Digital image processing for speckle images
by Rodney D Huffman

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science
Montana State University

Abstract:
The turbulent atmosphere makes it virtually impossible to image objects using conventional stellar photography. Various digital techniques have been developed to reconstruct the actual object from a series of short exposures. A. Labeyrie discovered a way to resolve angular distances for simple star patterns by computing the average autocorrelation from the snapshots. Because all phase information is lost from computing the autocorrelation, distances may be resolved but the object cannot be imaged. Imaging techniques have been developed for both the spatial and frequency domains. The Knox-Thompson and Input-Output algorithms estimate the phase for a given modulus. Other methods, such as shift-and-add, process the speckle images in the spatial domain. The shift-and-add method has been used to obtain diffraction limited images for objects containing a point that is much brighter than the other points. The shift-and-add process sometimes fails to properly align the images, resulting in "ghost" stars and other false images. This thesis reviews various techniques and describes the process by which they may be implemented. A software package was developed and instructions are given concerning its use in implementing those techniques.

A pew method, called correlate-shift-and-add, solves the shift-and-add problem by correlating with respect to the entire image. Results have shown that correlate-shift-and-add produces a truer image than shift-and-add when the object does not contain a point that is much brighter than the others.

DIGITAL IMAGE PROCESSING TECHNIQUES

FOR SPECKLE IMAGES

by

Rodney D. Huffman

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

June 1984

APPROVAL

of a thesis submitted by

Rodney D. Huffman

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

May 24, 1984	Fredrick M Cady
Date	Chairperson, Graduate Committee

Approved for the Major Department

May 24, 1984	D. A. Pierre
Date	Head, Major Department

Approved for the College of Graduate Studies

June 5, 1984	Henry L Parsons
Date	Graduate Dean

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Director of Libraries when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without written permission.

Signature Rod Hoffman

Date 5-24-84

TABLE OF CONTENTS

TABLE OF CONTENTS--Continued

TABLE OF CONTENTS--Continued

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The turbulent atmosphere makes it virtually impossible to image objects using conventional stellar photography. Various digital techniques have been developed to reconstruct the actual object from a series of short exposures. A. Labeyrie discovered a way to resolve angular distances for simple star patterns by computing the average autocorrelation from the snapshots. Because all phase information is lost from computing the autocorrelation, distances may be resolved but the object cannot be imaged. Imaging techniques have been developed for both the spatial and frequency domains. The Knox-Thompson and Input-Output algorithms estimate the phase for a given modulus. Other methods, such as shift-and-add, process the speckle images in the spatial domain. The shift-and-add method has been used to obtain diffraction limited images for objects containing a point that is much brighter than the other points. The shift-and-add process sometimes fails to properly align the images, resulting in "ghost" stars and other false images. This thesis reviews various techniques and describes the process by which they may be implemented. A software package was developed and instructions are given concerning its use in implementing those techniques.

A new method, called correlate-shift-and-add, solves the shift-and-add problem by correlating with respect to the entire image. Results have shown that correlate-shift-and-add produces a truer image than shift-and-add when the object does not contain a point that is much brighter than the others.

CHAPTER 1

WHAT IS DIGITAL IMAGE PROCESSING?

A Historical View

Image processing has been performed on digital computers for over 20 years. NASA first used digital image processing techniques to analyze images of the lunar surface. The Surveyor missions in the 1960s produced thousands of lunar images to evaluate landing sites for later manned missions. Later, NASA applied digital imagery to view other planets. The Voyager spacecrafts were launched in 1977 and provided the first color close-up images of Saturn and Jupiter.

Digital image processing proved to be a success in space exploration, so NASA extended its use in imaging the earth's surface with the LANDSAT series, which started with the launching of LANDSAT 1 on July 23, 1972. These satellites were equipped with digital multispectral imaging systems and were originally intended for agricultural applications. Four spectral bands (green, red, and 2 near infrared) were selected based on agricultural needs, where they were found to be useful in monitoring crops and soil moisture content. Another earth-orbiting series of satellites is the GOES series that provides images used in weather forecasting and weather pattern analysis.

