', 'SAMPLING INCREMENT ( 1,2,4,8,16,32,64,128 ) : ')
      READ (5,*,END=900) SMPINC
      WRITE (6,6)
      6      FORMAT ('$', 'DO YOU WISH SIMILAR CLUSTERS TO BE MERGED ? ')
      READ (5,*(A),END=900) ANS1
      IF ( ANS1 .EQ. 'Y' .OR. ANS1 .EQ. 'y' ) ANS1 = 'Y'
      IF ( ANS1 .NE. 'Y' ) GOTO 8
      WRITE (6,7)
      7      FORMAT ('$', 'MINIMUM INTER-CLUSTER DISTANCE : ')
      READ (5,*,END=900) TOL
      WRITE (6,9)
      8      FORMAT ('$', 'ELIMINATE CLUSTERS WITH 0 PIXELS ? ')
      9

```

```

      READ (5, '(A)', END=900) ANS2
      IF ( ANS2 .EQ. 'Y' .OR. ANS2 .EQ. 'y' ) ANS2 = 'Y'
      IF ( ANS2 .NE. 'Y' ) GOTO 10
*
*.....      OBTAIN BAND IMAGERY NAMES (AND CHECK IF THEY EXIST)
*
10  WRITE (6,*)
      DEFAULT = 'PICS:[CNS9024.LANDSAT]*.PIX'
      DO 30 K = 1, NUMBND, 1
11  WRITE (6,20) K
20  FORMAT ('$', 'ENTER FILENAME FOR BAND', I3, ': ')
      READ (5, '(A)', END=900) FILNAM( K )
      INNAME = FILNAM( K )
      STATUS=LIB$FIND_FILE(INNAME, OUTNAME, CONTEXT, DEFAULT, OUTNAME)
      CALL LIB$FIND_FILE_END( CONTEXT )
      IF ( STATUS .EQ. 65537 ) THEN
          FILNAM( K ) = OUTNAME
      ELSE
          WRITE (6,*) BELL
          WRITE (6,*) '*** ERROR *** FILE NOT FOUND'
          WRITE (6,*)
          GOTO 11
      END IF
30  CONTINUE
*
*.....      OPEN THE 'NUMBND' IMAGE FILES
*
      DO 40 LUN = LUNSFT+1, NUMBND+LUNSFT, 1
          OPEN ( LUN, FILE=FILNAM( LUN-9 ), ACCESS='DIRECT',
+              STATUS='OLD', RECL=N/4, FORM='UNFORMATTED', READONLY )
40  CONTINUE
*
*.....      INITIALIZE THE CENTROID VECTORS
*
      ISEED = SECNDS(0.0)          !!!!! INITIALIZE SEED WITH THE SECONDS
                                   !!!!! SINCE MIDNIGHT
      DO 60 I = 1, NUMCLS, 1
          IRND = RAN( ISEED )
          JRND = RAN( ISEED )
          IROW = INT( IRND * 511.0 + 1 )
          JCOL = INT( JRND * 511.0 + 1 )
          DO 50 J = 1, NUMBND, 1
              READ ( J+LUNSFT, REC = IROW ) ROWBUF
              CVTB = ROWBUF( JCOL )
              CNTRD( I, J ) = CVTB
          50 CONTINUE
      60 CONTINUE
*
*.....      CHECK IF ANY OF THE CENTROIDS ARE EQUAL AND MERGE THEM
*
61  IF ( ANS1 .EQ. 'Y' ) THEN
      CALL MERGE( CNTRD, BIN, NUMCLS, NUMBND, TOL, MAXBND,

```

```

      *                               MAXCLS, DST, NUMMRG )
      END IF

*
*..... ELIMINATE ANY CLUSTER WITH 0 PIXELS
*

      IF ( ANS2 .EQ. 'Y' ) THEN
        IF ( ITER .EQ. 0 ) GOTO 66
        NUMELM = 0
        DO 64 I = 1, NUMCLS, 1
          IF ( BIN( I ) .EQ. 0 ) THEN
            NUMELM = NUMELM + 1
            CALL ELIM( CNTROD, BIN, NUMCLS, NUMBND, I, MAXBND, MAXCLS )
          END IF
        64 CONTINUE
      END IF

*
*..... PRINT OUT THE CURRENT CENTROIDS
*

      66 WRITE (6,*)
      WRITE (6,67) ITER
      67 FORMAT ('$', 'ITERATION #', I3)
      ITER = ITER + 1
      WRITE (6,62) NUMMRG
      62 FORMAT ('$', 'NUMBER OF CLUSTERS MERGED      = ', I3)
      WRITE (6,65) NUMELM
      65 FORMAT ('$', 'NUMBER OF CLUSTERS ELIMINATED = ', I3)
      WRITE (6,68)
      68 FORMAT ('$', 'CURRENT CENTROID VALUES :')
      WRITE (6,*)
      DO 80 I = 1, NUMCLS, 1
        WRITE (6,69) I
      69 FORMAT ('$', 'CLUSTER(', I3, ', ' )')
        WRITE (6,70) ( CNTROD( I, J ), J = 1, NUMBND, 1 ), BIN( I )
      70 FORMAT ('+', T5, (NUMBND)F7.1, I9, ' PIXELS')
      80 CONTINUE

*
*..... CLEAR THE BINS AND SUMS
*

      DO 86 I = 1, MAXCLS, 1
        BIN( I ) = 0
        DO 85 J = 1, MAXBND, 1
          SUM( I, J ) = 0
        85 CONTINUE
      86 CONTINUE

*
*..... GET THE CURRENT PIXEL
*

      DO 110 J = 1, N, SMPINC
        DO 100 K = 1, M, SMPINC
          DO 90 L = 1, NUMBND, 1
            READ ( L+LUNSFT, REC = J ) ROWBUF
            CVTB = ROWBUF( K )
          90 CONTINUE
        100 CONTINUE
      110 CONTINUE

```

!!!! ROW LOOP
!!!! COLUMN LOOP
!!!! BAND LOOP

```

90      PIXEL( L ) = CVTI
      CONTINUE                                     !!!! END BAND LOOP
*
*****      FIND WHICH CLUSTER THE CURRENT PIXEL IS CLOSEST TO
*
      CALL MINDST( PIXEL, CNTRD, NUMBND, NUMCLS, MINPNT,
+             MAXBND, MAXCLS, DIST )
*
*****      UPDATE THE PIXEL/CLUSTER COUNT AND SUM/CLUSTER VALUE
*
      BIN( MINPNT ) = BIN( MINPNT ) + 1
      DO 95 L = 1, NUMBND, 1
        SUM( MINPNT,L ) = SUM( MINPNT,L ) + PIXEL( L )
95      CONTINUE
100     CONTINUE                                     !!!! END COLUMN LOOP
110     CONTINUE                                     !!!! END ROW LOOP
*
*****      STORE OLD CENTROIDS AND COMPUTE NEW CENTROIDS
*
      DO 130 I = 1, NUMCLS, 1
        DO 120 J = 1, NUMBND, 1
          IF ( BIN( I ) .EQ. 0 ) GoTo 130
          OLD CNT( I,J ) = CNTRD( I,J )
          CNTRD( I,J ) = FLOAT( SUM( I,J ) ) / FLOAT( BIN( I ) )
120      CONTINUE
130      CONTINUE
*
*****      CHECK FOR CONVERGENCE OF THE CENTROIDS
*
      DO 150 I = 1, NUMCLS, 1
        DO 140 J = 1, NUMBND, 1
          IF ( BIN( I ) .EQ. 0 ) GoTo 150
          IF ( ABS(OLD CNT( I,J ) - CNTRD( I,J ) ) .GT. TOLNCE) THEN
            GoTo 61
          END IF
140      CONTINUE
150      CONTINUE
      WRITE (6,*)
      WRITE (6,160) TOLNCE
160      FORMAT ( '$', 'CONVERGENCE TOLERANCE OF ', F7.4, ' EXCEEDED' )
*
*****      CLOSE THE IMAGE FILES
*
      DO 190 LUN = LUNSFT+1, NUMBND+LUNSFT, 1
        CLOSE( LUN )
190      CONTINUE
*
*****      WRITE THE FINAL CENTROIDS TO A FILE

```

```

      OPEN ( 1, FILE='CLUSTER.OUT', STATUS='NEW' )
      DO 220 I = 1, NUMCLS, 1
        WRITE (1,210) ( CNTROD( I,J ), J = 1, NUMBND, 1 )
        FORMAT ( (NUMBND)F7.1 )
210    CONTINUE
220    CLOSE( 1 )

.....      RUN THE COLOR IMAGE GENERATION PROGRAM
.....

      WRITE (6,*)
      WRITE (6,230)
230    FORMAT ( '$', 'GENERATE A COLOR COMPOSITE IMAGE ( Y OR N ) ? ' )
      READ (5, '(A)', END=900) ANS

      IF ( ANS .EQ. 'Y' .OR. ANS .EQ. 'Y' ) THEN
235    WRITE (6,240)
240    FORMAT ( '$', 'ENTER UNIT NUMBER : ' )
      READ (5, '(A)', END=900) UNIT
      IF (UNIT.NE.'0' .AND. UNIT.NE.'1' .AND. UNIT.NE.'2') GOTO 235
      CMD = 'USER:[CNS9024.IMGOPS]DISPLAY '
      CMD( 31:31 ) = UNIT
      CALL LIB$SPAWN ( 'RUN USER:[CNS9024.IMGOPS]DISPLAY' )
      CALL LIB$SPAWN ( CMD )
      END IF

.....      PROGRAM TERMINATION
.....

900    WRITE (6,*)
      END

.....
      SUBROUTINE MINDst( POINT, CNTROD, NUMBND, NUMCLS, MINPNT,
+      MaxBND, MaxCLS, DIST )
.....
      MINDst -      THIS SUBROUTINE WILL CALCULATE THE DISTANCE
      OF THE CURRENT PIXEL TO EACH OF THE
      CENTROIDS. THE ALGORITHM WILL THEN
      DETERMINE WHICH OF THE DISTANCE IS THE MINIMUM
      AND RETURN THIS CLUSTER NUMBER TO THE CALLING
      PROGRAM.
.....

      VARIABLE DECLARATION

      CNTROD( , ) - THE MATRIX OF CURRENT CENTROID VECTORS
      DIST( ) - THE ITH DISTANCE FROM 'POINT' TO 'CNTROD( , )'
      MaxBND - MAXIMUM NUMBER OF IMAGE BANDS ALLOWED
      MaxCLS - MAXIMUM NUMBER OF CLUSTERS ALLOWED
      MINPNT - THE NUMBER OF THE CLUSTER CLOSEST TO 'POINT( )'
      NUMBND - THE NUMBER OF IMAGE BANDS
      NUMCLS - THE DESIRED NUMBER OF CLUSTERS

```

```

*      POINT( ) - THE CURRENT PIXEL VECTOR WITH 'MaxBnd' ELEMENTS
*      SUM      - A RUNNING TOTAL FOR DISTANCE CALCULATIONS
*.....
*      REQUIRED SUBROUTINES:          NONE
*.....
*      AUTHOR: CARL SALVAGGIO  NOVEMBER 11, 1986
*                        ROCHESTER INSTITUTE OF TECHNOLOGY
*                        CENTER FOR IMAGING SCIENCE
*.....

      INTEGER*2      MAXBND, MAXCLS
      INTEGER*2      MINPNT, NUMBND, NUMCLS
      INTEGER*2      POINT(MAXBND)
      REAL*4          CNTROD(MAXCLS,MAXBND), DIST(MAXCLS)
      REAL*4          SUM
*.....
*      CALCULATE THE DISTANCE FORM THE POINT TO EACH CENTROID
*
      DO 20 I = 1, NUMCLS, 1
        SUM = 0.0
        DO 10 J = 1, NUMBND, 1
          SUM = SUM + ( FLOAT( POINT( J ) ) - CNTROD( I,J ) )**2
10        CONTINUE
        DIST( I ) = SQRT( SUM )
20      CONTINUE
*.....
*      DETERMINE WHICH OF THE ITH DISTANCES IS A MINIMUM
*
      MINPNT = 1
      DO 30 I = 2, NUMCLS, 1
        IF ( DIST( MINPNT ) .GT. DIST( I ) ) MINPNT = I
30      CONTINUE
*.....
*      RETURN THE CLOSEST CLUSTER NUMBER TO CALLING PROGRAM
*
      RETURN
      END
*.....
*      SUBROUTINE MERGE( CNTROD, BIN, NUMCLS, NUMBND, TOL,
*      +               MAXBND, MAXCLS, DST, NUMMRG )
*.....
*      MERGE - THIS SUBROUTINE WILL EXAMINE ALL THE CENTROIDS
*              AND IF ANY TWO OF THEM ARE WITHIN A SPECIFIED
*              DISTANCE OF EACH OTHER, THIS PROGRAM WILL
*              MERGE THEM INTO ONE CENTROID
*.....
*      VARIABLE DECLARATION
*
*      TOL - THE MAXIMUM DISTANCE FOR EACH BAND VALUE

```

```

*          TO ALLOW CONVERGENCE
*
*          ALL OTHERS ARE THE SAME AS CALLING PROGRAM
*
*-----
*          AUTHOR: CARL SALVAGGIO  NOVEMBER 1, 1986
*                               ROCHESTER INSTITUTE OF TECHNOLOGY
*                               CENTER FOR IMAGING SCIENCE
*-----

      INTEGER*2      NUMCLS, NUMBND, MAXBND, MAXCLS, TOL
      INTEGER*2      NUMMRG
      REAL*4          SUM, DST(MAXCLS,MAXCLS)
      REAL*4          CNTRD(MAXCLS,MAXBND), BIN(MAXCLS)
      REAL*4          BINI, BINJ

*
*-----      COMPUTE A DISTANCE MATRIX
*
      NUMMRG = 0
5     DO 30 I = 1, NUMCLS, 1
        DO 20 J = 1, NUMCLS, 1
            SUM = 0.0
            DO 10 K = 1, NUMBND, 1
                SUM = SUM + ( CNTRD( I,K ) - CNTRD( J,K ) )**2
            10    CONTINUE
            DST( I,J ) = SQRT( SUM )
        20    CONTINUE
    30    CONTINUE

*
*-----      TEST FOR EQUAL CENTROIDS
*
      DO 60 I = 1, NUMCLS, 1
          DO 50 J = I+1, NUMCLS, 1
              !!!!! ROW LOOP
              !!!!! COLUMN LOOP

              IF ( DST( I,J ) .LE. TOL ) THEN
                  NUMCLS = NUMCLS - 1
                  NUMMRG = NUMMRG + 1
                  IF ( BIN( I ) .EQ. 0 .AND. BIN( J ) .EQ. 0 ) THEN
                      BINI = 1.0
                      BINJ = 1.0
                      DIVISOR = 2.0
                  ELSE
                      BINI = BIN( I )
                      BINJ = BIN( J )
                      BIN( I ) = BIN( I ) + BIN( J )
                      DIVISOR = BIN( I ) + BIN( J )
                  END IF
                  DO 40 K = 1, NUMBND, 1
                      CNTRD( I,K ) = ( BINI * CNTRD( I,K ) +
                                      BINJ * CNTRD( J,K ) ) /
                                      ( DIVISOR )
                  40    CONTINUE
              50    DO 35 L = J, NUMCLS, 1
                  DO 34 M = 1, NUMBND, 1
                      +
                      +
          40

```

```

          CNTROD( L,M ) = CNTROD( L+1,M )
34          CONTINUE
          BIN( L ) = BIN( L+1 )
35          CONTINUE
          GOTO 5
          END IF

50          CONTINUE
60          CONTINUE                                     !!!! END COLUMN LOOP
                                                    !!!! END ROW LOOP

*
*****      RETURN THE AMENDED CENTROIDS TO THE CALLING PROGRAM
*
      RETURN
      END

*****
+      SUBROUTINE ELIM( CNTROD, BIN, NUMCLS, NUMBND, NUM,
+                     MAXBND, MAXCLS )
*****

      INTEGER*2      NUMCLS, MAXBND, MAXCLS
      INTEGER*4      NUM
      REAL*4         CNTROD(MAXCLS,MAXBND), BIN(MAXCLS)

*
*****      REDUCE CLUSTERS BY 1
*

      NUMCLS = NUMCLS - 1

*
*****      MOVE THE REMAINING CENTROIDS UP ONE IN THE LIST
*

      DO 20 I = NUM, NUMCLS, 1
        DO 10 J = 1, NUMBND, 1
          CNTROD( I,J ) = CNTROD( I+1,J )
10          CONTINUE
          BIN( I ) = BIN( I+1 )
20          CONTINUE
*
*****      RETURN TO CALLING PROGRAM
*

      RETURN
      END

```

```

.....
.
.   DISPLAY -      THIS PROGRAM WILL TAKE THE FINAL CENTROID
.                   VALUES COMPUTED IN CLUSTER.FOR AND CREATE
.                   A COLOR COMPOSITE REPRESENTATION OF THE
.                   CLASSIFIED IMAGE. THE IMAGE IS CLASSIFIED
.                   ACCORDING TO A MINIMUM DISTANCE TO THE MEAN
.                   CLASSIFIER SINCE ONLY THE CENTROID DATA IS
.                   KNOWN. THE COLORS THAT ARE ASSIGNED TO EACH OF
.                   THE CLASSES ARE ARBITRARILY FIXED AND CAN BE
.                   SEEN IN THIS CODE. AS OF NOW A MAXIMUM OF 10
.                   CLUSTERS CAN BE DISPLAYED.
.
.....

```

VARIABLE DECLARATION

