CUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCE

MASTER THESIS

Gülçin YİĞİTLER

New Techniques for Classification, Enhancement, and Fusion Of Multispectral Remote Sensing Images

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ADANA, 2005

ÖZ

YÜKSEK LİSANS TEZİ

UZAKTAN ALGILANMIŞ GÖRÜNTÜLERDE SINIFLANDIRMA, NETLEŞTİRME VE FÜZYON TEKNİKLERİNE YENİ BİR YAKLAŞIM

Gülçin YİĞİTLER

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK ANA BİLİM DALI

Danışman : Yard. Doç. Dr. Turgay İBRİKÇİ
Prof . Dr. Okan ERSOY
Yıl: 2005 Sayfa: 69
Jüri : Yard. Doç. Dr. Turgay İBRİKÇİ
Prof. Dr. Vedat PEŞTEMALCI
Yrd. Doç. Dr. Kerem ÜN

İşlenmemiş uydu görüntüleri analistlerin göremediği çok büyük miktarlarda bilgi içerir. Bu yüzden, görüntüdeki detayları ve hedef bileşenlerdeki gizli farklılıkları ortaya çıkarmak için görüntü netleştirme yöntemleri kullanılır.

Bu çalışmada yeni bir çok bantlı görüntü netleştirme ve füzyon yöntemi geliştirilmektedir. Bugüne kadar kullanılan görüntü netleştirme yöntemlerinde bantlar ayrı ayrı işlendiğinden görüntüde bozulmalar meydana geliyordu. Bantların içerdiği çok miktarda bilgi verimli bir şekilde kullanılmıyordu. Bu çalışmanın amacı bantlarda bulunan bilgiden en iyi şekilde faydalanıp, bozulmaları ortadan kaldırmak ve resmin görünümünü iyileştirmektir. Bu amacı gerçekleştirmek için dönüşüm teknikleri kullanılarak görüntü netleştirme ve görüntü füzyonu birlikte düşünülüp iki yeni algoritma geliştirildi. Geliştirilen birinci algoritma RGB2NTSC , RGB (Red Green Blue) renk sisteminden IHS (Intensity Hue Saturation) renk sistemine dönüşüm metodunun ve morfolojik süzgeçlemenin kullanıldığı bir metottur. RGB2NTSC algoritmasında sadece 3 görüntü ile çalışılabildiğinden, 3'ten fazla görüntü ile çalışabilmek için ikinci olarak gerçek ayrık Fourier analiz dönüşümü (RDFT) algoritması geliştirildi. Bu metotlar sonucunda elde edilen görüntüler daha az gürültülü ve beklenen özelliklere (kesin kenar vb.) sahiptir.

Anahtar Kelimeler: Görüntü Netleştirme, Görüntü Füzyonu, Çok Bantlı Görüntüler, IHS (Intensity Hue Saturation) Dönüşümü, Gerçek Ayrık Fourier Dönüşümü (RDFT)

ABSTRACT

MSc THESIS

NEW TECHNIQUES FOR CLASSIFICATION, ENHANCEMENT, AND FUSION OF MULTISPECTRAL REMOTE SENSING IMAGES

Gülçin YİĞİTLER

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
INSTITUTE OF NATURAL AND APPLIED SCIENCES
UNIVERSITY OF ÇUKUROVA

Supervisor : Asst. Prof. Dr. Turgay İBRİKÇİ
Prof. Dr. Okan ERSOY
Year: 2005 Pages: 69
Jury : Asst. Prof. Dr. Turgay İBRİKÇİ
Prof. Dr. Vedat PEŞTEMALCI
Asst. Prof. Dr. Kerem ÜN

Raw satellite images often contain a vast amount of information that is not readily apparent to the analyst. Therefore, image enhancement techniques are used to highlight features of interest and expose subtle differences in the spectral signature of the components of the target.

This thesis presents a new approach to the development of multispectral / hyperspectral image enhancement and fusion algorithms. In approaches used up to now for image enhancement, the bands are typically processed separately, and this results in considerable distortion. The amounts of information in many bands are also not very efficiently used. The objective of this work is to utilize the amount of information available more effectively, remove such distortions and to improve the appearance of the images. To realize this goal, we developed two new algorithms using transform techniques. In the resulting algorithms developed, image enhancement and image fusion are considered together. The first algorithm developed is RGB (Red Green Blue) to IHS (Intensity Hue Saturation) (RGB2NTSC) algorithm which uses RGB to IHS transformation method and a morphological filter. The second algorithm developed is the real discrete Fourier transform (RDFT) algorithm. Because the RGB2NTSC algorithm allows a transformation to work only with three images, to be able to combine more than three images, we used the RDFT. The final images obtained from these algorithms are much less noisy and has more desirable properties such as sharp edges.

Keywords: Image Enhancement, Image Fusion, Multispectral and Hyperspectral Images, IHS Transform, RDFT Transform.

ACKNOWLEDGEMENTS

I would like to express my respects and gratitude to my supervisor Asst. Prof. Dr. Turgay İBRİKÇİ. Many thanks for all of his supports, guidance, patience, cooperation, suggestions on initiating, improving and completing this study.

I would like to thank to Prof. Dr. Okan K. ERSOY School of Electrical and Computer Engineering, Purdue University for sharing his knowledge, and for his support, and assistance throughout this project.

I would like to thank to my committee members Prof. Dr. Vedat PEŞTEMALCI and Asst. Prof. Dr. Kerem ÜN for their supports and very valuable discussions.

I would like to thank to my family, my father Hasan, my mother Ayşe, and my brother Mahir. They always encouraged and supported me with their loves and inspirations.

Thanks to my lover İnan, for his love, tolerance, helps, patience and moral support. Finally, I thank all my friends, great people that I couldn't mention the name of here, for their good wishes and encouragements.

CONTENTS

NOTATIONS

g(x, y)	The processed image
f(x, y)	The input image
T	An operator on f
I	Image
I_Z	The opening of the image
I^Z	The closing of the image
Z	Structuring element (SE)
S_j	The result of smoothing I
d_j	SE diameter
D_j	Residual Image
T_j	The positive residual the “tophat” image
B_j	The negative residual the “bothat” image
R_j	either T_j or B_j
p_{Rj}	The gray level probability distribution of R_j
m_{Rj}	The square root of the second moment of p_{Rj}
t	Threshold
J	The cleaned image
$\hat{T}_j$	jth noise cleaned tophat
$\hat{B}_j$	The jth noise cleaned bothat
f	A factor
s	The minimum area per 3×3 neighborhood of a region in the cleaned up residual
E $\{n^2\}$	The second moment of the gray level probability distribution of R
Y, C_1, C_2	Tristimulus values
I	Intensity
H	Hue
S	Saturation

$X_c(n)$	The complex discrete Fourier transform
$X(n)$	The real discrete Fourier transform
TM	Transform Matrix
ITM	Inverse Transform Matrix
L	Number of possible combinations per band
M	TM size
X	Feature vector
$g_m(x)$	Discriminant function
w_i	i th class
$P(w_i\|x)$	Conditional probabilities
x	Vector of brightness values
$P(w_i)$	Prior probabilities
$P(x\|w_i)$	Multivariate normal density function
m_i	Mean vector
Σ_i	Covariance matrix

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Raw satellite data often contain a vast amount of information that is not readily apparent to the analyst. Image enhancement techniques are used to highlight features of interest and expose subtle differences in the spectral signature of the components of the target (Pfister, 2004). Image enhancement processes consist of a collection of techniques that seek to improve the visual appearance of an image, or to convert the image to a form better suited for analysis by a human or a machine (Pratt, 2001). Image fusion is the process of combining complementary information from multiple sources to obtain a new image which is more suitable for human and machine perception or further image processing (Ersoy, 2005).

This thesis presents a new approach to multispectral / hyperspectral image enhancement and fusion algorithms. In approaches used up to now for image enhancement, bands were usually processed separately, and this results in considerable amount of distortion. The objective of this work is to make better use of all the bands available, thereby remove such distortions and improve the appearance of the images. To realize this goal, we developed two new algorithms. In the algorithms developed, image enhancement and image fusion are considered together. Even though the experimental results utilize multispectral remote sensing images, the techniques are also applicable to other multispectral images, such as biomedical multispectral images, and hyperspectral images.

In multispectral and hyperspectral image processing, each band constitutes an image. Hence, at each pixel, there is a vector generated by different bands. We make use of two concepts to make maximal use of the information present. The first one is to choose M bands out of a total of N bands at each processing step. This generates vectors of length M at each pixel. These vectors are transformed with a transformation matrix to a transform space, processed and inverse transformed. This corresponds to image fusion which is further enhanced by a second concept discussed below. Signal processing, such as morphological or other types of filtering is applied in the transformation domain. Then, the inverse transform is applied to return to the original space.

The second crucial concept is averaging over many different images obtained. When we use M out of N images, there are many possible combinations. For example, when we choose 3 out of 7 images, there are 15 possible combinations per band. When we use all these combinations, each band is regenerated 15 times. When the resulting 15 images are averaged per band, the final image per band is much less noisy and has more desirable properties such as sharp edges.

The first algorithm developed is RGB (Red Green Blue) to IHS (Intensity Hue Saturation) (RGB2NTSC) algorithm which uses RGB to IHS transformation method and a morphological filter. An ideal filter should satisfy two goals which are preserving edges, thin lines, and small features, and smoothing areas between these features. The morphological filter used is chosen for this purpose (Peters, 1995).

The second algorithm developed is the real discrete Fourier transform (RDFT) algorithm. The RGB2NTSC algorithm allows a transformation to work only with three band color composite images. To be able to combine more than three images, we used the RDFT, which allows any number of band composites.

The thesis is organized as follows:

The image enhancement concept and smoothing spatial filter types, morphologic filters and morphological image cleaning filters are reviewed in Chapter 2.

In Chapter 3, image fusion method is discussed, and it is compared with RGB2NTSC algorithm.

RDFT is introduced in Chapter 4.

The new transform techniques (RGB2NTSC and RDFT algorithms) and experimental results are discussed in Chapter 5.

The classification concept, maximum likelihood method and classification accuracies are reviewed in Chapter 6.

The conclusions are discussed in Chapter 7.

2. IMAGE ENHANCEMENT

Image enhancement methods improve the appearance of the imagery. The advantage of digital imagery is that allows us to manipulate the digital pixel values in an image. Examples of enhancement functions include contrast stretching to increase the tonal distinction between various features in a scene, and spatial filtering to enhance (or suppress) specific spatial patterns in an image (Natural Resources Canada, 2004).

2.1. Spatial Domain

Spatial domain constitutes the collection of pixels composing an image. Spatial domain methods are procedures that operate directly on these pixels. Spatial domain processes will be denoted by the expression

$$g(x,y) = T\left[f\left(x,y\right)\right] \tag{1}$$

where f(x, y) is the input image, g(x, y) is the processed image, and *T* is an operator on *f*, defined over some neighborhood of (x, y). In addition, *T* can operate on a set of input images, such as performing the pixel-by-pixel sum of *K* images for noise reduction.

The principal approach in defining a neighborhood about a point (x, y) is to use a square or rectangular subimage area centered at (x, y), as seen in Figure 2.1.

The center of the subimage is moved from pixel to pixel starting, say, at the top left corner. The operator *T* is applied at each location (x, y) to yield the output, *g*, at that location. The process utilizes only the pixels in the area of the image spanned by the neighborhood (Gonzalez and Woods, 2002).

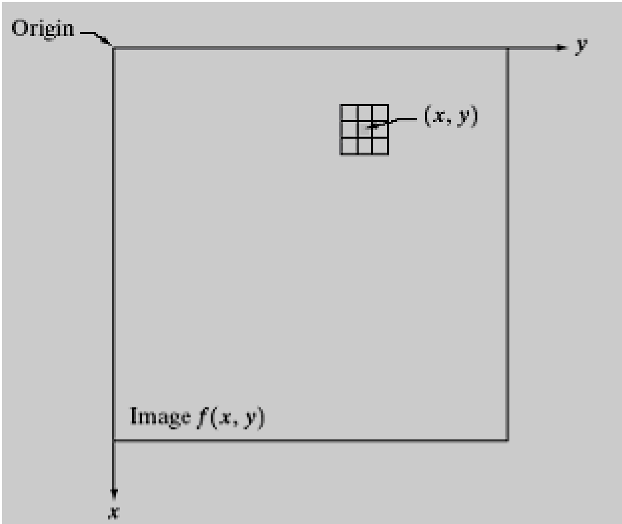


Figure 2.1. A 3x3 neigborhood about a point (x,y) in an image (Gonzalez and Woods, 2002)

2.1.1. Spatial Filtering

The techniques of spatial filtering are illustrated in Figure 2.2. The filter mask is moved from point to point in an image. At each point (x, y), the response of the filter at that point is calculated using a predefined relationship.

2.1.1.1. Smoothing Spatial Filters

Smoothing filters are used for blurring and for noise reduction. Blurring is used in preprocessing steps, such as removal of small details from an image, and bridging of small gaps in lines or curves. Linear filters and nonlinear filters are used for noise reduction.

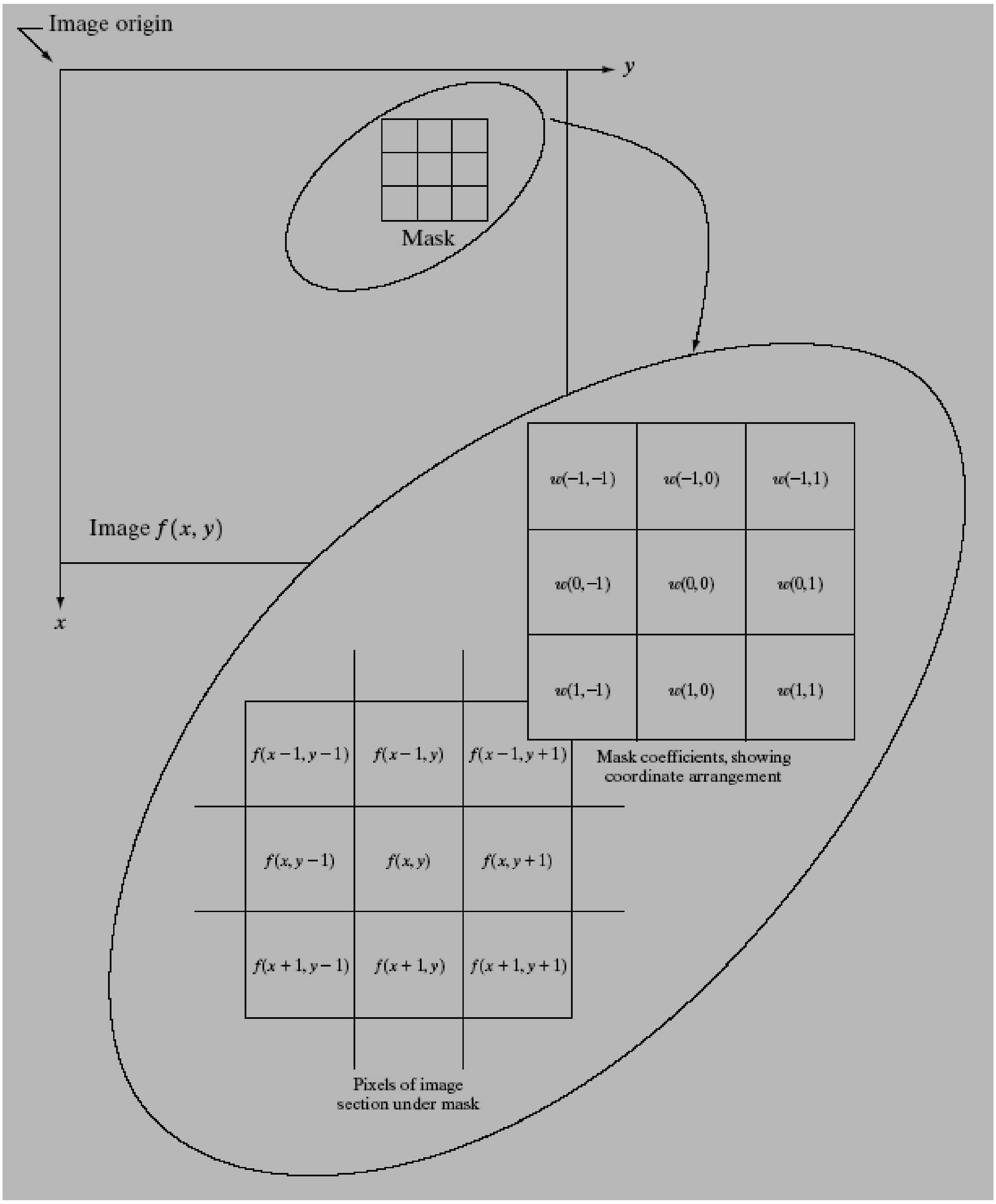


Figure 2.2. The techniques of spatial filtering. The magnified drawing shows a 3x3 mask and the image section directly under it; the image section is shown displaced out from under the mask for ease of readability (Gonzalez and Woods, 2002)

2.1.1.1.(1). Linear Filters

The output of a smoothing, linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask. Linear filters also are known as averaging filters or lowpass filters.

