DEVELOPING TOOLS FOR THE ENHANCEMENT OF REMOTE SENSING IMAGES AND PARALLEL PROGRAMMING APPLICATIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ÇANKAYA UNIVERSITY

BY

HÜSEYİN ŞAHİN AKBAL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF COMPUTER ENGINEERING

OCTOBER 2005

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Yurdahan Güler
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

M. R.

Prof. Dr. Mehmet Reşit Tolun
Head of the Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Cem Özdoğan
Supervisor

Examining Committee Members

Prof. Dr. Ziya B. Güvenç — Ziya B.

Asst. Prof. Dr. Cem Özdoğan

Dr. Cevat Şener — Şener

ABSTRACT

DEVELOPING TOOLS FOR THE ENHANCEMENT OF REMOTE SENSING IMAGES AND PARALLEL PROGRAMMING APPLICATIONS

Akbal, Hüseyin Şahin

M. S., Department of Computer Engineering

Supervisor: Asst. Prof. Dr. Cem Özdoğan

October 2005, 47 pages

The principal goal of enhancement techniques is to process an image so that the result is more suitable than the original image for a specific application.

In this study the remote sensing is chosen as the area of that specific application. First a survey is made on the enhancement techniques used for the enhancement of remote sensing images. Then optimization of these techniques are tried using parallel computation. Although optimization efforts cover a subdivision of the techniques, the results offer a definite idea about what can be achieved from the parallelization of the enhancement techniques used in the area of the remote sensing.

Keywords: parallel programming, parallel computation, remote sensing, image processing, image enhancement, Fourier transform.

ÖZ

UZAKTAN ALGILAMA İLE ELDE EDİLMİŞ GÖRÜNTÜLERİN İYİLEŞTİRİLMESİ VE PARALEL PROGRAMLAMA UYGULAMALARI İÇİN ARAÇLARIN GELİŞTİRİLMESİ

Akbal, Hüseyin Şahin

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Cem Özdoğan

Ekim 2005, 47 sayfa

Görüntü iyileştirme tekniklerinin temel amacı, bir görüntüyü belli bir amaca yönelik özel bir uygulama için orijinal halinden daha uygun bir hale getirmektir.

Bu çalışmada, özel uygulama alanı olarak uzaktan algılama seçildi. Öncelikle uzaktan algılama ile elde edilmiş görüntülerin iyileştirilmesinde kullanılan görüntü iyileştirme tekniklerini incelendi. Sonrasında da bu teknikleri paralel hesaplama kullanarak bir adım ileriye götürmek için denemeler yapıldı. Her ne kadar alan içerisindeki tekniklerin bir kısmının paralelliği üzerine çalışılmış olsa da; elde edilen sonuçlar uzaktan algılama alanında kullanılan iyileştirme tekniklerinin paralelleştirilmesi ile nasıl bir kazanım elde edilebileceği konusunda bize kesin bir fikir vermektedir.

Anahtar Kelimeler: parallel programlama, parallel hesaplama, uzaktan algılama, görüntü işleme, görüntü iyileştirme, Fourier dönüşümü.

ACKNOWLEDGMENTS

I would like to thank to my advisor Asst. Prof. Dr. Cem Özdoğan for his guidance and patience to me throughout the study of this thesis work.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Independent of the area of the study, image processing systems show the same characteristics. First comes the acquisition of the image. In the remote sensing this step is performed by the sensors on the aircraft or spacecraft platforms and that are not in contact with the target of interest. Then preprocessing of the image data is performed. This step includes enhancement of the data before the next step, which is the processing of the enhanced image data to extract some useful information for a specific purpose.

The principal goal of enhancement techniques is to process an image so that the result is more suitable than the original image for a specific application. The word *specific* is important, because it establishes at the outset that the techniques discussed in Chapter 3 are very much problem oriented. Thus, for example, a method that is quite useful for enhancing x-ray images may not necessarily be the best approach for enhancing pictures of Mars transmitted by a space probe [1].

Enhancement operations that are used in the area of the remote sensing are examined in two broad categories: point operations and neighborhood operations. Enhancement techniques using point operations are called radiometric enhancement techniques. Enhancement techniques using neighborhood operations are called geometric enhancement techniques.

Radiometric enhancement techniques use the original value of the pixel to produce the new brightness value of the pixel. The aim of these operations is to alter the contrast of the image to make a specified feature more clear.

Geometric enhancement techniques use both the existing value of the pixel and values of a specified neighborhood of pixels to produce the new brightness value of the pixel. Geometric enhancement techniques are performed to remove the noise from image data or to make the edges better.

Geometric enhancement techniques are performed using templates in the spatial domain. They are also performed in the frequency domain using Fourier transform. The parallelization part of this study includes the parallelization of the geometric enhancement techniques in the frequency domain.

The aim of this thesis study is developing the parallel versions of the enhancement algorithms used in the area of the remote sensing.

After reviewing the enhancement techniques which are used for the enhancement of the remote sensing imagery, the results of attempts to use parallel computing techniques for the enhancement of the remote sensing imagery are presented.

This thesis consists of five chapters except introduction: chapter 2 gives us remote sensing basics; description of the remote sensing, how images are taken in the remote sensing, what kind of data the imagery includes etc.

Chapter 3 covers the range of radiometric and geometric enhancement techniques commonly adopted in practice. This chapter also includes the brief explanation of Fourier transform and its usage for the filtering of images in the spatial domain.

Chapter 4 includes parallelization efforts. This chapter gives both the sequential and parallel algorithms that are tested.

Chapter 5 covers the graphical presentation of the results that are obtained from the performed tests, how the programs are tested and the comments on the results.

Chapter 6 gives a brief conclusion about this study.

CHAPTER 2

REMOTE SENSING BASICS

It is probably the most used question phrase in the world: "What is ...?". So, let's start asking another what question, what is remote sensing?. Of course this question has many different answers. Two of them will be used throughout this text: the first one says that, remote sensing is the science and art of obtaining useful information about an object, area or phenomenon through the analysis of data acquired by a device that is not in contact with the object, area, or phenomenon under investigation [2]. The second one says that, remote sensing is the science of acquiring, processing, and interpreting images, and related data, acquired from spacecraft and satellites that record the interaction between matter and electromagnetic energy [3].

Both of the definitions mentioned are obtaining information about an object of interest without being in contact. Just like hearing or seeing something. In both sensing activities, the terminology of remote sensing assumes us doing passive sensing. In remote sensing there are two types of sensing, active and passive. In both types the electromagnetic energy is measured. Passive sensors measure the energy that is naturally available. This natural availability requires an external energy source, which is generally the Sun. The Sun's energy is either reflected or absorbed and then emitted. During the time when the Sun is illuminating the object of interest, i.e. the

Earth, passive sensors can measure energy. But when the object of interest is not illuminated by the Sun, the passive sensors can detect emitted energy, if the amount of energy is large enough to be recorded. On the other hand, active sensors do not need an external energy source. They include their own energy sources. The remote sensing platform emits radiation towards the target to be investigated. Then the radiation reflected from the target is measured by the sensor. Active sensors can be used for examining wavelengths that are not sufficiently provided by the Sun, such as microwaves. Also the illumination of the target is controlled better. They can make measurements independently from the time of the day and season. However, active sensing platforms require the generation of a huge amount of energy to illuminate the targets sufficiently.

According to CCRS (Canada Centre for Remote Sensing) [4] the remote sensing process involves 7 basic steps:

1. **Energy Source or Illumination:** The first requirement for the remote sensing is to have an energy source which illuminates or provides electromagnetic energy to the target of interest.

2. **Radiation and the Atmosphere:** As the energy travels from its source to the target, it will come in contact and interact with the atmosphere it passes through. This interaction also occurs as the energy travels from the target to the sensor.

3. **Interaction with the Target:** Once the energy makes its way to the target through the atmosphere, it interacts with the target depending on the properties of both the target and the radiation.

4. **Recording of Energy by the Sensor:** After the energy has been scattered by, or emitted from the target, a sensor (remote - not in contact with the target) is used to collect and record the electromagnetic radiation.

5. **Transmission, Reception and Processing:** The energy recorded by the sensor has to be transmitted, often in electronic form, to a receiving and processing station where the data are processed into an image (hard copy and/or digital).

6. **Interpretation and Analysis:** The processed image is interpreted, visually and/or digitally or electronically, to extract information about the target which was illuminated.

7. **Application:** The final element of the remote sensing process is achieved when the information which is extracted from the imagery about the target is applied for better understanding of it, reveal some new information, or assist in solving a particular problem.

Although any sensible range of wavelengths may be used in the remote sensing, the details of the second step makes us to exclude certain wavelengths. During the radiation passing from the atmosphere, it is affected in two ways; scattering and absorption. Scattering means redirection of the radiation from its original path. How much scattering takes place depends on the wavelength of the radiation, the wealth of particles or gases, and the distance which the radiation travels through the atmosphere [4]. The latter one, absorption, occurs when the molecules in the atmosphere absorb energy at various wavelengths. Three main constituents of the atmosphere which absorb radiation are ozone, carbon dioxide, and water molecules.

The following spectral ranges are used in remote sensing:

- **Ultraviolet (UV):** The UV portion of the spectrum covers wavelengths from 10 nm to 400 nm and has the shortest wavelengths which are practical for the remote sensing. Some Earth surface materials, primarily rocks and minerals, fluoresce or emit visible light when illuminated by UV radiation.

- **Visible:** The visible portion of the spectrum, which takes its name because our eyes can detect the electromagnetic energy of this portion as light, covers wavelengths from 400 nm to 700 nm.