```

.
.   CNTROD( , ) -   THE MATRIX OF CURRENT CENTROID VECTORS
.   CVTBR,CVTIR -   RED BYTE TO INTEGER TEMPORARY CONVERSION VALUES
.   CVTBG,CVTIG -   GREEN BYTE TO INTEGER TEMPORARY CONVERSION VALUES
.   CVTBB,CVTIB -   BLUE BYTE TO INTEGER TEMPORARY CONVERSION VALUES
.   CVTB,CVTI  -   BYTE TO INTEGER TEMPORARY CONVERSION VALUES
.   DIST( )     -   THE ITH DISTANCE FROM 'POINT' TO 'CNTROD( , )'
.   IMGSEC(,,) -   SECTION OF IMAGE IN CORE MEMORY THAT IS BEING
.                   OPERATED ON
.   MAXBND      -   MAXIMUM NUMBER OF IMAGE BANDS ALLOWED
.   MAXCLS      -   MAXIMUM NUMBER OF CLUSTERS ALLOWED
.   MINBND      -   MINIMUM NUMBER OF IMAGE BANDS ALLOWED
.   MINPNT      -   THE NUMBER OF THE CLUSTER CLOSEST TO 'POINT( )'
.   NUMBND      -   THE NUMBER OF IMAGE BANDS
.   NUMCLS      -   THE DESIRED NUMBER OF CLUSTERS
.   PIXEL( )    -   THE CURRENT PIXEL VECTOR WITH 'MAXBND' ELEMENTS
.   RDIGCNT,    -
.   GDIGCNT,    -
.   BDIGCNT     -   THE R,G,B DIGITAL COUNTS FOR THE OUTPUT IMAGE
.   ROUTROW(,,) -
.   GOUTROW(,,) -
.   BOUTROW(,,) -   BYTE R,G,B ROW OF DATA FOR THE OUTPUT IMAGE
.   ROWBUF( )   -   TEMPORARY BYTE BUFFER TO STORE IMAGE ROW DATA
.
.....

```

```

.
.   REQUIRED SUBROUTINES:      MINDST
.
.....

```

```

.
.   AUTHOR: CARL SALVAGGIO   NOVEMBER 11, 1986
.                           ROCHESTER INSTITUTE OF TECHNOLOGY
.                           CENTER FOR IMAGING SCIENCE
.
.....

```

```

.
.   BYTE          CVTBR, CVTBG, CVTBB, CVTB
.   CHARACTER*1    BELL
.   CHARACTER*80    INNAME, OUTNAME, DEFAULT
.   INTEGER*2      CVTIR, CVTIG, CVTIB, CVTI
.   INTEGER*2      NUMBND, NUMCLS, LUNSFT, MINPNT
.   INTEGER*2      RDIGCNT, GDIGCNT, BDIGCNT, SECINC
.   INTEGER*4      N, M, B, MAXBND, MINBND, SMPINC, MAXCLS
.   INTEGER*4      CONTEXT, STATUS, LIB$FIND_FILE
.

```

```

PARAMETER      ( N = 512, M = 512, B = 1536 )
PARAMETER      ( SMPINC = 1 )
PARAMETER      ( MAXBND = 10 , MINBND = 2 )
PARAMETER      ( MAXCLS = 10 )
PARAMETER      ( BELL = CHAR(7) )
PARAMETER      ( LUNSFT = 9 )
PARAMETER      ( SECINC = 64 )

BYTE            LINBUF(B), LINONE(M), LINTWO(M), LINTHR(M)
BYTE            ROWBUF(M)
BYTE            ROUTROW(SECINC,M), GOUTROW(SECINC,M)
BYTE            BOUTROW(SECINC,M)
CHARACTER*80    FILNAM(MAXBND)
INTEGER*2       PIXEL(MAXBND)
INTEGER*2       IMGSEC(SECINC,M,MAXBND)
REAL*4          CNTRD(MAXCLS, MAXBND)
REAL*4          DIST(MAXCLS)

EQUIVALENCE     ( CVTIR, CVTBR )
EQUIVALENCE     ( CVTIG, CVTBG )
EQUIVALENCE     ( CVTIB, CVTBB )
EQUIVALENCE     ( CVTI, CVTB )
EQUIVALENCE     ( LINBUF( 1 ), LINONE( 1 ) )
EQUIVALENCE     ( LINBUF( 513 ), LINTWO( 1 ) )
EQUIVALENCE     ( LINBUF( 1025 ), LINTHR( 1 ) )

```

```

.
.....      OBTAIN THE NUMBER OF BANDS AND CLUSTERS
.

```

```

      CALL LIB$ERASE_PAGE(1,1)
3    WRITE (6,4)
4    FORMAT ('$', 'HOW MANY IMAGE BANDS DID YOU HAVE ? ')
      READ (5,*,END=900) NUMBND
      IF (( NUMBND .LT. MINBND ) .OR. ( NUMBND .GT. MAXBND )) THEN
        WRITE (6,*) BELL
        WRITE (6,*) '**** ERROR **** ILLEGAL NUMBER OF BANDS'
        WRITE (6,*)
        GOTO 3
      END IF
5    WRITE (6,6)
6    FORMAT ('$', 'HOW MANY CLUSTERS DID YOU PRODUCE ? ')
      READ (5,*,END=900) NUMCLS
      IF (( NUMCLS .LT. 1 ) .OR. ( NUMCLS .GT. MAXCLS )) THEN
        WRITE (6,*) BELL
        WRITE (6,*) '**** ERROR **** ILLEGAL NUMBER OF CLUSTERS'
        WRITE (6,*)
        GOTO 5
      END IF

```

```

.
.....      OBTAIN BAND IMAGERY NAMES (AND CHECK IF THEY EXIST)
.

```

```

      WRITE (6,*)
      DEFAULT = 'PICS:[CNS9024.LANDSAT]*.Pix'
      DO 30 K = 1, NUMBND, 1
10     WRITE (6,20) K
20     FORMAT ('$', 'ENTER FILENAME FOR BAND', I3, ': ')

```

```

      READ (5, '(A)', END=900) FILNAM( K )
      INNAME = FILNAM( K )
      STATUS=LIB$FIND_FILE( INNAME, OUTNAME, CONTEXT, DEFAULT, OUTNAME )
      CALL LIB$FIND_FILE_END( CONTEXT )
      IF ( STATUS .EQ. 65537 ) THEN
        FILNAM( K ) = OUTNAME
      ELSE
        WRITE (6,*) BELL
        WRITE (6,*) '*** ERROR *** FILE NOT FOUND'
        WRITE (6,*)
        GOTO 10
      END IF
30    CONTINUE

*
*.....    DEFINE THE CENTROID VECTORS
*

      OPEN ( 1, FILE='CLUSTER.OUT', STATUS='OLD' )
      DO 40 I = 1, NUMCLS, 1
        READ (1,*) ( CNTRD( I,J ), J = 1, NUMBND, 1 )
40    CONTINUE
      CLOSE( 1 )

*
*.....    OPEN THE 'NUMBND' IMAGE FILES
*

      DO 50 LUN = LUNSFT+1, NUMBND+LUNSFT, 1
        OPEN ( LUN, FILE=FILNAM( LUN-9 ), ACCESS='DIRECT',
          + STATUS='OLD', RECL=N/4, FORM='UNFORMATTED', READONLY )
50    CONTINUE

*
*.....    BUILD THE OUTPUT IMAGE
*

      + OPEN ( 1, FILE='DISPLAY.PIX', ACCESS='DIRECT',
        STATUS='NEW', RECL=N/4, FORM='UNFORMATTED' )

*
*.....    GET THE CURRENT SECTION
*

      WRITE (6,*)
      WRITE (6,*) '... BUILDING R,G,B OUTPUT IMAGE'
      DO 130 J = 1, N, SECINC          !!!! SECTION LOOP
        DO 80 K = 1, SECINC, SMPINC    !!!! ROW LOOP
          DO 70 L = 1, NUMBND, 1       !!!! BAND LOOP
            READ ( L+LUNSFT, REC = J+K-1 ) ROWBUF
            DO 60 I = 1, M, 1           !!!! COLUMN LOOP
              CVTB = ROWBUF( I )
              IMGSEC( K,I,L ) = CVTI
60            CONTINUE                !!!! END COLUMN LOOP
70          CONTINUE                  !!!! END BAND LOOP
80        CONTINUE                    !!!! END ROW LOOP

*
*.....    FIND WHICH CLUSTER THE CURRENT PIXEL IS CLOSEST TO
*
```

```

Do 110 K = 1, SECINC, SMPINC          !!!! ROW LOOP
Do 100 I = 1, M, 1                    !!!! COLUMN LOOP
Do 90 II = 1, NUMBND, 1                !!!! BAND LOOP
    PIXEL( II ) = IMGSEC( K, I, II )
    CONTINUE                           !!!! END BAND LOOP
    CALL MINDST( PIXEL, CNTROD, NUMBND, NUMCLS, MINPNT,
                MAXBND, MAXCLS, DIST )
90
+
* .....
*
BUILD CURRENT ROW OF THE DISPLAY IMAGE

IF ( MINPNT .EQ. 1 ) THEN              !!!! GROUP 1 RED
    RDIGCNT = 255
    GDIGCNT = 0
    BDIGCNT = 0
ELSEIF ( MINPNT .EQ. 2 ) THEN          !!!! GROUP 2 GREEN
    RDIGCNT = 0
    GDIGCNT = 255
    BDIGCNT = 0
ELSEIF ( MINPNT .EQ. 3 ) THEN          !!!! GROUP 3 BLUE
    RDIGCNT = 0
    GDIGCNT = 0
    BDIGCNT = 255
ELSEIF ( MINPNT .EQ. 4 ) THEN          !!!! GROUP 4 YELLOW
    RDIGCNT = 255
    GDIGCNT = 255
    BDIGCNT = 0
ELSEIF ( MINPNT .EQ. 5 ) THEN          !!!! GROUP 5 MAGENTA
    RDIGCNT = 255
    GDIGCNT = 0
    BDIGCNT = 255
ELSEIF ( MINPNT .EQ. 6 ) THEN          !!!! GROUP 6 CYAN
    RDIGCNT = 0
    GDIGCNT = 255
    BDIGCNT = 255
ELSEIF ( MINPNT .EQ. 7 ) THEN          !!!! GROUP 7 WHITE
    RDIGCNT = 255
    GDIGCNT = 255
    BDIGCNT = 255
ELSEIF ( MINPNT .EQ. 8 ) THEN          !!!! GROUP 8 BLACK
    RDIGCNT = 0
    GDIGCNT = 0
    BDIGCNT = 0
ELSEIF ( MINPNT .EQ. 9 ) THEN          !!!! GROUP 9 DARK GREEN
    RDIGCNT = 0
    GDIGCNT = 100
    BDIGCNT = 0
ELSEIF ( MINPNT .EQ. 10 ) THEN         !!!! GROUP 10 BARK BLUE
    RDIGCNT = 0
    GDIGCNT = 0
    BDIGCNT = 100
END IF
CvTIR = RDIGCNT
CvTIG = GDIGCNT
CvTIB = BDIGCNT
ROUTROW( K,I ) = CvTBR
GOUTROW( K,I ) = CvTBG
BOUTROW( K,I ) = CvTBB
100
CONTINUE                               !!!! END COLUMN LOOP

```

```

110      CONTINUE
.
+-----+      OUTPUT CURRENT DISPLAY IMAGE ROW
.
      DO 120 K = 1, SECINC, SMPINC
      DO 115 L = 1, B, 3
        LINBUF( L ) = ROUTROW( K, L / 3 + 1 )
        LINBUF( L + 1 ) = GOUTROW( K, L / 3 + 1 )
        LINBUF( L + 2 ) = BOUTROW( K, L / 3 + 1 )
115      CONTINUE
        RECNUM = RECNUM + 1
        WRITE ( 1, REC=RECNUM ) LINONE
        RECNUM = RECNUM + 1
        WRITE ( 1, REC=RECNUM ) LINTWO
        RECNUM = RECNUM + 1
        WRITE ( 1, REC=RECNUM ) LINTHR
120      CONTINUE
130      CONTINUE                                     !!!! END SECTION LOOP
      CLOSE( 1 )
.
+-----+      CLOSE THE IMAGE FILES
.
      DO 140 LUN = LUNSFT+1, NUMBND+LUNSFT, 1
        CLOSE( LUN )
140      CONTINUE
.
+-----+      PROGRAM TERMINATION
.
900      WRITE ( 6, *)
      END

+-----+
+      SUBROUTINE MINDST( POINT, CNTROD, NUMBND, NUMCLS, MINPNT,
+      MAXBND, MAXCLS, DIST )
+-----+
.
.      MINDST -      THIS SUBROUTINE WILL CALCULATE THE DISTANCE
.                      OF THE CURRENT PIXEL TO EACH OF THE
.                      CENTROIDS. THE ALGORITHM WILL THEN
.                      DETERMINE WHICH OF THE DISTANCE IS THE MINIMUM
.                      AND RETURN THIS CLUSTER NUMBER TO THE CALLING
.                      PROGRAM.
.
+-----+
.
.      VARIABLE DECLARATION
.
.      CNTROD( , ) -   THE MATRIX OF CURRENT CENTROID VECTORS
.      DIST( )      -   THE ITH DISTANCE FROM 'POINT' TO 'CNTROD( , )'
.      MAXBND       -   MAXIMUM NUMBER OF IMAGE BANDS ALLOWED
.      MAXCLS       -   MAXIMUM NUMBER OF CLUSTERS ALLOWED

```

```

*      MINPNT      -   THE NUMBER OF THE CLUSTER CLOSEST TO 'POINT( )'
*      NUMBND      -   THE NUMBER OF IMAGE BANDS
*      NUMCLS      -   THE DESIRED NUMBER OF CLUSTERS
*      POINT( )    -   THE CURRENT PIXEL VECTOR WITH 'MaxBND' ELEMENTS
*      SUM         -   A RUNNING TOTAL FOR DISTANCE CALCULATIONS
*
*.....
*      REQUIRED SUBROUTINES:          NONE
*
*.....
*      AUTHOR: CARL SALVAGGIO  NOVEMBER 11, 1986
*                        ROCHESTER INSTITUTE OF TECHNOLOGY
*                        CENTER FOR IMAGING SCIENCE
*
*.....

      INTEGER*2      MAXBND, MAXCLS
      INTEGER*2      MINPNT, NUMBND, NUMCLS
      INTEGER*2      POINT(MAXBND)
      REAL*4          CNTRD(MAXCLS,MAXBND), DIST(MAXCLS)
      REAL*4          SUM

*
*.....      CALCULATE THE DISTANCE FORM THE POINT TO EACH CENTROID
*
      DO 20 I = 1, NUMCLS, 1
        SUM = 0.0
        DO 10 J = 1, NUMBND, 1
          SUM = SUM + ( FLOAT( POINT( J ) ) - CNTRD( I,J ) )**2
10        CONTINUE
        DIST( I ) = SORT( SUM )
20      CONTINUE

*
*.....      DETERMINE WHICH OF THE ITH DISTANCES IS A MINIMUM
*
      MINPNT = 1
      DO 30 I = 2, NUMCLS, 1
        IF ( DIST( MINPNT ) .GT. DIST( I ) ) MINPNT = I
30      CONTINUE

*
*.....      RETURN THE CLOSEST CLUSTER NUMBER TO CALLING PROGRAM
*
      RETURN
      END

```

Appendix C

Description of the Automated Rate of Change Segmentation Algorithm

**BldPIF
FindPIF
Gradient
Interval
ScaleDivide**

BLDPif - THIS IS AN AUTOMATED PROGRAM WHICH REQUIRES THE INPUT OF AN INFRARED, RED AND FAR INFRARED BAND IMAGERY AND WILL OUTPUT AN IMAGE WHICH CONSISTS OF ONLY PSEUDO INVARIANT FEATURES (PIF's). THE PROGRAM WILL UTILIZE THE RATE OF CHANGE INFORMATION IN THREE-DIMENSIONS OF THE NUMBER OF PIXELS IN ORDER TO LOCATE APPROPRIATE 4/3 AND BAND 7 THRESHOLD VALUES. THE APPROPRIATE BANDS WILL THEN BE THRESHOLDED AND LOGICALLY COMBINED TO PRODUCE THE PIF MASK FOR THE CURRENT IMAGE SET.

VARIABLE DECLARATION:

```

ANDCHAN - THE IMAGE MEMORY CHANNEL THAT WILL CONTAIN THE
          RESULT OF THE LOGICAL .AND. OF THE THRESHOLDED
          4/3 IMAGE AND THE THRESHOLDED BAND 7 IMAGE
BAND3 - THE NAME OF THE FILE CONTAINING THE BAND 3 IMAGE
BAND4 - THE NAME OF THE FILE CONTAINING THE BAND 4 IMAGE
BAND7 - THE NAME OF THE FILE CONTAINING THE BAND 7 IMAGE
CHANNEL4TO3 - THE IMAGE MEMORY PLANE THAT WILL CONTAIN THE
              4/3 (INFRARED TO RED ) RATIO IMAGE
CHANNEL7 - THE IMAGE MEMORY PLANE THAT WILL CONTAIN THE
            BAND 7 (FAR INFRARED) IMAGE
EXISTS - THE LOGICAL VARIABLE THAT CHECKS FOR FILE
          EXISTENCE
FOURTOTHREE - THE NAME OF THE FILE CONTAINING THE BAND 4/3
              RATIO IMAGE
HIST - THE 256 ELEMENT VECTOR THAT CONTAINS THE HISTOGRAM
        OF THE ANDCHAN AFTER RETURN FROM THE HISTOGRAM
        SUBROUTINE
INC4TO3 - THE INCREMENT VALUE FOR THE THRESHOLDING FOR THE
          BAND 4/3 RATIO IMAGE
INC7 - THE INCREMENT VALUE FOR THE THRESHOLDING FOR THE
        BAND 7 IMAGE
MASKFIL - THE NAME OF THE FILE CONTAINING TO WRITE THE
          PIF MASK TO
STP4TO3 - THE STOPPING VALUE FOR THRESHOLDING FOR THE BAND
          4/3 RATIO IMAGE
STR4TO3 - THE STARTING VALUE FOR THRESHOLDING FOR THE BAND
          4/3 RATIO IMAGE
STP7 - THE STOPPING VALUE FOR THRESHOLDING FOR THE BAND
        7 IMAGE
STR7 - THE STARTING VALUE FOR THRESHOLDING FOR THE BAND
        7 IMAGE
THRESH4TO3 - THE CURRENT THRESHOLD VALUE FOR THE BAND 4/3 RATIO
             IMAGE
THRESH7 - THE CURRENT THRESHOLD VALUE FOR THE BAND 7 IMAGE
TOTAL - THE TOTAL NUMBER OF PIXELS IN A 512 x 512 IMAGE
UNIT - THE HOLDER VARIABLE THAT WILL CONTAIN THE UNIT
        NUMBER ATTACHED DURING THE CALL TO IPI.ATTUNIT

```

REQUIRED SUBROUTINES: USER: [CNS9024.DEANZA] SHOMNO
THRESHOLD

Appendix C