The aim of smoothing filters is to replace the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask. This process results in an image with reduced “sharp” transitions in gray levels. Because random noise typically consists of sharp transitions in gray levels, the most obvious application of smoothing is noise reduction. However, edges (which almost always are desirable features of an image) also are characterized by sharp transitions in gray levels, so averaging filters have the undesirable side effect of blurring edges.

Figure 2.3 shows 3X3 smoothing filter. Use of the this filter yields the standard average of the pixels under the mask. This can best be seen by substituting the coefficients of the mask into Eq. (2);

$$R = \frac{1}{9}\sum_{i=1}^{9} z_i \qquad (2)$$

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Figure 2.3. 3X3 smoothing filters

which is the average of the gray levels of the pixels in the 3X3 neighborhood defined by the mask. Note that, instead of being 1/9, the coefficients of the filter are all 1’s.The idea here is that it is computationally more efficient to have coefficients valued 1. At the end of the filtering process the entire image is divided by 9. An mXn mask would have a normalizing constant equal to 1/mn.

2.1.1.1.(2). Smoothing Nonlinear Filters

Linear averaging filters blur edges. They also perform very poorly in case of binary noise. Binary noise (e.g. outliers in a homogeneous image region) causes wrong gray values for a few randomly distributed pixels. The image is blurred, but the error caused by the binary noise is not eliminated but only distributed to neighboring pixels. So the proper operation to process such distortions is to detect these pixels and to eliminate them.

Although nonlinear filters have other deficiencies, they can solve the above mentioned problem better than linear filters (Pratt, 2001).

2.1.1.1.(3). Order-Statistics Filters (Median Filter)

Order-statistics filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result. The best-known order-statistics filter is the median filter, which replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel (the original value of the pixel is included in the computation of the median). Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of impulsive noise, also called salt-and-pepper noise because of its appearance as white and black dots superimposed on an image.

The median filter is a neighborhood operation, similar to convolution, except that the calculation is not a weighted sum. Instead, the pixels in the neighborhood are ranked in the order of their gray levels, and the median value of the group is stored in the output pixel (Castleman, 1996). As in the case of the kernel operations, this is used to produce a new image and only the original pixel values are used in the ranking for the neighborhood around each pixel (Russ, 1999).

The median filter is excellent for removal of binary noise. Since a binary noise pixel tends to have values in the extreme rank positions, it is very unlikely that it will show the medium gray value in the neighborhood. So, if a pixel is accidentally changed to an extreme value, it will be eliminated from the image and replaced by a reasonable value, the median value in the neighborhood as seen in Figure 2.4. (Jaehne, 1997; Hussain, 1991).

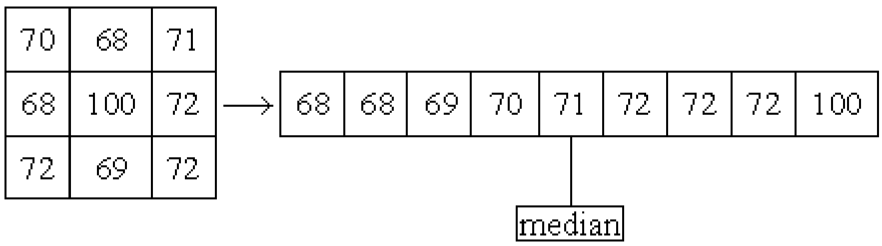


Figure 2.4. Median Filtering

The following example illustrate the effect of a 1x3 median filter M:

M [… 1 2 3 7 8 9 …] → [… 1 2 3 7 8 9 …]

M [… 1 2 15 4 5 6…] → [… 1 2 4 5 5 6 …]

M [… 0 0 0 9 9 9 …] → [… 0 0 0 9 9 9 …]

As expected, the median filter eliminates outliers. The two other gray value structures, a monotonically increasing ramp and edge between two constant neighborhoods are preserved. In this way, a median filter effectively eliminates binary noise while preserving edge sharpness (Jaehne, 1997).

The median filter not only smooths noise in homogeneous image regions, but also tends to produce regions of constant or nearly constant intensity. Iterative filtering of an image with a median filter results in an image containing only constant neighborhoods and edges. If only single pixels are distorted, a 3X3 median filter is sufficient to eliminate them. If clusters of distorted pixels occur, larger median filters must be used (Jaehne, 1997; Jain, 1989).

Because of the requirement for sorting all the pixels in each neighborhood by gray level, median filtering is slower than convolving (Castleman,1996). There are, however, algorithms that speed up the process (Huang, 1979).

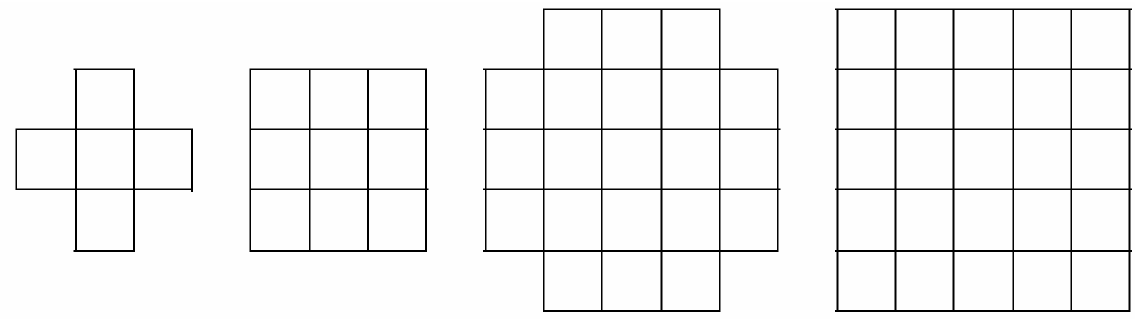


Figure 2.5. Neighborhood patterns used for median filtering

One important variable in the use of a rank operator is the spatial extent of the neighborhood (mask). Generally, shapes that are squares (for convenience of computation) or approximations to a circle (to minimize directional effects) are used. Figure 2.5 shows several of the masks often used for ranking. As the size of the neighborhood is increased, however, the computational effort in performing the ranking increases rapidly (Russ, 1999; Castleman, 1996).

Although the median filter is by far the most useful order-statistics filter in image processing, it is by no means the only one. The median represents the 50th percentile of a ranked set of numbers, but there are many other ranking possibilities. For example, using the 100th percentile results in the so-called max filter, which is useful in finding the brightest points in an image. The 0th percentile filter is the min filter, used for the opposite purpose (Gonzalez and Woods, 2002).

2.2. Morphological Filters

Preserving edges, thin lines, and small features, and smoothing areas between these features are two goals of an perfect filter. Classical spatial filters can not achieve these two goals concurrently. They usually replace each pixel with some function of the pixel's neighborhoods. They usually use an edge detection algorithm as a first step of the adaptive filtering algorithm to preserve edge deformation, however, there are many problems during edge detecting such as misplaced edges, undetected edges etc. Although median filters satisfy both noise cleaning and edge detection in some applications (salt and pepper noise), they are not sufficiently qualified to remove dense noise. They may remove some thin lines and small features. Instead of using adaptive filtering algorithms, morphological filters are the most well-known nonlinear filters for image enhancement (Yıldırım and Ersoy, 2004).

Common applications of morphological operators include filling small holes in objects, separating adjacent or slightly overlapping objects, and joining broken boundaries into continuous segments. Morphological operators have been successfully used in remote sensing applications. Structuring elements such as disk, ball, square etc. are used with the four most basic operations: dilation, erosion, opening, and closing. Opening and closing can be produced through combinations of multiple applications of dilation and erosion. There have been many filters which are introduced using various combinations of these operators. Dilation allows objects to expand, thus potentially filling in small holes and connecting disjoint objects. Erosion shrinks objects by etching away (eroding) their boundaries. These operations can be customized for an application by the proper selection of the structuring element, which determines exactly how the objects wil be dilated or eroded. The dilation process is performed by laying the structuring element on the image and sliding it across the image in a manner similar to convolution. The difference is in the operation performed. It is best described in a sequence of steps as follows:

1. If the origin of the structuring element coincides with a '0' in the image, there is no change; move to the next pixel.

2. If the origin of the structuring element coincides with a ‘1’ in the image, perform the OR logic operation on all pixels within the structuring element.

An example is shown in Figure 2.6. Note that with a dilation operation, all the ‘1’ pixels in the original image will be retained, any boundaries will be expanded, and small holes will be filled. The erosion process is similar to dilation, but we turn pixels to ‘0’, not ‘1’. As before, slide the structuring element across the image and then follow these steps:

1. If the origin of the structuring element coincides with a ‘0’ in the image, there is no change; move to the next pixel.
2. If the origin of the structuring element coincides with a ‘1’ in the image, and any of the ‘1’ pixels in the structuring element extend beyond the object (‘1’ pixels) in the image, then change the ‘1’ pixel in the image to a ‘0’.

In Figure 2.7, the only remaining pixels are those that coincide with the origin of the structuring element where the entire structuring element was contained in the existing object. Because the structuring element is 3 pixels wide, the 2-pixel-wide right leg of the image object was eroded away, but the 3-pixel-wide left leg retained some of its center pixels.

Opening consists of an erosion followed by a dilation and can be used to eliminate all pixel regions that are too small to contain the structuring element. In this case the structuring element is often called a probe, because it is probing the image looking for small objects to filter out of the image as seen in Figure 2.8.

Closing consists of a dilation followed by erosion and can be used to fill in holes and small gaps. The closing operation has the effect of filling in holes and closing gaps as seen in Figure 2.9.

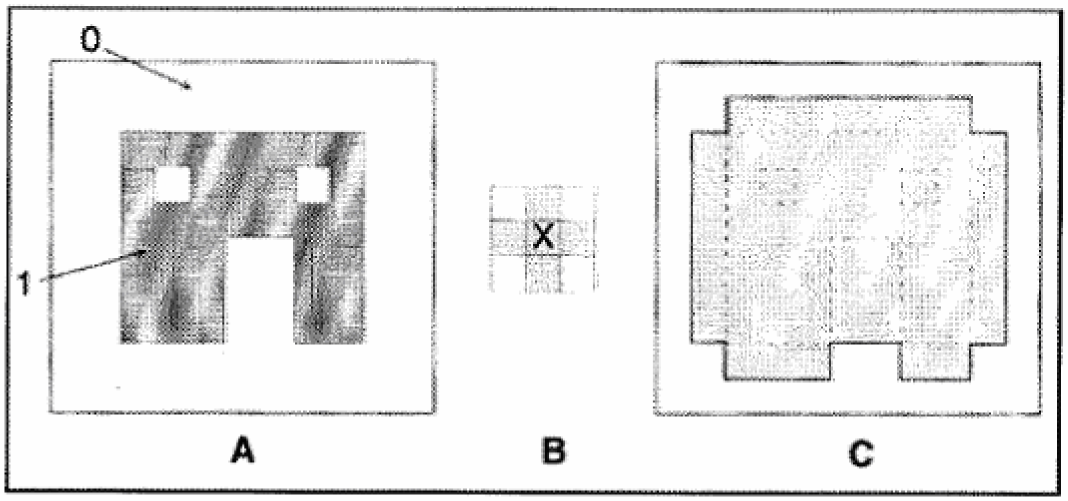


Figure 2.6. Dilation a) Original image b) Structural element; x=origin, c) Image after dilation; original in dashes

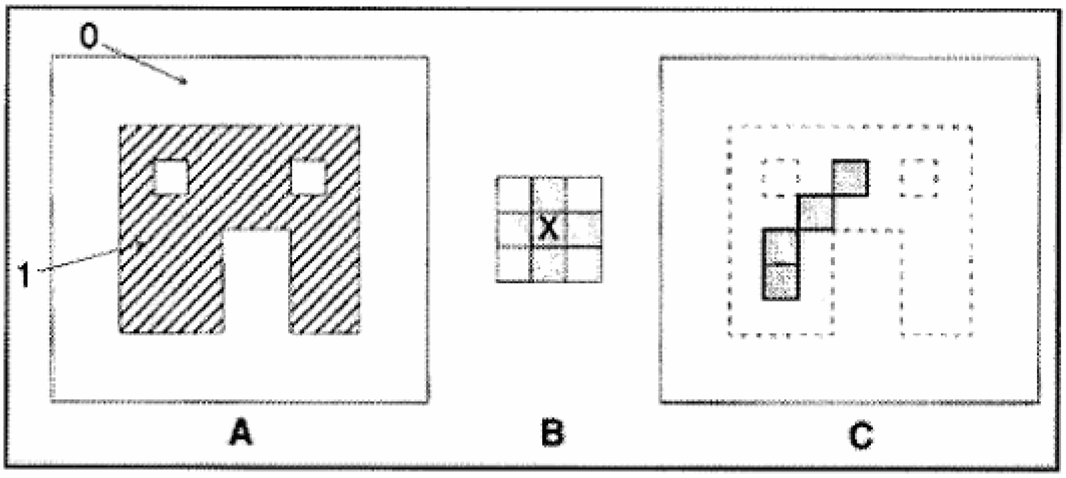


Figure 2.7. Erosion: a) Original Image, b) Structural element; x=origin, c) Image after erosion; original in dashes

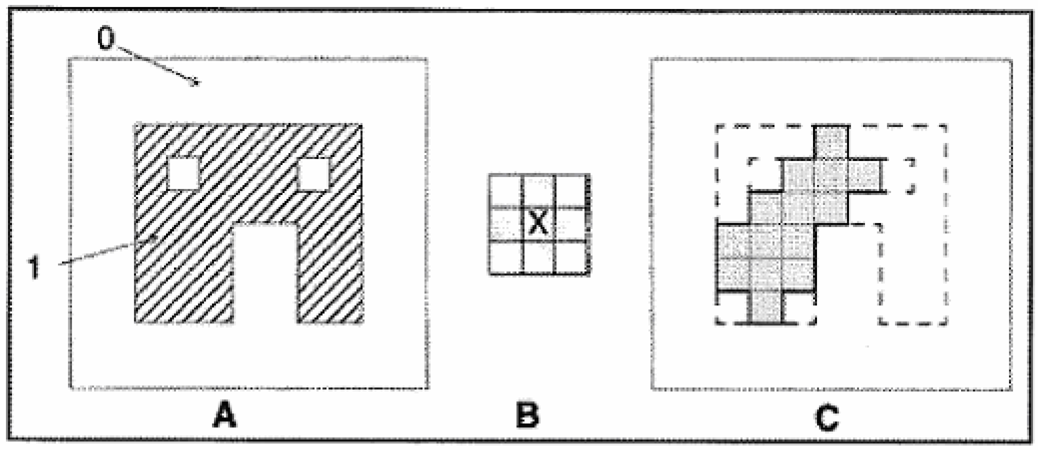


Figure 2.8. Opening a) Original Image, b) Structural element; x=origin, c) Image after opening; erosion followed by dilation

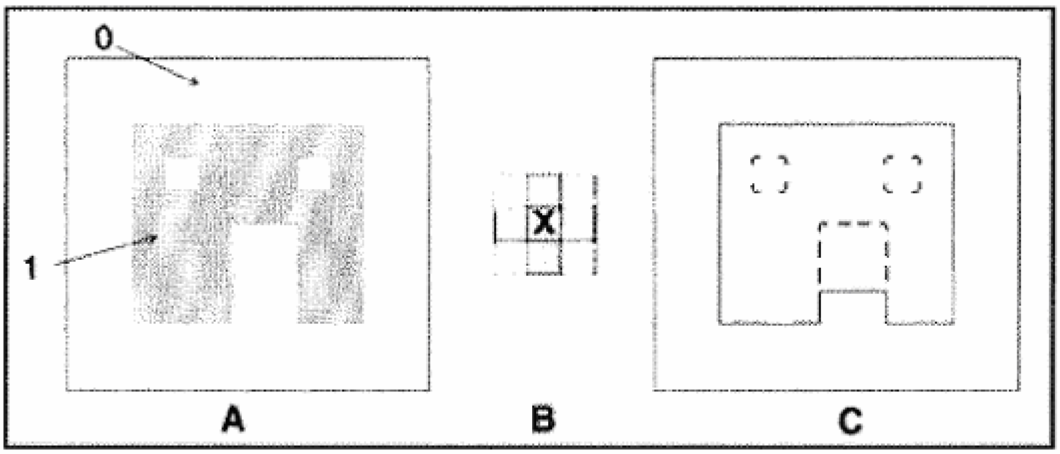


Figure 2.9. Closing: a) Original Image, b) Structural element; x=origin, c) Image after closing; dilation followed by erosion; original in dashes

2.3. Morphological Image Cleaning Filter

Many noise suppressing filters have been introduced using morphological operators by Strenberg (Strenberg, 1986), and Song and Delp (Song and Delp, 1990). Strenberg (Strenberg, 1986) proposed a new noise reduction algorithm using iterative application of opening and closing operators with successively larger structuring elements. Song and Delp (Song and Delp, 1990) devised generalized morphological filter which takes linear combinations of the results of openings and closings with multiple structuring elements. This filter works well in the presence of impulsive noise. However, in the presence of dense noise, there is a trade-off between noise smoothing and detail preservation. Moreover, there is no systematic approach for the exact specification of the structuring elements. Good results in different images probably depend on the choice of structuring elements.