- **Infrared (IR):** The IR portion of the spectrum covers wavelengths from 700 nm to 1 mm. As you can see easily this range is much more larger than the visible portion. The IR portion is divided into two sub portions based on their radiation properties. The first one is the ”reflected IR” which covers wavelengths from 700 nm to 3000 nm. The second one is the ”emitted IR”, or ”thermal IR” which covers wavelengths from 3000 nm to 1 mm. The thermal IR region is essentially the radiation that is emitted from the Earth’s surface in the form of heat.

- **Microwave:** The microwave portion of the spectrum covers wavelengths from 1 mm to 100 mm and has the longest wavelengths used for remote sensing.

The significance of these different ranges lies in the interaction mechanism between the electromagnetic radiation and the materials being examined [5].

- In the visible/infrared range the reflected energy measured by a sensor depends upon properties such as the pigmentation, moisture content and cellular structure of vegetation, the mineral and moisture contents of soils and the level of sedimentation of water.

- At the thermal end of the infrared range it is heat capacity and other thermal properties of the surface and near subsurface that control the strength of radiation detected.
- In the microwave range, using active imaging systems based upon radar techniques, the roughness of the cover type being detected and its electrical properties, expressed in complex permittivity (which in turn strongly influenced by the moisture content) determine the magnitude of the reflected signal.

Radiation that is not absorbed or scattered in the atmosphere will interact with the target of interest in three ways depending on the wavelength of the energy and the material and condition of the target:

1. Absorption occurs when radiation is absorbed into the target.
2. Transmission occurs when radiation passes through a target.
3. Reflection occurs when the radiation is redirected.

Reflection occurs in many different ways. As can be seen from the Figure (2.1), smooth surfaces cause mirror-like reflection which is called *specular* reflection. They appear darker areas in the image data. Rough surfaces cause *diffuse* reflection which scatters the radiation in all directions. They appear light in the image data. Manufactured features such as buildings cause *corner* reflection, which is generally observed in microwave image data. In the image data they appear as very bright areas. Features, such as vegetation canopies and sea ice, cause volume scattering behaviour which leads to light tonal appearance in radar imagery.

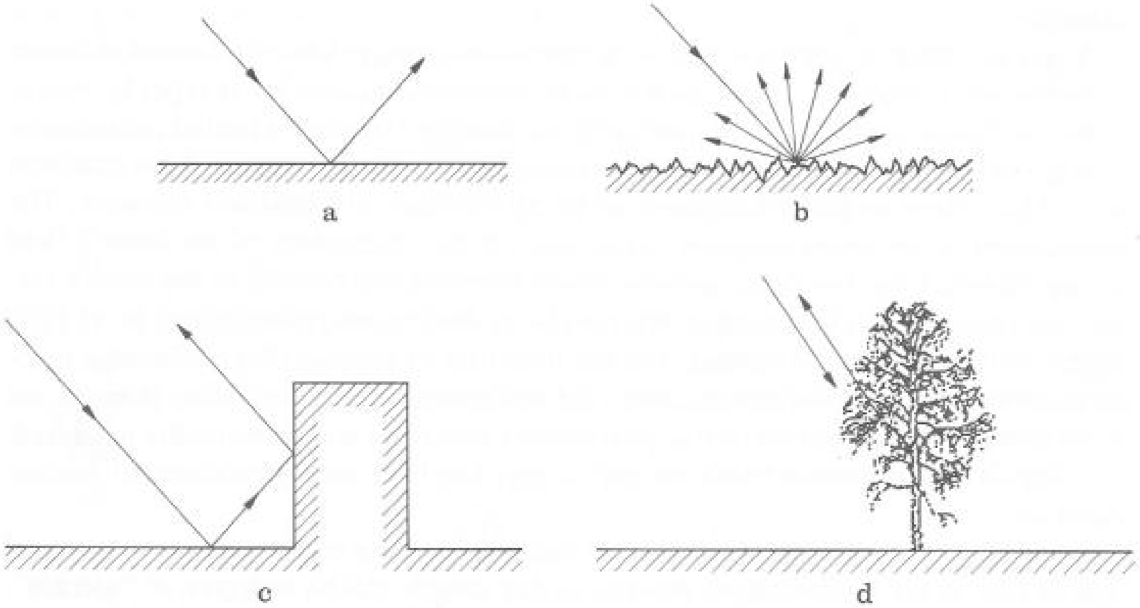


Figure 2.1: **a** Specular, **b** diffuse, **c** corner reflector and **d** volume scattering behaviour, encountered in the information of microwave image data

It is important to know how common Earth surface materials interact with the radiation in the used wavelength. For visible and infrared portions of the spectrum reflectance characteristics of common Earth surface materials can be seen in the Figure (2.2). It is seen that water reflects about 10% or less in the blue-green range, a smaller percentage in the red and certainly no energy in the infrared range. If there is suspended sediment present in the upper layers of the water body, then this will allow better reflectivity and a brighter appearance of the water. Suspended sediment can be easily confused with shallow, and clear, water, since these two phenomena appear very similar. Chlorophyll in algae absorbs more of the blue wavelengths and reflects the green, making the water appear more green in color when algae is present. The

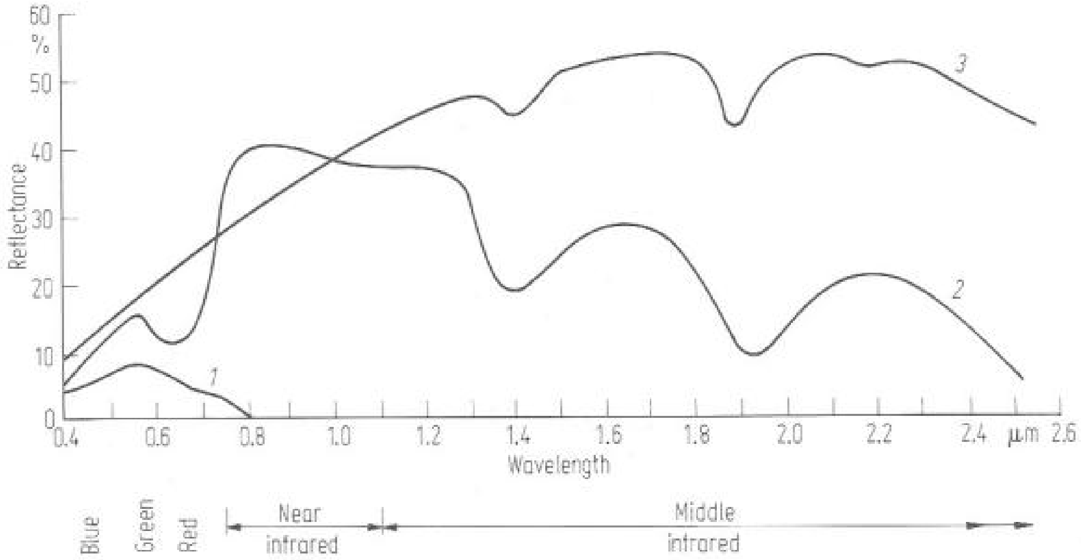


Figure 2.2: Spectral reflectance characteristics of common earth surface materials in the visible and near-to-mid infrared range. *1* Water, *2* vegetation, *3* soil.

topography of the water surface (rough, smooth, floating materials, etc.) can also lead to complications for water-related interpretation due to potential problems of specular reflection and other influences on color and brightness.

The vegetation curve is relatively more complex than the other two. In the middle infrared range it is dominated by the water absorption bands at 1.4 μm, 1.9 μm and 2.5 μm. Between 0.7 μm and 1.3 μm it is affected mainly by plant cell structure while in the visible range of wavelengths plant pigmentation, the amount of chlorophyll, is the dominant factor. The vegetation curve is drawn for healthy green vegetation, which has chlorophyll absorption bands in the blue and red regions leaving only green reflection of any significance. The internal structure of healthy leaves act as excellent diffuse reflectors of near infrared wavelengths. If our eyes were sensitive to near infrared, trees would appear extremely bright to us at these wavelengths. In fact, measuring and monitoring the near infrared reflectance is one way that scientists can determine how healthy or unhealthy vegetation may be.

Resolution or resolving power is the key term to determine the ability of an optical system. It is the measure of the ability of an optical system to distinguish between signals that are spatially near and spectrally similar. Resolution may be examined under four titles [6]:

1. Spectral resolution refers to the number and dimension of specific wavelength intervals in the spectrum to which a sensor is sensitive. The size of the interval or band may be large, as with panchromatic black-and-white aerial photography (0.4 μm to 0.7 μm), or relatively small, as with band 3 of the Landsat 5 Thematic Mapper (TM) sensor system (0.63 μm to 0.69 μm). The bands are selected to maximize the contrast between the object of interest and its background. Careful selection of the spectral bands may improve the probability that a feature will be detected and identified.

2. Spatial resolution is described as pixel size equivalent to ground meters. For the Landsat TM it is 30 x 30 m. The smaller the spatial resolution, the greater the resolving power of the sensor system.

3. Temporal resolution of a sensor system refers to how often it records imagery of a particular area. Ideally, the sensor obtains data repetitively to capture unique discriminating characteristics of the object under investigation. For example [7], agricultural crops have unique crop calendars in each geographic region. To measure specific agricultural variables, it is necessary to acquire remotely sensed data at critical dates in the phenological cycle. Analysis of multiple-date imagery provides information on how the variables are changing through time. Change information provides insight into process influencing the development

of the crop.

4. Radiometric resolution describes the range and discernable number of discrete brightness values. Frequently the radiometric resolution is expressed in terms of the number of binary digits, or bits, necessary to represent the range of available brightness values. Thus data with 8 bit radiometric resolution has 256 levels of brightness values.