```

WRITTT
LOGICAL AND
HISTOGRAM
BINARY
TRIMNO
USER:[Cns9024.P1F] FINDPIF
INTERVAL
SCALEDIVIDE
GRADIENT
USER:[Sls4255.C1s.IPI] IPI_ATTUNIT
IPI_DETUNIT

```

```

AUTHOR:      CARL SALVAGGIO      CENTER FOR IMAGING SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
JANUARY 31, 1987

```

```

CHARACTER*1      BELL, ANS
CHARACTER*80     FOURTOHREE, BAND3, BAND4, BAND7, MASKFIL
INTEGER*4        CHANNEL4TO3, CHANNEL7, ANDCHAN, UNIT, TOTAL
INTEGER*4        HIST(0:255)
INTEGER*4        SAVEX, SAVEY
INTEGER*2        STR4TO3, STP4TO3, INC4TO3, STR7, STP7, INC7
INTEGER*2        THRESH4TO3, THRESH7
LOGICAL*1        EXISTS
REAL*4           NUMBERS(256,256), GRAD(256,256)

```

```

.
..... INITIALIZE CONSTANTS

```

PARAMETER (BELL = CHAR(7))

```
DATA CHANNEL4TO3      / 0 /
DATA CHANNEL7          / 1 /
DATA ANDCHAN          / 2 /
DATA TOTAL             / 262144 /
DATA FOURTOTHREE      / 'TEMP. OUT' /
```

```
1  FORMAT ('+',A1)                                ! BELL RING
```

```

*
*****      GET NAME FOR THE BAND 3 IMAGE
*

```

```

2  WRITE (6,*)
5  WRITE (6,5)
  FORMAT ('$',_,3: ')
  READ (5, '(A)', END=900) BAND3
  INQUIRE ( FILE = BAND3, EXIST = EXISTS )
  IF ( EXISTS .EQ. .FALSE. ) THEN
    WRITE (6,1) BELL
    GO TO 2
  END IF

```

..... GET NAME FOR THE BAND 4 IMAGE

```

.
10 WRITE (6,20)
20 FORMAT ('$', '_4: ')
  READ (5, '(A)', END=900) BAND4
  INQUIRE ( FILE = BAND4, EXIST = EXISTS )
  IF ( EXISTS .EQ. .FALSE. ) THEN
    WRITE (6,1) BELL
    GOTO 10
  END IF

.
..... GET NAME FOR THE BAND 7 IMAGE
.

30 WRITE (6,40)
40 FORMAT ('$', '_7: ')
  READ (5, '(A)', END=900) BAND7
  INQUIRE ( FILE = BAND7, EXIST = EXISTS )
  IF ( EXISTS .EQ. .FALSE. ) THEN
    WRITE (6,1) BELL
    GOTO 30
  END IF

.
..... CALCULATE THE BAND 4/3 RATIO IMAGE
.

  CALL SCALEDIVIDE ( BAND4, BAND3, FOURTOTHREE )

.
..... GET THE THRESHOLDING REGION FOR 4/3 RATIO AND BAND 7 IMAGE
.

  CALL INTERVAL ( FOURTOTHREE, BAND7, STR4TO3, STP4TO3,
    .              INC4TO3, STR7, STP7, INC7 )

.
..... ATTACH THE UNIT TO THE CURRENT PROGRAM
.

  CALL IPI_ATTUNIT ( UNIT )

.
..... PLACE THE 4/3 RATIO IMAGE IN CHANNEL 0
.

  CALL SHOWN0 ( UNIT, FOURTOTHREE, CHANNEL4TO3 )

.
..... 4/3 RATIO THRESHOLD LOOP
.

  DO 200 THRESH4TO3 = STR4TO3, STP4TO3, INC4TO3

.
..... PLACE REFRESHED BAND 7 IMAGE IN CHANNEL 1
.

  CALL SHOWN0 ( UNIT, BAND7, CHANNEL7 )

```

```

.
:..... THRESHOLD THE 4/3 RATIO IMAGE AND SAVE TO CHANNEL 0
.

CALL THRESHOLD ( UNIT, CHANNEL4TO3, THRESH4TO3, 1 )
CALL WRITTT ( UNIT, CHANNEL4TO3, CHANNEL4TO3 )

.
:..... BAND 7 IMAGE THRESHOLD LOOP
.

DO 100 THRESH7 = STR7, STP7, INC7

.
:..... THRESHOLD THE BAND 7 IMAGE AND SAVE TO CHANNEL 1
.

CALL THRESHOLD ( UNIT, CHANNEL7, THRESH7, 0 )
CALL WRITTT ( UNIT, CHANNEL7, CHANNEL7 )

.
:..... .AND. THE TWO THRESHOLDED IMAGES (CHANNEL 0 & 1 )
.

CALL LOGICAL_AND ( UNIT, CHANNEL7, CHANNEL4TO3, ANDCHAN )
CALL BINARY ( UNIT, ANDCHAN )
CALL WRITTT ( UNIT, ANDCHAN, ANDCHAN )

.
:..... HISTOGRAM THE RESULT OF THE .AND. (CHANNEL 2)
.

CALL HISTOGRAM ( UNIT, ANDCHAN, HIST )

.
:..... DISPLAY THE THRESHOLD VALUES AND # OF PIXELS
.

WRITE (6,*) THRESH4TO3, THRESH7, TOTAL - HIST(0)

.
:..... FILL THE # OF PIXEL SURFACE MATRIX
.

NUMBERS(THRESH4TO3,THRESH7) = TOTAL - HIST(0)

.
:..... END THE BAND 7 THRESHOLD LOOP
.

100 CONTINUE

.
:..... END THE 4/3 RATIO THRESHOLD LOOP
.

200 CONTINUE
.

```

```

.....    TEMPORARY OUTPUT
.

D      OPEN ( 1, FILE='TEMP.NUM', STATUS='NEW' )
D      DO 700 I = STR7, STP7, INC7
D      DO 600 J = STP4TO3, STR4TO3, -INC4TO3
D      WRITE (1,*) J, I, NUMBERS(J,I)
D 600    CONTINUE
D 700    CONTINUE
D      CLOSE ( 1 )

.
.....    COMPUTE THE GRADIENT
.

.    CALL GRADIENT ( STR4TO3, STP4TO3, INC4TO3,
.                  STR7, STP7, INC7, NUMBERS, GRAD )

.
.....    TEMPORARY OUTPUT
.

D      OPEN ( 1, FILE='TEMP.GRD', STATUS='NEW' )
D      DO 850 I = STR7, STP7, INC7
D      DO 800 J = STP4TO3, STR4TO3, -INC4TO3
D      WRITE (1,*) J, I, GRAD(J,I)
D 800    CONTINUE
D 850    CONTINUE
D      CLOSE ( 1 )

.
.....    FIND THE PIF THRESHOLD VALUES
.

.    CALL FINDPIF ( STR4TO3, STP4TO3, INC4TO3,
.                  STR7, STP7, INC7, GRAD, SAVEX, SAVEY )

WRITE (6,*)
WRITE (6,*) 'THRESHOLD FOR 4/3 RATIO = ',SAVEX
WRITE (6,*) 'THRESHOLD FOR BAND 7   = ',SAVEY
WRITE (6,*)

.
.....    RELOAD AND THRESHOLD THE BAND 4/3 RATIO IMAGE
.

CALL SHOMNO ( UNIT, FOURTOHREE, CHANNEL4TO3 )
CALL THRESHOLD ( UNIT, CHANNEL4TO3, SAVEX, 1 )
CALL WRTITT ( UNIT, CHANNEL4TO3, CHANNEL4TO3 )

.
.....    RELOAD AND THRESHOLD THE BAND 7 IMAGE
.

CALL SHOMNO ( UNIT, BAND7, CHANNEL7 )
CALL THRESHOLD ( UNIT, CHANNEL7, SAVEY, 0 )
CALL WRTITT ( UNIT, CHANNEL7, CHANNEL7 )

.
.....    .AND. THE TWO IMAGES TO FIND THE PIF MASK

```

Appendix C

```

*
      CALL LOGICAL_AND ( UNIT, CHANNEL7, CHANNEL4TO3, ANDCHAN )
*
*.....      MAKE THE PIF MASK A BINARY IMAGE
*
      CALL BINARY ( UNIT, ANDCHAN )
      CALL WRITTT ( UNIT, ANDCHAN, ANDCHAN )
      CALL TRIMNO ( UNIT, ANDCHAN )
*
*.....      SAVE THE PIF MASK
*
      WRITE (6,*)
      WRITE (6,870)
870   FORMAT ('$', 'DO YOU WISH TO SAVE THIS PIF MASK (Y OR N)? ')
      READ (5, '(A)', END=900) ANS
      CALL STR$UPCASE (ANS, ANS)
      IF ( ANS .EQ. 'Y' ) THEN
         WRITE (6,*)
         WRITE (6,880)
880   FORMAT ('$', 'ENTER FILE NAME TO STORE MASK TO: ')
         READ (5, '(A)', END=900) MASKFIL
         CALL SAVMNO ( UNIT, MASKFIL, ANDCHAN )
      END IF
*
*.....      TERMINATE THE PROGRAM
*
900   CALL LIB$SPAWN ( 'DELETE/NOCONFIRM TEMP.OUT;*' )
      CALL TRIMNO ( UNIT, -1 )
      CALL IPI_DETUNIT ( UNIT )
      WRITE (6,*)
      END
```

```

SUBROUTINE FINDPIF ( STR4TO3, STP4TO3, INC4TO3,
                   STR7, STP7, INC7, GRAD, SAVEX, SAVEY )
.....
FINDPIF -      WILL EXAMINE THE GRADIENT OF THE NUMBER OF
                PIXEL IN THE HISTOGRAM DATA TO FIND THE PLATEAU
                POINT WHERE THE THRESHOLD POINTS FOR THE PIF
                MASK SHOULD BE LOCATED.
.....

VARIABLE DECLARATION -
STR4TO3 -      THE STARTING VALUE FOR THRESHOLD OF THE BAND
                4/3 RATIO
STP4TO3 -      THE STOPPING VALUE FOR THRESHOLD OF THE BAND
                4/3 RATIO
INC4TO3 -      THE INCREMENT VALUE FOR THE THRESHOLD OF THE BAND
                4/3 RATIO
STR7 -        THE STARTING VALUE FOR THRESHOLD OF THE BAND
                7 IMAGE
STP7 -        THE STOPPING VALUE FOR THRESHOLD OF THE BAND
                7 IMAGE
INC7 -        THE INCREMENT VALUE FOR THE THRESHOLD OF THE BAND
                7 IMAGE
GRAD -        THE ARRAY CONTAINING THE GRADIENT DATA
SAVEX -       WILL RETURN THE VALUE OF THE 4/3 THRESHOLD
SAVEY -       WILL RETURN THE VALUE OF THE 7 THRESHOLD
.....

REQUIRED SUBROUTINES:      NONE
.....

AUTHOR: CARL SALVAGGIO    JANUARY 8, 1987
                           ROCHESTER INSTITUTE OF TECHNOLOGY
                           CENTER FOR IMAGING SCIENCE
MODIFIED      JANUARY 31, 1987  C.S.
MODIFIED      FEBRUARY 6, 1987  C.S.
.....

INTEGER*2      STR4TO3, STP4TO3, INC4TO3, STR7, STP7, INC7
INTEGER*4      SAVEX, SAVEY, I
REAL*4         GRAD(256,256)
REAL*4         SAVEMAX, MAXIMUM, SAVEMIN, MINIMUM, SAVEGRAD
REAL*4         INC

.....
SEARCH PARALLEL TO THE THRESH 7 AXIS TO FIND MAX

SAVEMAX = 0.0
MAXIMUM = 0.0

DO 10 I = STR7+(2*INC7), STP7, INC7
    MAXIMUM = MAX( MAXIMUM, GRAD(STR4TO3+2*(INC4TO3),I) )
D    WRITE (6,*) GRAD(STR4TO3+(2*INC4TO3),I),
D    STR4TO3+(2*INC4TO3), I

```

```

      IF ( MAXIMUM .NE. SAVEMax ) THEN
        SAVEY = 1
        WRITE (6,*) '..... SAVEY = ',SAVEY
      END IF
      SAVEMax = MAXIMUM
10    CONTINUE

*
*..... SEARCH PARALLEL TO THRESH 4/3 TO FIND LOCAL MINIMA
*

      MINIMUM = 1.0E32
      SAVEMIN = 0.0
      SAVEGRAD = 0.0

D     WRITE (6,*)
D     DO 20 I = STP4TO3-INC4TO3, STR4TO3+INC4TO3, -INC4TO3
D       WRITE (6,*) GRAD(I,SAVEY)
D       IF ( GRAD(I,SAVEY) .GE. SAVEGRAD ) THEN
D         SAVEGRAD = GRAD(I,SAVEY)
D         GOTO 20
D       END IF
D       MINIMUM = MIN( MINIMUM, GRAD(I,SAVEY) )
D       IF ( MINIMUM .NE. SAVEMIN ) THEN
D         IFLAG = 1
D         SAVEX = 1
D         WRITE (6,*) '..... SAVEX = ',SAVEX
D       END IF
D       SAVEMIN = MINIMUM
20    CONTINUE

*
*..... IF NO VALLEY FOUND SEARCH FOR FIN PARALLEL TO BAND 7 AXIS
*

      IF ( IFLAG .EQ. 0 ) THEN
        SAVEMax = 0.0
        MAXIMUM = 0.0
D       WRITE (6,*)
D       DO 30 I = STP4TO3-INC4TO3, STR4TO3+INC4TO3, -INC4TO3
D         WRITE (6,*) GRAD(I,STR7+(2*INC7))
D         MAXIMUM = MAX( MAXIMUM, GRAD(I,STR7+(2*INC7)) )
D         IF ( MAXIMUM .NE. SAVEMax ) THEN
D           SAVEX = 1
D           WRITE (6,*) '..... SAVEX = ',SAVEX
D         END IF
D         SAVEMax = MAXIMUM
30    CONTINUE
      END IF

*
*..... RETURN TO CALLING PROGRAM
*

      RETURN
      END

```

Appendix C

```

SUBROUTINE GRADIENT ( STR4TO3, STP4TO3, INC4TO3,
                     STR7, STP7, INC7, SURF, GRAD )
.....
GRADIENT -      WILL CALCULATE THE GRADIENT AT EVERY POINT
                  ON A SURFACE. THE ORIGINAL DATA MUST BE IN
                  THREE COLUMNS, X, Y, Z, WITH THE X'S CHANGING
                  WHILE Y IS CONSTANT
.....

VARIABLE DECLARATION :
STR4TO3 -      THE STARTING VALUE FOR THRESHOLD OF THE BAND
                4/3 RATIO
STP4TO3 -      THE STOPPING VALUE FOR THRESHOLD OF THE BAND
                4/3 RATIO
INC4TO3 -      THE INCREMENT VALUE FOR THE THRESHOLD OF THE BAND
                4/3 RATIO
STR7 -        THE STARTING VALUE FOR THRESHOLD OF THE BAND
                7 IMAGE
STP7 -        THE STOPPING VALUE FOR THRESHOLD OF THE BAND
                7 IMAGE
INC7 -        THE INCREMENT VALUE FOR THE THRESHOLD OF THE BAND
                7 IMAGE
SURF -        THE ARRAY CONTAINING THE Z-DIMENSION VALUES
                OF THE SURFACE WHICH THE GRADIENT IS TO BE
                TAKEN OF
GRAD -        THE ARRAY CONTAINING THE RESULTING GRADIENT
.....

REQUIRED SUBROUTINES :      NONE
.....

WRITTEN BY CARL SALVAGGIO      CENTER FOR IMAGING SCIENCE
                                ROCHESTER INSTITUTE OF TECHNOLOGY
                                DECEMBER 19, 1986
                                MODIFIED      JANUARY 31, 1987      BY C.S.
.....