The filters introduced by Strenberg, and Song- Delp remove features in the image if their support region does not cover the structuring element, and the features whose support regions cover the structuring element of opening and closing operators remain. To choose most favorable structuring element size is a big problem in these algorithms, because there is always a trade of between removing noise and preserving thin lines and small features in the image. The Strenberg and the Song-Delp delete small or thin features along with the noise. If there were some way to adapt a morphological smoothing filter so that it would not eliminate these features then it could, theoretically, meet both the goals for image cleaning. This idea, together with the observation that 1D object appear relatively insensitive to low levels of noise inspired the morphological image cleaning (MIC) algorithm (Peters, 1995).

2.3.1. The MIC Algorithm

The fundamental idea behind the morphological image-cleaning algorithm is to segment the image into features and noise, using the residual image that is the difference between an original image and a smoothed version using morphological opening and closing operators. The features from the residual are added back to the smoothed image. Ideally, this results in an image whose edges and other one dimensional features are as sharp as the original yet has smooth regions between them. Because morphological openings and closings do not blur some edge-like features, the algorithm uses them to smooth the image.

Morphology also provides the tools for differentiating between noise and features in the residual image. All the problems associated with image segmentation can affect this step. However, MIC performs well under relatively low levels of spatially dense noise, and the technique is suited for images where the standard deviation of the noise is less than half the standard deviation of the features (Peters, 1995).

2.3.1.1. Morphological Smoothing of Images

The OCCO filter is the pixel wise average of open-close and close-open of the image. Let I represent an image. Let I_Z represent the opening of the image with structuring element (SE) Z. Let I^Z represent the closing of the image with Z. The OCCO filter is calculated as:

$$\text{OCCO}(I; Z) = \frac{1}{2}(I_Z)^{Z} + \frac{1}{2}(I^Z)_{Z} \tag{3}$$

OCCO (I; Z) is approximately smooth with respect to the SE and, unlike its constituent operators, unbiased with respect to gray level. MIC uses an OCCO filter for smoothing. MIC's OCCO filter has two parameters associated with its structuring element, its size (i.e. diameter) d, and the choice of whether to use a flat or a curved top (i.e. function-set processing). It is suggested that slightly smoother results can be had using the curved SE.

The choice of SE size is of fundamental importance. In fact, no single size will do well. It is necessary to filter the image with a range of sizes and operate on the various residuals separately. It appears to be a common characteristic of the images for which this algorithm was designed that the visible noise is size limited. The variance of noise should be smaller than half of the variance of signal. That is, if one smoothes an image with successively larger OCCO filters, one of the smoothed images, for some SE diameter d, appears to be noise free whereas its predecessor (with a smaller SE diameter) shows evidence of noise. By "noise free" it is meant that large, constant or smoothly varying regions of the smoothed image show no small perturbations from the overall gray level trend of the region. A sufficiently large SE appears to filter all the noise out of the image. Thus, for a sufficiently large diameter SE, the residual difference image contains all the noise as well as all the features that were too small to be approximated by the structuring element.

To preserve the small features, it is necessary to segment the residual image into features and noise. However, for most images, the SE diameter d needed to smooth the image, is so large that the residual contains much of the important feature information. The more features there are in the residual image, the more difficult it is to segment. For a large enough structuring element, the residual image is the original image itself. In the residual, I minus a small diameter OCCO filtered image, the noise regions and feature regions are clearly identifiable and easily separable by the technique to be described later. But, the amount of noise present in the small SE OCCO image causes the resultant image to remain noisy. This noise is visible in smoother regions of the image. Thus, if the d is too large it is difficult to segment the residual image, and if the d is too small the algorithm fails to clean the image.

Finally, the best result has been obtained with smoothing the image with several OCCO filters of different diameters. The result is a morphological size distribution . Let S_j be the result of smoothing I with an OCCO filter with SE diameter d_j. Let $S_0 = I$. Assume $d_1 < d_2 < \cdots < d_k$ and that $d_0 = 1$ since the OCCO of I with $d_0 = 1$ is I itself. Then, $D_j = S_{j-1} - S_j$ is the jth residual image. D_j has, for the most part, no features larger than d_j and none smaller than d_{j-1}. That is, none of the features in D_j could cover the SE with diameter $d_{j\text{-}}1$. Each Dj is an intermediate residual image that constitutes a "size-band".

2.3.1.2. Processing of the Residuals

Each of the residuals D_j is assumed to contain both features and noise. The qualitative difference between D_j and S_{j-1} is that D_j is a "flattened out" S_{j-1}. The large amplitude regions of Dj indicate large variations in S_{j-1} from the behavior of S_j. Such regions are undoubtedly of more visual significance than the small amplitude regions. If the amplitude standard deviation of the noise in I is smaller than that of the features then it is reasonable to assume that this is true within the various size-bands. It is reasonable for white noise since its spatial frequency characteristic is flat.

(The assumption may not be true across all size-bands for images corrupted with patterned noise; in this case one size-band may be dominated by the noise, masking all features. But this is an advantage when it comes to removing patterned noise.) If the noise standard deviation is smaller than the amplitudes of the most important features on a given scale, it is possible to remove the noise from D_j by thresholding.

Because residual image D_j is the difference between two OCCO filter outputs, it is a signed image. The zero regions of D_j indicate areas where S_{j-1} and S_j coincide. The positive regions of D_j are the areas in S_{j-1} that are brighter than S_j. The negative regions are dark areas in S_{j-1} with respect to D_j. The light features may be perceptually more important than the dark features or vice-versa. Let T_j show the positive residual image called the "tophat" image, (that is the term normally used for the difference between an image and its opening by a flat topped structuring element).

$$T_j(x,y) = \begin{cases} D_j(x,y) & \text{if } D_j(x,y) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Call the negative residual image, B_j, the "bothat" image:

$$B_j(x,y) = \begin{cases} -D_j(x,y) & \text{if } D_j(x,y) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Note that B_j is a nonnegative image. MIC performs the same operation on both T_j and B_j using (possibly) different parameters. Let R_j represent either T_j or B_j . The goal of the processing of R_j is to segment it into feature regions and noise regions and then to discard the noise. Because of the assumptions about the characteristics of the noise and the important features in the image, and because of the way that R_j is constructed, the first step in the segmentation of R_j is to threshold it. Like many other algorithms, MIC computes thresholds from p_{Rj} (n), the graylevel probability distribution of R_j. Let m_{Rj} be the square root of the second moment of $p_{Rj}(n)$

$$m_{Rj} = \sqrt{\sum_{n=0}^{255} n^2 p_{Rj}(n)} \tag{6}$$

Experimentation has shown that thresholds

$$t = f \cdot m_{Rj} \tag{7}$$

where $f \in [1, 2]$ yield the best results for most images. The value of f is a free parameter in the MIC algorithm which defaults to one.

Let X_{Rj} be an image and t a threshold such that

$$X_{R_j}(x, y) = \begin{cases} 1 & \text{if } R_j(x, y) \geq t \\ 0 & \text{if } R_j(x, y) < t \end{cases} \tag{8}$$

Then X_{Rj}, is the support map of those pixels in R_j that are equal to or greater than the threshold. In general X_{Rj} will contain many isolated ones (ones surrounded by zeros in a 3 × 3 neighborhood) and many regions of only two adjoining ones. Isolated one or two pixel regions are more likely to be seen by a person as noise than features. Therefore the algorithm removes them. It does this as follows: First it applies a rank filter to X_{Rj} to find the locations of all 3×3 neighborhoods with 3 or more ones. Then it dilates the location map with a 3 × 3 square of ones and forms the logical AND of the result with X_{Rj} . The result of this is a support map of the pixels in R_j that are at least as large as the threshold and are near at least two other such pixels in a 3 × 3 neighborhood. There can be isolated pixels in this map. Consider the following 5 × 5 region:

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{9}$$

The 3×3 neighborhoods in each of the corners have four ones in it, yet the one in the center of the 5×5 region is isolated. Such isolated pixels are removed with a simple deletion operation. The isolated delete can cause some neighborhoods to have less than three ones. Thus, it is necessary to iterate the rank filtering and isolated deletions until there are no more isolated pixels. The minimum number of support pixels (i.e. 1's) inside each 3 × 3 neighborhood is another free parameter of the algorithm. It defaults to three pixels.

At this point, it would be possible to use X_{Rj} to select those pixels from R_j which correspond, presumably, to important features. These features could be added back to S_j to restore some of its detail. However, thresholding R_j trims off the "bases" of large features. When the features are added back to S_j, the trimming introduces distortion that was not present in the original image. Figure 2.10 is a 1D example of this (It uses opening and the tophat to simplify the demonstration of the problem. The distortion with the OCCO transform is analogous).

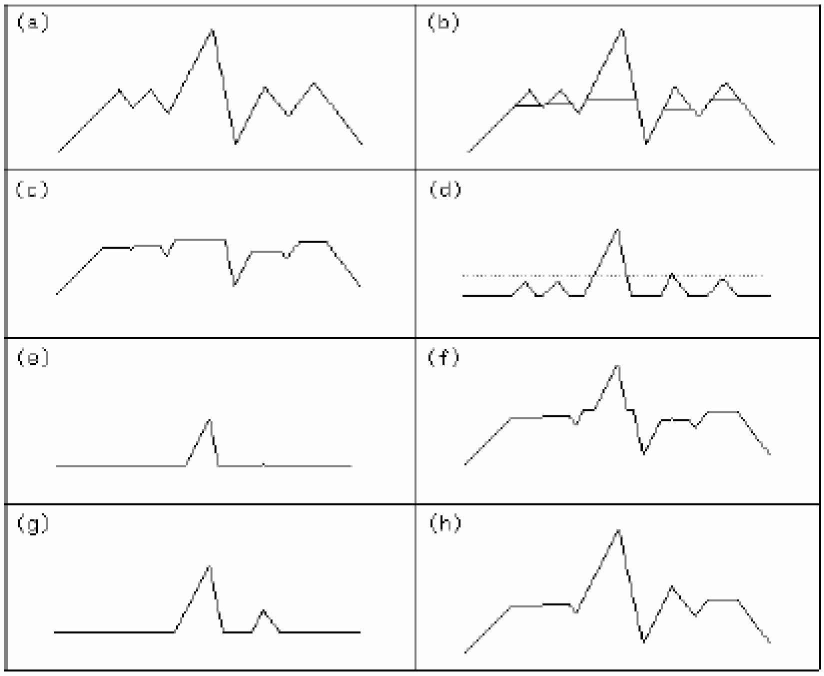


Figure 2.10. Recovery of the "base" of a center clipped residual (Peters, 1995)

(a) is the original 1D image.

(b) and (c) show the opening of (a) by a flat topped structuring element.

(b) shows the opening and the original superimposed.

(c) is the opening alone.

(d) is the tophat image showing the threshold (dotted line).

(e) is the thresholded tophat.

(f) shows the result of adding (e) directly to (c).

Notice the distortion of the large peak; essentially the middle of it has been removed. A way around this problem is to grow back the bases of those peaks which survived the thresholding procedure.

(g) is the thresholded tophat with the bases of the peaks restored.

(h) shows the sum of (g) and (c).

Note that the distortion of (f) is not present in (h). The large features are complete. To recover the bases of the features, MIC expands the nonzero regions of X_{Rj} through the processing of their morphological skeletons. The skeleton of a region is its medial axis. This dilates the skeleton with a binary structuring element of diameter d_j . This is a reasonable approach because of the characteristics of the size-band, D_j, from which X_{Rj} is computed. The largest feature in D_j must be smaller than the structuring element of radius d_j. Thus by dilating the skeleton with the structuring element of radius d_j, the algorithm finds the support of the largest possible area that could include both the peaks and their bases. This unavoidably reintroduces the support of some noise pixels. To complete the processing of the residual, the algorithm zeros any pixel in R_j whose support is not in the dilated skeleton. Finally, the algorithm recombines the information in the cleaned up residuals with the smooth image. Assume that the algorithm computed smooth images $S_1, \ldots, S_k$ using structuring elements with diameters $d_1, \ldots, d_k$ such that $d_1 < \cdots < d_k$. Assume it removed the noise from tophats, $T_1, \ldots, T_k$ and from bothats $B_1, \ldots, B_k$. The most coarsely smoothed image, namely S_k is the base image in the reconstruction. The bright features are put back in S_k by adding to it the sum of all the cleaned-up tophats. The dark features are put back in S_k by subtracting from it the sum of all the cleaned-up bothats. That is,

$$J = S_k + \sum_{j=1}^{k} \hat{T}_j - \sum_{j=1}^{k} \hat{B}_j \tag{10}$$

Where J is the cleaned image, $\hat{T}_j$ is the jth noise cleaned tophat and $\hat{B}_j$ is the jth noise cleaned bothat. Note that if nothing were done to the original T_j ’s and B_j ’s, then $J \equiv I$.

2.3.1.3. Formal Statement of the Algorithm

The following is a formal statement of the image cleaning algorithm. It is divided into three parts; a main routine and two subroutines. The main routine controls the decomposition of the image into size-bands, creates the residuals, sends them off to be cleaned, and recombines the results. The first subroutine is the image smoothing procedure. The second is the residual processor.

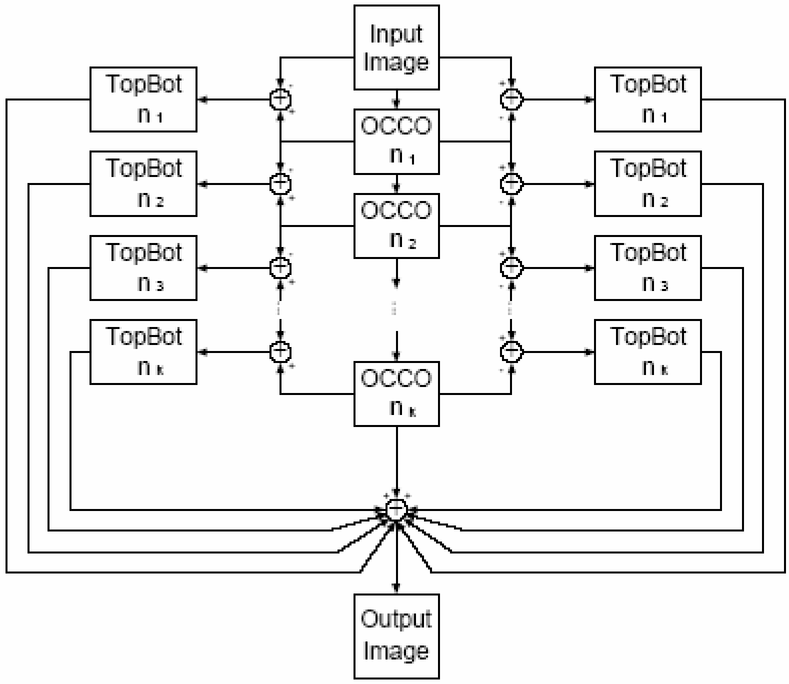


Figure 2.11. Block diagram of the main loop of the noise reduction algorithm (Peters, 1995)

2.3.1.3.(1). Algorithm 1 : Morphological Image Cleaning (MIC)

Let I be a grayscale image and let $\hat{I}$ be a cleaned version of I. Let S = OCCO(I; d) be the pixelwise average of the openclose of I and the closeopen of I where the morphological operations are performed with a structuring element with disk-shaped support of diameter d. OCCO(I; d) is computed by subroutine 1. Let T be the tophat of I with respect to S and let B be the bothat of I with respect to S. Let

$\hat{T}$ = TOPBOT(T; d, f, s) be the cleaned up tophat T, and let $\hat{B}$ = TOPBOT(B; d, f, s) be the cleaned up bothat B. Parameter f is a factor by which to multiply the square root of the second moment when calculating the threshold, and s is the minimum area per 3×3 neighborhood of a region in the cleaned up residual. TOPBOT () is computed by subroutine 2. Let T_{acc} be the accumulated cleaned tophat and let B_{acc} be the accumulated cleaned bothat. Let k be the number of size-bands to be computed. Let $\{d_1, . . ., d_k\}$ be integers such that $d_1 < \cdot \cdot \cdot < d_k$. Let j be an index variable. All additions and subtractions of images are pixelwise. The usual choices of parameters are $d_j = 2^{j+1}+1$, f = 1, s = 3. Occasionally, it is useful to specify one value of f for the tophat and another for the bothat. The same applies to s.