Together with the frame size of an image, in equivalent ground kilometers, spectral resolution, radiometric resolution and spatial resolution determine the amount of data to be processed. As an example, Landsat TM instrument [5] can be used. It has 7 wavelength bands with 8 bit radiometric resolution, six of which have 30 m spatial resolution and one of which has a spatial resolution of 120 m (the thermal infrared band). An image frame of 185 km x 185 km therefore represents 2.37 million pixels in the thermal band and 38 million pixels in the other six bands. At 8 bits per pixel, a complete 7 band image is composed of 1.848 x 10^9 bits or 1.848 giga bits; alternatively and more commonly the data value would be expressed as 231 million bytes, where 1 byte is equivalent to 8 bits.

CHAPTER 3

IMAGE PROCESSING BASICS

Remote sensing image data in digital form has two components. Spatially the data is composed of discrete picture elements, or pixels, and radiometrically it is quantised into discrete brightness levels. Several approaches are possible to extract information from the digital image data. In quantitative analysis, computers ar used and in photointerpretation (or image interpretation) a human analyst/interpreter extracts information from the digital image data. Photointerpretation includes interpretation of the image's spatial, temporal and spectral elements. When many different images of the same geographical area recorded at different dates, there may be some changes. There is nothing to do with the temporal details. Image's spatial elements include shape, size, orientation and texture. Spectral elements includes the brightness values. To interpret spectral elements, the spectral reflectance of typical ground cover types and how those characteristics are sampled by the sensor have to be known.

Generally photointerpretation requires some preprocessing of the images. This preprocessing may be done in two categories: radiometric enhancement and geometric enhancement. Radiometric enhancement includes altering the contrast range occupied by the pixels. Only point operations are used to do that. Geometric enhancement includes smoothing noise, enhancing and highlighting edges, detecting and enhancing lines. Performing these operations require usage of a specified neighborhood of the pixels. These operations are usually time consuming to evaluate. Radiometric enhancement changes spectral and spatial characteristics of the image data while the geometric enhancement only changes spatial characteristics.

3.1 Radiometric Enhancement

Remote sensors record reflected and emitted radiant flux exiting from Earth's surface materials. Ideally, one material would reflect a tremendous amount of energy in a certain wavelength, while another material would reflect much less energy in the same wavelength. This would result in contrast between the two types of materials when recorded by a remote sensing system. Unfortunately, different materials often reflect similar amounts of radiant flux throughout the visible, near-infrared, and mid-infrared portion of the electromagnetic spectrum, resulting in a relatively low contrast image.

To improve the contrast of the digital remotely sensed data, it is desirable to utilize the entire brightness range of the display medium, which is generally a video display or hard copy output device. There are *linear* and *nonlinear* digital contrast enhancement techniques.

3.1.1 Linear Contrast Enhancement

Contrast enhancement (or contrast stretching) expands the original input brightness values to make use of the total range of the output device. To make a decision about the contrast of the image data its histogram is used. If each pixel in the image is examined and its brightness value noted, a graph of number of pixels with a given brightness versus brightness value can be constructed. This is referred to as histogram of the image. A 24 bit color image includes 3 components which are blue, green and red. Each of these components uses 8 bits to show different brightness values. This means that 256 levels may be used, from 0 to 255, in each band. A 0 brightness value means darkness and a 255 brightness value means lightness. If only one of these bands is considered, whose range is 255, highest value minus lowest value, and in the original image the range is 100; then a limited portion of the range is used. This limitation can be seen from the histogram of the image. For example if your minimum brightness value is 0 and the highest brightness value is 100, then you have a dark image. If your minimum brightness value is 155 and the highest brightness value is 255, then you have a bright image. But both of them are low contrast images. It is difficult to visually interpret such images. A more useful display can be produced if the range of original brightness values are expanded to use the full dynamic range of the video display.

Linear contrast enhancement is the best applied to remotely sensed images with Gaussian or near-Gaussian histograms, that is when all brightness values fall generally within a single, relatively narrow range of the histogram and only one mode is apparent. Unfortunately, this is rarely the case especially for scenes that contain both land and water bodies. To perform a linear contrast enhancement, the analyst examines the image statistics and determines the minimum and maximum brightness values in the band, min_k and max_k, respectively. The output brightness value BV_{out}, is computed according to the equation

$$BV_{out} = \left(\frac{BV_{in} - min_k}{max_k - min_k} \right) quant_k \tag{3.1}$$

where BV$_{in}$ is the original input brightness value and quant$_k$ is the range of brightness values that can be displayed on the video display. There are many approaches used to select min$_k$ and max$_k$:

- **Min-max Contrast Stretch:** For an 8 bit color system 0 is used as min$_k$ and 255 is used as max$_k$. If the image involves minimum value of 4 and maximum value of 105, then in the new image all 4 values become 0 and all 105 values become 255.
- **Percentage Linear Contrast Stretch:** min$_k$ and max$_k$ that lie a certain percentage of pixels from the mean of the histogram may be specified. For example setting the minimum and maximum ±1 standard deviation ($\pm 1\sigma$) from the mean. Values smaller than the min$_k$, which is now mean-1σ, go to 0, while values larger than the max$_k$ go to 255.
- **Specific Percentage Linear Contrast Stretch (Saturating Linear Contrast Enhancement [5]):** Sometimes it may be more useful for the user to be able to select min$_k$ and max$_k$ freely. For example if in the image data the wetlands are interested in and the brightness values from 13 to 27 shows those areas, min$_k$ is chosen as 13 and max$_k$ is chosen as 27. Then as in the percentage linear contrast stretch, values smaller than min$_k$ go to 0 and values larger than max$_k$ go to 255.
- **Piecewise Linear Contrast Stretch:** When the histogram of an image is not Gaussian in nature (i.e., it is bimodal, trimodal, etc.), it is possible to perform

a piecewise linear contrast stretch. Here the analyst identifies a number of linear enhancement steps that expand the brightness ranges in the modes of the histogram. In effect, this corresponds to setting up a series of $\min_k$ and $\max_k$ and using the above equation within user-selected regions of the histogram.

3.1.2 Nonlinear Contrast Enhancement

Two methods will be examined in this section; *histogram equalization* and *logarithmic contrast enhancement.* Histogram equalization applies the greatest contrast enhancement to the most populated range of the brightness values in the image. It automatically reduces the contrast in the very light and very dark parts of the image. Logarithmic and exponential mappings of brightness values between original and modified images are useful for enhancing dark and light features respectively.

- **Histogram Equalization:** In many situations it is desirable to modify the contrast of an image so that its histogram matches a preconceived shape. A particular and important modified shape is the uniform histogram in which, in principle, each bar has the same height. Such a histogram has associated with it an image that utilises the available brightness levels equally and thus should give a display in which there is a good representation of the detail at all brightness values. In practice a perfectly uniform histogram cannot be achieved for digital image data; the procedure following however produces a histogram that is quasi-uniform on the average. The method of producing a uniform histogram is known generally as histogram equalization.

  Let $h_i(x)$ be the histogram function of the original image and $h_o(y)$ represent the modified histogram, which is to be uniform. If the image contains a total of N pixels and there are L brightness values, then each of the brightness values in the modified histogram should have a bar of *N/L* pixels associated with it. Recall also that the bars in a discrete histogram have the values $h_o(y)\mathrm{d}y$. In the case of L available brightness values, dy = *(L - 1)/L* so that for a uniform histogram

$$h_o(y)(L-1)/L = N/L \tag{3.2}$$

giving $h_o(y) = N/(L-1)$. From

$$h_o(y) = h_i(x)\frac{dx}{dy} \tag{3.3}$$

therefore

$$\frac{dy}{dx} = \frac{d}{dx}\{f(x)\} = \frac{L-1}{N}h_i(x) \tag{3.4}$$

in which y = *f(x)* is the sought-for mapping or transformation of brightness values that takes the original histogram of an image into a uniform histogram. Consequently,

$$y = f(x) = \frac{L-1}{N}\int h_i(x)dx \tag{3.5}$$

The histogram equalization transform therefore is the integral of the original histogram function times a scaling factor. The integral is just the continuous cumulative histogram; this can be replaced by the discrete cumulative histogram in the case of imagery with quantised brightness values, in which case Equation (3.5) will yield a look-up table that can be used to move histogram bars to new brightness value locations.

- **Logarithmic Contrast Enhancement:** Logarithmic mapping of brightness values between original and modified images are useful for enhancing dark light features. It could be reversed to enhance bright features mapping the brightness values using an inverse log function.

The selection of a contrast enhancement algorithm depends on the nature of the original histogram and the elements of the scene that are of greatest interest to the user. An experienced image analyst can usually identify an appropriate contrast enhancement algorithm by examining the image histogram and then experimenting until satisfactory results are obtained. Most contrast enhancements cause some useful information to be lost. However, that which remains should be of value. Contrast enhancement is applied primarily to improve visual image analysis. Contrast stretching can distort the original pixel values, often in a nonlinear fashion.

3.2 Geometric Enhancement

Geometric enhancements of most interest in remote sensing generally relate to smoothing, edge detection and enhancement, and line detection. Enhancement of edges and lines leads to image sharpening. Each of these operations is considered in the following sections. Most of the methods to be presented are, or can be expressed as, template techniques in which a template, box or window is defined and then moved over the image row by row and column by column. The products of the pixel brightness values, covered by the template at a particular position, and the template entries are taken and summed to give the template response. This response is then used to define a new brightness value for the pixel currently at the centre of the template. When this is done for every pixel in the image, a radiometrically modified image is produced that enhances or smooths geometric features according to the specific numbers loaded into the template. A 3x3 template is illustrated in Figure (3.1). Templates of any size can be defined, and for an M by N pixel sized template, the response for image pixel *i, j* is

$$r(i,j) = \sum_{m=1}^{M}\sum_{n=1}^{N}\phi(m,n)t(m,n) \tag{3.6}$$

where *ϕ(m, n)* is the pixel brightness value, addressed according to the template position and *t(m, n)* is the template entry at that location. Often the template entries collectively are referred to as the 'kernel' of the template and the template technique generally is called convolution, in view of its similarity to time domain convolution in linear system theory [5].