INTEGER*2      STR4TO3, STP4TO3, INC4TO3, STR7, STP7, INC7
INTEGER*4      I, J
REAL           SURF(256,256), GRAD(256,256)

.....
CALCULATE THE GRADIENT OF THE SURFACE

DO 20 I = STR4TO3+INC4TO3, STP4TO3, INC4TO3
DO 10 J = STR7+INC7, STP7, INC7
GRAD(I,J) = SQRT(( (SURF(I,J)-SURF(I-INC4TO3,J))
                  / REAL(INC4TO3))**2 +
                  ( (SURF(I,J)-SURF(I,J-INC7))
                  / REAL(INC7))**2 )
10 CONTINUE
20 CONTINUE

.....
RETURN TO CALLING PROGRAM

RETURN
END

```

Appendix C

```

SUBROUTINE INTERVAL ( FOURTOHREE, BAND7, STR4TO3, STP4TO3,
                     INC4TO3, STR7, STP7, INC7 )

```

```

INTERVAL -      THIS ROUTINE WILL DETERMINE THE INTERVAL OVER WHICH
                  THE THRESHOLD REGIONS SHOULD BE SEARCHED.  FOR BOTH
                  THE 4/3 RATIO AND BAND 7 THRESHOLD REGIONS, THE
                  INTERVAL IS DEFINED AS FROM THE MEAN OF THE
                  HISTOGRAM TO 2 TIME THE STANDARD DEVIATION BELOW
                  THE MEAN VALUE.  THIS WAS DETERMINED FROM EMPIRICAL
                  EVALUATION OF MANY DIFFERENT IMAGE HISTOGRAMS.

```

VARIABLE DECLARATION -

```

FOURTOHREE -    THE NAME OF THE FILE CONTAINING THE BAND 4/3
                  RATIO IMAGE
BAND7 -         THE NAME OF THE FILE CONTAINING THE BAND 7 IMAGE
STR4TO3 -       THE STARTING VALUE FOR THRESHOLD OF THE BAND
                  4/3 RATIO
STP4TO3 -       THE STOPPING VALUE FOR THRESHOLD OF THE BAND
                  4/3 RATIO
INC4TO3 -       THE INCREMENT VALUE FOR THE THRESHOLD OF THE BAND
                  4/3 RATIO
STR7 -         THE STARTING VALUE FOR THRESHOLD OF THE BAND
                  7 IMAGE
STP7 -         THE STOPPING VALUE FOR THRESHOLD OF THE BAND
                  7 IMAGE
INC7 -         THE INCREMENT VALUE FOR THE THRESHOLD OF THE BAND
                  7 IMAGE

```

```

REQUIRED SUBROUTINES:      HISTSTATS

```

```

AUTHOR: CARL SALVAGGIO    FEBRUARY 6, 1987
                          ROCHESTER INSTITUTE OF TECHNOLOGY
                          CENTER FOR IMAGING SCIENCE

```

```

CHARACTER*80      FOURTOHREE, BAND7
INTEGER*4         CHANNEL4TO3, CHANNEL7, UNIT
INTEGER*4         HIST4TO3(0:255), HIST7(0:255)
INTEGER*2         STR4TO3, STP4TO3, INC4TO3, STR7, STP7, INC7
REAL*4           MEAN4TO3, MEAN7, SD4TO3, SD7

```

```

DATA CHANNEL4TO3      / 0 /
DATA CHANNEL7         / 1 /

```

```

..... ATTACH THE UNIT TO THE CURRENT PROGRAM

```

```

CALL IPI_ATTUNIT ( UNIT )

```

```

..... PLACE THE 4/3 RATIO IMAGE IN CHANNEL 0
.
      CALL SHOMNO ( UNIT, FOURTOHREE, CHANNEL4TO3 )
.
..... PLACE REFRESHED BAND 7 IMAGE IN CHANNEL 1
.
      CALL SHOMNO ( UNIT, BAND7, CHANNEL7 )
.
..... HISTOGRAM THE BAND 4/3 CHANNEL ( CHANNEL 0 )
.
      CALL HISTOGRAM ( UNIT, CHANNEL4TO3, HIST4TO3 )
.
..... HISTOGRAM THE BAND 7 CHANNEL ( CHANNEL 1 )
.
      CALL HISTOGRAM ( UNIT, CHANNEL7, HIST7 )
.
..... FIND HISTOGRAM STATISTICS
.
      CALL HISTSTATS ( HIST4TO3, MEAN4TO3, SD4TO3 )
      CALL HISTSTATS ( HIST7, MEAN7, SD7 )
.
..... DETERMINE SAMPLING INTERVALS
.
      STR4TO3 = INT( MEAN4TO3 )
      STP4TO3 = INT( MEAN4TO3 - 2.0*SD4TO3 )
      INC4TO3 = -2
      STR7 = INT( MEAN7 - 2.5*SD7 )
      STP7 = INT( MEAN7 + SD7 )
      INC7 = 2
.
..... CHECK FOR OUT OF BOUNDS ERRORS
.
      IF ( STR4TO3 .GT. 255 ) STR4TO3 = 255
      IF ( STP4TO3 .LT. 1 ) STP4TO3 = 1
      IF ( STR7 .LT. 1 ) STR7 = 1
      IF ( STP7 .GT. 255 ) STP7 = 255
.
      IF ( REAL( STR4TO3 - STP4TO3 ) / 2.0 .NE.
        • INT( REAL( STR4TO3 - STP4TO3 ) / 2.0 ) ) STR4TO3=STR4TO3+1
      IF ( REAL( STP7 - STR7 ) / 2.0 .NE.
        • INT( REAL( STP7 - STR7 ) / 2.0 ) ) STP7 = STP7 + 1
.
..... RETURN TO CALLING PROGRAM
.
      RETURN

```

Appendix C

END

```

.....
SUBROUTINE HISTSTATS ( HIST, MEAN, STDDEV )
.....
*
*   HISTSTATS -   WILL CALCULATE THE MEAN AND STANDARD DEVIATION
*                 OF AN IMAGE HISTOGRAM
*
*
*   VARIABLE DECLARATION
*
*   HIST -        THE INTEGER*4 VECTOR DEFINED AS (0:255) WHICH
*                 CONTAINS THE IMAGE HISTOGRAM
*   MEAN -        THE REAL*4 VARIABLE RETURNING THE HISTOGRAM MEAN
*   STDDEV -      THE REAL*4 VARIABLE RETURNING THE HISTOGRAM STANDARD
*                 DEVIATION
*
*
*   REQUIRED SUBROUTINES          NONE
*
*
*   AUTHOR: CARL SALVAGGIO   FEBRUARY 6, 1987
*                        ROCHESTER INSTITUTE OF TECHNOLOGY
*                        CENTER FOR IMAGING SCIENCE
*
*
*
*   INTEGER*4   HIST(0:255)
*   REAL*4      MEAN, STDDEV
*   REAL*4      SUMX, SUMX2
*
*   SUMX = 0.0
*   SUMX2 = 0.0
*   TOTNUM = 0.0
*
*
*   .....
*   DETERMINE HISTOGRAM MEAN AND STANDARD DEVIATION
*
*
*   DO 10 I = 0, 255, 1
*       TOTNUM = TOTNUM + HIST( I )
*       SUMX = SUMX + ( I * HIST( I ) )
*       SUMX2 = SUMX2 + HIST( I ) * I**2
* 10  CONTINUE
*
*   MEAN = SUMX / TOTNUM
*   STDDEV = SQRT ( ( TOTNUM * SUMX2 - SUMX**2 ) /
*                  ( TOTNUM * ( TOTNUM - 1 ) ) )
*
*   D   WRITE (6,*) 'TOTNUM = ',TOTNUM
*   D   WRITE (6,*) 'MEAN = ',MEAN
*   D   WRITE (6,*) 'STDDEV = ',STDDEV
*
*
*   .....
*   RETURN TO CALLING PROGRAM
*
*
*   RETURN
*   END

```

```

SUBROUTINE SCALEDIVIDE ( FILNAMA, FILNAMB, QUOTNAM )
.....
.
.   SCALEDIVIDE -   DIVIDES TWO IMAGES AND FORCES THE QUOTIENT TO
.                   FILL THE FULL DYNAMIC RANGE
.
.....
.
.   VARIABLE DECLARATION
.
.   FILNAMA -       THE FILENAME CONTAIN THE IMAGE FOR THE NUMERATOR
.   FILNAMB -       THE FILENAME CONTAIN THE IMAGE FOR THE DENOMINATOR
.   FILNAMA -       THE FILENAME CONTAIN WHICH SHOULD CONTAIN THE
.                   QUOTIENT
.
.....
.
.   REQUIRED SUBROUTINES           PIXIN
.                                   PIXOUT
.
.....
.
.   AUTHOR:           CARL SALVAGGIO  CENTER FOR IMAGING SCIENCE
.                                   ROCHESTER INSTITUTE OF TECHNOLOGY
.                                   FEBRUARY 7, 1987
.
.....

CHARACTER*80  FILNAMA, FILNAMB, QUOTNAM
INTEGER*2     IMAGEA(512,512), IMAGEB(512,512)
REAL*4        BIGA, BIGB, QUOT, FACTOR, MAXQUOT
INTEGER*2     QUOTIENT(512,512)

.
.....   PLACE THE IMAGES IN CORE MEMORY
.
CALL PIXIN ( IMAGEA, FILNAMA )
CALL PIXIN ( IMAGEB, FILNAMB )

.
.....   FIND THE LARGEST QUOTIENT VALUE
.
MAXQUOT = 0.0
DO 7 J = 1, 512, 1
  DO 5 I = 1, 512, 1
    BIGA = REAL( IMAGEA(I,J) )
    BIGB = REAL( IMAGEB(I,J) )
    QUOT = BIGA / BIGB
    IF ( QUOT .GT. MAXQUOT ) THEN
      MAXQUOT = QUOT
    END IF
  5 CONTINUE
7 CONTINUE

.
.....   DETERMINE SCALING FACTOR TO FILL DYNAMIC RANGE
.

```

```

      FACTOR = 255.0 / MaxQuot
D      WRITE (6,*) 'FACTOR = ', FACTOR
      *
      .....      PERFORM THE DIVISION AND SCALE BY THE SCALING FACTOR
      *
      Do 20 J = 1, 512, 1
      Do 10 I = 1, 512, 1
      BIGA = REAL( IMAGEA(I,J) )
      BIGA = BIGA * FACTOR
      BIGB = REAL( IMAGEB(I,J) )
      QUOTIENT(I,J) = IFIX( BIGA / BIGB )
10      CONTINUE
20      CONTINUE
      *
      .....      WRITE THE QUOTIENT IMAGE OUT TO DISK
      *
      CALL PIXOUT ( QUOTIENT, QUOTNAM )
      *
      .....      RETURN TO CALLING PROGRAM
      *

      RETURN
      END

```

Appendix D

Description of the PIF Normalization Code

**Normalize
HistStats**

NORMALIZE - THIS PROGRAM WILL PERFORM THE PIF NORMALIZATION PROCESS. THE PIF IMAGES WILL BE CALCULATED, THE PIF HISTOGRAMS DETERMINED FOR EACH BAND IMAGE, THE HISTOGRAM STATISTICS COMPUTED, AND THE APPROPRIATE TRANSFORMS DETERMINED. THIS PROGRAM IS DESIGNED FOR USE AFTER DETERMINING THE PIF MASK USING EITHER BLDPIF OR MANPIF.

VARIABLE DECLARATION

ANDCHAN - THE MEMORY PLANE TO DISPLAY THE RESULT OF THE LOGICAL AND OPERATION TO
 CLEARCHAN - THE MEMORY PLANE THAT IS USED IN THE DVP CLEARING PROCESS
 DAY1 - THE NAMES OF THE DAY 1 IMAGE FILES
 DAY2 - THE NAMES OF THE DAY 2 IMAGE FILES
 FILNAM - THE NAME OF THE FILE TO STORE THE PIF TRANSFORMS TO
 HIST - THE PIF HISTOGRAM DATA ARRAY
 HISTNAM - THE NAME OF THE FILE TO STORE THE PIF HISTOGRAMS TO
 IMAGECHAN - THE MEMORY PLANE TO DISPLAY THE IMAGE IN
 INTERCEPT - THE PIF TRANSFORM INTERCEPT ARRAY
 MASK - THE NAME OF THE FILE CONTAINING THE PIF MASK IMAGES
 MASKCHAN - THE MEMORY PLANE TO DISPLAY THE PIF MASK IN
 MEAN - THE PIF HISTOGRAM MEAN ARRAY
 SLOPE - THE PIF TRANSFORM SLOPE ARRAY
 STDDEV - THE PIF HISTOGRAM STANDARD DEVIATION ARRAY

REQUIRED SUBROUTINES: USER:[SLS4255.CIS.IPI] IPI_ATTUNIT
 USER:[CNS9024.DEANZA] SHOMNO
 LOGICAL AND
 HISTOGRAM
 HISTSTATS
 TRIMNO
 MxBITT
 WRTITT
 SAVMNO
 VAX/VMS RTL LIB\$ERASEPAGE
 STR\$UPCASE

AUTHOR: CARL SALVAGGIO CENTER FOR IMAGING SCIENCE
 ROCHESTER INSTITUTE OF TECHNOLOGY
 MARCH 2, 1987

CHARACTER*1 BELL, CONT, ANS
 CHARACTER*80 DAY1(6), DAY2(6), FILNAM, HISTNAM
 CHARACTER*80 MASK(2)
 INTEGER*4 IMAGECHAN, MASKCHAN, ANDCHAN, CLEARCHAN
 INTEGER*4 HIST(0:255)
 LOGICAL*1 EXISTS
 REAL*4 MEAN(2,6), STDDEV(2,6)

Appendix D

```

REAL*4          SLOPE(6), INTERCEPT(6)

.....
      INITIALIZE CONSTANTS

      PARAMETER ( BELL = CHAR(7) )

      DATA IMAGECHAN      / 0 /
      DATA MASKCHAN       / 1 /
      DATA ANDCHAN        / 2 /
      DATA CLEARCHAN      / 3 /
      DATA HISTNAM        / 'IMAGE#DAY#.HST' /

1  FORMAT ('+',A1)                                ! BELL RING

.....
      GET THE NUMBER OF BANDS THAT ARE TO BE NORMALIZED

      CALL LIB$ERASE_PAGE(1,1)
      WRITE (6,*)
      WRITE (6,*) '      --- PIF SCENE NORMALIZATION ---'
      WRITE (6,*)
      WRITE (6,*) 'NOTE: THE DAY 1 IMAGE THAT IS REFERRED TO IN'
      WRITE (6,*) '      THIS PROGRAM IS THE DATA SET THAT YOU WISH'
      WRITE (6,*) '      TO TRANSFORM. THE DAY 2 IMAGE IS THE'
      WRITE (6,*) '      IMAGE YOU WISH THE DAY 1 IMAGE TO LOOK'
      WRITE (6,*) '      LIKE AFTER THE TRANSFORMATION IS PERFORMED.'
      WRITE (6,*)
51  WRITE (6,52)
52  FORMAT ('$', 'ENTER THE NUMBER OF BAND IMAGES YOUR DATA HAS: ')
      READ (5,*,END=900) NUMBANDS
      IF ( NUMBANDS .GT. 6 ) THEN
          WRITE (6,1) BELL
          WRITE (6,*)
          WRITE (6,*) '*** ERROR *** MAXIMUM NUMBER OF BANDS IS 6'
          WRITE (6,*)
          GOTO 51
      END IF
      IF ( NUMBANDS .LT. 1 ) THEN
          WRITE (6,1) BELL
          WRITE (6,*)
          WRITE (6,*) '*** ERROR *** YOU NEED AT LEAST ONE BAND'
          WRITE (6,*)
          GOTO 51
      END IF

.....
      GET NAMES FOR THE DAY 1 IMAGES

      WRITE (6,*)
      DO 79 I = 1, NUMBANDS, 1
2  WRITE (6,5) I
5  FORMAT ('$', 'ENTER FILE NAME FOR DAY 1 IMAGE',I2,' : ')
      READ (5, '(A)',END=900) DAY1( I )
      INQUIRE ( FILE=DAY1( I ), EXIST=EXISTS )
      IF ( EXISTS .EQ. .FALSE. ) THEN
          WRITE (6,*)

```

Appendix D

```

        WRITE (6,1) BELL
        WRITE (6,*) '*** ERROR *** FILE DOES NOT EXIST'
        WRITE (6,*)
        GOTO 2
    END IF
79    CONTINUE

.
.....    GET NAMES FOR THE DAY 2 IMAGES
.

    WRITE (6,*)
    DO 10 I = 1, NUMBANDS, 1
20      WRITE (6,78) I
78      FORMAT ('$', 'ENTER FILE NAME FOR DAY 2 IMAGE', I2, ' : ')
        READ (5, '(A)', END=900) DAY2( I )
        INQUIRE ( FILE=DAY2( I ), EXIST=EXISTS )
        IF ( EXISTS .EQ. .FALSE. ) THEN
            WRITE (6,*)
            WRITE (6,1) BELL
            WRITE (6,*) '*** ERROR *** FILE DOES NOT EXIST'
            WRITE (6,*)
            GOTO 20
        END IF
    10    CONTINUE

.
.....    GET THE MASK NAME FOR EACH DAY
.

    WRITE (6,*)
    DO 68 K = 1, 2, 1
67      WRITE (6,69) K
69      FORMAT ('$', 'ENTER FILE NAME FOR DAY', I2, ' MASK: ')
        READ (5, '(A)', END=900) MASK( K )
        INQUIRE ( FILE=MASK( K ), EXIST=EXISTS )
        IF ( EXISTS .EQ. .FALSE. ) THEN
            WRITE (6,*)
            WRITE (6,1) BELL
            WRITE (6,*) '*** ERROR *** FILE DOES NOT EXIST'
            WRITE (6,*)
            GOTO 67
        END IF
    68    CONTINUE
        WRITE (6,*)
        WRITE (6,42)
42      FORMAT ('$', 'DO YOU WISH TO SAVE THE HISTOGRAM DATA (Y OR N)? ')
        READ (5, '(A)', END=900) ANS
        CALL STR$UPCASE(ANS,ANS)

.
.....    ATTACH THE UNIT TO THE CURRENT PROCESS
.