1. Initialize variables and images: $j \leftarrow 0$; $T_{acc} \leftarrow 0$; $B_{acc} \leftarrow 0$;
2. $j \leftarrow j + 1$;
3. $S \leftarrow OCCO (I; d_j)$;
4. for all pixels (x, y),

$$T(x, y) \leftarrow \begin{cases} I(x, y) - S(x, y) & \text{if } I(x, y) - S(x, y) > 0 \\ 0 & \text{if } I(x, y) - S(x, y) \leq 0; \end{cases}$$

5. $\hat{T} \leftarrow$ TOPBOT (T; dj, f, s);
6. $T_{acc} \leftarrow T_{acc} + \hat{T}$;

7. for all pixels (x, y),

$$B(x, y) \leftarrow \begin{cases} S(x, y) - I(x, y) & \text{if } S(x, y) - I(x, y) > 0 \\ 0 & \text{if } S(x, y) - I(x, y) \leq 0; \end{cases}$$

8. $\hat{B} \leftarrow$ TOPBOT (B; dj, f, s);
9. $B_{acc} \leftarrow B_{acc} + \hat{B}$;
10. $I \leftarrow S$;
11. If $j < k$ go to 2
12. $\hat{I} = T_{acc} + S - B_{acc}$;

2.3.1.3.(2). Algorithm 2: OCCO (I, d)

Let I, C, O, CO, OC, and S represent images. Let OPEN (I, d) and CLOSE (I, d) represent morphological opening and closing, respectively, using a structuring element with disk-shaped support of diameter d.

1. O ← OPEN (I, d); 2. OC ← CLOSE (O, d); 3. C ← CLOSE (I, d);

4. CO ← OPEN(C, d); 5. For all pixels (x, y), $S(x, y) = \frac{1}{2}\,[OC(x, y) + CO(x, y)]$;

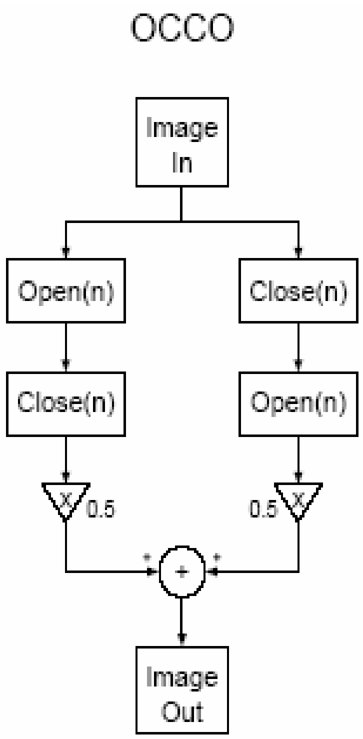


Figure 2.12. Block diagram of subroutine subroutine OCCO (Peters, 1995)

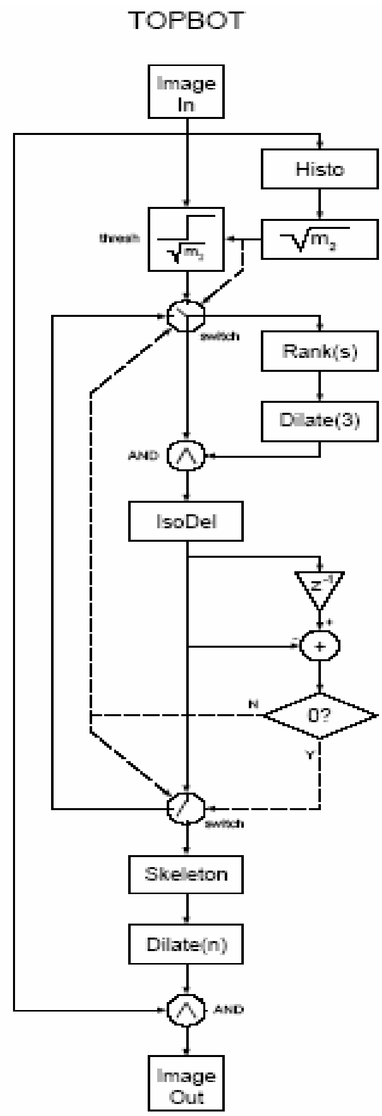


Figure 2.13. Block diagram of TOPBOT (Peters, 1995)

2.3.1.3.(3). Algorithm 3: TOPBOT (R, d, f, s)

Let R designate the residual image (either a tophat or a bothat). Let d be the diameter of the structuring element used to create the smooth image. Let histo(R) be the graylevel histogram of R. Let E $\{n^2\}$ be the second moment of the graylevel probability distribution of R. Let THRESH(R, t) represent the thresholding of R at t as defined in Eq. (7). X_i is a binary image mask. Parameter f is a real number in the interval [1, 2], and s is an integer in [1, . . . , 9]. RANK (X_i, s) is the rank filter of

order s for sets. (This returns a binary image X_{i+1} such that if $X_{i+1}(x, y) = 1$ then $X_i(x, y)$ is the center of a 3 × 3 neighborhood in X_i that contains at least s white pixels. If $X_{i+1}(x, y) = 0$ then the 3 × 3 neighborhood in X_i centered at (x, y) contains fewer than s pixels.) DILATE (X_i, d) is the dilation of (X_i) with a disk of radius d (when d = 3, the "disk" is a 3×3 square). AND is the pixelwise logical AND operator. ISODEL (X_i) removes isolated pixels. SKELETON (X_i) computes the morphological skeleton.

1. h ← histo(R);
2. m ← $\sqrt{E\{n^2\}}$ (calculated from h);
3. t ← f · m;
4. X_0 ← THRESH(R, t);
5. X_1 ← RANK(X_0, s)
6. X_2 ← DILATE(X_1, 3)
7. X_3 ← X_0 AND X_2
8. X_4 ← ISODEL(X_3)
9. If there were deletions in the last step, let X_0 ← X_4 and GOTO 5.
10. X_5 ← SKELETON(X_4)
11. X_6 ← DILATE(X_5, d)
12. $\hat{R}$ ← R AND X_6

Figure 2.11 is a block diagram of the main algorithm loop. Figures 2.12 and 2.13 have diagrams of the two subroutines (Peters, 1995).

3. IMAGE FUSION

Image combination from different sensors with different spatial and spectral resolutions, has become a significant technique in the field of digital remote sensing. Image fusion is the process of combining complementary information from multiple sources to obtain a new image which is more suitable for human and machine perception or further image processing (Ersoy, 2005).

Fusion can be done with two images having different spatial resolutions such as 20 meter resolution SPOT multispectral image with corresponding 10 meter resolution panchromatic image, the 30 meter resolution LANDSAT-TM multispectral images with then corresponding 15 meter resolution panchromatic images, TM image with SPOT panchromatic image and air photos (Qin and Liu, 2001), as well as JERS-1 SAR image with Landsat TM data (Rokhmatuloh, Tateishi, Wikantika, Munadi, and Aslam, 2003). In all cases, while the high spatial resolution panchromatic images provide better spatial characterization of the landscape, the low resolution multispectral images provide better characterization of the physical characteristics of the earth. Fusing these two types of images by forming new images integrates both the spectral aspects of the low resolution images and the spatial aspects of the high resolution images (Chen, Wang, Amani, 2004).

Image fusion can be performed at three different levels, namely pixel level, feature level, and decision level (Ersoy, 2005). We focused on pixel level in our work. The advantage of pixel fusion is that the images use the most original information (Parcharidis, Kazi-Tani, 2000).

The most commonly used image fusion techniques are Principal Components Analysis (PCA), Intensity-Hue-Saturation Transformation (IHS), High Pass Filter (HPF) and Wavelet transformation (WT) (Carper, Lillesand, and Kiefer, 1990).

IHS is the most widely used method for fusing remote sensing image data with different spatial resolutions. The details of the this transformation method will be described later.

PCA is a relevant method for merging remotely sensed imagery because PCA reduces the dimensionality of the original data from n to 2 or 3 transformed principal component images, which contains majority of information.

The HPF method can be used to combine two images with different resolutions but the result is sensitive to the filtering used (filtering type, filter window size, etc.) and the mathematical operation used. It is this difficult to adopt a standard procedure for all users and the results depend highly on each application.

The entire procedure of WT approach is based on a complex and sophisticated pyramidal transformation where the result also depends on the level of decomposition and the filtering technique used to construct the wavelet coefficients. The two methods, i.e. HPF and WT are still in the development stage(Beauchemin and Fung, 1999).

3.1. Intensity-Hue Saturation Transformation

RGB representation is far from the human concept of color. In the RGB space, color features are highly correlated, and it is impossible to evaluate the similarity of two colors from their distance in this space. Various kinds of color features can be calculated from components R, G and B. Each color feature has its own characteristics. In remote sensing, the IHS colour transform has been widely used to display information.

IHS is a color mode in which H, S, and I denote hue, saturation and intensity, respectively. The process of transforming the color image from red green blue space to the intensity, hue, and saturation space is called IHS transformation (Qin and Liu, 2001). IHS transformations permit the separation of spatial information as an intensity component from the spectral information in the hue and saturation components of a three-color composite image. The analysist is than able to manipulate independently the spatial information while maintaining the overall color balance of the original scene (Beauchemin and Fung, 1999).

Many IHS transformation algorithms have been developed for RGB tristimulus values into the parameters of human color perception and vice versa.

One of the IHS transformation algorithms is given below. To map the input values into the IHS space, RGB values are first converted into the Y, C_1, and C_2 tristimulus values (Carron and Lambert, 1994):

$$\begin{pmatrix} Y \\ C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1 & -1/2 & -1/2 \\ 0 & -\sqrt{3}/2 & \sqrt{3}/2 \end{pmatrix} . \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{11}$$

Tristimulus values Y, C_1, and C_2, are then transformed into IHS coordinates by means of the following equations:

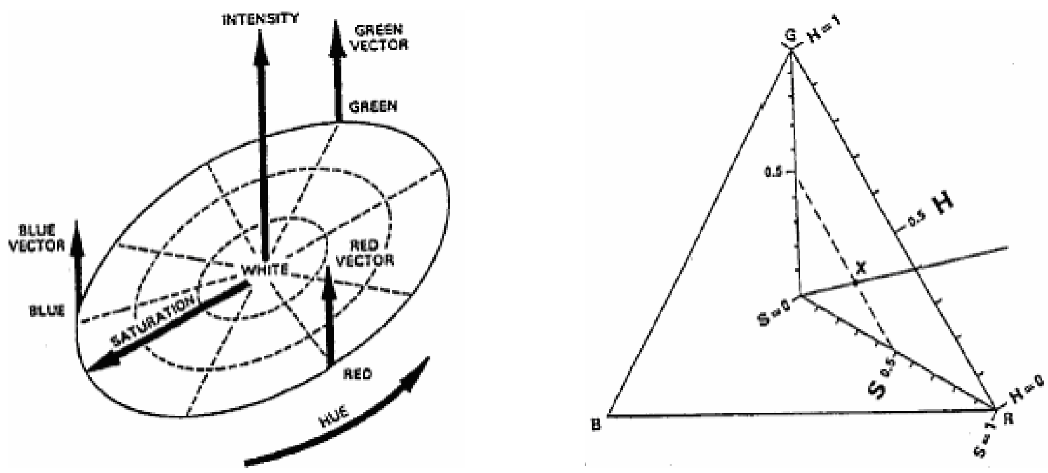
- $I = Y$ (12)
- $S = \sqrt{C_1^2 + C_2^2}$ (13)
- If $C_2 \geq 0$ then $H = \cos^{-1}(C_2/S)$ (14)

  else $H = 2\pi - \cos^{-1}(C_2/S)$

Intensity, hue and saturation refer to parameters of human color perception. “Intensity” refers to the total brightness of a color. “Hue” generally refers to the dominant or average wavelength of light contributing to a color. When applied to data displayed on an RGB monitor, hue can be described on a circular scale progressing from red to green to blue and back to red. A pixel’s hue is determined by the relative proportion of its red, green and blue inputs. “Saturation” specifies the purity of a color relative to gray. Vivid colors are highly saturated while pale, pastel colors have low saturation.

The intensity of an RGB color is a function of the magnitude of the input primaries. Some algorithms define an RGB color's intensity as the magnitude of the largest input.

Calculation of a pixel's intensity, hue, and saturation necessitates the definition of a color space in which the relationships between the RGB and IHS coordinates of a color are known, as seen Figure 3.1.(a) Smith's "Triangle" model (Figure3.1.(b)) was used for the SPOT data mergers (Carper, Lillesand, and Kiefer, 1990).



(a) RGB and IHS coordinate systems **(b)** Simplified IHS representation

Figure 3.1. IHS coordinate system

In image fusion, IHS concept is based on the representation of low-resolution multispectral images in the IHS system and then substituting the I component with the high spatial resolution image. The intensity component-the sum of the bands- is replaced with a stretched higher spatial resolution value, and this is followed by performing an inverse IHS transform. An inverse IHS transformation produces a new high resolution multispectral image. The fused image has abundant information, and also preserves high spatial resolution. (Qin and Lui, 2001). The IHS transformation used in this thesis is given in Section 5.2.

3.2. Comparing Image Fusion and RGB2NTSC Algorithms

In this work we developed a new algorithm RGB2NTSC (Red Green Blue to Intensity Hue Saturation). The details of the algorithm will be described in section 5.1. Here, the differences between RGB2NTSC and previous image fusion algorithms are explained shortly.

Figure 3.2 shows an example image fusion algorithm. There are images from different sources. For example, band1, band2, and band3 are multispecral images from Landsat; Pan is panchromatic image from SPOT. First, the three bands are transformed with IHS transformation. Obtained Intensity value is replaced with PAN image and the inverse IHS transformation is applied. The new image integrates both the spectral aspects of the low resolution images (MS) and spatial aspects of the high resolution images.

Figure 3.3 shows one loop of the RGB2NTSC algorithm, the same process is repeated for all three composite images. In this algorithm, we just used images from one source (Landsat MS). Similarly, we applied IHS transformation to three band composite images of MS. Then we filtered the I, H, and S values separately. Then, the inverse IHS transformation is applied to obtain new band values. The final procedure is to average all the images for each band. This is very effective since there are many images obtained for each band in multispectral and hyperspectral images.

RGB2NTSC use image transform technique to involve the manipulation of multiple bands of the same image, whereas image fusion is combine complementary information from multiple sources.

The RGB2NTSC algorithm and also the RDFT algorithm discussed in Chapter 5 have been applied to multispectral images so far. Fusion of information from different bands is inherent in these algorithms.