3.2.1 Smoothing

Smoothing operations are also known as low pass filtering operations. There are two kind of smoothing operations, *mean value smoothing* and *median filtering*:

- **Mean Value Smoothing:** Images can contain random noise superimposed on the pixel brightness values owing to noise generated in the transducers which acquire the image data, systematic quantisation noise in the signal digitising electronics and noise added to the video signal during transmission. This will show as a speckled 'salt and pepper' pattern on the image in regions of homogeneity; it can be removed by the process of low pass filtering or smoothing, unfortunately usually at the expense of some high frequency information in the image. To smooth an image a uniform template in the Equation (3.5) is used with entries

$$t(m,n) = 1/MN \qquad for \quad all \qquad m,n \tag{3.7}$$

so that the template response is a simple average of the pixel brightness values currently within the template,

$$r(i,j) = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \phi(m,n) \tag{3.8}$$

The pixel at the centre of the template is thus represented by the average brightness level in a neighborhood defined by the template dimensions. This is an intuitively obvious template for smoothing and is equivalent to using running averages for smoothing time series information.

It is evident that high frequency information such as edges will also be averaged and lost. This loss of high frequency detail can be circumvented somewhat if a threshold is applied to the template response in the following manner. Let

$$\varrho(i,j) = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \phi(i,j) \tag{3.9}$$

then

$$\begin{aligned} r(i,j) &= \varrho(i,j) \qquad \text{if} \qquad |\phi(i,j) - \varrho(i,j)| < T \\ &= \phi(i,j) \qquad \text{otherwise} \end{aligned} \tag{3.10}$$

where T is prespecified threshold. T could be determined *a priori* based upon knowledge of or an estimate of scene signal to noise ratio.

Eliason and McEwan [8] recommend choosing the threshold as a multiple of the standard deviation of brightness within the template window. This provides better noise removal in homogeneous regions while allowing better preservation of edges and other valid high spatial frequency detail.

In principle, templates of any shape and size can be used. Larger templates give more smoothing (and greater loss of high frequency detail) whereas horizontal rectangular templates will smooth horizontal noise but leave noise and high frequency detail in the vertical direction relatively unaffected by comparison.

Commonly, smoothing by template methods is referred to as box car filtering. When based upon the Equation (3.6) it is also called mean value smoothing, or averaging.

- **Median Filtering:** Disadvantages of the thresholding method for avoiding edge deterioration are that it adds to the computational cost of the smoothing operation and T must be determined. An alternative technique for smoothing in which the edges in an image are maintained is that of median filtering. In this the pixel at the centre of the template is given the median brightness value of all the pixels covered by the template – i.e. that value which has as many values higher and lower. (For example, the median of 4, 6, 3, 7, 9, 2, 1, 8, 8 is 6, whereas the mean is 5.3).

  An application for which median filtering is well suited is the removal of impulse-like noise. This is because pixels corresponding to noise are atypical in their neighborhood and will be replaced by the most typical pixel in that neighborhood.

  Finally, it should be noted that median filtering is not a linear function of the brightness values of the image pixels. Consequently it is not a convolution operation.

3.2.2 Edge Detection and Enhancement

Edge enhancement is a particularly simple and effective means for increasing geometric detail in an image. It is performed by first detecting edges and then either adding these back into the original image to increase contrast in the vicinity of an edge, or highlighting edges using saturated (black, white or color) overlays on borders.

There are essentially three economical techniques for detecting edges using image domain techniques. These are

(i) by using an edge detecting template,

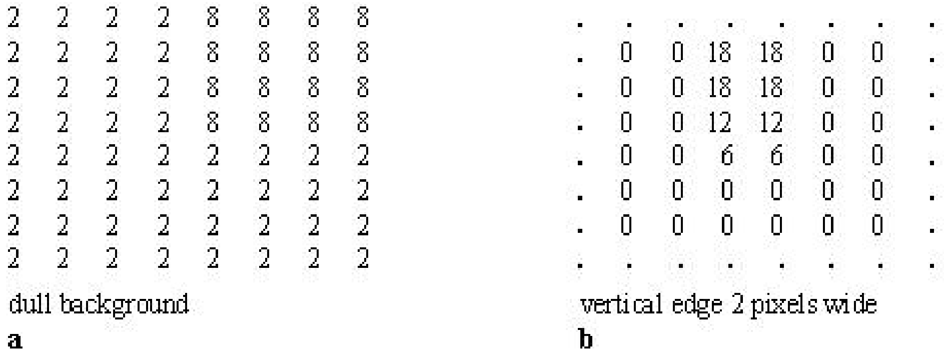
(ii) by calculating spatial derivatives, or

(iii) by subtracting a smoothed image from its original.

- **Edge Detecting Templates:** A 3x3 template that detects vertical edges in image data is

$$t(m,n) = \begin{matrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{matrix} \tag{3.11}$$

  As can be inferred from its structure it computes a value for the central pixel under the template that is the accumulated difference horizontally between pixels on three adjacent rows. To see this, consider a region of an image which is basically dull (brightness value 2) into which protrudes a bright object (brightness value 8) as depicted in Figure (3.2 a). Application of the template yields the responses shown in Figure (3.2 b), wherein the vertical edge between the object and background has been detected but not the horizontal edge. Note that the edge is defined by two columns of pixels, one on either side of the true edge position. A threshold would normally be applied to the template response (say 9 in the case of Figure (3.2)) to define the edge pixels.



2	2	2	2	8	8	8	8
2	2	2	2	8	8	8	8
2	2	2	2	8	8	8	8
2	2	2	2	8	8	8	8
2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2

dull background

a

.	.	.	.	.	.	.	.
.	0	0	18	18	0	0	.
.	0	0	18	18	0	0	.
.	0	0	12	12	0	0	.
.	0	0	6	6	0	0	.
.	0	0	0	0	0	0	.
.	0	0	0	0	0	0	.
.	.	.	.	.	.	.	.

vertical edge 2 pixels wide

b

Figure 3.1: Image **a** and edges detected by a vertically sensitive template **b**; Dots indicate indeterminate edge responses for this example

Templates for detecting edges in other orientations are:

$$\begin{array}{rrr} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{array} \qquad \begin{array}{rrr} 0 & +1 & +1 \\ -1 & 0 & +1 \\ -1 & -1 & 0 \end{array} \qquad \begin{array}{rrr} +1 & +1 & 0 \\ +1 & 0 & -1 \\ 0 & -1 & -1 \end{array}$$

horizontal *diagonal*

Clearly all four 3x3 templates have to be applied to an image to detect its edges in all orientations. This requires four passes over the image data, computing each template response for each pixel. At the completion of all processing the four template responses for each pixel are compared and the pixel labelled (as an edge in a particular direction) according to the largest template response provided that the response is also above a user specified threshold. Choosing a threshold too low will lead to many false edge counts. These contribute to noise in the processed image. Conversely, if the threshold is set too high, there will be little continuity in the detected edges.

- **Spatial Derivatives (Roberts and Sobel):** If an image consists of a continuous brightness function of a pair of continuous coordinates, x and y, say $\phi(x,y)$, then a vector gradient can be defined in the image according to

$$\nabla\phi(x,y) = \frac{\partial}{\partial x}\phi(x,y)\mathbf{i} + \frac{\partial}{\partial y}\phi(x,y)\mathbf{j} \tag{3.12}$$

where **i,j** are a pair of unit vectors. The direction of the vector gradient is the direction of maximum upward slope and its amplitude is the value of the slope. For edge detection operations usually only the magnitude of gradient, defined by

$$|\nabla| = \sqrt{\nabla_1^2 + \nabla_2^2} \tag{3.13}$$

is retained, in which

$$\nabla_1 = \frac{\partial}{\partial x}\phi(x,y) \qquad \nabla_2 = \frac{\partial}{\partial y}\phi(x,y) \tag{3.14}$$

The direction of the gradient is usually of interest only in contouring applications or in determining aspect in digital terrain models.

- **The Roberts Operator:** For digital image data, in which x and y are discretised, the continuous derivatives in the Equations (3.12) and (3.13) are replaced by differences. For example, it is possible to define

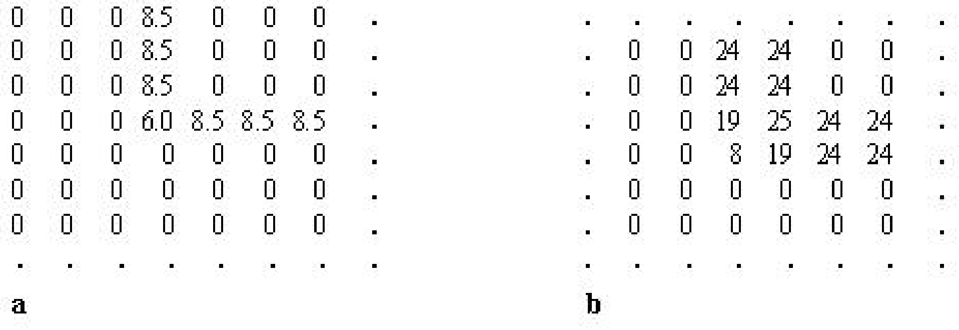
  $$\nabla_1 = \phi(i,j) - \phi(i+1,j+1) \tag{3.15}$$

  and

  $$\nabla_2 = \phi(i+1,j) - \phi(i,j+1) \tag{3.16}$$

  which are the discrete components of the vector derivative at the point $i+\frac{1}{2}$, $j+\frac{1}{2}$, in the diagonal directions. This estimate of gradient is called the Roberts operator, and is by definition associated with the pixel *i,j*.