        CALL IPI_ATTUNIT ( UNIT )

.
.....    FIND THE HISTOGRAM STATISTICS FOR EACH BAND IMAGE
.

```

```

Do 864 J = 1, 2, 1
  CALL SHOMNO ( UNIT, MASK( J ), MASKCHAN )
  Do 863 I = 1, NUMBANDS, 1
    IF ( J .EQ. 1 ) THEN
      CALL SHOMNO ( UNIT, Day1( I ), IMAGECHAN )
    ELSE
      CALL SHOMNO ( UNIT, Day2( I ), IMAGECHAN )
    END IF
    CALL LOGICAL_AND ( UNIT, IMAGECHAN, MASKCHAN, ANDCHAN )
    CALL HISTOGRAM ( UNIT, ANDCHAN, HIST )
    HIST(0) = 0
    CALL HISTSTATS ( HIST, MEAN(J,I), STDDEV(J,I) )
    IF ( ANS .EQ. 'Y' ) THEN
      HISTNAM(6:6) = CHAR( I+48 )
      HISTNAM(10:10) = CHAR( J+48 )
      OPEN ( 1, FILE=HISTNAM, STATUS='NEW' )
      Do 861 M = 0, 255, 1
        WRITE (1,*) M, HIST( M )
      861 CONTINUE
      CLOSE ( 1 )
    END IF
  863 CONTINUE
864 CONTINUE

*
*****      DISPLAY THE HISTOGRAM STATS TO THE USER
*

WRITE (6,*)
WRITE (6,*)
WRITE (6,*) '      --- PIF HISTOGRAM STATISTICS ---'
Do 743 I = 1, 2, 1
  WRITE (6,*)
  741 WRITE (6,741) I
  FORMAT ( '$', ' THE HISTOGRAM STATS FOR Day', I2, ' ARE: ' )
  WRITE (6,*)
  WRITE (6,*) 'IMAGE          MEAN          STANDARD DEVIATION'
  Do 742 J = 1, NUMBANDS, 1
    WRITE (6,739) J, MEAN(I,J), STDDEV(I,J)
  739 FORMAT ( '$', I3, 10X, F6.2, 10X, F6.2 )
  742 CONTINUE
  WRITE (6,*)
  743 CONTINUE
  READ (5, '(A)', END=900) CONT

*
*****      CALCULATE THE LINEAR HISTOGRAM TRANSFORMATIONS
*

Do 895 J = 1, NUMBANDS, 1
  SLOPE(J) = STDDEV(2,J) / STDDEV(1,J)
  INTERCEPT(J) = MEAN(2,J) - SLOPE(J) * MEAN(1,J)
895 CONTINUE

*
*****      DISPLAY THE TRANSFORMS TO THE USER
*

WRITE (6,*)
WRITE (6,*)

```

```

WRITE (6,*)
WRITE (6,*) '    --- PIF LINEAR TRANSFORMS ---'
WRITE (6,*)
WRITE (6,*) 'IMAGE          SLOPE          INTERCEPT'
WRITE (6,*)
DO 836 J= 1, NUMBANDS, 1
835   WRITE (6,835) J, SLOPE(J), INTERCEPT(J)
      FORMAT ('$',I3,10X,F6.2,10X,F7.2 )
836   CONTINUE
      WRITE (6,*)

*
*.....      WRITE TRANSFORMS OUT TO FILE
*
      WRITE (6,837)
837   FORMAT ('$', 'DO YOU WISH TO SAVE TRANSFORMS TO A FILE (Y OR N)? ')
      READ (5, '(A)', END=900) ANS
      CALL STR$UPCASE (ANS, ANS)
      IF ( ANS .EQ. 'Y' ) THEN
          WRITE (6,838)
838   FORMAT ('$', 'ENTER FILENAME TO STORE TO: ')
          READ (5, '(A)', END=900) FILNAM
          OPEN ( 1, FILE=FILNAM, STATUS='NEW' )
          DO 839 J = 1, NUMBANDS, 1
              WRITE (1,*) SLOPE(J), INTERCEPT(J)
839   CONTINUE
          END IF

*
*.....      CLEAR THE CHANNELS
*
      CALL LOGICAL_AND ( UNIT, IMAGECHAN, CLEARCHAN, IMAGECHAN )
      CALL LOGICAL_AND ( UNIT, MASKCHAN, CLEARCHAN, MASKCHAN )
      CALL LOGICAL_AND ( UNIT, ANDCHAN, CLEARCHAN, ANDCHAN )

*
*.....      TRANSFORM THE DAY 1 IMAGERY
*
      WRITE (6,*)
      WRITE (6,*)
      WRITE (6,*)
      WRITE (6,*) '    --- PIF TRANSFORMATIONS ---'
      WRITE (6,*)
      CALL TRIMNO ( UNIT, IMAGECHAN )
      DO 915 I = 1, NUMBANDS, 1
          CALL SHOMNO ( UNIT, DAY1( I ), IMAGECHAN )
          CALL MXBITT ( UNIT, IMAGECHAN, SLOPE(I), INTERCEPT(I) )
          CALL WRITTT ( UNIT, IMAGECHAN, IMAGECHAN )
          WRITE (6,977) I
977   FORMAT ('$', 'SAVE TRANSFORMED IMAGE', I2, ' (Y OR N)? ')
          READ (5, '(A)', END=900) ANS
          CALL STR$UPCASE ( ANS, ANS )
          IF ( ANS .EQ. 'Y' ) THEN
              WRITE (6,979)
979   FORMAT ('$', 'ENTER FILE NAME TO STORE IN: ')
              READ (5, '(A)', END=900) FILNAM
              CALL SAVMNO ( UNIT, FILNAM, IMAGECHAN )

```

Appendix D

```

        WRITE (6,*)
    END IF
915  CONTINUE

*
*.....
*      DETACH UNIT FROM THE CURRENT PROCESS AND TERMINATE
*
900  CALL LOGICAL_AND ( UNIT, IMAGECHAN, CLEARCHAN, IMAGECHAN )
    CALL LOGICAL_AND ( UNIT, MASKCHAN, CLEARCHAN, MASKCHAN )
    CALL LOGICAL_AND ( UNIT, ANDCHAN, CLEARCHAN, ANDCHAN )

    CALL TRIMNO ( UNIT, -1 )
    CALL IPI_DETUNIT ( UNIT )
    WRITE (6,*)
    END

SUBROUTINE HISTSTATS ( HIST, MEAN, STDDEV )
*.....
*
*      HISTSTATS -      WILL CALCULATE THE MEAN AND STANDARD DEVIATION
*                        OF AN IMAGE HISTOGRAM
*.....
*
*      VARIABLE DECLARATION
*
*      HIST -          THE INTEGER*4 VECTOR DEFINED AS (0:255) WHICH
*                        CONTAINS THE IMAGE HISTOGRAM
*      MEAN -          THE REAL*4 VARIABLE RETURNING THE HISTOGRAM MEAN
*      STDDEV -        THE REAL*4 VARIABLE RETURNING THE HISTOGRAM STANDARD
*                        DEVIATION
*.....
*
*      REQUIRED SUBROUTINES          NONE
*.....
*
*      AUTHOR: CARL SALVAGGIO  FEBRUARY 6, 1987
*                        ROCHESTER INSTITUTE OF TECHNOLOGY
*                        CENTER FOR IMAGING SCIENCE
*.....

    INTEGER*4      HIST(0:255)
    REAL*4          MEAN, STDDEV
    REAL*4          SUMX, SUMX2

    SUMX = 0.0
    SUMX2 = 0.0
    TOTNUM = 0.0

*
*.....
*      DETERMINE HISTOGRAM MEAN AND STANDARD DEVIATION
*

```

```

      DO 10 I = 0.255, 1
        TOTNUM = TOTNUM + HIST( I )
        SUMX = SUMX + ( I * HIST( I ) )
        SUMX2 = SUMX2 + HIST( I ) * I**2
10    CONTINUE

      MEAN = SUMX / TOTNUM
      STDDEV = SQRT ( ( TOTNUM * SUMX2 - SUMX**2 ) /
                     ( TOTNUM * ( TOTNUM - 1 ) ) )

D      WRITE (6,*) 'TOTNUM = ',TOTNUM
D      WRITE (6,*) 'MEAN = ',MEAN
D      WRITE (6,*) 'STDDEV = ',STDDEV
      .
      .....      RETURN TO CALLING PROGRAM
      .

      RETURN
      END

```

Appendix E

**Description of the Code Used to Read Data
Off of TM Computer Compatible Tape**

**LT4Read
LandFull
Land512**

```

.....
*
*   LT4READ - WILL LOOK AT A LANDSAT-4 IMAGERY FILE THAT HAS
*   BEEN COPIED FROM MAGTAPE TO DISK, AND EXTRACT
*   EITHER A FULL SCENE OR A FULL RESOLUTION SUBSECTION.
*   THE IMAGE DATA THAT THIS PROGRAM IS
*   INTENDED TO READ IS LOCATED IN 28672 BYTE RECORDS,
*   OF WHICH THERE ARE 1492. THE RECORD CONTAINS FOUR
*   IMAGE LINES, EACH OCCUPYING 7168 BYTES OF THE
*   RECORD. THIS IS THE FORMAT THAT NASA USED BEFORE
*   THEY STANDARDIZED ON THEIR CURRENT TAPE FORMAT.
*
*

```

```

*   WRITTEN BY CARL SALVAGGIO      OCTOBER 28, 1986
*
*
.....

```

```

CHARACTER*1      ANS
INTEGER*2        ICHOICE

CALL LIB$ERASE_PAGE(1,1)
WRITE (5,9)
9  FORMAT ('$', '.....')
WRITE (5,10)
10 FORMAT ('$', '*** LANDSAT 4 (UNCONVENTIONAL) TAPE READ ***')
WRITE (5,11)
11 FORMAT ('$', '.....')
WRITE (5,*)
WRITE (5,*)
WRITE (5,50)
50 FORMAT ('$', 'THIS PROGRAM REQUIRES THE DATA TO BE ARRANGED'//
*          'IN THE FOLLOWING DATA STRUCTURE. THE IMAGERY'//
*          'FILE SHOULD BE ON DISK (COPIED DIRECTLY FROM'//
*          'MAGTAPE). THIS FILE SHOULD CONTAIN 1492 IMAGE'//
*          'RECORDS OF LENGTH 28672 BYTES.')
WRITE (5,*)
WRITE (5,60)
60 FORMAT ('$', 'DO YOU HAVE THIS DATA READY ( Y OR N ) ? ')
READ (5, '(A)', END=900) ANS
IF ( ANS .NE. 'Y' .AND. ANS .NE. 'Y' ) THEN
  WRITE (5,*)
  WRITE (5,70)
70  FORMAT ('$', 'YOU MUST HAVE THIS DATA READY FIRST !!!')
  GoTo 900
END IF

WRITE (5,*)
WRITE (5,20)
20  FORMAT ('$', 'SAMPLING CHOICES:')
WRITE (5,*)
WRITE (5,*) :      (1) SUBSAMPLE A FULL SCENE'
WRITE (5,*) :      (2) EXTRACT 512 X 512 FULL RESOLUTION SCENE'
WRITE (5,*)
30  WRITE (5,40)
40  FORMAT ('$', 'ENTER CHOICE (1 OR 2) : ')
READ (5, '(A)', END=900) ICHOICE
IF ( ICHOICE .LT. 1 .OR. ICHOICE .GT. 2 ) GoTo 30

IF ( ICHOICE .EQ. 1 ) THEN
  CALL LIB$ERASE_PAGE(1,1)
  WRITE (5,100)

```

```

100   FORMAT ('$', '.....')
      WRITE (5,101)
101   FORMAT ('$', '***          SUBSAMPLE A FULL SCENE          ***')
      WRITE (5,102)
102   FORMAT ('$', '.....')
      WRITE (5,*)
      CALL LANDFULL
      ELSEIF ( ICHOICE .EQ. 2 ) THEN
      CALL LIB$ERASE_PAGE(1,1)
      WRITE (5,103)
103   FORMAT ('$', '.....')
      WRITE (5,104)
104   FORMAT ('$', '***          EXTRACT 512 x 512 FULL RES SCENE          ***')
      WRITE (5,105)
105   FORMAT ('$', '.....')
      WRITE (5,*)
      CALL LAND512
      ENDIF

900   WRITE (5,*)
      STOP 'LT4READ COMPLETED.'
      END

```

.....
 SUBROUTINE LANDFULL


```

*
*   LANDFULL - WILL LOOK AT A LANDSAT-4 IMAGERY FILE THAT HAS
*               BEEN COPIED FROM MAGTAPE TO DISK, AND SUBSAMPLE
*               AT AN APPROPRIATE RATE TO OBTAIN A FULL SCENE
*               IMAGE.  THE IMAGE DATA THAT THIS PROGRAM IS
*               INTENDED TO READ IS LOCATED IN 28672 BYTE RECORDS,
*               OF WHICH THERE ARE 1492.  THE RECORD CONTAINS FOUR
*               IMAGE LINES, EACH OCCUPYING 7168 BYTES OF THE
*               RECORD.  THIS IS THE FORMAT THAT NASA USED BEFORE
*               THEY STANDARDIZED ON THEIR CURRENT TAPE FORMAT.
*

```

WRITTEN BY CARL SALVAGGIO OCTOBER 28, 1986

```

*
*
*   BYTE          RECBUF( 28672 )
*   CHARACTER*80  FILNAM, LANDFIL
*   INTEGER*4     NUMIMGREC, RECLIN, NUMREC
*   INTEGER*4     IMGPIXLIN, NUMBYTREC
*   INTEGER*4     STRREC, STPREC, STRPIX, STPPIX
*   INTEGER*4     SCAN, LINE
*   INTEGER*4     RECRD, PIXINC, RECINC, OUTSIZ
*
*   PARAMETER     ( OUTSIZ = 512 )
*   PARAMETER     ( NUMREC = 1492 )
*   PARAMETER     ( NUMPIXLIN = 7168 )
*   PARAMETER     ( NUMRECLIN = 4 )
*   PARAMETER     ( NUMRECIMG = NUMRECLIN*NUMREC )

```

..... GET THE FILESPEC TO STORE THE 512x512 IMAGE TO

```

1      WRITE (5,1)
      FORMAT ('$', 'ENTER LANDSAT IMAGERY FILENAME : ')
      READ (5, '(A)', END=900) LANDFIL
      WRITE (5,2)
2      FORMAT ('$', 'ENTER FILENAME TO STORE IMAGE TO : ')
      READ (5, '(A)', END=900) FILNAM

*
*****      OPEN INPUT AND OUTPUT FILES
*

      INQUIRE (FILE=LANDFIL, RECL=RECLLEN )

D      WRITE (*,*) 'RECLLEN = ', RECLLEN

      OPEN ( 1, FILE=LANDFIL, FORM='UNFORMATTED',
1/2      ORGANIZATION='SEQUENTIAL', ACCESS='DIRECT',
        STATUS='OLD', RECL=RECLLEN/4 )

      OPEN ( 2, FILE=FILNAM, FORM='UNFORMATTED',
1/2      ORGANIZATION='SEQUENTIAL', ACCESS='DIRECT',
        STATUS='NEW', RECL=OUTSIZ/4 )

*
*****      GRAB IMAGE SUBSECTION AND STORE TO DISK
*

      PIXINC = INT( REAL( NUMPIXLIN ) / REAL( OUTSIZ ) )
      RECINC = INT( REAL( NUMRECIMG ) / REAL( OUTSIZ ) )

      STRPIX = 1
      STPPIX = ( OUTSIZ - 1 ) * PIXINC + STRPIX

      STRREC = ( 2 * NUMRECLIN ) + 1          !! SECOND LINE, FIRST SCAN
      STPREC = NUMRECIMG

      WRITE (5,*)
      WRITE (5,3) NUMRECIMG, NUMPIXLIN
3      FORMAT ('$', 'IMAGE SIZE : ', I5, ' x ', I5)
      WRITE (5,4) RECINC, PIXINC
4      FORMAT ('$', 'SUBSAMPLING RATE : ', I3, ' x ', I3)
      WRITE (5,*)

      RECRD = 1
      DO 10 N = STRREC, STPREC, RECINC
        LINE = N / 4
        SCAN = N - LINE*4
        READ (1, REC=LINE) RECBUF

        JOFFSET = SCAN * NUMPIXLIN
        WRITE (2, REC=RECRD) ( RECBUF( I ), I = STRPIX+JOFFSET,
+          STPPIX+JOFFSET, PIXINC )

        RECRD = RECRD + 1
      IF ( RECRD .GT. OUTSIZ ) GOTO 900
10     CONTINUE

*
*****      RETURN TO CALLING PROGRAM

```

```

900      CLOSE( 1 )
          CLOSE( 2 )

```

RETURN
END

SUBROUTINE LAND512

LANDFULL - WILL LOOK AT A LANDSAT-4 IMAGERY FILE THAT HAS BEEN COPIED FROM MAGTAPE TO DISK, AND SAMPLE A SPECIFIED 512 X 512 SUBSECTION TO CREATE A FULL RESOLUTION IMAGE. THE IMAGE DATA THAT THIS PROGRAM IS INTENDED TO READ IS LOCATED IN 28672 BYTE RECORDS, OF WHICH THERE ARE 1492. THE RECORD CONTAINS FOUR IMAGE LINES, EACH OCCUPYING 7168 BYTES OF THE RECORD. THIS IS THE FORMAT THAT NASA USED BEFORE THEY STANDARDIZED ON THEIR CURRENT TAPE FORMAT.

WRITTEN BY CARL SALVAGGIO OCTOBER 28, 1986

```

BYTE          RECBUF( 28672 )
CHARACTER*80  FILNAM, LANDFIL
CHARACTER*1   BELL
INTEGER*4     NUMIMGREC, RECLEN, NUMREC
INTEGER*4     IMGPIXLIN, NUMBYTREC
INTEGER*4     STRREC, STRPXC, STRPIX, STPPIX
INTEGER*4     SCAN, LINE
INTEGER*4     RECRD, PIXINC, RECINC, OUTSIZ
INTEGER*4     XCOORD, YCOORD