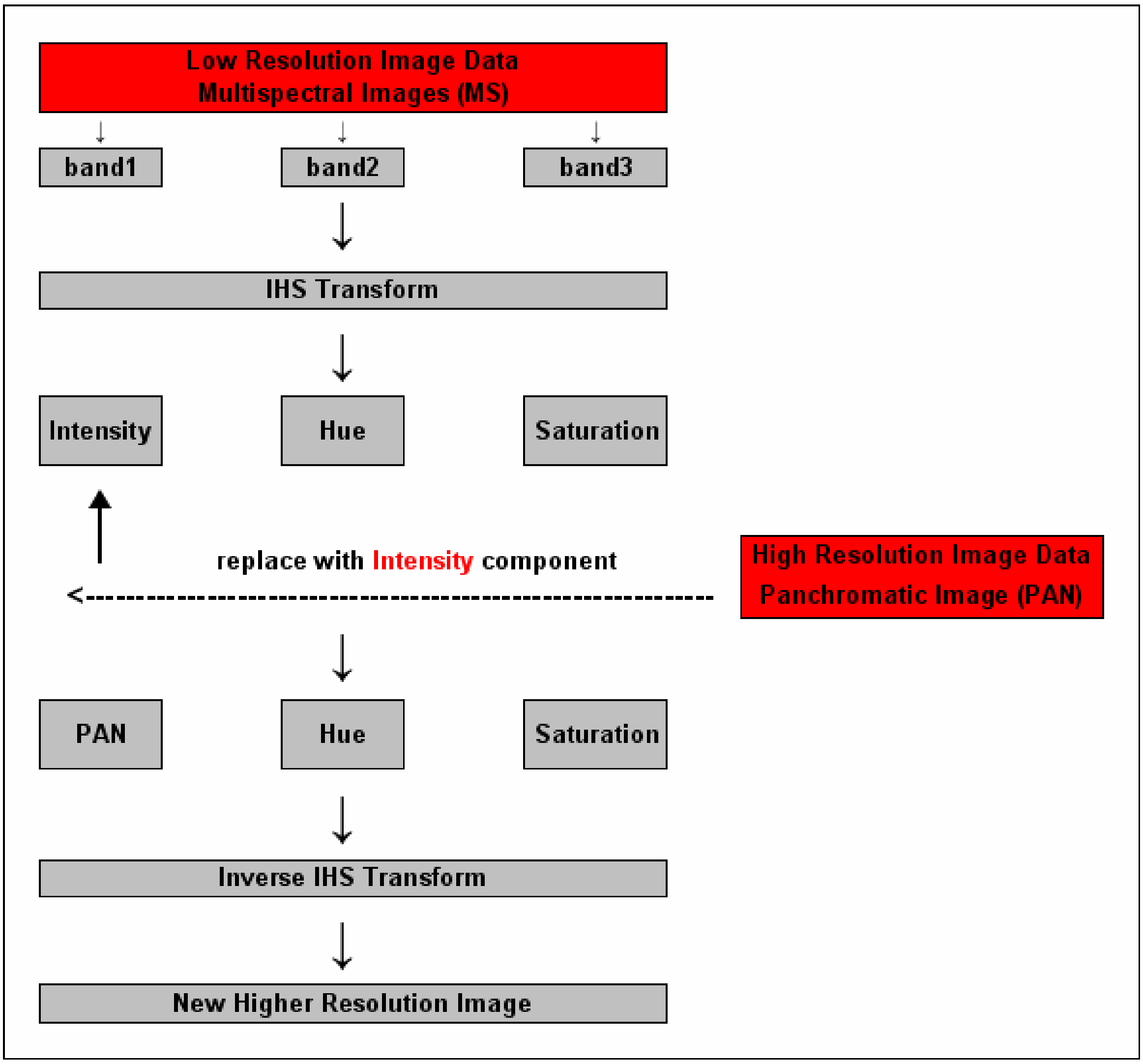


Figure 3.2. Image Fusion Algorithm

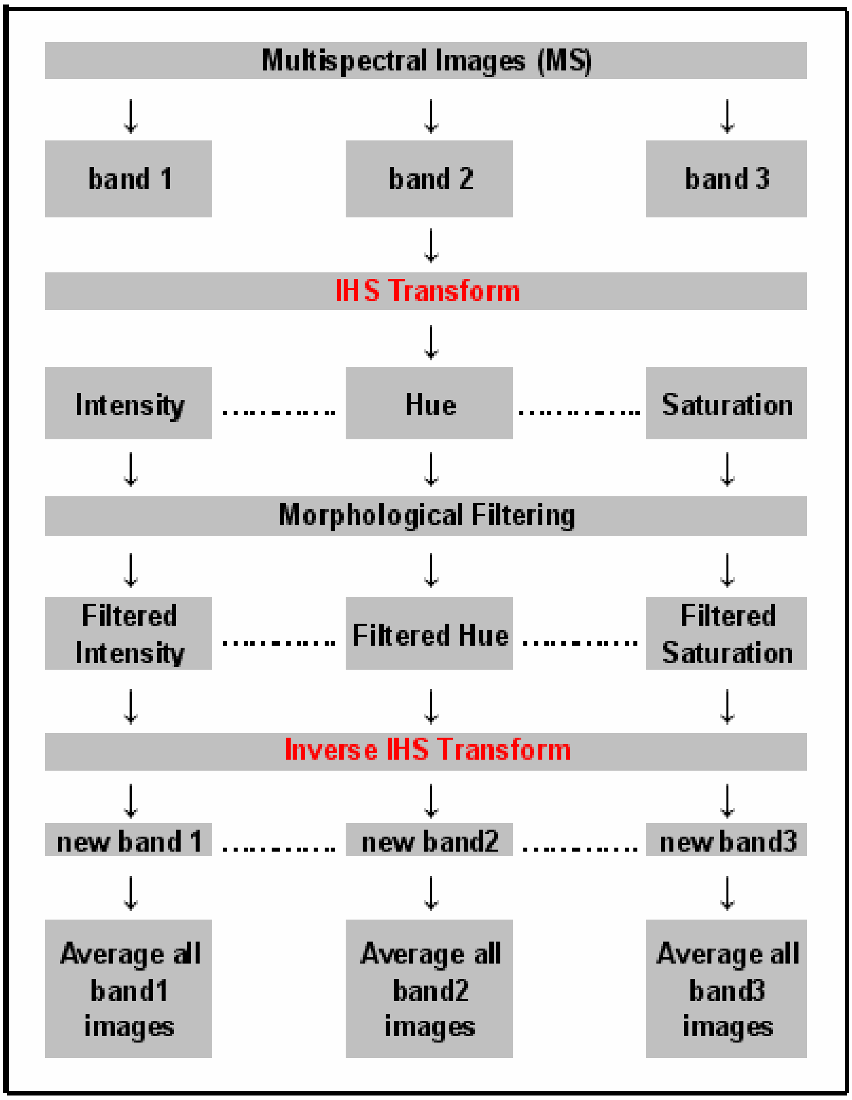


Figure 3.3. One loop of the RGB2NTSC Algorithm

4. REAL DISCRETE FOURIER TRANSFORM

4.1. Fourier Transform

The Fourier transform (FT) is a generalization of the Fourier series. The Fourier transform is usually represented with complex exponentials (Fourier Transforms, DFTs, and FFTs, 2003). Fourier-related transforms are of central importance in diverse applications of science, engineering, and technology (Ersoy, 1994).

For a signal or function $\chi(t)$, $-\infty \leq t \ \infty$, the complex Fourier transform (CFT) is defined as;

$$X_c(f) = \int_{-\infty}^{\infty} \chi(t) e^{-j2\pi ft} dt \tag{15}$$

The inverse complex Fourier transform (ICFT) is given by

$$\chi(t) = \int_{-\infty}^{\infty} X_c(t) e^{j2\pi ft} df \tag{16}$$

The real Fourier transform (RFT) of $\chi(t)$ can be defined as (Ersoy, 1994)

$$X(f) = 2\int_{-\infty}^{\infty} \chi(t)\cos(2\pi ft + \theta(f))dt \tag{17}$$

where

$$\theta(f) = \begin{cases} 0 & f \geq 0 \\ \pi/2 & f < 0 \end{cases} \tag{18}$$

The inverse real Fourier transform (IRFT) is given by

$$\chi(\mathrm{t}) = 2\int_{-\infty}^{\infty} X(\mathrm{f})\cos(2\pi\mathrm{ft} + \theta(\mathrm{f}))\mathrm{df} \tag{19}$$

4.2. The Discrete Fourier Transforms

Consider a possibly complex series χ (n) with N samples of the form $x_0, x_1, x_2, x_3, ..., x_n, ..., x_{N-1}$.

Further, assume that that the series outside the range 0, N-1 is extended N-periodic, that is, χ (n+N) = χ (n), for all n (Bourke, 1993).

The orthonormalized complex discrete Fourier transform (CDFT) of x(n) is defined as

$$X_c(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \chi(k) e^{-2\pi jnk/N} \tag{20}$$

The inverse CDFT is given by

$$\chi(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_c(k) e^{2\pi jnk/N} \tag{21}$$

4.3. Real Discrete Fourier Transform

The real discrete Fourier transform (RDFT) corresponds to the real Fourier series for sampled periodic signals with sampled periodic frequency responses whereas the discrete Fourier transform (DFT) corresponds to the complex Fourier series for the same type of signals. The DFT yields complex results. When the input data is real, the output actually has complex conjugate symmetry. Thus, half of the output data is redundant. This means twice as many computations are generated. The real DFT (RDFT) has real output for real input and totally avoids redundancy.

The orthonormalized real discrete Fourier transform (RDFT) is defined as (Ersoy, 1994)

$$\frac{X(n)}{\upsilon(n)} = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} \chi(k) \cos\left(\frac{2\pi nk}{N} + \theta(n)\right) \tag{21}$$

where

$$v(n) = \begin{cases} 1/\sqrt{2} & n = 0 \text{ or } N/2 \\ 1 & \text{otherwise} \end{cases} \tag{22}$$

$$\theta(n) = \begin{cases} 0 & 0 \,£\, n \,£\, N_1 \\ \pi/2 & N_1 < n < N \end{cases} \tag{23}$$

N_1 equals N/2 if N is even and (N - 1)/2 if N is odd. The inverse RDFT is given by

$$\chi(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} X(k) v(k) \cos\left(\frac{2\pi nk}{N} + \theta(k)\right) \tag{24}$$

Equations (21) and (24) can also be written for $0 \leq n \leq N_1$ as

$$\frac{X_1(n)}{v(n)} = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} \chi(k) \cos\left(\frac{2\pi nk}{N}\right) \tag{25}$$

$$X_0(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} \chi(k) \sin\left(\frac{2\pi nk}{N}\right) \tag{26}$$

and

$$\chi(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N_1} [X_1(k)v(k)\cos\left(\frac{2\pi nk}{N}\right) + X_0(k)\sin\left(\frac{2\pi nk}{N}\right)] \tag{27}$$

Thus X (n) equals $X_1(n)$ for $0 \le n \le N_1$, and $X_0(N-n)$ for $N_1 < n < N$. This ordering of X (n) gives the best results in various applications.

The RDFT has better performance than the DFT in data compression and filtering. The RDFT also has better performance than the DFT in the computation of real convolution because of the reduced number of operations, and the fact that forward and inverse transforms can be implemented with the same signal flow graph, thereby facilitating hardware and software design (Ersoy, 1985).

There are other real discrete transforms as well, such as the discrete cosine transform. They are often used in multidimensional applications such as image processing (Ersoy, 1994).

5. NEW TRANSFORM TECHNIQUES

5.1. RGB2NTSC Transform Algorithm

Computers often use RGB (Red, Green, and Blue) output to create color images. In an RGB display, all of the colors that make up an image are made up of a combination of red, green, and blue at varying levels of intensity, each ranging from 0-255. Each unique color has its own combination of red, green, and blue levels. With all of the possible combinations of red, green, and blue values, this provides for a display system capable of using millions of different colors.

The way the multispectral data are mapped into three colors in the output image depends on the information that one wishes to highlight in the images. The spectral characteristics of the target being observed and the type of information a researcher hopes to extract from the raw data determine which bands will be used in the composite and which color (red, green, or blue) will be assigned to each band. For example, in some applications, it may be desirable that land cover classes be associated with familiar colors (e.g., grass is green). In other cases contrasting colors are preferred to highlight objects of interest from the background.

Regardless of the combination of bands used, the mapping of multispectral sensor data to the RGB color display is the same. Three bands are selected, each is assigned to one of the three primary RGB colors, and the value of each color level is mapped to the measured value of each pixel in the appropriate band. For example, to create a composite image that maps the measured values of multispectral bands 3, 2, and 1 to the colors red, green, and blue respectively, the color of each pixel would be calculated using the following rules:

- The red value of the pixel would be set equal to the measured energy level of that pixel in band 3,
- The green value of the pixel would be set equal to the measured energy level of that pixel in band 2,
- The blue value of the pixel would be set equal to the measured energy level of that pixel in band 1 (Pfister, 2004).

The commonly used RGB color space is not suitable for a merging process, as the correlation of the image channels is not clearly emphasized. The IHS system offers an advantage in that the separate channels outline certain color properties, namely intensity I, hue H, and saturation S. Hue (H), Saturation (S) and Intensity (I) are more suited to the segmentation processing of color images. IHS transformations permit the separation of spatial information as an intensity component from the spectral information in the hue and saturation components of a three-color composite image (Beauchemin and Fung, 1999).

We illustrate the multispectral case below, assuming the number of bands N equals 7 and a Landsat remote sensing image is used. The main processing steps of the algorithm are as follows:

1) Input the set of images.
2) Constitute 3 band color composite image combinations of seven bands (1,2,3); (1,2,4); (2,3,4);...(5,6,7).
3) Transform each combination, using the RGB2NTSC transformation.
4) Filter transform domain images, for example, using morphological filtering.
5) Inverse transform.
6) Calculate average value of each band, using all the images of each band.

Figure 5.1 shows RGB2NTSC Algorithm.

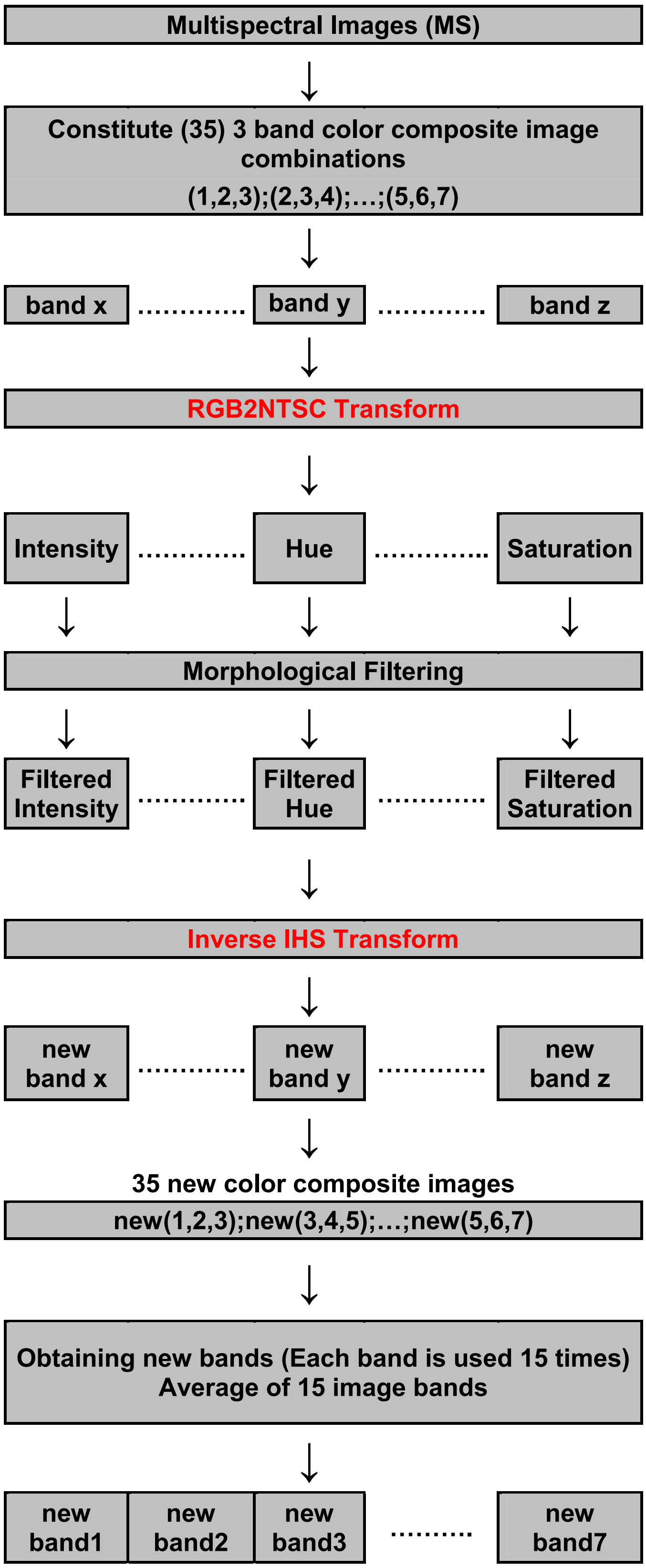


Figure 5. 1. RGB2NTSC Algorithm with seven LANDSAT images

The details of each step are further given below.