  Application of the Roberts operator to the model image of Figure (3.2 a) yields the results shown in Figure (3.3 a), in which it will be seen that both horizontal and vertical edges are detected, as will be diagonal edges. Since this procedure computes a local gradient it is necessary to choose a threshold value above which edge gradients are said to occur. This is usually chosen with experience of a particular image. Frequently however it is useful to produce gradient maps in which pixels, for which the local gradient lies within prespecified upper and lower bounds, are displayed. Conventionally, the responses are placed to the left and upper sides of the edges.



0	0	0	8.5	0	0	0	.
0	0	0	8.5	0	0	0	.
0	0	0	8.5	0	0	0	.
0	0	0	6.0	8.5	8.5	8.5	.
0	0	0	0	0	0	0	.
0	0	0	0	0	0	0	.
0	0	0	0	0	0	0	.
.	.	.	.	.	.	.	.

a

.	.	.	.	.	.	.	.
.	0	0	24	24	0	0	.
.	0	0	24	24	0	0	.
.	0	0	19	25	24	24	.
.	0	0	8	19	24	24	.
.	0	0	0	0	0	0	.
.	0	0	0	0	0	0	.
.	.	.	.	.	.	.	.

b

Figure 3.2: Response of **a** the Robert's operator and **b** the Sobel operator to the model image data of Figure (3.2 a). Dots are indeterminate responses from edge pixels

– **The Sobel Operator:** A better edge estimator than the Roberts operator is the Sobel operator, which computes discrete gradient in the horizontal and vertical directions *at* the pixel location *i,j.* For this, which is clearly more costly to evaluate, the orthogonal components of gradient are

$$\begin{aligned} \nabla_1 = &\{\phi(i-1,j+1) + 2\phi(i-1,j) + \phi(i-1,j-1)\} \\ &-\{\phi(i+1,j+1) + 2\phi(i+1,j) + \phi(i+1,j-1)\} \end{aligned} \tag{3.17}$$

and

$$\begin{aligned} \nabla_2 = &\{\phi(i-1,j+1) + 2\phi(i,j+1) + \phi(i+1,j+1)\} \\ &-\{\phi(i-1,j-1) + 2\phi(i,j-1) + \phi(i+1,j-1)\} \end{aligned} \tag{3.18}$$

Applying this to the example of Figure (3.2 a) produces the responses shown in Figure (3.3 b). Again, both horizontal ad vertical edges are detected as will be edges on a diagonal slope. As before, a threshold on the responses is generally chosen to allow an edge map to be produced in which small responses, resulting from noise or minor gradients, are suppressed. Also gradient maps can be produced illustrating regions in which the local slope lies within user specified bounds.

It can be seen that the Sobel operator is equivalent to simultaneous application of the templates:

$$\nabla_1 = \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix} \qquad \nabla_2 = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \tag{3.19}$$

- **Subtracting a Smoothed Image From Its Original:** While treated in the context of edge enhancement this technique really leads to the enhancement of all high spatial frequency detail in an image including edges, lines, and points of high gradient. It is probably better regarded therefore as a sharpening technique.

A smoothed image retains all low spatial frequency information but has its high frequency features, such as edges and lines, attenuated (unless edge preservation procedures such as thresholding are employed). Consequently, if a smoothed image is subtracted from its original the resultant difference image will have only the edges and lines substantially remaining.

The difference operation to create a high spatial frequency image can give negative brightness values. Provided the image is not displayed, this produces no problems. For display however it is common to scale the difference image such that a zero difference is displayed as mid-grey with positive differences towards white and negative differences towards black. When the difference image is added back to the original, negative brightnesses can again result. Again, this can be handled by level shifting or scaling, or simply by setting negative brightness values to zero.

3.2.3 Line Detection

Lines are extended edges. Analysts may use *linear* and *nonlinear* line detecting templates mentioned below for line detection:

- **Linear Line Detecting Templates:** Line features such as rivers and roads in satellite images can be detected as pairs of edges if they are more than one pixel wide or alternatively, if they are a single pixel in width, they can be detected using the following line detecting templates:

  $$\begin{array}{ccc@{\qquad}ccc@{\qquad}ccc@{\qquad}ccc}
  -1 & 2 & -1 & -1 & -1 & -1 & -1 & -1 & 2 & 2 & -1 & -1 \\
  -1 & 2 & -1 & 2 & 2 & 2 & -1 & 2 & -1 & -1 & 2 & -1 \\
  -1 & 2 & -1 & -1 & -1 & -1 & 2 & -1 & -1 & -1 & -1 & 2
  \end{array}$$

  vertical *horizontal* *diagonal*

  These templates seem not to have been used to any great extent in remote sensing image processing since lines, in addition to edges, are enhanced using the gradient and subtractive smoothing techniques of the previous section.

- **Nonlinear and Semi-linear Line Detecting Templates:** The line detecting templates mentioned above are regarded as linear since their convolution with image data is a linear mathematical operation. Some nonlinear line detecting template operations have also been proposed. To describe these it is of value to denote a 3x3 neighborhood of pixels in an image as

$$\begin{matrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{matrix}$$

A nonlinear line detector algorithm, proposed by Rosenfeld and Thurston [9] establishes pixel B_2 as part of a dark vertical line if

$$A_1, C_i > B_i \tag{3.20}$$

by a prespecified threshold. Similar expressions apply for lines of other orientations and for bright lines on dark backgrounds.

Vanderbrug [10] has proposed what he calls a semilinear detector. For the pixel array above this determines B_2 as part of a dark vertical line if

$$\sum_{i=1}^{3} A_i \qquad \text{and} \qquad \sum_{i=1}^{3} C_i > \sum_{i=1}^{3} B_i \tag{3.21}$$

by some prespecified threshold.

Gurney [11] has noted that the semilinear detector works better than the non linear algorithm although line thickening results and computational cost is high. These disadvantages are obviated by the use of the additional constraint with the semilinear algorithm:

$$A_2 > B_2 \qquad \text{and} \qquad C_2 > B_2 \tag{3.22}$$

Gurney also discusses means by which the thresholds for the semilinear detector can be effectively established.

3.3 The Fourier Transform

Fourier analysis is mathematical technique for separating an image into its various spatial frequency components. First, let us consider a continuous function *f(x)*. The Fourier theorem states that any function *f(x)* can be represented by a summation of a series of sinusoidal terms of varying spatial frequencies. These terms can be obtained by the Fourier transform of *f(x)*, which is written as:

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi iux} dx \tag{3.23}$$

where u is spatial frequency. This means that *F(u)* is a frequency domain function. The spatial domain function *f(x)* can be recovered from *F(u)* by the inverse Fourier transform

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{2\pi iux} du \tag{3.24}$$

To utilize Fourier analysis in digital image processing, two extensions of these equations must be considered. First, both transforms can be extended from one dimensional functions to two-dimensional functions *f(x, y)* and *F(u, v)* [12]. For Equation (3.23) this becomes

$$F(u,v) = \int_{-\infty}^{\infty} \int f(x,y) e^{-2\pi i(ux+vy)} dxdy \tag{3.25}$$

Furthermore, both transforms to discrete functions can be extended. The two-dimensional discrete Fourier transform is written as

$$F(u,v) = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) e^{-2\pi i(\frac{ux}{N} + \frac{vy}{M})} \tag{3.26}$$

where N is the number of pixels in the x direction and M is the number of pixels in the y direction. Every remotely sensed image may be described as a two-dimensional discrete function. Therefore, Equation (3.26) may be used to compute the Fourier transform of an image. The image can be reconstructed using the inverse transform.

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F(u,v) e^{2\pi i (\frac{ux}{N} + \frac{vy}{M})} \tag{3.27}$$

The *F(u, v)* contains the spatial frequency information of the original image *f(x, y)* and is called the *frequency spectrum.* Note that it is a complex function because it contains i which equals $\sqrt{-1}$. Any complex function can be written as the sum of a real part and an imaginary part.

$$F(u,v) = R(u,v) + iI(u,v) \tag{3.28}$$

which is equivalent to

$$F(u,v) = |F(u,v)| e^{i\phi(u,v)} \tag{3.29}$$

where $|F(u,v)|$ is a real function and

$$|F(u,v)| = \sqrt{R(u,v)^2 + I(u,v)^2} \tag{3.30}$$

$|F(u,v)|$ is called the magnitude of the Fourier transform and can be displayed as a two-dimensional image. It represents the magnitude and the direction of the different frequency components in the image *f(x,y).* The variable ϕ in Equation (3.29) represents *phase* information in the image *f(x, y).* Although the phase information is usually ignored when the Fourier transform is displayed, the original image cannot be recovered without it.

It is important to remember that the Fourier transformed image *F(u, v)* contains all the information found in the original image. It provides a mechanism for analyzing and manipulating images according to their spatial frequency. It is useful for image restoration, filtering, and radiometric correction. For example, the Fourier transform can be used to remove periodic noise in remotely sensed data. When the pattern of periodic noise is unchanged throughout the image, it is called stationary periodic noise. Striping in remotely sensed imagery is usually composed of stationary periodic noise.