PARAMETER     ( BELL = CHAR( 7 ) )
PARAMETER     ( OUTSIZ = 512 )
PARAMETER     ( NUMREC = 1492 )
PARAMETER     ( NUMPIXLIN = 7168 )
PARAMETER     ( NUMRECLIN = 4 )
PARAMETER     ( NUMRECLIMG = NUMRECLIN*NUMREC )

```

GET THE FILESPEC TO STORE THE 512x512 IMAGE TO

```

1  WRITE (5,1)
   FORMAT ('$','ENTER LANDSAT IMAGERY FILENAME : ')
   READ (5, '(A)', END=900) LANDFIL
   WRITE (5,2)
2  FORMAT ('$','ENTER FILENAME TO STORE IMAGE TO : ')
   READ (5, '(A)', END=900) FILNAM

```

OPEN INPUT AND OUTPUT FILES

INQUIRE (FILE=LANDFIL, RECL=RECLN)

```

D      WRITE (",") 'RECL = ',RECL
      OPEN ( 1, FILE=LANDFIL, FORM='UNFORMATTED',
1        ORGANIZATION='SEQUENTIAL',ACCESS='DIRECT',
2        STATUS='OLD', RECL=RECL/4 )

      OPEN ( 2, FILE=FilNAM, FORM='UNFORMATTED',
1        ORGANIZATION='SEQUENTIAL',ACCESS='DIRECT',
2        STATUS='NEW', RECL=OUTSIZ/4 )

      *
      *****      DISPLAY THE IMAGE SIZE
      *

      WRITE (5,*)
      WRITE (5,3) NUMRECI MG, NUMPIXLIN
3      FORMAT ('$', 'IMAGE SIZE      : ',I5,' x ',I5)

      *
      *****      OBTAIN COORDINATES FOR THE IMAGE TO BE EXTRACTED
      *

      6      WRITE (5,*)
      WRITE (5,4)
      4      FORMAT ('$', 'ENTER COORDINATES OF UPPER LEFT CORNER : ')
      READ (5,*,END=900) XCOORD, YCOORD

      *
      *****      CHECK IF COORDINATES ARE IN IMAGE BOUNDS
      *

      IF ( ((XCOORD+OUTSIZ-1 .GT. NUMRECI MG) .OR.
      *      (XCOORD .LT. 0)) .OR. ((YCOORD+OUTSIZ-1 .GT.
      *      NUMPIXLIN) .OR. (YCOORD .LT. 0)) ) THEN
      WRITE (5,5)
      WRITE (5,*) BELL
      5      FORMAT ('$', '*** ERROR *** IMAGE COORDINATE OUT-OF-BOUNDS')
      GOTO 6
      END IF

      *
      *****      GRAB IMAGE SUBSECTION AND STORE TO DISK
      *

      PIXINC = 1
      RECINC = 1

      STRPIX = YCOORD
      STPIX = STRPIX + OUTSIZ - 1

      STRREC = ( 2 * NUMRECLIN ) + XCOORD + 1
      STPREC = STRREC + OUTSIZ - 1

      RECRD = 1
      DO 10 N = STRREC, STPREC, RECINC
        LINE = N / 4
        SCAN = N - LINE*4
        READ (1,REC=LINE) RECBUF

```

Appendix E

```
      JOFFSET = SCAN * NUMPIXLIN
      +      WRITE (2,REC=RECRD) ( RECBUF( I ), I = STRPIX+JOFFSET,
                                   STPPIX+JOFFSET, PIXINC )

      RECRD = RECRD + 1
10      IF ( RECRD .GT. OUTSIZ ) GOTO 900
      CONTINUE

*
*.....      RETURN TO CALLING PROGRAM
*

900      CLOSE( 1 )
      CLOSE( 2 )

      RETURN
      END
```

Appendix F

Summary of Digital Count Data and Reflectance Conversion Data Used in the Control Point Analysis

Table F-1

Summary of the digital count values associated with the control points for the urban Rochester imagery

Control Point No.	TM Spectral Channel		
	1	2	3
	'82 '84 T('82) _A T('82) _I	'82 '84 T('82) _A T('82) _I	'82 '84 T('82) _A T('82) _I
1	109 144 130 132	48 69 65 66	54 84 81 84
2	116 149 144 147	49 70 67 68	51 82 75 77
3	103 112 117 119	40 45 47 48	41 47 54 55
4	112 139 136 139	47 64 62 64	50 75 73 75
5	117 164 146 150	52 78 73 75	58 94 90 92
6	96 107 102 103	38 45 43 44	38 48 48 49
7	121 167 155 158	52 79 73 75	58 99 90 92
8	124 162 161 165	55 81 80 82	62 101 98 101

	4	5	7
	'82 '84 T('82) _A T('82) _I	'82 '84 T('82) _A T('82) _I	'82 '84 T('82) _A T('82) _I
1	55 73 90 84	84 122 137 133	50 77 86 86
2	50 71 80 74	76 114 124 120	45 70 77 77
3	34 43 48 42	33 48 56 50	20 36 33 33
4	51 67 82 76	82 123 134 130	46 82 79 79
5	55 81 90 84	86 127 140 136	51 79 87 88
6	34 53 48 42	36 71 61 55	21 43 35 35
7	55 88 90 84	88 131 143 140	47 81 80 81
8	59 87 98 91	95 136 154 151	54 79 93 93

Table F-2

Summary of the digital count values associated with the control points for the urban Buffalo imagery

Control Point No.	1				2*				3			
	'82	'84	T('82) _A	T('82) _I	'82	'84	T('82) _A	T('82) _I	'82	'84	T('82) _A	T('82) _I
1	89	107	114	114	-	-	-	-	31	41	43	44
2	106	138	144	144	-	-	-	-	49	71	74	73
3	88	104	112	112	-	-	-	-	29	38	40	41
4	107	155	146	146	-	-	-	-	50	82	75	75
5	108	153	148	148	-	-	-	-	49	77	74	73
6	100	123	133	134	-	-	-	-	42	57	62	62
7	79	98	96	96	-	-	-	-	20	25	24	26
	4				5				7			
	'82	'84	T('82) _A	T('82) _I	'82	'84	T('82) _A	T('82) _I	'82	'84	T('82) _A	T('82) _I
1	26	27	39	40	33	46	51	54	21	30	33	34
2	46	68	73	73	57	86	85	85	31	48	48	48
3	23	33	34	35	29	41	46	49	18	24	28	30
4	51	79	81	81	68	96	100	99	38	54	58	57
5	47	69	74	75	68	96	100	99	39	58	60	59
6	37	48	57	58	57	72	85	85	35	46	54	53
7	11	20	14	16	7	15	15	21	4	9	7	11

* the band 2 images were unable to be read from the magnetic tape

Table F-3

Summary of digital count values associated with the control points
for high resolution airphoto Buffalo imagery

Control Point No.	IR			CIR Spectral Region						G		
	'70	'72	$T('72)_A$	$T('72)_I$	'70	'72	$T('72)_A$	$T('72)_I$	'70	'72	$T('72)_A$	$T('72)_I$
1	95	52	85	99	63	39	57	70	71	46	65	73
2	126	84	133	161	130	87	132	148	143	86	147	156
3	255	158	242	255	214	138	211	231	188	105	186	195
4	253	144	221	255	200	126	192	212	177	96	168	176
5	255	147	226	255	202	131	200	220	180	102	180	189
6	207	119	184	228	165	104	158	176	157	88	151	160
7	221	123	190	235	180	115	175	194	169	97	170	178
8	249	133	205	254	159	108	164	182	161	93	162	170

$T('70)_A$ and $T('70)_I$ represent the transformed Day 1 digital count values utilizing the automated and interactive segmentation algorithms, respectively

Table F-4

Summary of the estimated reflectance data for the 1984 urban Rochester imagery

	TM Spectral Channel							
	1 (Blue)		2 (Green)		3 (Red)		4 (Near IR)	
	%R	DC	%R	DC	%R	DC	%R	DC
	28	164	28	79	30	105	30	127
	25	159	27	82	28	94	28	85
	16	147	17	65	18	75	18	67
	15	126	16	58	17	65	14	60
	12	127	13	55	14	64	8	39
	8	110	8	45	8	47	1	9
	3	83	5	33	3	25		
2	78	4	29	2	29			
$\alpha_k =$	3.3		2.1		2.9		3.4	
$\beta_k =$	79.1		24.9		19.0		8.1	
$r^2 =$	0.93		0.97		0.98		0.92	

Table F-5

Summary of the estimated reflectance data for the 1984 urban Buffalo imagery

		TM Spectral Channel					
1 (Blue)		2 (Green)*		3 (Red)		4 (Near IR)	
%R	DC	%R	DC	%R	DC	%R	DC
36	174	-	-	37	91	60	108
30	151	-	-	31	75	38	82
13	110	-	-	14	42	32	69
10	100	-	-	11	35	16	36
6	91	-	-	10	32	11	32
4	98	-	-	2	24	1	19
$\alpha_k =$		2.5		2.0		1.6	
$\beta_k =$		79.3		15.3		15.7	
$r^2 =$		0.98		0.99		0.99	

* the band 2 images were unable to be read from the magnetic tape

Table F-6

Summary of the estimated reflectance data for the 1972 high resolution airphoto imagery

CIR Spectral Region

IR		R		G	
%R	DC	%R	DC	%R	DC
1	91	2	46	4	70
13	102	14	82	8	108
13	116	14	102	13	120
17	119	18	119	18	137
34	158	34	152	30	159
50	211	44	173	35	166

 $\alpha_k =$

2.5

2.9

2.7

 $\beta_k =$

79.7

50.4

77.3

 $r^2 =$

0.97

0.94

0.90

Appendix G

Summary of the Results Obtained Utilizing the Multivariate Segmentation Algorithm

This appendix is a summary of the work done in the development of the multivariate segmentation algorithm. As was stated in Section 2.2 this approach was dropped after little investigation in favor of the rate of change segmentation algorithm that was the major focus of this study. The results presented here are for the urban Rochester TM scenes. After studying these two image data sets, it was decided that further pursuit of this means of segmentation would prove futile.

Table G-1 is a summary of the principal components computed using TM bands 1,2,3,4,5 and 7. As can be seen from this data, as was predicted by other investigators, the first three principal components explained about 97% of the variability in the six band image data. Figure G-1 shows these principal component images for the 1984 Rochester data. These images confirm the interpretation of these principal components as explained by Crist and Kauth where the first principal component represents overall image brightness, the second component represents greenness (vegetation cover areas are highlighted) and the third principal component represents wetness (wet areas are highlighted).

These first three principal component images were then used as input to an unsupervised multivariate clustering routine using the k-means algorithm. The results of 50 iterations through this clustering algorithm yielded the cluster means described in Table G-2.

Table G-1

Summary of the principal components data computed for the
1982 and 1984 urban Rochester reflective TM data

	Eigenvalues *					
	1	2	3	4	5	6
1982 Image Data	320.72 (57.05)	200.90 (35.74)	30.24 (5.38)	7.10 (1.26)	2.26 (0.40)	0.87 (0.16)
1984 Image Data	1199.20 (58.25)	675.23 (32.80)	146.52 (7.11)	20.18 (0.98)	15.71 (0.76)	1.73 (0.08)

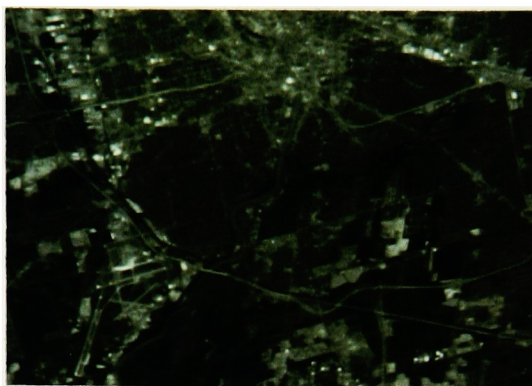
Associated 1982 Eigenvectors

	1	2	3	4	5	6
Band 1	0.140	0.372	0.628	0.319	-0.582	0.086
Band 2	0.113	0.185	0.293	0.046	0.360	-0.858
Band 3	0.171	0.328	0.355	-0.026	0.694	0.504
Band 4	0.349	-0.764	0.486	-0.240	-0.006	0.032
Band 5	0.823	-0.002	-0.386	0.417	0.009	0.002
Band 7	0.374	0.370	-0.090	-0.815	-0.223	-0.039

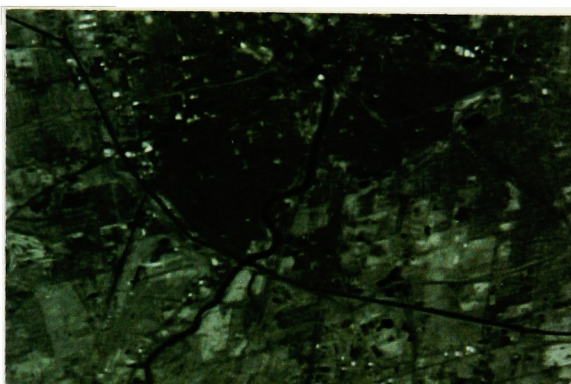
Associated 1984 Eigenvectors

	1	2	3	4	5	6
Band 1	0.466	-0.005	0.528	0.293	0.635	-0.122
Band 2	0.249	-0.052	0.305	-0.014	-0.262	0.879
Band 3	0.432	-0.044	0.379	-0.116	-0.667	-0.458
Band 4	-0.518	-0.681	0.481	-0.176	0.047	-0.049
Band 5	0.306	-0.677	-0.437	0.498	-0.095	0.004
Band 7	0.414	-0.269	-0.248	-0.788	0.269	0.020

* Values in parentheses represent the percent of the total variability explained by each of the individual eigenvalues / eigenvectors



First Principal Component Image



Second Principal Component Image



Third Principal Component Image

Figure G-1 The first three principal component images derived from the six reflective Landsat TM bands of the 1984 urban Rochester data set

Table G-2

Summary of the cluster means determined from the unsupervised multivariate classifier run on the first three principal component images of the 1982 and 1984 urban Rochester TM data

Cluster Means For 1982 Image *

	PC 1	PC 2	PC 3
Cluster 1	3.2	4.7	42.5
Cluster 2	5.4	12.0	42.4
Cluster 3	6.1	17.7	46.1
Cluster 4	3.7	5.4	40.7
Cluster 5	6.3	14.6	43.8
Cluster 6	4.6	3.8	43.2
Cluster 7	5.1	10.5	42.0
Cluster 8	8.7	8.4	41.2

Cluster Means For 1984 Image *

	PC 1	PC 2	PC 3
Cluster 1	64.4	74.1	59.4
Cluster 2	42.8	91.0	57.0
Cluster 3	81.0	102.0	58.8
Cluster 4	20.5	113.4	61.4
Cluster 5	111.7	139.8	62.3
Cluster 6	50.9	114.4	54.0

* Above cluster means based on 50 samples

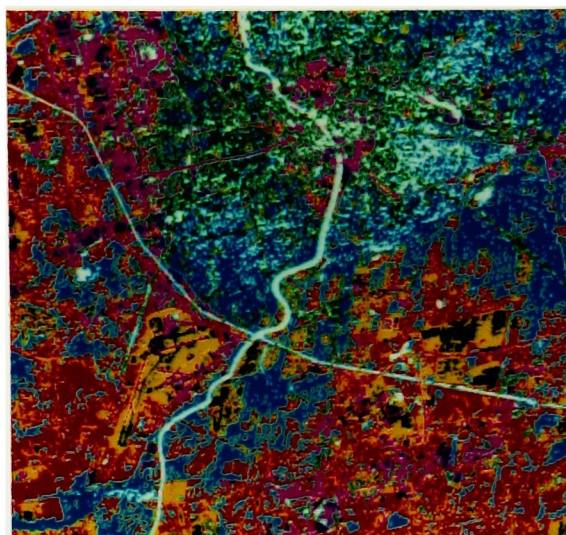
Cluster Image Key

Cluster 1 - Red	Cluster 5 - Magenta
Cluster 2 - Green	Cluster 6 - Cyan
Cluster 3 - Blue	Cluster 7 - White
Cluster 4 - Yellow	Cluster 8 - Black

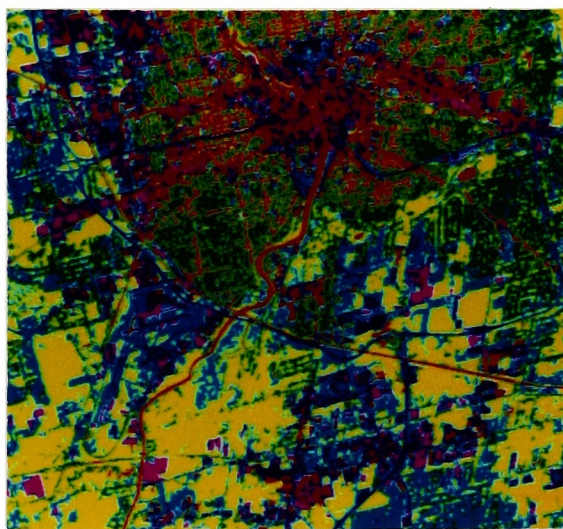
Color composite images were made using the cluster means described in Table G-2 and appear in Figure G-2. The colors in this image correspond to the color key defined in Table G-2. As can be seen from these images, there is no individual spectral class that can be classified as urban features. This can be expected since the spectral signatures of all urban features are certainly not alike.

Several of the spectral classes could be identified as urban features but these also contained pixels which were certainly not pseudo invariant features. These non-PIF pixels were included in the spectral classes since their spectral signatures were close enough to those of the respective urban features in principal component space.

At this point it was realized that the following problems existed. First, no individual spectral class resulting from the unsupervised multivariate classifier could be identified as urban features. Second, several spectral classes were found to contain urban features but these same classes also contained non-PIF pixels. Third, upon individual runs of the clustering algorithm, the same spectral classes were not obtained each time (due to the initial choice of cluster means) which resulted in a method that may produce different results each time. For these reasons it was decided to abandon this multivariate segmentation method in favor of the rate of change algorithm.



1982 Color Composite Image



1984 Color Composite Image

Figure G-2 Color composite images of the spectral cluster formed by the unsupervised multivariate clustering algorithm on the first three principal component images of the 1982 and 1984 urban Rochester TM scenes

Appendix H

Description of the Array Processor Based Image Utility Subroutines Used in the Major Programs

Binary
Histogram
Logical_AND
MxbITT
PixIn
PixOut
ShoMno
SavMno
TriMno
Threshold
WrtITT

```

SUBROUTINE BINARY ( UNIT, CHANNEL )
.....
*
*   BINARY -      THIS PROGRAM WILL TAKE AN IMAGE AND SET IT TO
*                 255 IF A PIXEL IS ON AND 0 IF A PIXEL IS 0
*
*.....
*
*   VARIABLE DECLARATION:
*
*   CHANNEL -     THE MEMORY CHANNEL THAT THE USER WISHES TO
*                 BINARIZE. THIS VARIABLE SHOULD BE DECLARED
*                 AS INTEGER*2 IN THE CALLING PROGRAM.
*
*.....
*
*   REQUIRED SUBROUTINES : [SLS4255.Cis.IPI]      PUTITT
*                                           ENABLEITT
*
*.....
*
*   AUTHOR:       CARL SALVAGGIO  CENTER FOR IMAGING SCIENCE
*                               ROCHESTER INSTITUTE OF TECHNOLOGY
*                               FEBRUARY 6, 1987
*
*.....