--In the first step, 3 band color composite images of seven band multispectral Landsat image are obtained.

When N=7, each band is used 15 times and 35 3-band color composite images exist. For example, Band1 and its combinations with other bands are as follows:

(1,2,3), (1,2,4), (1,2,5), (1,2,6), (1,2,7), (1,3,4), (1,3,5), (1,3,6), (1,3,7), (1,4,5), (1,4,6), (1,4,7), (1,5,6), (1,5,7), (1,6,7)

--Then, each composite image is transformed from RGB format to IHS format. We assign R = band3, G = band2, B = band1.

The RGB➔IHS transformation matrix is given by

$$TM = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ 0,299 & -0,274 & -0,322 \\ 0,211 & -0,523 & 0,312 \end{pmatrix} \tag{28}$$

e.g. , (band1, band2, band3)➔

[(0.299Band1+ 0.587Band2 + 0.114Band3), Intensity

(0.596 Band1 + (- 0.274) Band2 + (-0.322) Band3), Hue

(0.211Band1 + (-0.523)Band2 + 0.312Band3)], Saturation

--Next, morphological filtering is applied to hue, saturation and luminance (intensity) images. The main idea behind the implementation of the morphological filter algorithm is to identify the features and noise of an image by first finding the residual image (which is the difference between an original image and a smoothened image) and separating the features from the noise and recombining it with the original image to get a noise free image. These thin features are very important to perceive the quality of the image by humans. Once the image has been smoothened, morphology is again re-utilized as an important tool in differentiating between noise and thin features in the residual image (Peters, 1995).

--We define the following:

FIV = Morphologic filter (Intensity)

FHV= Morphologic filter (Hue)

FSV= Morphologic filter (Saturation)

(FIV, FHV, FSV) denotes the image in the filtered domain.

--IHS to RGB conversion: the modified IHS data set (hue, saturation, and intensity bands) are inverse transformed into red, green and blue bands. The inverse transform matrix is given by

$$\text{ITM} = \begin{pmatrix} 1 & 0{,}9560 & 0{,}6210 \\ 1 & -0{,}2720 & -0{,}6470 \\ 1 & -1{,}1060 & 1{,}7030 \end{pmatrix} \tag{29}$$

--At the end of each inverse transform, new band values are obtained.

--The same process is repeated for all three band color composite images (35 times). In this process, each band is used 15 times with other bands.

The new image band is calculated as the average of 15 image bands in the case of 7-bands LANDSAT images:

$$\text{band1\_final} = \frac{\text{band1\_new}_1 + \text{band1\_new}_2 + \ldots + \text{band1\_new}_{15}}{15} \tag{30}$$

5.2. Real Discrete Fourier Transform (RDFT) Algorithm

The RGB2NTSC method allows a transformation to work only with three band color composite images whereas the RDFT permits any number of band composites (three, four…). When we compare the results between the RGB2NTSC and RDFT algorithm, the best results were also obtained with the RDFT transform.

The main processing steps of the algorithm are the same as in the RGB2NTSC algorithm. The major difference is the different transform which allows combining any number of bands. The number of possible combinations per band is given by

$$L = \binom{N-1}{M-1} = \frac{(N-1)!}{(M-1)!\big((N-1)-(M-1)\big)!} \tag{31}$$

where M denotes the transform matrix (TM) size. The number of possible combinations per band as a function of transform matrix size is shown in Table 5.1.

Table 5. 1. Number of Combinations per Band

TM Size	3X3	4X4	5X5	6X6	7X7
Number of possible combinations per band (L)	15	20	15	6	1

5.3. Experimental Results

5.3.1. RGB2NTSC Test Results

Our test data include different images from Landsat Thematic Mapper (TM) satellite data. The Landsat TM satellite data have a spatial resolution of 30m and include seven spectral bands. The Landsat TM sensor includes one near-infrared band, two middle-infrared bands, and one thermal infrared band. The specific wavelengths are shown in Table II (Geo InSight International, 1999).

The new enhancement method was applied first to the “mississippi.lan” image shown in Figure 5.2. Each image of ‘mississippi.lan’ (Matlab 7.0, 2004) has 512 lines and 512 columns, with a total number of 1,835,008 pixels and 7 bands (Geo InSight International, 1999). When we first applied the morphological filter, we observed the resulting image had a lot of distortions, and lost edges as seen in

Figure 5.3. With the new enhancement method, the distortions were prevented, and the edges were preserved, as seen in Figure 5.4.

Table 5. 2. Bands in Landsat Thematic Mapper Satellite Data

Band	Wavelength in micrometers (µm)
Blue	0.45- 0.52
Green	0.52-0.60
Red	0.63-0.69
Near-IR	0.76-0.90
Mid-IR	1.55-1.75
Thermal-IR	10.4-12.5
Mid-IR	2.08-2.35

Figure 5. 2. Original Landsat TM image

Figure 5. 3. The image obtained with Morphologic filtering

Figure 5. 4. The image obtained with the RGB2NTSC algorithm

Below, we used the landsat images "newyork.lan", "rio.lan" and Landsat TM image of İstanbul, Turkey. The data for these images have 512 lines and 512 columns, with a total number of 1,835,008 pixels and 7 bands. In these applications, there is no significiant difference between the original image and the morphological filtered image. But the details of some regions become more clear with the RGB2NTSC method as seen in Figure 5.5-5.12.

Figure 5. 5. Original New York image

Figure 5. 6. The image obtained with the RGB2NTSC Algorithm

Figure 5. 7. The image obtained with Morphologic filtering

Figure 5. 8. Original RIO image

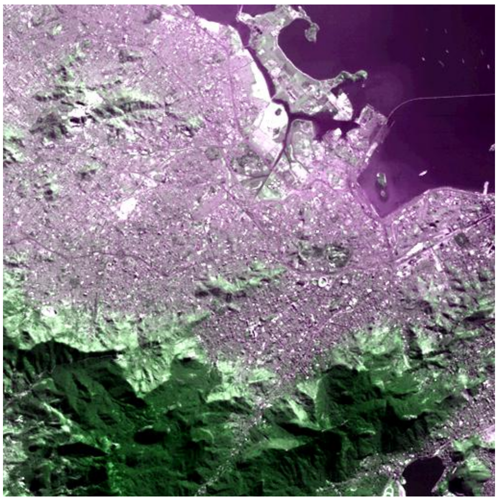


Figure 5. 9. The image obtained with the RGB2NTSC Algorithm

Figure 5. 10. The image obtained with Morphologic filtering

Figure 5. 11. Original Landsat TM image

Figure 5. 12. The image obtained with RGB2NTSC

5.3.2. RDFT Test Results

The RDFT method was first applied to the "mississippi.lan" image. RDFT transform matrices were applied with M band color composite images. Each band was regenerated by averaging L images, where L is given in Table 5.1. The resulting images are shown in Figure 5.13.

From Figure 5.13, we conclude the following:

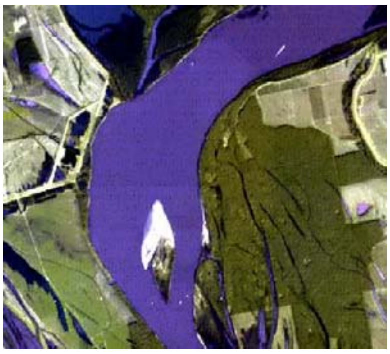
--The best result was obtained with the 4X4 RDFT transform, which correlates well with the fact that its number of combinations per band is maximum, namely 20 images per band.

--The 3X3 RDFT transform and 5X5 RDFT transform gives similar results, where the number of combinations per band is 15.

--The image obtained with the 7X7 RDFT transform includes some distortions like the morphological filter result, especially in the river region. This also correlates well with the fact that the number of combinations per band is 1, and hence there is no averaging.

(a) Original image

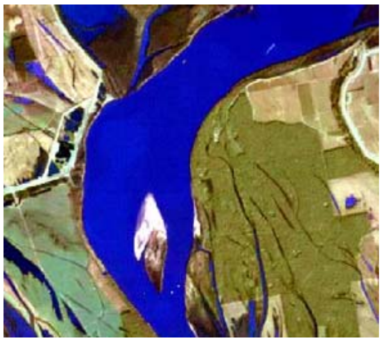
(b) 3X3 transform result



(c) 4X4 transform result

(d) 5X5 transform result

(e) 6X6 transform result



(f) 7X7 transform result

Figure 5. 13. The images obtained with the RDFT Algorithm

5.3.2.1. 3X3 RDFT Processing

RDFT transform matrix is applied to 3 band color composite images. Each band is regenerated 15 times (Table 5.1).

- The transform matrix (TM) for RDFT is given by

$$TM = 1/\sqrt{3}\begin{pmatrix} 1 & 1 & 1 \\ \sqrt{2} & -1/\sqrt{2} & -1/\sqrt{2} \\ 0 & -\sqrt{3/2} & \sqrt{3/2} \end{pmatrix} \tag{32}$$

- The inverse transform matrix (ITM) is given by

$$ITM = 1/\sqrt{3}\begin{pmatrix} 1 & \sqrt{2} & 0 \\ 1 & -1/\sqrt{2} & -\sqrt{3/2} \\ 1 & -1/\sqrt{2} & \sqrt{3/2} \end{pmatrix} \tag{33}$$

Figure 5. 14. The image obtained with the RDFT Algorithm (3X3 Transform Matrix)

5.3.2.2. 4X4 RDFT Processing

RDFT transform matrix is applied to 4 band color composite images. Each band is regenerated 20 times (Table 5.1).

- The transform matrix (TM) for RDFT is given by

$$TM = \begin{pmatrix} 0,5 & 0,5 & 0,5 & 0,5 \\ 0,7071 & 0 & -0,7071 & 0 \\ 0,5 & -0,5 & 0,5 & -0,5 \\ 0 & 0,7071 & 0 & -0,7071 \end{pmatrix} \tag{34}$$

- The inverse transform matrix (ITM) is given by

$$ITM = \begin{pmatrix} 0,5 & 0,7071 & 0,5 & 0 \\ 0,5 & 0 & -0,5 & 0,7071 \\ 0,5 & -0,7071 & 0,5 & 0 \\ 0,5 & 0 & -0,5 & -0,7071 \end{pmatrix} \tag{35}$$

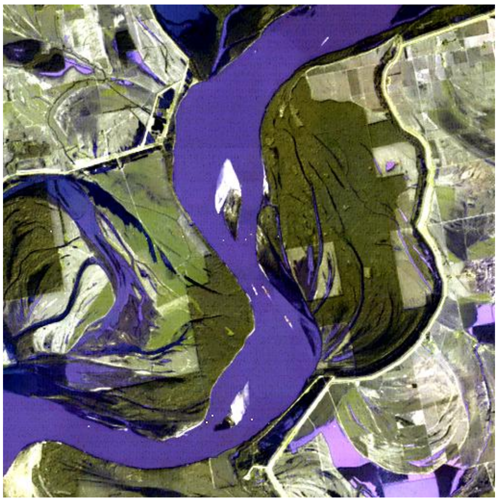


Figure 5. 15. The image obtained with the RDFT Algorithm (4X4 Transform Matrix)

5.3.2.3. 5X5 RDFT Processing

RDFT transform matrix is applied to 5 band color composite images. Each band is regenerated 15 times (Table 5.1).

- The transform matrix (TM) for RDFT is given by

$$TM = \begin{pmatrix} 0,4472 & 0,4472 & 0,4472 & 0,4472 & 0,4472 \\ 0.6325 & 0.1954 & -0.5117 & -0.5117 & 0.1954 \\ 0.6325 & -0.5117 & 0.1954 & 0.1954 & -0.5117 \\ 0 & 0.3717 & -0.6015 & 0.6015 & -0.3717 \\ 0 & 0.6015 & 0.3717 & -0.3717 & -0.6015 \end{pmatrix} \quad (36)$$

- The inverse transform matrix (ITM) is given by

$$ITM = \begin{pmatrix} 0,4472 & 0.6325 & 0.6325 & 0 & 0 \\ 0,4472 & 0.1954 & -0.5117 & 0.3717 & 0.6015 \\ 0,4472 & -0.5117 & 0.1954 & -0.6015 & 0.3717 \\ 0,4472 & -0.5117 & 0.1954 & 0.6015 & -0.3717 \\ 0,4472 & 0.1954 & -0.5117 & -0.3717 & -0.6015 \end{pmatrix} \quad (37)$$

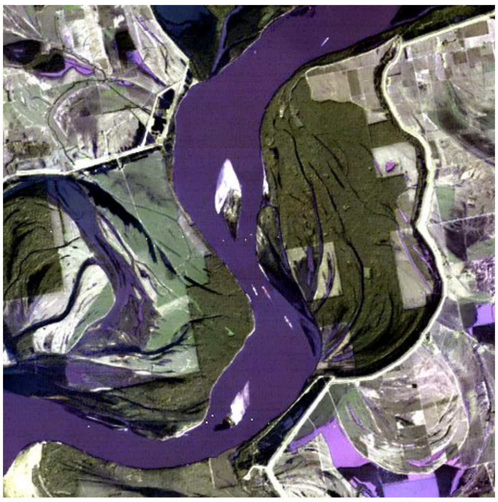


Figure 5. 16. The image obtained with the RDFT Algorithm (5X5 Transform Matrix)

5.3.2.4. 6X6 RDFT PROCESSING

RDFT transform matrix is applied to 6 band color composite images. Each band is regenerated 6 times (Table 5.1)

- The transform matrix (TM) for RDFT is given by

$$TM = \begin{pmatrix} 0,4082 & 0,4082 & 0,4082 & 0,4082 & 0,4082 & 0,4082 \\ 0,5774 & 0,2887 & -0,2887 & 0,5774 & -0,2887 & 0,2887 \\ 0,5774 & -0,2887 & -0,2887 & 0,5774 & -0,2887 & -0,2887 \\ 0,4082 & -0,4082 & 0,4082 & 0 & 0,4082 & 0 \\ 0 & 0,5 & -0,5 & 0 & 0,5 & -0,5 \\ 0 & 0,5 & 0,5 & -0,5774 & -0,5 & -0,5 \end{pmatrix} \quad (38)$$

- The inverse transform matrix (ITM) is given by

$$ITM = \begin{pmatrix} 0,4082 & 0,5774 & 0,5774 & 0,4082 & 0 & 0 \\ 0,4082 & 0,2887 & -0,2887 & -0,4082 & 0,5 & 0,5 \\ 0,4082 & -0,2887 & -0,2887 & 0,4082 & -0,5 & 0,5 \\ 0,4082 & -0,5774 & 0,5774 & 0 & 0 & -0,5774 \\ 0,4082 & -0,2887 & -0,2887 & 0,4082 & 0,5 & -0,5 \\ 0,4082 & 0,2887 & -0,2887 & 0 & -0,5 & -0,5 \end{pmatrix} \quad (39)$$

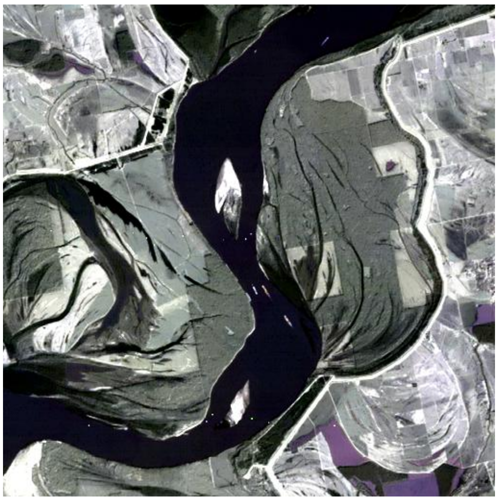


Figure 5. 17. The image obtained with the RDFT Algorithm (6X6 Transform Matrix)

5.3.2.5. 7X7 RDFT PROCESSING

RDFT transform matrix is applied to 7 band color composite images. Each band is regenerated 1 times (Table 5.1)

- The transform matrix (TM) for RDFT is given by

$$TM = \begin{pmatrix} 0,3780 & 0,3780 & 0,3780 & 0,3780 & 0,3780 & 0,3780 & 0,3780 \\ 0,5345 & 0,3333 & -0,1189 & -0,4816 & -0,4816 & -0,1189 & 0,3333 \\ 0,5345 & -0,1189 & -0,4816 & 0,3333 & 0,3333 & -0,4816 & -0,1189 \\ 0,3780 & -0,4816 & 0,3333 & -0,1189 & 0,1189 & 0,3333 & -0,4816 \\ 0 & 0,2319 & -0,4179 & 0,5211 & -0,5211 & 0,4179 & -0,2319 \\ 0 & 0,5211 & -0,2319 & -0,4179 & 0,4179 & 0,2319 & -0,5211 \\ 0 & -0,4052 & 0,5211 & 0,2319 & -0,2319 & -0,5211 & 0,4179 \end{pmatrix} \quad (40)$$