Digital image processing has not been limited to astronomic and atmospheric applications. Its use has been very important in medical applications as well. Digital imagery has been combined with pattern recognition techniques to classify chromosomes [15]. It has also been useful in Radiography, where X-ray images are enhanced and averaged to accentuate relevant portions in the images.

Digital Image Processing Components

A typical digital image processing system consists of a vision component, a processing component, and a display component.

Vision Component

This component involves the conversion of a real scene into a form that can be interpreted by a computer. For a digital computer, continuous images must be converted into discrete samples. There have been a number of digital camera systems developed over the years.

The two Viking spacecraft were equiped with Vidicon tube systems [15]. With this system, an electron beam scans a charged surface that is charged as a result of its sensitivity to light. The electric charge produced at a particular point is proportional to the amount of light detected at that point. Unfortunately, this system introduces a certain amount of distortion in its images. Geometric distortion is introduced as a result of the electron beam's non-linearity. "Leaking" also occurs with this system where charges deteriorate before they are scanned.

Charge Coupled Devices (CCDs) have become popular image devices. A CCD is a solid-state device comprised of a fixed array of potential wells. A potential well accumulates charge proportional to the amount of light it detects. The wells are then read sequentially through an analog-to-digital converter. CCDs have many advantages over Vidicon systems. Their spectral response is broader and they are very sensitive to small changes in light. They are also smaller and require no electron beam.

Video digitizers are used to acquire data from conventional television signals. The signal is transmitted at a high rate where one frame (typically a 512 by 525 matrix of pixels) is sent in one-thirtieth of one second. It is the task of the video digitizer to grab a frame from the analog signal, digitize it, and make the digitized form available to a host computer for processing.

Display Component

This component involves the display or presentation of an image after it has been acquired, digitized, and processed. A common visual display unit is the video monitor. A video monitor reads data directly from memory. Images are stored in memory in the form of an array composed of many digital elements called pixels. A pixel is the smallest conceptual information unit and designates a single point on the video screen. Each point on the screen maps to a unique location in memory. A video monitor reads the contents of memory and converts its digital readings to an analog signal which causes an electron beam to excite a photosensitive substance that coats the display screen. Video monitors typically refresh a full screen every thirtieth of a second. An image is usually transfered from memory in a host computer to private memory for the video monitor. The video memory can also be manipulated through commands from the host computer. What is transferred and when it is transferred is controlled by a user on the host computer.

There are several devices that can be added to a video display system to aid in input and or output. Character and vector generators make it easier for a host computer to create sophisticated graphics. Trackballs, joysticks, and tablets are typical input devices to complement the usually standard keyboard.

Hard-copy devices have been developed to acquire relatively quick results compared to film processing. However, their quality of imagery is poor. On the other hand, images can be photo-processed and printed to produce higher quality images but processing time is much longer.

Processing Component

This component processes the data acquired by the vision component. The continuous image is digitized by the vision component yielding a form that can be processed by a computer. Among the three components, this component more directly involves the topic of this paper.

A typical image as seen by a computer consists of a two dimensional array of discrete quantities. Each point in the array (called a pixel) represents a unique point in the image and their binary values denote degrees of grayness. The human eye can only resolve approximately 40 shades of gray, but a computer, with a pixel size of 8 bits, can represent 256 shades of gray. The micro-based image processing system (MIPS) in the Electrical Engineering Department at Montana State University has a gray level accuracy of 6 bits (0 to 255 in steps of 4).

Any distortion caused by the vision element can be corrected in the processing element. An example of this is where lunar images acquired by the Mariner spacecraft were corrected to compensate for the geometric distortion produced by the camera systems. In this case, a program was executed to perform a simple geometric transformation.

Images can also be enhanced to accentuate desired portions. Supervised and automatic contrasting techniques have been developed to bring out detail that is barely resolvable to the naked eye prior to processing. Most enhancing procedures involve the image histogram which is simply a distribution function of the number of pixels with respect to gray level. Contrasting techniques broadens the gray range for images that have little range, resulting in more discernable detail.