INTEGER*2      CHANNEL, MEMVOC, DVPMEM
INTEGER*2      ITT(0:255)
INTEGER*4      UNIT

DATA MEMVOC / 1 /           ! MEMORY VOC NUMBER
DATA DVPMEM / 1 /           ! DVP ALLOCATION NUMBER

*
*.....      ERROR CHECKING
*
IF ( CHANNEL .LT. 0 .OR. CHANNEL .GT. 2 ) THEN
  WRITE (6,*)
  WRITE (6,*) '*** ERROR IN THRESHOLD *** ILLEGAL CHANNEL NUMBER'
  WRITE (6,*)
  RETURN
END IF

*
*.....      DEFINE THE ITT
*
DO 10 I = 0, 255, 1
  IF ( I .EQ. 0 ) THEN
    ITT( I ) = 0
  ELSE
    ITT( I ) = 255
  END IF
10 CONTINUE

*
*.....      WRITE TO THE ITT AND ENABLE IT
*
CALL IPI_PUTITT ( UNIT, CHANNEL, ITT )
CALL IPI_ENABLEITT ( UNIT, CHANNEL, MEMVOC, DVPMEM )

*
*.....      RETURN TO CALLING PROGRAM
*
RETURN
END

```

Appendix H

```

SUBROUTINE HISTOGRAM ( UNIT, CHANNEL, HISTARRAY )
.....
*
*   HISTOGRAM -   THIS PROGRAM WILL COMPUTE THE HISTOGRAM OF A
*                 SPECIFIED CHANNEL AND RETURN THE VALUES OF THE
*                 HISTOGRAM IN AN ARRAY.
*
*.....
*
*   VARIABLE DECLARATION:
*
*   CHANNEL -     THIS IS THE CHANNEL NUMBER THAT THE USER
*                 WISHES TO TAKE THE HISTOGRAM OF. THIS VARIABLE
*                 SHOULD BE DECLARED AS INTEGER*2 IN THE CALLING
*                 PROGRAM.
*
*   HISTARRAY -   THIS IS THE ARRAY THAT THE HISTOGRAM VALUES ARE
*                 RETURNED IN. THIS VARIABLE SHOULD BE DECLARED
*                 AS FOLLOWS IN THE CALLING PROGRAM:
*                 INTEGER*4 HISTARRAY(0:255)
*
*.....
*
*   REQUIRED SUBROUTINES:  [SLS4255.Cis.IPI]      IPI_ATTDPV
*                                                    IPI_DETDPV
*                                                    IPI_CLEARHST
*                                                    IPI_CALCHST
*                                                    IPI_GETHST
*                                                    IPI_SETSIZE
*
*.....
*
*   AUTHOR:        CARL SALVAGGIO  CENTER FOR IMAGING SCIENCE
*                               ROCHESTER INSTITUTE OF TECHNOLOGY
*                               JANUARY 16, 1987
*
*.....

INTEGER*4      UNIT, STATUS, CHANNEL, HstOps
INTEGER*4      HISTARRAY( 0:255 )

DATA HstOps / 0 /

*
*.....      ERROR CHECKING
*
      IF ( CHANNEL .LT. 0 .OR. CHANNEL .GT. 2 ) THEN
        WRITE (6,*)
        WRITE (6,*) '*** ERROR IN HISTOGRAM *** ILLEGAL CHANNEL NUMBER'
        WRITE (6,*)
        RETURN
      END IF

*
*.....      GET THE HISTOGRAM
*
      CALL IPI_ATTDPV ( UNIT, 1 )           ! ATTACH THE DVP
      CALL IPI_SETSIZE ( UNIT, 512 )       ! SET IMAGE SIZE
      CALL IPI_CLEARHST ( UNIT )           ! CLEAR THE HISTOGRAM

      CALL IPI_CALCHST ( UNIT, CHANNEL, HstOps ) ! CALCULATE HISTOGRAM
      CALL IPI_GETHST ( UNIT, HISTARRAY )      ! GET HISTOGRAM FROM DVP
      CALL IPI_DETDPV ( UNIT )                ! DETACH THE DVP

*
*.....      RETURN TO THE CALLING PROGRAM
*
      RETURN
      END

```

```

SUBROUTINE LOGICAL_AND ( UNIT, INCHANA, INCHANB, OUTCHAN )
.....
*
*   LOGICAL_AND - WILL TAKE THE IMAGES IN TWO SEPARATE CHANNELS AND
*                 PERFORM A LOGICAL .AND. ON THEM AND PLACE THE
*                 RESULT IN A THIRD CHANNEL.
*
*.....
*
*   VARIABLE DECLARATION
*
*   ALL THE FOLLOWING SHOULD BE DECLARED INTEGER*2
*
*   INCHANA - THE FIRST OF THE INPUT IMAGE CHANNELS
*   INCHANB - THE SECOND OF THE INPUT IMAGE CHANNELS
*   OUTCHAN - THE IMAGE CHANNEL FOR THE OUTPUT RESULT IMAGE
*
*.....
*
*   REQUIRED SUBROUTINES:  [SLS4255.C is IPI]      IPI_ATTDP
*                                                           IPI_DETDP
*                                                           IPI_SETSIZE
*                                                           IPI_DVPMATH
*                                                           IPI_CONSTANTS
*
*.....
*
*   AUTHOR:          CARL SALVAGGIO  CENTER FOR IMAGING SCIENCE
*                                                           ROCHESTER INSTITUTE OF TECHNOLOGY
*                                                           JANUARY 23, 1987
*
*.....
*
*   INTEGER*2      INCHANA, INCHANB, OUTCHAN
*   INTEGER*4      UNIT
*
*
*.....
*   ASSIGN THE OPCode VALUES
*
*
*   EXTERNAL      IPI_AND
*
*
*.....
*   ERROR CHECKING
*
*
*   IF ( INCHANA .LT. 0 .OR. INCHANA .GT. 3 ) THEN
*     WRITE (6,*)
*     WRITE (6,*) '*** ERROR IN LOGICAL_AND *** ILLEGAL INCHANA'
*     WRITE (6,*)
*     RETURN
*   END IF
*
*   IF ( INCHANB .LT. 0 .OR. INCHANB .GT. 3 ) THEN
*     WRITE (6,*)
*     WRITE (6,*) '*** ERROR IN LOGICAL_AND *** ILLEGAL INCHANB'
*     WRITE (6,*)
*     RETURN
*   END IF

```

```

IF ( OUTCHAN .LT. 0 .OR. OUTCHAN .GT. 3 ) THEN
  WRITE (6,*)
  WRITE (6,*) '*** ERROR IN LOGICAL AND *** ILLEGAL OUTCHAN'
  WRITE (6,*)
  RETURN
END IF

*
*-----*      PERFORM LOGICAL .AND.
*

CALL IPI_ATTDP ( UNIT, 1 )           ! ATTACH THE DVP
CALL IPI_SETSIZE ( UNIT, 512 )       ! SET IMAGE SIZE
CALL IPI_DVPMATH ( UNIT, IPI_AND,    ! PERFORM .AND.
  *      INCHANA, INCHANB, OUTCHAN )
CALL IPI_DETDP ( UNIT )             ! DETACH THE DVP

*
*-----*      RETURN TO CALLING PROGRAM
*

RETURN
END

```

```

SUBROUTINE MxBITT ( UNIT, CHANNEL, M, B )
.....
*
*   MxBITT -      THIS PROGRAM WILL COMPUTE A LINEAR ITT AND
*                  APPLY IT TO THE IMAGE IN THE SELECTED CHANNEL.
*                  THE ITT WILL BE OF THE FORM  $Y = MX + B$ .
*
*.....
*
*   VARIABLE DECLARATION:
*
*   CHANNEL -      THE MEMORY CHANNEL THAT THE USER WISHES TO
*                  THRESHOLD IN. THIS VARIABLE SHOULD BE DECLARED
*                  AS INTEGER*2 IN THE CALLING PROGRAM.
*   M -           THE SLOPE OF THE DESIRED ITT. THIS VARIABLE
*                  SHOULD BE DECLARED AS REAL*4 IN THE CALLING
*                  PROGRAM.
*   B -           THE INTERCEPT OF THE DESIRED ITT. THIS VARIABLE
*                  SHOULD BE DECLARED AS REAL*4 IN THE CALLING
*                  PROGRAM.
*
*.....
*
*   REQUIRED SUBROUTINES : [SLS4255.Cis.Ipi]      PUTITT
*                                     ENABLEITT
*
*.....
*
*   AUTHOR:        CARL SALVAGGIO  CENTER FOR IMAGING SCIENCE
*                                     ROCHESTER INSTITUTE OF TECHNOLOGY
*                                     JANUARY 16, 1987
*
*.....

INTEGER*2      CHANNEL, MEMVOC, DVPMEM
INTEGER*2      ITT(0:255)
INTEGER*2      UNIT
INTEGER*4      M, B, VALUE

DATA MEMVOC / 1 /
DATA DVPMEM / 1 /

*
*.....      ERROR CHECKING
*
      IF ( CHANNEL .LT. 0 .OR. CHANNEL .GT. 2 ) THEN
        WRITE (6,*)
        WRITE (6,*) '*** ERROR IN MxBITT *** ILLEGAL CHANNEL NUMBER'
        WRITE (6,*)
        RETURN
      END IF

*
*.....      DEFINE THE ITT
*
      DO 10 I = 0, 255, 1
        VALUE = M * REAL(I) + B
        IF ( VALUE .GT. 255 ) THEN

```

```

      ITT( I ) = 255
      ELSE IF ( VALUE .LT. 0 ) THEN
        ITT( I ) = 0
      ELSE
        ITT( I ) = IFIX( VALUE )
      END IF
10  CONTINUE

*
*-----  WRITE TO THE ITT AND ENABLE IT
*

      CALL IPI_PUTITT ( UNIT, CHANNEL, ITT )
      CALL IPI_ENABLEITT ( UNIT, CHANNEL, MEMVOC, DVPMEM )

*
*-----  RETURN TO CALLING PROGRAM
*

      RETURN
      END

```

```

SUBROUTINE PixIn( IMAGE, FILNAM )
.....
*
*   THIS SUBROUTINE ACCESSES AN IMAGE FILE THAT EXISTS IN UNFORMATTED,
*   LOGICAL*1 STORAGE AND CONVERTS THE IMAGE DATA INTO INTEGER*2 DATA,
*   STORED IN THE 512 x 512 ARRAY IMAGE(I,J)
*
*   WRITTEN BY CARL SALVAGGIO      10/3/86
*   MODIFIED FROM EXISTING CODE BY VOLCHOK, BIEGEL, SCHIMMINGER
*                                   AND GORZYNSKI
*
.....

      INTEGER      ROW, COLUMN
      PARAMETER    ( ROW = 512, COLUMN = 512 )

      INTEGER*2    IMAGE( ROW, COLUMN )
      LOGICAL*1    LOGIC( ROW )
      CHARACTER*80  FILNAM
      CHARACTER*1   ANS

      OPEN( 4, FILE=FILNAM, ACCESS='DIRECT', STATUS='OLD',
x      RECL=ROW/4, FORM='UNFORMATTED' )

      DO 120 I=1,ROW
        READ (4,REC=I) ( LOGIC(N), N = 1,COLUMN )
        DO 120 J = 1,COLUMN
          IMAGE(I,J) = LOGIC( J )
          IF( IMAGE(I,J) .LT. 0 ) IMAGE(I,J)=IMAGE(I,J)+256
120      CONTINUE

      RETURN
      END

```

```

.....
SUBROUTINE PixOut( IMAGE, FILNAM )
.....
*
*   THIS SUBROUTINE WRITES AN IMAGE FILE BY CONVERTING INTEGER*2
*   TO BYTE DATA AND STORING TO AN EXTERNAL BINARY FILE WITH 512
*   RECORDS, EACH OF LENGTH 128 LONGWORDS
*
*   REQUIRED SUBROUTINE :  MACRO-32 SUBROUTINE  MOVBYT
*
*   WRITTEN BY CARL SALVAGGIO      10/6/86
*   MODIFIED FROM EXISTING CODE BY VOLCHOK, BIEGEL, SCHIMMINGER
*   AND GORZYNSKI
*
.....

      INTEGER*4      ROW, COLUMN
      PARAMETER      ( ROW = 512, COLUMN = 512 )

      BYTE            LOGIC( COLUMN )
      INTEGER*2       IMAGE( ROW, COLUMN )
      CHARACTER*80     FILNAM

      OPEN( 4, FILE=FILNAM, ACCESS='DIRECT', STATUS='NEW'
X          RECL=ROW/4, FORM='UNFORMATTED',
X          ORGANIZATION='SEQUENTIAL' )

      DO 130 I = 1, ROW
        DO 120 J = 1, COLUMN
          CALL MOVBYT( IMAGE( I,J ), LOGIC( J ) )
120        CONTINUE
        WRITE(4,REC=I) ( LOGIC(N), N = 1, COLUMN )
130        CONTINUE

      CLOSE( 4 )

      RETURN
      END

```

```

.....
SUBROUTINE SHOMNO ( UNIT, NAME, CHANNEL )
.....
*
*   SHOMNO -      WILL DISPLAY A 512x512 MONOCHROME IMAGE IN THE
*                  SPECIFIED CHANNEL.
*
.....
*
*   VARIABLE DECLARATION:
*
*   NAME -        THE FILENAME CONTAINING THE 512x512 MONOCHROME
*                  IMAGE. THIS SHOULD BE DECLARED AS CHARACTER*80
*                  IN THE CALLING PROGRAM.
*   CHANNEL -     THE CHANNEL THAT THE IMAGE SHOULD BE PLACED IN.
*                  THIS SHOULD BE DECLARED AS INTEGER*2 IN THE
*                  CALLING PROGRAM.
*
.....
*
*   REQUIRED SUBROUTINES:  [SLS4255.Cis.IPI]      IPI_OPENFILE
*                                                           IPI_DISKPIC
*                                                           IPI_PUTPIC
*                                                           IPI_CLOSEFILE
*
.....
*
*   AUTHOR:        CARL SALVAGGIO  CENTER FOR IMAGING SCIENCE
*                               ROCHESTER INSTITUTE OF TECHNOLOGY
*                               JANUARY 20, 1987
*
.....

INTEGER*2      CHANNEL
INTEGER*4      IPIBLK(8), FILEPTR, PICOPS, CMRS, UNIT
CHARACTER*80   NAME

DATA PICOPS / 0 /

*
*.....      ERROR CHECKING
*
IF ( CHANNEL .LT. -1 .OR. CHANNEL .GT. 3 ) THEN
  WRITE (6,*)
  WRITE (6,*) '*** ERROR IN SHOMNO *** ILLEGAL CHANNEL'
  WRITE (6,*)
  RETURN
END IF

*
*.....      ASSIGN THE CMR THE PROPER VALUE
*

IF ( CHANNEL .EQ. -1 ) CMRS = 7
IF ( CHANNEL .EQ. 0 ) CMRS = 1
IF ( CHANNEL .EQ. 1 ) CMRS = 2
IF ( CHANNEL .EQ. 2 ) CMRS = 4
IF ( CHANNEL .EQ. 3 ) CMRS = 8

```

```
*****      DISPLAY THE IMAGE
*
      CALL IPI_OPENFILE ( IPIBLK, NAME, 0 )
      CALL IPI_DISKPIC  ( IPIBLK, FILEPTR, PicOps )
      CALL IPI_PUTPIC   ( UNIT, CMRS, ZVAL( FILEPTR ), PicOps )
      CALL IPI_CLOSEFILE ( IPIBLK )
*
*****      RETURN TO CALLING PROGRAM
*
      RETURN
      END
```

Appendix H

```

.....
SUBROUTINE SAVMNO ( UNIT, NAME, CHANNEL )
.....
*
*   SAVMNO -   WILL SAVE A 512x512 MONOCHROME IMAGE TO DISK
*              FROM THE CHANNEL.
*
.....
*
*   VARIABLE DECLARATION:
*
*   NAME -     THE FILENAME CONTAINING THE 512x512 MONOCHROME
*              IMAGE. THIS SHOULD BE DECLARED AS CHARACTER*80
*              IN THE CALLING PROGRAM.
*   CHANNEL -  THE CHANNEL THAT THE IMAGE SHOULD BE PLACED IN.
*              THIS SHOULD BE DECLARED AS INTEGER*2 IN THE
*              CALLING PROGRAM.
*
.....
*
*   REQUIRED SUBROUTINES:  [SLS4255.Cis.IPI]      IPI_OPENFILE
*                                                           IPI_DISKPIC
*                                                           IPI_GETPIC
*                                                           IPI_SETSIZE
*                                                           IPI_CLOSEFILE
*
.....
*
*   AUTHOR:      CARL SALVAGGIO  CENTER FOR IMAGING SCIENCE
*                               ROCHESTER INSTITUTE OF TECHNOLOGY
*                               FEBRUARY 6, 1987
*
.....