- The inverse transform matrix (ITM) is given by

$$ITM = \begin{pmatrix} 0,3780 & 0,5345 & 0,5345 & 0,3780 & 0 & 0 & 0 \\ 0,3780 & 0,3333 & -0,1189 & -0,4816 & 0,2319 & 0,5211 & -0,4052 \\ 0,3780 & -0,1189 & -0,4816 & 0,3333 & -0,4179 & -0,2319 & 0,5211 \\ 0,3780 & -0,4816 & 0,3333 & -0,1189 & 0,5211 & -0,4179 & 0,2319 \\ 0,3780 & -0,4816 & 0,3333 & -0,1189 & -0,5211 & 0,4179 & -0,2319 \\ 0,3780 & -0,1189 & -0,4816 & 0,3333 & 0,4179 & 0,2319 & -0,5211 \\ 0,3780 & 0,3333 & -0,1189 & -0,4816 & -0,2319 & -0,5211 & -0,4179 \end{pmatrix} \quad (41)$$

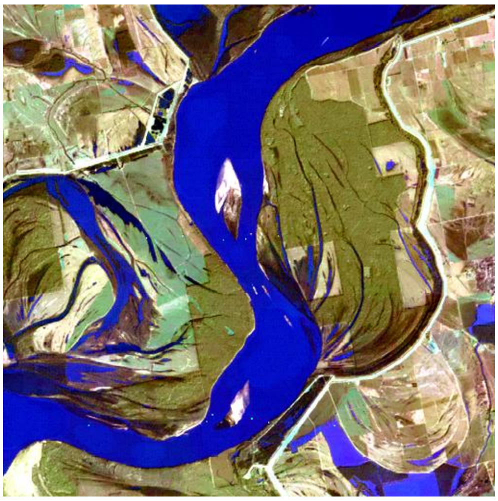


Figure 5. 18. The image obtained with the RDFT Algorithm (7X7 Transform Matrix)

5.3.3. Comparing RGB2NTSC and RDFT Algorithm Results

We compared the results of the RDFT algorithm with the results of the RGB2NTSC algorithm. We observe that the RDFT method gives better results than the RGB2NTSC algorithm. This is especially clear with bands [5, 6, 7], as seen in Figure 5.19-5.21.

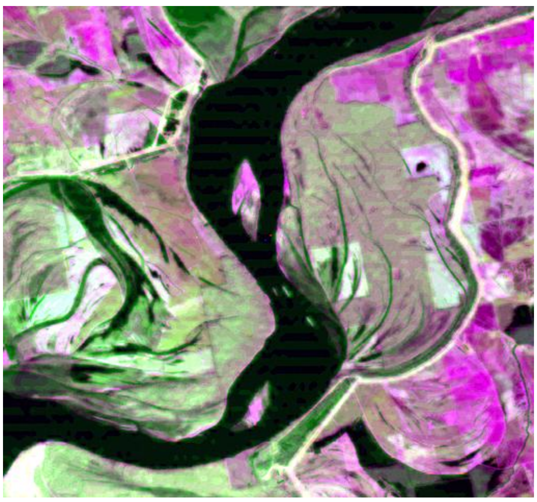


Figure 5. 19. Original Landsat TM image

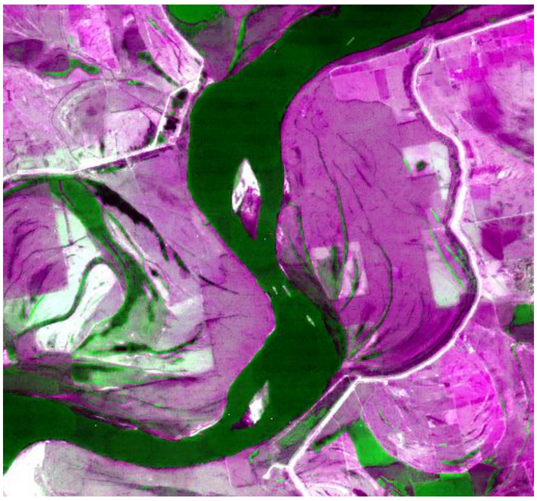


Figure 5. 20. The image obtained (Bands [5,6,7]) with the RGB2NTSC

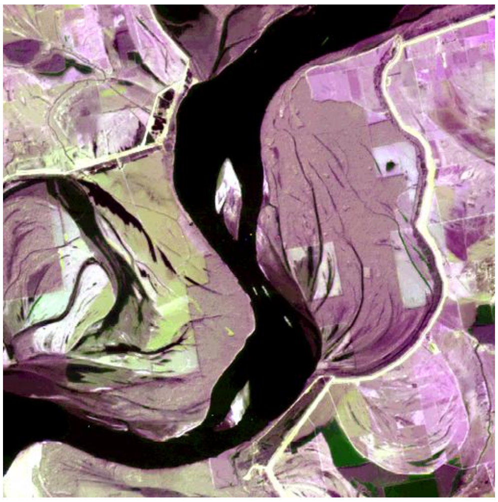


Figure 5. 21. The image obtained (bands [5, 6, 7]) with the RDFT

6. CLASSIFICATION

6.1. Classifier Concept

Classification is identifying groups of pixels that have similar spectral characteristics and determining the various features or land cover classes represented by these groups. Visual classification relies on the analyst's ability to use visual elements (tone, contrast, shape, etc) to classify an image. One digital image classification approach is based on the spectral information used to create the image and classifies each individual pixel based on its spectral characteristics. The result of classification is that all pixels in an image are assigned to particular classes or themes (e.g. water, coniferous forest, deciduous forest, corn, wheat, etc.), resulting in a classified image that is essentially a thematic map of the original image. The theme of classification is selectable, thus a classification can be performed to observe land use patterns, geology, vegetation types, or rainfall (Pfister, 2004).

Training of the pixels for classification can be either supervised or unsupervised (Remote Sensing Tutorials, Arizona University).

Supervised classification is performed when some prior or acquired knowledge of the classes in a scene is used to identify representative samples of different surface cover types. These samples, known as training sites, are set up to identify the spectral characteristics of each class of interest. The determination of training sites is based on the analyst's knowledge of the geographical region and the surface cover types present in the image. Once the training sites have been established, the numerical information in all of the image's spectral bands are used to define the spectral "signature" of each class. Once the computer has determined the signatures for each class, it will compare every pixel to the signatures and label it as the class that it is mathematically closest to. Thus, in supervised classification, the analyst starts with information classes and uses these to define spectral classes. Each pixel in the image is then assigned to the class which it most closely resembles.

Unsupervised classification is essentially the opposite of supervised classification. The pixels in an image are examined by the computer and classified into spectral classes. The grouping is based solely on the numerical information in the data and the spectral classes are later matched by the analyst to information classes. In order to create unsupervised classification, the analyst typically determines the number of spectral classes to identify and a computer algorithm will find pixels with similar spectral properties and group them accordingly (Pfister, 2004).

Classification is the same as pattern recognition in which the task is to classify a given pattern or signal such as a signal waveform or an image into one of several classes. Patterns are often measured quantities, which are random in nature. As a result of the random nature of patterns, statistical decision theory is often used for the purpose of classification (Theodoridis and Koutroumbas, 1999).

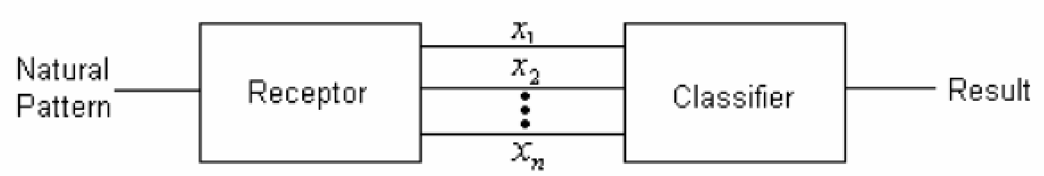


Figure 6. 1. Basic scheme of a pattern recognition system (Landgrebe, 2003)

As Figure 6.1 shows, the design of a pattern recognition system involves three main aspects: (i) acquisition of the data, (ii) representation of the data, and (iii) decision making. A pattern recognition problem dictates the choice of a sensor, a representation scheme and a decision making model (Jain, Duin, and Mao, 2000).

In the statistical approach, each pattern is represented in terms of n measurements (features) and is viewed as a point in an n-dimensional space, which is called feature space (Duda, Hart, and Stork, 2001). Figure 6.2 shows a simple example for n = 2 case. Any point in the feature space can then be represented as an n-dimensional feature vector **x** as

$$X=[X_1, X_2,.., X_n]^T \tag{42}$$

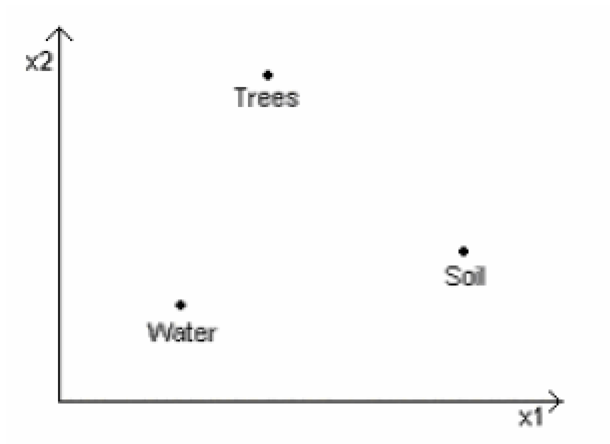


Figure 6. 2. An Example Feature Space

What makes pattern recognition problems hard is that there can be a large degree of variability of inputs that belong to the same class, relative to the differences between patterns in different classes (Duda, Hart, and Stork, 2001). Because of the variations present in the spectral response of earth surface materials, patterns belonging to a given class form a localized cluster of points as shown in Figure 6.3, rather than just a point as shown in Figure 6.2.

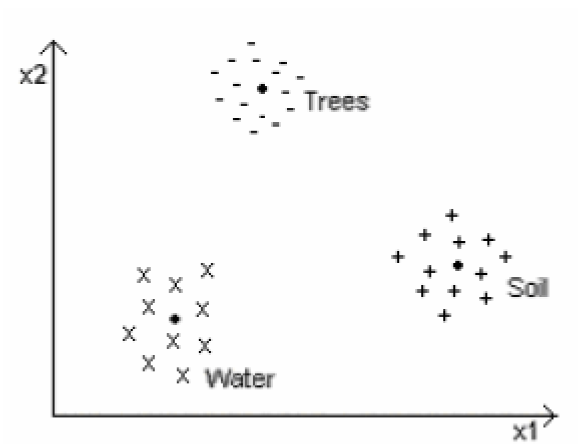


Figure 6. 3. Clusters consisting of data points

The objective of classification is to find the decision rules, that partition the feature space into volumes called decision regions, each one corresponding to a given class. If a feature vector is located in a particular decision region, it will be

assigned to the class associated with that region. The decision regions are separated by surfaces called the decision boundaries. The decision boundaries in feature space, which divide the space into regions that are to be associated with each class, cannot be easily explicitly and directly defined. It involves a difficult computational problem of determinining on which side of a boundary a given unknown pixel falls, which can be solved with the following algorithm. Assume that one can determine a set of m functions $\{g_1(x), g_2(x), \ldots, g_m(x)\}$
Such that $g_i(x)$ is larger than all others whenever x is from class i.
This suggests that the appropriate classification rule as follows:

Let ω_i denote ith class.

Decide x € ω_i if (i.e., x is in class ω_i if and only if)

$g_i(x) \geq g_j(x)$ for all j=1,2,…m.

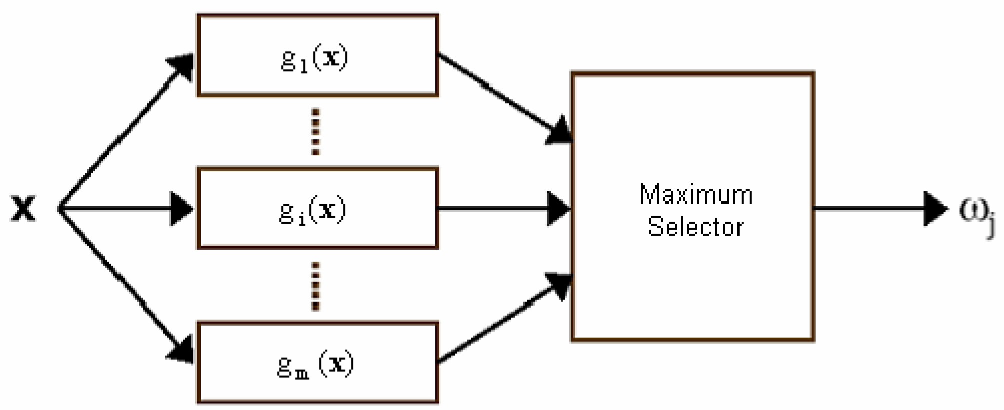


Figure 6. 4. A pattern classifier defined in terms of discriminant functions, $g_i(x)$ (Landgrebe, 2003)

The collection of functions $\{g_1(x), g_2(x),\ldots, g_m(x)\}$ are called discriminant functions. As shown in Figure 6.4, to classify an unknown pixel value x, one must evaluate each discriminant function at x to determine which discriminant function is the largest to assign the pixel to a class, thus making the classification process simple and fast (Landgrebe, 2003).

6.2. Maximum Likelihood Method

Maximum likelihood is the most common supervised classification method. When the distributions of the spectra are normal, the maximum likelihood method gives the best performance. The principles of the maximum likelihood method are described in the following: The classes are defined by

$$\omega_i, \ i = 1, \ldots, M \tag{43}$$

M is total number of classes. The method determines the class of a pixel at a location x by conditional probabilities

$$P\,(\omega_i|x), i=1,\ldots,M \tag{44}$$

Here, x is a vector of brightness values for the pixel. The probability $P(\omega_i|x)$, gives the likelihood that the correct class is ω_i for a pixel at position x. Then, classification can be performed according to

$$x \ € \ \omega_i \text{ if } P(\omega_i|x) > P(\omega_j|x) \text{ for all } j \neq i \tag{45}$$

The pixel at x belongs to class ω_i if $P\,(\omega_i|x)$ is highest. This intuitive decision rule is a special case of a more general rule in which decisions can be biased according to different degrees of significance being attached to different incorrect classifications. The general approach is called Bayes' classification.

However, $P\,(\omega_i|x)$ values are generally unknown. If sufficient training data are available for each land cover type, these data can be used to estimate the probability distribution $P\,(\omega_i|x)$ for a cover type that describes the chance of finding pixel from class ω_i at the position x. There will be as many $P\,(\omega_i|x)$ as there are ground cover classes. In other words, for a pixel at a position x in multispectral space, a set of probabilities can be computed that give the relative likelihood that belongs to each available class.

The desired probabilities P (ω_i|x) in Eq. 45 and P (x|ω_i) estimated from training data are related by Baye's theorem:

$$P(\omega_i|x) = P(x|\omega_i)\,P(\omega_i)/P(x) \tag{46}$$

where P(ω_i) is the probability that class ω_i occurs in the image, and P(x) is the probability of finding a pixel from any class at location x. Although P(x) itself is not important for classification, it can be calculated as

$$p(x) = \sum_{i=1}^{M} P(x\,|\,\omega_i)P(\omega_i) \tag{47}$$

P (ω_i) are called a priori probabilities, P (ω_i|x) are called posteriori probabilities. Using Eq. 46, it can be seen that the classification rule of Eq. 45 can be written as

$$x \in \omega_i \text{ if } P(x\,|\,\omega_i)\,P(\omega_i) > P(x|\omega_j)\,P(\omega_j) \text{ for all } j \neq i \tag{48}$$

where P(x) has been removed as a common factor. Eq. 48 is more commonly used than Eq. 45. Since P (ω_i|x) and P (ω_i) estimated with training data, for mathematical convenience, if the definition

$$g_i(x) = \mathrm{In}\left\{P(x\,|\,\omega_i)P(\omega_i)\right\} = \mathrm{In}P(x\,|\,\omega_i) + \mathrm{In}P(\omega_i) \tag{49}$$

is used, where In is the natural logarithm, then (48) can be restated as

$$\text{if } g_i(x) \geq g_j(x) \text{ for all } j \neq i \tag{50}$$

where $g_i(x)$ are referred to as discriminant functions. This is, with one modification to follow, the decision rule used in maximum likelihood classification.