3.3.1 Spatial Filtering In Frequency Domain

Filtering in the spatial domain using convolution filters is discussed. It can also be performed in the frequency domain. Using the Fourier transform, the frequency information of the image can be directly manipulated. The manipulation can be performed by multiplying the Fourier transform of the original image by a mask image called frequency domain filter, which will block or weaken certain frequency components by making the values or certain parts of the frequency spectrum become smaller or even zero. Then the inverse Fourier transform of the manipulated frequency spectrum is computed to obtain a filtered image in the frequency domain. Numerous algorithms are available for computing the Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) [13]. Spatial filtering in the frequency domain generally involves computing the FFT of the original image, multiplying the FFT of a convolution mask of the analyst's choice (e.g., a low pass filter) with the FFT, and inverting the resultant image with the IFFT; that is,

$$f(x,y) \xrightarrow{FFT} F(u,v) \rightarrow F(u,v)G(u,v) \rightarrow F'(u,v) \xrightarrow{IFFT} f'(x,y)$$

The convolution theorem states that the convolution of two images is equivalent to the multiplication of their Fourier transformations. If

$$f'(x,y) = f(x,y) * g(x,y) \tag{3.31}$$

where * represents the operation of convolution, *f(x, y)* is the original image and g(x, y) is a convolution mask filter, then

$$F'(u,v) = F(u,v)G(u,v) \tag{3.32}$$

where F', F, and G are Fourier transforms of f', *f*, and g, respectively.

An example of such manipulation is shown Figure (3.3) [6]. A high-pass filter were used to construct the filter function *g(x, y)* in Figure (3.3). In practice, one problem must be solved. Usually, the dimensions of *f(x, y)* and *g(x, y)* are different; for example, the high-pass filter in Figure (3.3) only has nine elements, while the image is composed of 128 x 128 pixels. Operation in the frequency domain requires that the sizes of *F(u, v)* and *G(u, v)* be the same. This means the sizes of f and g must be made the same because the Fourier transform of an image as the same size as the original image. The solution of this problem is to construct *g(x, y)* by putting the convolution mask at the center of a zero-value image that has the same size as *f*. The multiplication of Fourier transforms *F(u, v)* and *G(u, v)* results in a new Fourier transform, $F'(u,v)$. Computing the inverse fast Fourier transform yields $f'(x,y)$, a filtered version of the original image. Thus, spatial filtering can be performed both in the spatial and frequency domain.

As demonstrated, filtering in the frequency domain involves one multiplication and two transformations. For general applications, convolution in the spatial domain may be more cost effective. Only when the size of *g(x, y)* is very large, does the

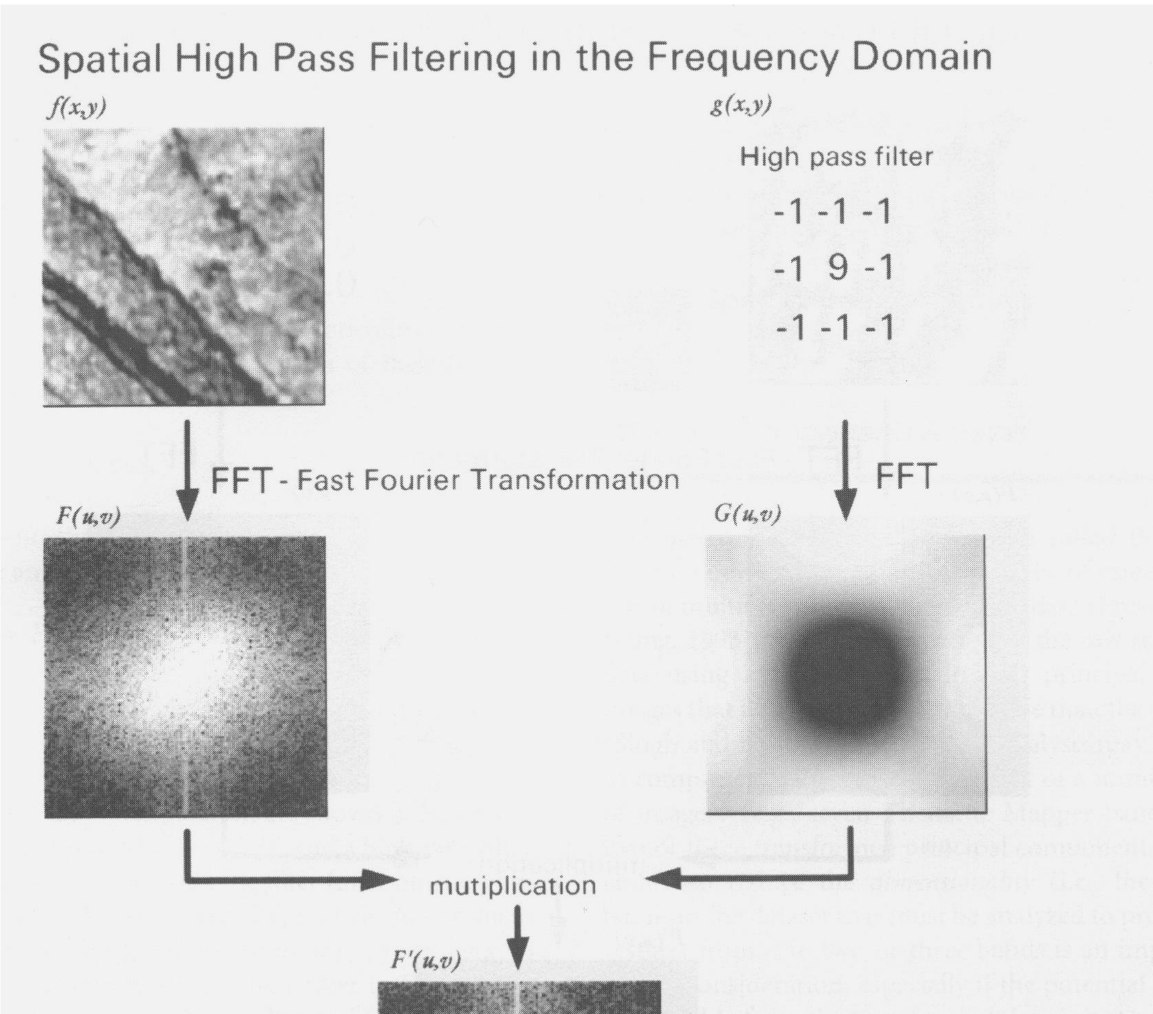

Spatial High Pass Filtering in the Frequency Domain
f(x,y)
g(x,y)
High pass filter
-1 -1 -1
-1 9 -1
-1 -1 -1
FFT - Fast Fourier Transformation
FFT
F(u,v)
G(u,v)
mutiplication
F'(u,v)
IFFT - Inverse Fast Fourier Transformation
f'(x,y)

Figure 3.3: Spatial high-pass filtering in the frequency domain using a Fourier transform [6]

Fourier method become cost effective. However, with the frequency domain method some filtering can also be done that is not easy to do in spatial domain. A frequency domain filter *G(u, v)* specifically designed to remove certain frequency components in the image may be constructed.

CHAPTER 4

IMPLEMENTATION AND PARALLELIZATION

Application development phase of the thesis started with the programming and testing of sequential algorithms mentioned in Chapter 3. First linear (min-max contrast stretching, percentage linear contrast stretching and specific percentage linear contrast stretching) and nonlinear contrast enhancement algorithms are coded. Then geometric enhancement (smoothing, edge detection and enhancement, line detection) algorithms are coded. Most of the sequential programs are joined with a graphical user interface (GUI).

All of the programs are written using C programming language and libraries on C. To form the GUI, Motif [14] toolkit and EasyMotif [15] are used. A pretentious Fourier transform library: "Fastest Fourier Transform in the West" (FFTW) [16] is used to calculate the forward and inverse Fourier transforms. MPI [17] library is used to develop parallel programs.

FFTW is a portable C package for computing the one and multidimensional discrete Fourier transform (DFT). Extensive benchmarking demonstrates that FFTW is typically faster than all other publicly available DFT software [18], including the well-known FFTPACK [19] and the code from Numerical Recipes [20]. More interestingly, FFTW is competitive with or better than proprietary, highly-tuned codes such as Suns Performance Library and IBMs ESSL library. FFTW is an implementation of the Cooley-Tukey [21] fast Fourier transform (FFT)

Three main ideas are the keys to FFTWs performance [18]. First, the computation of the transform is performed by an executor consisting of highly-optimized, composable blocks of C code called codelets. Second, at runtime, a planner finds an efficient way (called a plan) to compose the codelets. Through the planner, FFTW adapts itself to the architecture of the machine it is running on. In this way, FFTW is a single program that performs efficiently on a variety of architectures. Third, the codelets are automatically generated by a codelet generator written in the Caml Light dialect of ML [22]. The codelet generator produces long, optimized, unreadable code, which is nevertheless easy to modify via simple changes to the generator.

FFTW includes both sequential and parallel routines for the calculation DFT. This is another reason why it is preferred in that application. By using FFTW both sequential and parallel versions of spatial filtering in frequency domain method explained in Chapter 3 are implemented.