INTEGER*2      CHANNEL
INTEGER*4      IPIBlk(8), FILEPTR, PICOPS, CMRS, UNIT, STATUS
CHARACTER*80   NAME

DATA PICOPS / 0 /

*
*.....      ERROR CHECKING
*
      IF ( CHANNEL .LT. 0 .OR. CHANNEL .GT. 2 ) THEN
        WRITE (6,*)
        WRITE (6,*) '*** ERROR IN SAVMNO *** ILLEGAL CHANNEL'
        WRITE (6,*)
        RETURN
      END IF

*
*.....      ASSIGN THE CMR THE PROPER VALUE
*
      IF ( CHANNEL .EQ. 0 ) CMRS = 1
      IF ( CHANNEL .EQ. 1 ) CMRS = 2
      IF ( CHANNEL .EQ. 2 ) CMRS = 4

*
*.....      DISPLAY THE IMAGE

```

```

*
STATUS = IPI_OPENFILE ( IPIBLK, NAME, IPI_SIZEPIC( PICOPS ) )
CALL IPI_ERRORCHECK ( STATUS, 'OPENFILE:' )
STATUS = IPI_DISKPIC ( IPIBLK, FILEPTR, PICOPS )
CALL IPI_ERRORCHECK ( STATUS, 'DISKPIC:' )
STATUS = IPI_GETPIC ( UNIT, CMRS, ZVAL( FILEPTR ), PICOPS )
CALL IPI_ERRORCHECK ( STATUS, 'GETPIC:' )
STATUS = IPI_CLOSEFILE ( IPIBLK )
CALL IPI_ERRORCHECK ( STATUS, 'CLOSEFILE:' )

*
*****      RETURN TO CALLING PROGRAM
*

RETURN
END

```

```

SUBROUTINE TRIMNO ( UNIT, CHAN )
.....
*
*   TRIMNO -      WILL DISPLAY A SPECIFIED CHANNEL IN BLACK AND
*                 WHITE.  JUST SEND THIS ROUTINE THE CHANNEL
*                 NUMBER, 0,1,2,3 TO VIEW THESE CHANNELS OR SEND
*                 -1 TO RESET TO NORMAL VIEWING
*
*.....
*
*   VARIABLE DECLARATION
*
*   CHAN - THE CHANNEL TO SET TO B & W ( INTEGER*4 )
*   UNIT - THE UNIT ATTACHED TO THE PROCESS ( INTEGER*4 )
*
*.....
*
*   REQUIRED SUBROUTINES      [SLS4255.Cis.IPI]VIEWCHAN
*
*.....
*
*   AUTHOR      CARL SALVAGGIO      CENTER FOR IMAGING SCIENCE
*                                     ROCHESTER INSTITUTE OF TECHNOLOGY
*                                     FEBRUARY 25, 1987
*
*.....
*
*   INTEGER*4 LENGTH, UNIT, STATUS, CHAN
*
*.....
*
*   SHOW THE SPECIFIED CHANNEL IN B & W
*
*   IF ( CHAN .EQ. -1 ) THEN
*     STATUS = IPI_VIEWCHAN( UNIT, CHAN, 0 )    ! RESET TRIMNO
*   ELSE
*     STATUS = IPI_VIEWCHAN( UNIT, CHAN )      ! SET TRIMNO
*   END IF
*   CALL IPI_ERRORCHECK ( STATUS, 'VIEWCHAN:' )
*
*.....
*
*   RETURN TO CALLING PROGRAM
*
*
*   RETURN
*   END

```

```

SUBROUTINE THRESHOLD ( UNIT, CHANNEL, THRESH, IOPT )
.....
*
*   THRESH -      THIS PROGRAM WILL THRESHOLD A MONOCHROME (ONE
*                  CHANNEL) IMAGE BY SETTING ALL DC'S ABOVE A
*                  SPECIFIED VALUE TO ZERO WHILE LEAVING ALL DC'S
*                  BELOW THE THRESHOLD VALUE THE SAME.
*
*.....
*
*   VARIABLE DECLARATION:
*
*   CHANNEL -      THE MEMORY CHANNEL THAT THE USER WISHES TO
*                  THRESHOLD IN. THIS VARIABLE SHOULD BE DECLARED
*                  AS INTEGER*2 IN THE CALLING PROGRAM.
*
*   THRESH -      THE DIGITAL COUNT VALUE AT WHICH THE THRESHOLD
*                  SHOULD OCCUR. THIS VARIABLE SHOULD BE DECLARED
*                  AS INTEGER*2 IN THE CALLING PROGRAM.
*
*   IOPT -        THE THRESHOLDING OPTION:
*                  IOPT = 0 WILL SET ALL VALUES BELOW 'THRESH'
*                  TO A DC OF ZERO.
*                  IOPT <> 0 WILL SET ALL VALUES ABOVE 'THRESH'
*                  TO A DC OF ZERO.
*
*.....
*
*   REQUIRED SUBROUTINES : [SLS4255.Cis.IPI]      PUTITT
*                                     ENABLEITT
*
*.....
*
*   AUTHOR:        CARL SALVAGGIO  CENTER FOR IMAGING SCIENCE
*                                     ROCHESTER INSTITUTE OF TECHNOLOGY
*                                     JANUARY 16, 1987
*
*.....

INTEGER*2  CHANNEL, THRESH, MEMVOC, DVPMEM
INTEGER*2  ITT(0:255)
INTEGER*4  UNIT

DATA MEMVOC / 1 /      ! MEMORY VOC NUMBER
DATA DVPMEM / 1 /      ! DVP ALLOCATION NUMBER

*
*.....      ERROR CHECKING
*
IF ( CHANNEL .LT. 0 .OR. CHANNEL .GT. 2 ) THEN
  WRITE (6,*)
  WRITE (6,*) '*** ERROR IN THRESHOLD *** ILLEGAL CHANNEL NUMBER'
  WRITE (6,*)
  RETURN
END IF

IF ( THRESH .LT. 0 .OR. THRESH .GT. 255 ) THEN
  WRITE (6,*)
  WRITE (6,*) '*** ERROR IN THRESHOLD *** ILLEGAL THRESHOLD'
  WRITE (6,*)
  RETURN

```

```

END IF

"
*****      DEFINE THE ITT
"

      IF ( IOPT .NE. 0 ) THEN
        DO 10 I = 0, 255, 1
          IF ( I .LT. THRESH ) THEN
            ITT( I ) = I
          ELSE
            ITT( I ) = 0
          END IF
10      CONTINUE
        ELSE
          DO 20 I = 0, 255, 1
            IF ( I .GE. THRESH ) THEN
              ITT( I ) = I
            ELSE
              ITT( I ) = 0
            END IF
20      CONTINUE
        END IF

"
*****      WRITE TO THE ITT AND ENABLE IT
"

      CALL IPI_PUTITT ( UNIT, CHANNEL, ITT )
      CALL IPI_ENABLEITT ( UNIT, CHANNEL, MEMVOC, DVPMEM )

"
*****      RETURN TO CALLING PROGRAM
"

      RETURN
      END

```

```

SUBROUTINE WRTITT ( UNIT, INCHAN, OUTCHAN )
.....
*
*   WRTITT -      THIS PROGRAM WILL SAVE THE IMAGE IN THE
*                 SPECIFIED CHANNEL THRU THE CURRENT ITT
*                 INTO A SPECIFIED CHANNEL.
*
*
*
*
*   VARIABLE DECLARATION:
*
*   INCHAN -      THE CHANNEL ON WHICH THE CURRENT ITT IS
*                 PRESENT. THIS VARIABLE SHOULD BE DECLARED AS
*                 INTEGER*2 IN THE CALLING PROGRAM.
*   OUTCHAN -     THE CHANNEL WHICH SHOULD BE WRITTEN THRU THE
*                 THE ITT IN 'INCHAN', THIS VARIABLE SHOULD BE
*                 DECLARED AS INTEGER*2 IN THE CALLING PROGRAM.
*
*
*
*   REQUIRED SUBROUTINES:  [SLS4255.Cis.IPI]      IPI_ATT DVP
*                                                         IPI_DET DVP
*                                                         IPI_ENABLE ITT
*                                                         IPI_DVP MATH
*                                                         IPI_CONSTANTS
*
*
*
*   AUTHOR:        CARL SALVAGGIO  CENTER FOR IMAGING SCIENCE
*                                     ROCHESTER INSTITUTE OF TECHNOLOGY
*                                     JANUARY 16, 1987
*
*
*
*   INTEGER*2      INCHAN, OUTCHAN
*   INTEGER*4      UNIT, STATUS, LENGTH
*
*   EXTERNAL       IPI__NOP
*
*
*
*   *****      ERROR CHECKING
*
*   IF ( INCHAN .LT. 0 .OR. INCHAN .GT. 2 ) THEN
*     WRITE (6,*)
*     WRITE (6,*) '*** ERROR IN WRTITT *** ILLEGAL IN CHANNEL'
*     WRITE (6,*)
*     RETURN
*   END IF
*
*   IF ( OUTCHAN .LT. 0 .OR. OUTCHAN .GT. 2 ) THEN
*     WRITE (6,*)
*     WRITE (6,*) '*** ERROR IN WRTITT *** ILLEGAL OUT CHANNEL'
*     WRITE (6,*)
*     RETURN
*   END IF
*
*
*
*   *****      SAVE THE CHANNEL THRU ITT TO MEMORY
*

```

Appendix H

```
CALL IPI_ATT DVP ( UNIT, 1 )           ! ATTACH THE DVP
CALL IPI_ENABLEITT ( UNIT, INCHAN, 0, 1 ) ! ENABLE THE ITT
! , , MEMVOC, DVP MEM )
CALL IPI_DVPMATH ( UNIT, IPI_NOP, INCHAN, ! WRITE THRU ITT
INCHAN, OUTCHAN )
CALL IPI_ENABLEITT ( UNIT, INCHAN, 0, 0 ) ! DISABLE THE ITT
CALL IPI_DET DVP ( UNIT )              ! DEATTACH THE DVP

-
***** RETURN TO CALLING PROGRAM
-

RETURN
END
```

Appendix I

Description Of Non-Linearity Problems Encounter When Digitizing Photographic Transparencies

When digitizing a photographic transparency, care must be taken to observe the non-linear nature of the photographic material. This becomes especially important when the resulting imagery is to be used with a process such as pseudo invariant feature analysis. It is essential that the assumption of linearity between the reflectivity of the scene elements and the brightness values in the digital image be satisfied or the resulting linear transformations become invalid. The following is a justification which allows the PIF normalization procedure to be used on digitized airphotos as long as the caveats mentioned are satisfied.

Linearity can be established between reflectivity of scene elements and the resulting brightness counts in a digitized image as follows:

If it is assumed that the radiance at the sensor, L , is a linear function of the reflectivity, R , of a scene element

$$L = \alpha R + \beta \quad (1)$$

where α and β are linear coefficients encompassing atmospheric effects

and we have the definition of intensity at the sensor

$$L = \frac{dI}{dA \cos\theta}$$

where I is the intensity associated with a scene element at the sensor

dA is the element of area associated with the scene element

and θ is the view angle measure to the normal

then if we assume that our sensing system is viewing straight down at the ground (i.e. $\theta = 0^\circ$) we have

$$I = \int L \, dA \cos\theta$$

$$I = \int L \, dA$$

$$I = L \int dA$$

and therefore

$$I = L A \quad (2)$$

which says that I at the sensor is a linear function of L at the sensor.

Now the irradiance at the sensor is found by taking equation (2) through the optical system of the sensor with a transmittance τ as

$$I = \tau L A$$

A can be rewritten as

$$A = \frac{\pi d^2}{4}$$

where d is the diameter of the aperture of the optical system.

The intensity on the sensor focal plane is

$$I = \frac{L \tau \pi d^2}{4}$$

so the irradiance on the focal plane is written as

$$E = \frac{L \tau \pi d^2}{4 f^2}$$

where f is the focal length of the optical system.

The term $G\#$ is defined as

$$G\# = \frac{L}{E} = \frac{4 f^2}{\tau \pi d^2}$$

and the irradiance on the sensor is simplified to

$$E = \frac{L}{G\#}$$

Thus, the irradiance on the film plane is a linear function of the radiance on the sensor.

By definition, the exposure on the film plane, H , is a linear function of the irradiance, E , namely that

$$H = E t \quad (3)$$

where t is the exposure time.

Therefore the exposure at the sensor is a linear function of the reflectivity of the scene elements on the ground where

$$H = E t$$

and,

$$H = \frac{L t}{G\#}$$

$$H = \frac{(\alpha R + \beta) t}{G\#}$$

$$H = \frac{\alpha t}{G\#} R + \frac{\beta t}{G\#}$$

and finally $H = m R + b$ (4)

with $m = \frac{\alpha t}{G\#}$

and $b = \frac{\beta t}{G\#}$

Now if the sensor is a photographic emulsion, the sensor response function is classically represented by the D-log H curve. This curve and the corresponding τ -H (transmittance vs. exposure) curve are depicted in Figure I-1. The relationships represented by these curves are highly non-linear. In order to overcome this non-

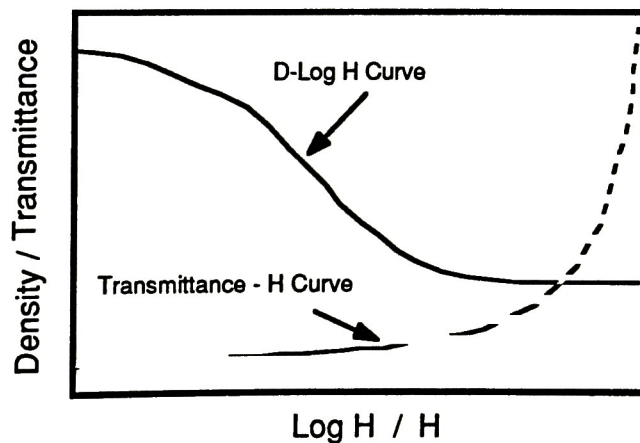


Figure I-1 Typical D-Log H Curve / τ vs. H Curve

linearity the following caveats must be instituted. The straight line portion of the characteristic D-log H curve is not an indication of linearity since both quantities are themselves logarithmic, however, in the exposure region corresponding to this straight line portion of the curve, the following can be said

Let γ be the slope of the straight line portion of the D-log H curve so

$$D = \gamma \log H - \log i \quad (5)^*$$

where $\log i$ is the intercept value of the density axis

we can rewrite (5) as

$$D = \gamma (\log H - c \log i)$$

where c is a constant.

Taking the antilogarithm of both sides of the above equation we get

$$\tau = C H^{-\gamma} \quad (6)$$

where C is a multiplicative constant.

* T.H. James, The Theory of the Photographic Process, Macmillan Publishing Co., Inc., 1966, p. 505.

When $\gamma = -1$ (i.e. for a positive working photographic material) we have $\tau = CH$ so that a photographic system can be made to be linear over a limited range under the following two caveats:

- 1) the exposure range is limited to that portion corresponding to the straight line portion of the D-log H curve, and
- 2) the value γ for the film is forced to be -1.

So now we have that the transmittance of a photographic emulsion is linearly related to the reflectivity of a scene element under the caveats listed above, namely

$$\tau = m' R + b' \quad .(7)$$

where m and b are linear coefficients.

Now if we can make the assumption that the digitizing system has a linear response function to incident radiance, L_D , then we have

$$DC = m'' L_D + b'' \quad (8)$$

where DC is the brightness value produced for a scene element by the digitizer, and
 m and b are gain and offset factors for the digitizing system.

Since we know that

$$L_D = \tau L_0$$

where L_0 is the constant radiance incident on the photographic emulsion

and τ is the transmittance of the transparency at any point

we can conclude that

$$DC = m''' R + b''' \quad (9)$$

where m''' and b''' are linear coefficients.

Equation (9) shows that the digital count of a digitized airphoto is linearly related to the reflectance of the scene elements included in the photograph over a limited dynamic range.

The previous discussion has shown that under specific caveats the brightness values of a digitized photographic transparency can be considered to be linearly related to the reflectance of corresponding scene elements. The caveats imposed are however very stringent. If the gamma of the photographic material significantly deviates from unity or the dynamic range of the scene exposures significantly deviate from the straight line region of the D-log H curve, the assumption of linearity is weakened. The weakening of this assumption can be prevented if the D-log H curve is known for the film and appropriate action is taken to correct the brightness values of the digital image for this non-linearity.

For this current study, the D-log H information was not available for the NHAP image used and therefore no correction could be applied. The caveats mentioned above were assumed to be true and the transparencies of Buffalo were digitized and assumed to represent a linear system.

Vita

The author of this thesis, Carl Salvaggio, was born on January 30, 1963 in Perth Amboy, New Jersey. After graduating from Cedar Ridge High School in Old Bridge, New Jersey in June 1981, Carl enrolled in the Biomedical Photographic Communications program at the Rochester Institute of Technology in September of that same year. After two quarters in this program, Carl decided to transfer to the Imaging Science program at RIT in June 1982.

In September 1985, the Digital Image Remote Sensing laboratory at RIT hired Carl as a graduate assistant. This position turned into full-time employment with this laboratory in May 1986 where Carl is still functioning as a Research Associate/Software Engineer under the direction of Dr. John R. Schott. This thesis is the culmination of a large body of work that has been done by Carl and this laboratory as part of governmental research grants and individual research projects.