At this stage, it is assumed that the probability distributions for each class are of the form of normal (Gaussian) multivariate models. This assumption, works well with multispectral and hyperspectral data. The multivariate normal density function is given by

$$p(x \mid \omega_i) = (2\pi)^{-N/2} \mid \Sigma_i \mid \exp\left\{-\frac{1}{2}(x - m_i)\Sigma_i^{-1}(x - m_i)\right\} \tag{51}$$

where m_i and Σ_i are the mean vector space and covariance matrix of the data in class ω_i, respectively. The resulting term $-\frac{N}{2}In(2\pi)$ is common to all $g_i(x)$. So the discrimination function for maximum likelihood classification can be written as

$$g_i(x) = -In \mid \sum_i \mid -(x - m_i)^t \Sigma_i^{-1}(x - m_i) \tag{52}$$

Implementation of the maximum likelihood decision rule involves using either (51) or (52) in (49).

The goal of classification is to determine the shapes of that separate one class from another in multispectral space. These surfaces can be determined in the following manner.

Spectral classes are defined by those regions in multispectral space where their discriminant functions have the highest values. Clearly these regions are separated by surfaces where the discriminant functions for adjoining spectral classes are equal:

$$g_i(x) - g_j(x) = 0 \tag{53}$$

This is referred to as a decision surface. If all the surfaces separating spectral classes are known, decisions about class membership of an image pixel can be made on the basis of the position to the complete set of surfaces.

The construction $-(x - m_i)^t \sum_i^{-1} (x - m_i)$ in (51) and (52) is a quadratic function of x. Consequently, the decision surfaces implemented by maximum likelihood classification are quadratic ans thus take the form of parabolas, circles, and ellipses. (Yamagata, 1997)

6.3. Experimental Results

The maximum likelihood classification method was applied to the "mississippi.lan" (Matlab 7.0, 2004) image. Each image of 'mississippi.lan' has 512 lines and 512 columns, with a total number of 1,835,008 pixels and 7 bands (Geo InSight International, 1999).

In our experiments, we test the result images of the RGB2NTSC and RDFT algorithm with Maximum Likelihood classification method to observe the effect of our algorithms in the classification accuracy. The number of classes is 5.

Table 6.1 show the test and training field accuracies of the images. Figure 6.6 shows thematic maps of images obtained with RGB2NTSC and RDFT algorithms.

Figure 6.5 show that our image enhancement/fusion methods improve performance. When the results are compared with the direct classification original image, RGB2NTSC and RDFT gives best results in classification.

Table 6. 1. Training and Test Field accuracies of ML classifier

Method	Training field accuracy	Test field accuracy
Original image	100	86,86
RGB2NTSC	100	89,79
RDFT (4X4 Matrix)	100	91,62
RDFT (5X5 Matrix)	100	95,57
RDFT (6X6 Matrix)	100	93,77

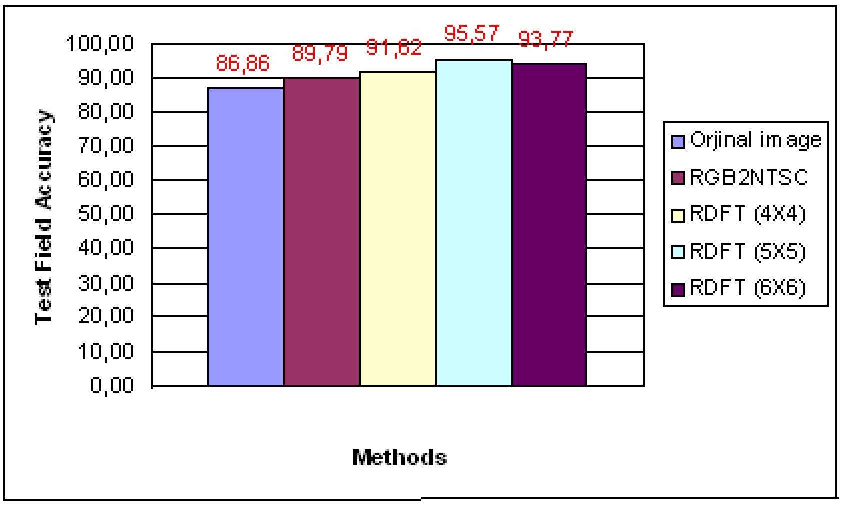
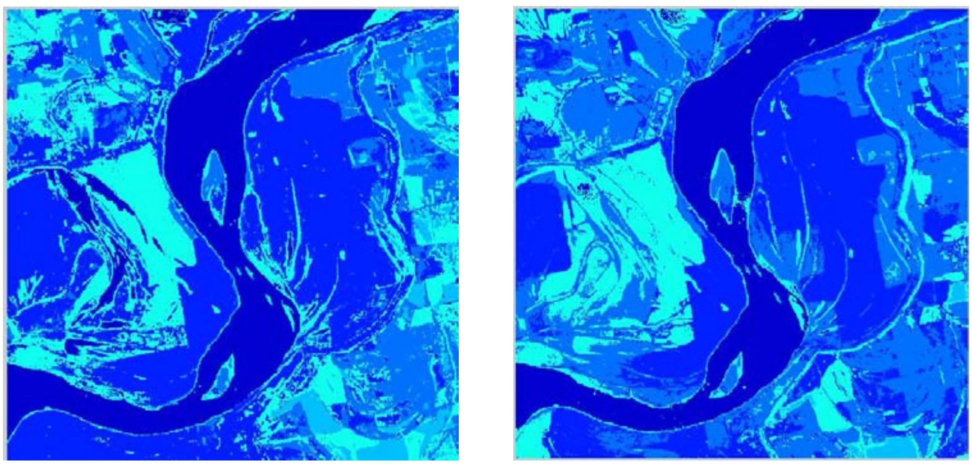
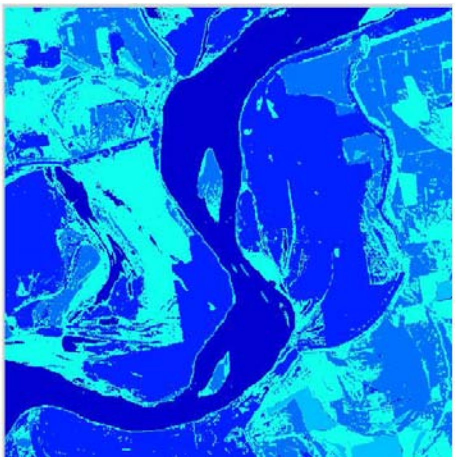


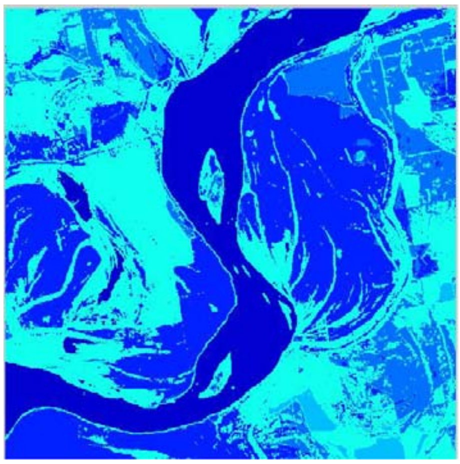
Figure 6. 5. Test Field Accuracies of ML classifier



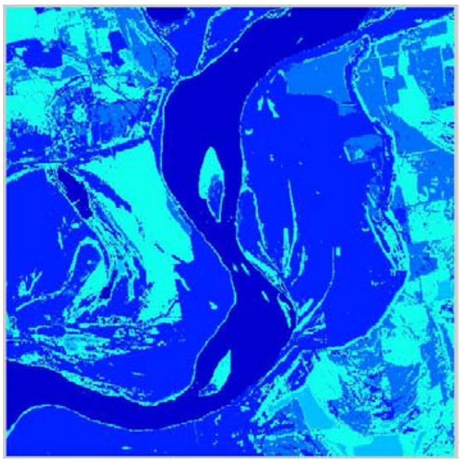
(a) (b)



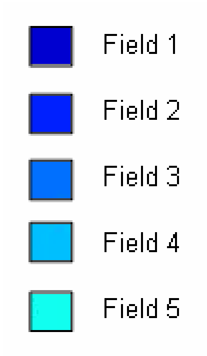
(c)



(d)



(e)



(f)

Figure 6. 6. Thematic maps generated by the ML **(a)** Original image; **(b)** RGB2NTSC, **(c)** RDFT (4x4), **(d)** RDFT (5x5), **(e)** RDFT (6x6), **(f)** Color coding map for classification results

7. CONCLUSIONS

There is a lot of information in multispectral / hyperspectral imagery. We illustrated in this thesis that the effective use of such information using transform techniques yield better image enhancement/fusion results. The key concept for this purpose is to combine M bands out of N bands, M < N at each step using a transform, do signal/image processing in the transform domain images, and inverse transform to obtain better images in the regular image domain. Averaging all the resulting images in this domain per band further enhances the results by a large margin. The RDFT approach is especially effective for this purpose.

With new enhnancement methods, the distortions that morphological filter causes were prevented and the edges preserved. The best result was obtained with the 4X4 RDFT transform, which correlates well with its number of combinations per band are maximum. The 3X3 RDFT transform and 5X5 RDFT transform gives similar results, where the number of combinations is 15. The images obtained with the 7X7 RDFT transform include some distortions like morphological filter result, especially in the river region. This also correlates well with the fact that the number of combinations per band is 1, and hence there is no averaging.

When the result images of the RGB2NTSC and RDFT algorithm are tested with maximum likelihood classification method to observe the effect of our algorithms in the classification accuracy. New image enhancement/fusion methods improve performance. When the results are compared with the direct classification original image, RGB2NTSC and RDFT gives best results in classification.

Even though the experimental results are with multispectral images, the techniques are also applicable to other multispectral images, such as biomedical multispectral images, and hyperspectral images.

REFERENCES

BEAUCHEMIN, M., FUNG, K. B., June 1999. Intensity-Hue-Saturation Color Display Transform for Hyperspectral Data. 21st Canadian Symposium on Remote Sensing, Ottawa, Ontario, Canada.

BOURKE, P., 1993. Discrete Fourier Transform, Fast Fourier Transform. Available: http://astronomy.swin.edu.au/~pbourke/other/dft/

BRETSCHNEIDER, T., KAO, O., Image Fusion in Remote Sensing. Online Symposium for Electronics Engineers.Available: http://www.techonline.com/community/tech_group/dsp/course/13503

CARPER, W. J., LILLESAND, T. M., KIEFER, R. W., April 1990. The Use of Intensity-Hue-Saturation Transformations for Merging SPOT Panchromatic and Multispectral Images. Programmetric Engineering and Remote Sensing, Vol. 56, No. 4, pp. 459-467.

CARRON T., LAMBERT, P., 1994. Color Edge Detector Using Jointly Hue, Saturation and Intensity. ICIP (3) 1994, pp. 977-981.

CASTLEMAN, K. R., 1996. Digital Image Processing. Springer, Prentice Hall, Englewood Cliffs, N.J.

CHEN, D.C., WANG, L., AMANI, M., 2004. A new technique for multi-resolution image fusion. IEEE, pp: 4901-4904.

DUDA, R.O., HART, P.E. and STORK, D.G., 2001. Pattern Classification. John Wiley & Sons, New York.

ERSOY, O. K., November 2005. DFT/RDFT Filter banks with Symmetric Zero-Phase Nonoverlapping Analysis/Synthesis Filters and Applications. Proc. Computational Intelligence and Systems Engineering Conf. (ANNIE 05), St. Louis, Missouri.

ERSOY, O. K., March 1994. A Comparative review of real and complex Fourier-related transforms. IEEE Proceedings, Vol. 82, No. 3, pp. 429-447.

ERSOY, O. K., August 1985. Real discrete Fourier transform. IEEE Tran. Acoustics, Speech, Signal Proc, Vol. ASSP-33, No. 4, pp. 880-882.

Fourier Transforms, DFTs, and FFTs, 2003. Available: http://www.me.psu.edu/me82/Learning/FFT/FFT.html

Geo InSight International, 1999. LANDSAT Thematic Mapper(TM). Available: http://www.geoinsight.com/Knowledgebase/RemoteSensing/Landsat.cfm.

GONZALEZ, R. C., WOODS, R. E., 2002. Digital Image Processing. M. J. Horton, Digital Image Processing Second Edition. Prentice-Hall, Upper Saddle River, New Jersey 07458, pp-75-142.

HUANG, T. S., YANG, G. J., and TANG, G. Y., 1979. A Fast Two-Dimensional Median Filtering Algorithm. IEEE Transactions on Acoustics, Speech and Signal Processing, 27(1), pp.13-18.

HUSSAIN, Z., 1991. Digital Image Processing: Practical Applications of Paralel Processing Techniques. Springer, E. Horwood, New York.

JAEHNE, B., 1997. Digital Image Processing: Concepts, Algorithms, and Scientific Applications. Springer, Berlin.

JAIN, A., DUIN, R., and MAO, J., 2000. Statistical Pattern Recognition. A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1), pp. 4-37.

JAIN, A. K., 1989. Fundamentals of Digital Image Processing. Prentice Hall, Englewood Cliffs, N.J.

LANDGREBE, D. A., 2003. Signal Theory Methods in Multispectral Remote Sensing. John Wiley & Sons, New York.

MATLAB 7.0-Image Analysis Toolbox, May 2004.

Natural Resources Canada, 2004. Fundamentals of Remote Sensing. Available : http://www.ccrs.nrcan.gc.ca/ccrs/learn/tutorials/fundam/chapter4

PARCHARIDIS, I., KAZI-TANI, L.M., 2000. Landsat TM and ERS Data Fusion: A Statistical Approach Evaluation for Four Different Methods. Geoscience and Remote Sensing Symposium, Proceedings. IGARSS 2000 Vol. 5, pp. 2120-2122.

PETERS, R.A., May 1995. A New Algorithm for Image Noise Reduction Using Mathematical Morphology. IEEE Transactions on Image Processing, Vol.4, No.5, pp. 554-568.

PFISTER, J., 2004. Using Landscape Metrics to Create an Index of Forest Fragmentation for the State of Maryland. M.S. Thesis, Towson University, Towson. http://chesapeake.towson.edu/data/all_classification.asp.

PRATT W. K., 2001. Digital Image Processing Third Edition. John Wiley & Sons, New York, pp. 241-297.

QIN, Q., LIU, D., and LIU, H., 2001. Images Fusing in Remote Sensing Mapping. pp. 315- 320.

RATHI M., 2004, Morphological Image Cleaning Filters, Purdue University, Research Report (Unpublished).

Remote Sensing Tutorials, Image Classification. The University of Arizona, available: http://aria.arizona.edu/courses/tutorials/class/html/class.html.

ROKHMATULOH, TATEISHI, R., WIKANTIKA, K., MUNADI, K., and ASLAM, M., 2003. Study on the Spectral Quality Preservation Derived from Multisensor Image Fusion Techniques between JERS-1 SAR and Landsat TM Data. 2003 IEEE International Geoscience and Remote Sensing Symposium,Toulouse-France, pp. 3656-3658.

RUSS, J. C., 1999. The Image Processing Handbook. CRC Press, Boca Raton, FL.

SONG, J., DELP, E. J., 1990. A study of the generalized morphological filter. Circuits,Syst., Signal Processing.

STRENBERG, S. R., 1986. Grayscale morphology. Comput., Vision, Graph., Image Processing, vol.35, pp. 333-355.

THEODORIDIS, S., KOUTROUMBAS, K., 1999. Pattern Recognition. Academic Press, San Diego.

YAMAGATA, Y., 1997. Advanced Remote Sensing techniques for monitoring complex ecosystems: Spectral Indices, Unmixing, and Classification of Wetlands. Phd thesis, University of Tokyo, pp. 11-14.

YILDIRIM, İ., ERSOY O. K., 18-24 July 2004. A New Multispectral Image Classification Method Using Adaptive Filtering. The Committee on Space Research (COSPAR) 35th Scientific Assembly, Paris.

BIOGRAPHY

Gülçin Yiğitler was born on July 19th. 1980 in Malatya, TURKEY. She graduated from Malatya Science High School in 1997. She received her B.Sc. degree from Çukurova University in the field of Electrical and Electronics Engineering in 2002.

After she graduated from University, she started as a master student at the Electrical and Electronics Engineering Department of Çukurova University. Meanwhile she has been working at Turk Telekom as an engineer since 2003.