The *sequential* program that is developed for the spatial filtering in frequency domain works as follows:

```
Read input image from file (in JPEG format).
Allocate space for input image data.
Allocate space for R, G, B color components of input image and
filter.
Read filter from file.
Separate input image into its color components.
Deallocate space for input image data.
Allocate space for responses of color components to filtering.
/* Perform filtering operation for the first color component */
```

```
Using FFTW library routines find the Fourier transforms of first
color component and filter.
Multiply transformed matrices.
Take inverse Fourier transform of the multiplication to return
to spatial domain.
/* Perform filtering operation for the second color component */
Using FFTW library routines find the Fourier transforms of second
color component and filter.
Multiply transformed matrices.
Take inverse Fourier transform of the multiplication to return
to spatial domain.
/* Perform filtering operation for the third color component */
Using FFTW library routines find the Fourier transforms of third
color component and filter.
Multiply transformed matrices.
Take inverse Fourier transform of the multiplication to return
to spatial domain.
Deallocate space for color components and filter.
Allocate space for output image.
Combine filtered colur components to produce output image.
Deallocate space for responses of color components.
Save the output image into file (in JPEG format).
Deallocate space for output image data.
```

The *parallel* program that is developed for the spatial filtering in frequency domain works as follows:

```
/*** MASTER ***/
Read input image from file (in JPEG format).
Allocate space for input image data.
Allocate space for R, G, B color components of input image and
filter.
Read filter from file.
Separate input image into its color components.
Deallocate space for input image data.
RECEIVE starting points and strip sizes of nodes.
/*** NODES ***/
SEND starting point and strip size.
/*** MASTER ***/
SEND necessary data strip of first color component and filter to
each node.
Deallocate space for the first color component.
/*** NODES ***/
Calculate Fourier transform of image data and filter using FFTW's
```

```
parallel routines.
Multiply transformed sub matrices.
Take inverse Fourier transform of the multiplication to return to spatial domain.
SEND filtered data to the master.
/*** MASTER ***/
GET filtered data strips and merge them to produce filtered first color component.
SEND necessary data strip of second color component and filter to each node.
Deallocate space for the second color component.
/*** NODES ***/
Calculate Fourier transform of image data and filter using FFTW's parallel routines.
Multiply transformed sub matrices.
Take inverse Fourier transform of the multiplication to return to spatial domain.
SEND filtered data to the master.
/*** MASTER ***/
GET filtered data strips and merge them to produce filtered second color component.
SEND necessary data strip of third color component and filter to each node.
Deallocate space for the third color component.
Deallocate space for the filter.
/*** NODES ***/
Calculate Fourier transform of image data and filter using FFTW's parallel routines.
Multiply transformed sub matrices.
Take inverse Fourier transform of the multiplication to return to spatial domain.
SEND filtered data to the master.
/*** MASTER ***/
GET filtered data strips and merge them to produce filtered third color component.
Combine filtered color components to produce output image.
Deallocate space for responses of color components.
Save the output image into file (in JPEG format).
Deallocate space for output image data.
```

CHAPTER 5

RESULTS AND DISCUSSIONS

The sequential version of the application of spatial domain filtering in frequency domain is tested using master PC on the cluster. Parallel version is completely tested up to 12 processing units including the master PC.

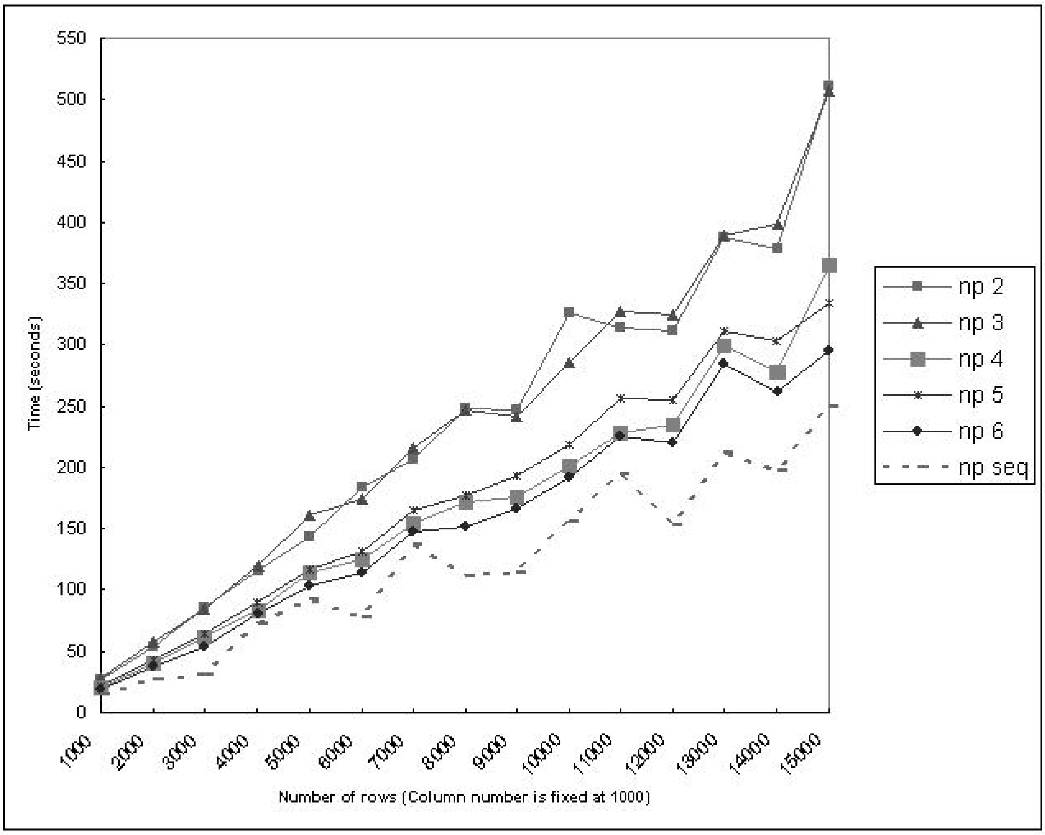


Figure 5.1: Run times (seconds) vs number of rows for different number of processors (np 2 to 6) and sequential (seq)

Figures (5.1) and (5.2) shows run times (seconds) versus number of rows for differ-

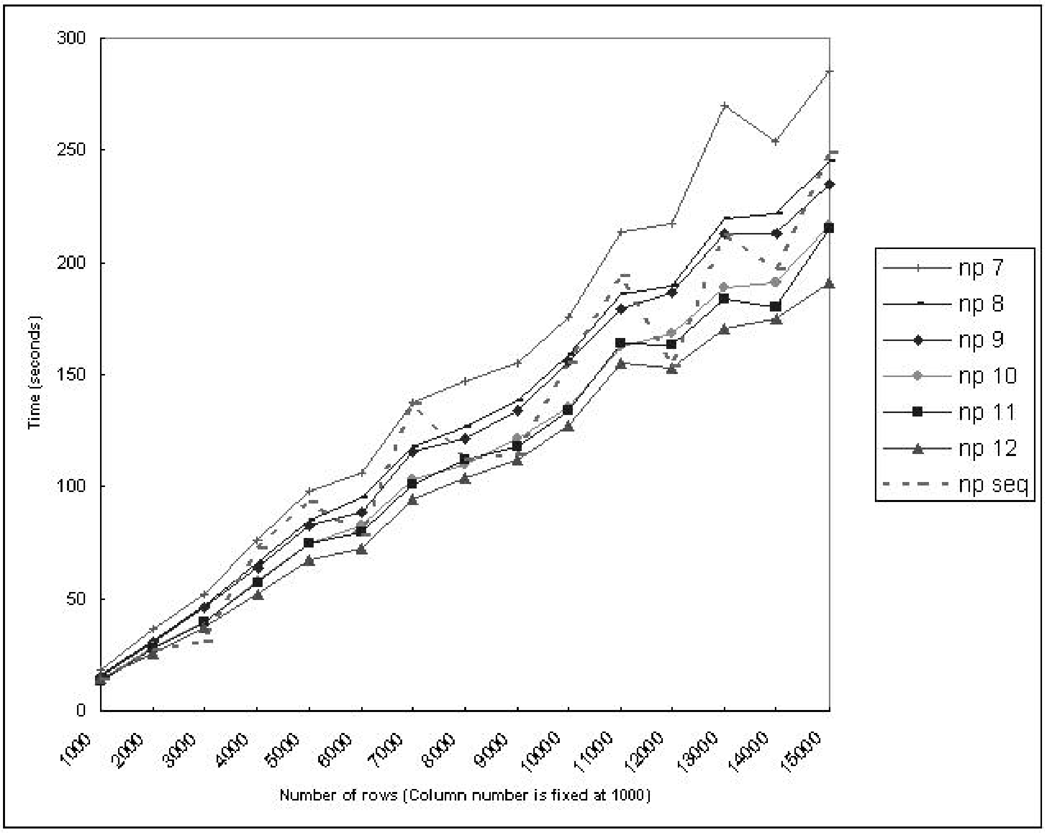


Figure 5.2: Run times (seconds) vs number of rows for different number of processors (np 7 to 12) and sequential (seq)

ent number of processors. During the tests 1000 is always used as column number of the image and the number of rows is changed from 1000 to 15000. The first advantage is taken at row number 4000 and number of processors is 8. At that point sequential run time is 72,6 seconds while parallel run time is 65,7 seconds. After that point on the parallel runs are usually advantageous. However, as seen at Figures (5.1) and (5.2) the sequential run times makes sharp decreases at row numbers 6000, 8000, 10000, 14000. For 12 processors nearly all values, except for row numbers 1000 and 3000, are better than sequential program. Although, the efficiency is low, the memory usage of machines extremely decreases in the parallel version. For example, in the case of 12 nodes and image size is 15000 x 1000; the maximum memory usage at the master node is 191 MB and at any other node is 62 MB. With the usage of sequential program maximum memory usage becomes 1030 MB at one PC. This differences shows that using parallel version gives us the ability of processing much bigger images.

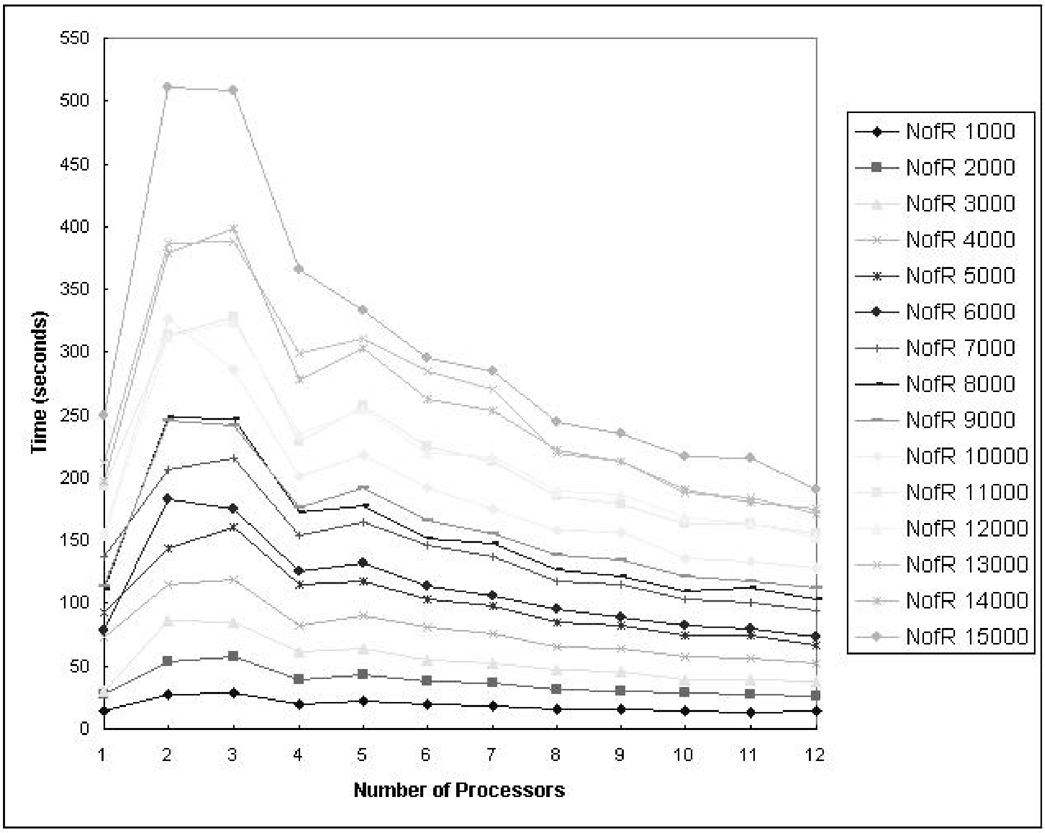


Figure 5.3: Run times (seconds) vs number of processors for different number of rows

Figure (5.3) shows run times (seconds) versus number of processors for different number of rows. As mentioned above the advantageous of parallel application starts at 8 number of processors. The upper line at Figure (5.3) shows the image of size 15000 x 1000. The sequential run for that image takes nearly 250 seconds. The parallel run starts from about 510 seconds and quickly decreases when the number of processors increased.

Figure (5.4) shows speedup values versus number of processors for different number of rows. At this graph the upper line shows the speedup values for the image size of 7000 x 1000. The lower line shows the speedup values for the image size of 3000

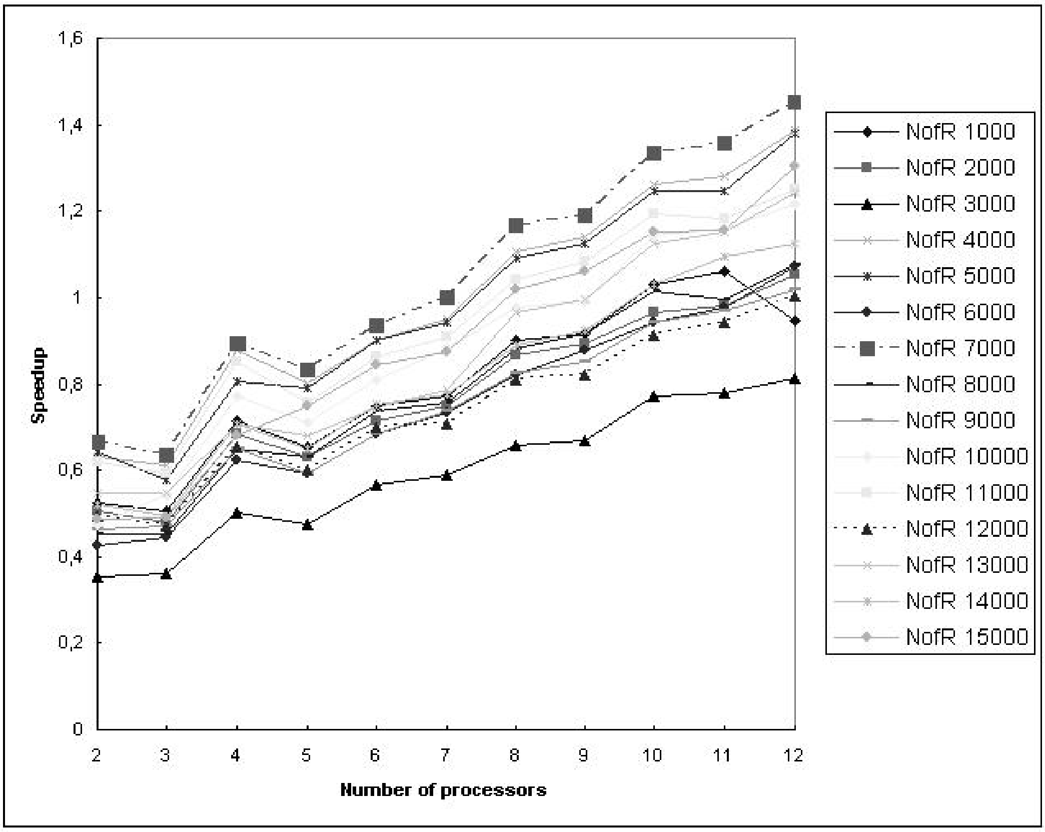


Figure 5.4: Speedup vs number of processors for different number of rows

x 1000. For all number of processors the image size of 7000 x 1000 gives the best speedup and the image size of 3000 x 1000 gives the worst speedup.

CHAPTER 6

CONCLUSION

Remote sensing is a way of collecting data about something that is interested in. Images are not all of the products of remote sensing, but they are the most useful ones. The use of image data requires the techniques of another area, image processing. Every image processing application includes a preprocessing step which covers the enhancement of acquired image data.

This thesis includes a complete survey of techniques used in the area of remote sensing for image enhancement and parallelization efforts on a subdivision of these techniques. Also in Chapter 1 a brief explanation is given about remote sensing that will be helpful before performing in this area.

As the test results show, there is a limited time advantage with the usage of parallelization on the method of spatial domain filtering using Fourier transform. However, parallelization has the advantage of using memory resources more efficiently. This advantage provides us to apply this technique on images larger than that can be used in the sequential version.

Finally it can be claimed easily, the parallelization of spatial domain filtering using Fourier transform is poor according to speedup and efficiency criterions of parallel computation.

During this study FFTW is used for the calculation of Fourier transform for both sequential and parallel versions of the application. Although FFTW has some fundamental advantages, as adaptation to the infrastructure of the system, for the calculation of Fourier transform; the parallel routines of the library are not as advantageous as the serial routines of the library. For the future work of this study, both serial and parallel versions of Fourier transform may also be implemented. This extra work will give us flexibility for load balancing.

REFERENCES

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, United States of America, 1993.

[2] T. M. Lillesand and R. W. Kiefer, *Remote Sensing and Image Interpretation*, John Wiley and Sons Inc., United States of America, 1979.

[3] F. F. Jr. Sabins, *Remote Sensing Principles and Interpretation, 3rd Edition*, W.H. Freeman and Company, New York, 1997.

[4] http://www.ccrs.nrcan.gc.ca/ccrs/learn/tutorials/fundam/fundam_e.html.

[5] J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis*, Springer, Germany, 1999.

[6] J. R. Jensen, *Introductory Digital Image Processing*, Prentice Hall, United States of America, 1996.

[7] M. D. Steven, "Satellite Remote Sensing for Agricultural Management: Oppotunities and Logistic Constraints", *ISPRS Journal of Photogrammetry & Remote Sensing* **48(4)**, 29–34, 1993.

[8] E. M. Eliason and A. S. McEwan, "Adaptive Box Filters for Removal of Random Noise from Digital Images", *Photogrammetric Engineering and Remote Sensing* **56**, 453–458, 1990.

[9] A. Rosenfeld and M. Thurston, "Edge and Curve Detection for Visual Scene Analysis", *IEEE Trans. Computers* **C–20**, 562–569, 1971.

[10] G. J. Vanderbrug, "Line Detection in Satellite Imagery", *IEEE Trans. Geoscience Electronics* **GE–14**, 37–44, 1976.

[11] C. M. Gurney, "Threshold Selection for Line Detection Algortihms", *IEEE Trans. Geoscience and Remote Sensing* **GE–18**, 204–211, 1980.

[12] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery *Numerical Recipes in FORTRAN: The Art of Scientific Computing, 2nd ed.*, Cambridge, London, 1992.

[13] J. C. Russ *The Image Processing Handbook*, CRC Press, Boca Raton FL, 1992.

[14] http://www.ist.co.uk/motif/.

[15] http://easymotif.sourceforge.net/EasyMotif.html.

[16] http://www.fftw.org/.

[17] http://www-unix.mcs.anl.gov/mpi/.

[18] M. Frigo and S. G. Johnson, *Fastest Fourier Transform in the West (MIT–LCS–TR–728)*, Technical Report of the MIT Laboratory for Computer Science, 1997.

[19] P. N. Swarztrauber, *Vectorizing the FFTs in Parallel Computations*, Academic Press, **pp.51–83**, 1982.

[20] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing, 2nd ed.*, Cambridge University Press, New York, 1992.

[21] J.W. Cooley and J.W. Tukey, "An algorithm for the machine computation of the complex Fourier series", *Mathematics of Computation* **Vol.19**, 297–301, 1965.

[22] X. Leroy, *The Caml Light system release 0.71*, Institut National de Recherche en Informatique at Automatique (INRIA), 1996.