Digital image processing is not limited to image enhancement. It is a useful restoration tool as well, which is the topic of this paper. Where enhancing techniques produce distorted versions of the actual object, restoration techniques perform the contrary operation of producing a true image from one or more degraded images. If enough _a priori_ information is known about an object, images of that object can be analyzed and processed to produce images that more closely resemble the actual object. For example, if the instantaneous point spread function were known, a severely blurred image can be deconvolved to produce a truer image of the actual object. This paper discusses restoration techniques using multiple images of the same object.

Besides processing images, this element must also contain resources for storing the images to be processed. Random accessed memory (RAM) may be adequate for processing techniques that require only a single image, but techniques that process multiple images require more storage. A more practical storage medium is disk storage (hard and or floppy). Both images and programs can be stored on disk and the disks can be interchanged to provide virtually an unlimited amount of storage. A storage medium that is both inexpensive and ideal for archiving images is magnetic tape. A good digital image system should have sufficient computing and storage facilities.

Software

So far much has been discussed concerning the hardware architecture of such a system but nothing has been mentioned about the software. A typical system should include programs for enhancing images, where supervised and automatic contrasting programs can be used. Geometric transformation programs can also be included to compensate for distortion inherent in images of curved surfaces. Geometric transformations were performed on many of the space images to compensate for the distortion induced by the camera systems.

Support programs should also be a part of the software package. In many cases it is desirable to zoom in on certain portions of an image or to shift portions around on the visual display. Overlay capabilities are desirable where a user may want to superimpose images to measure the amount of correlation, or may want to superimpose a grid pattern over an image to measure distances.

Fast-Fourier-transform programs provide an excellent imaging tool for both restoration and enhancing techniques. If the system does not contain floating point hardware, efficient software routines should be written to support these programs.

Access to the various software programs should be simple, and the programs should be "user friendly". There should be as little redundancy as possible. Thus, input and output should be standardized so that related programs may be executed in sequence.

A Specific Image Processing System

The Electrical Engineering Department at Montana State University has a micro-based image processing system. The main computer system consists of an Intel 8080 microprocessor and 64K bytes of RAM memory. For I/O, the system contains a keyboard, a black-and-white video monitor, and a dual floppy disk unit. The system contains a video camera for input and a video recorder for input or output. The display element consists of a black-and-white television monitor. The system also contains a video digitizer to convert an analog signal to digital form and a digital-to-analog converter to convert a digital image to an analog signal to be transmitted to the television monitor.

To aid in processing and storing, the micro system is linked with a VAX 780 system thru a serial port. All the programs for this paper were written for the VAX system and the language used was FORTRAN 77. A diagram of the whole system is shown in figure 2.

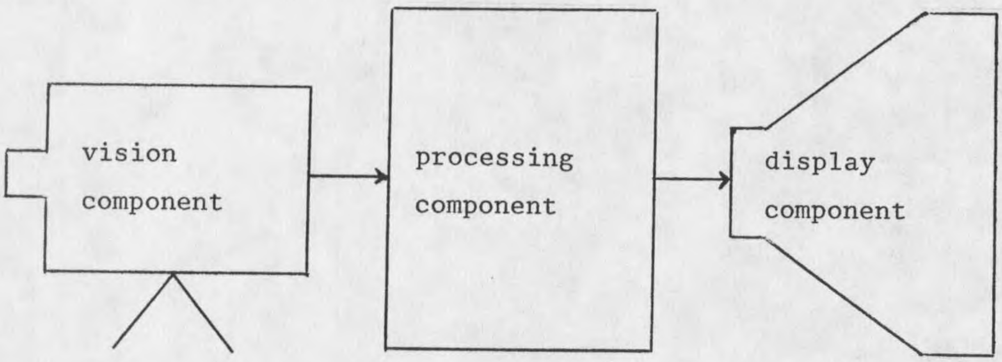


Figure 1. Conceptual diagram of an image system.

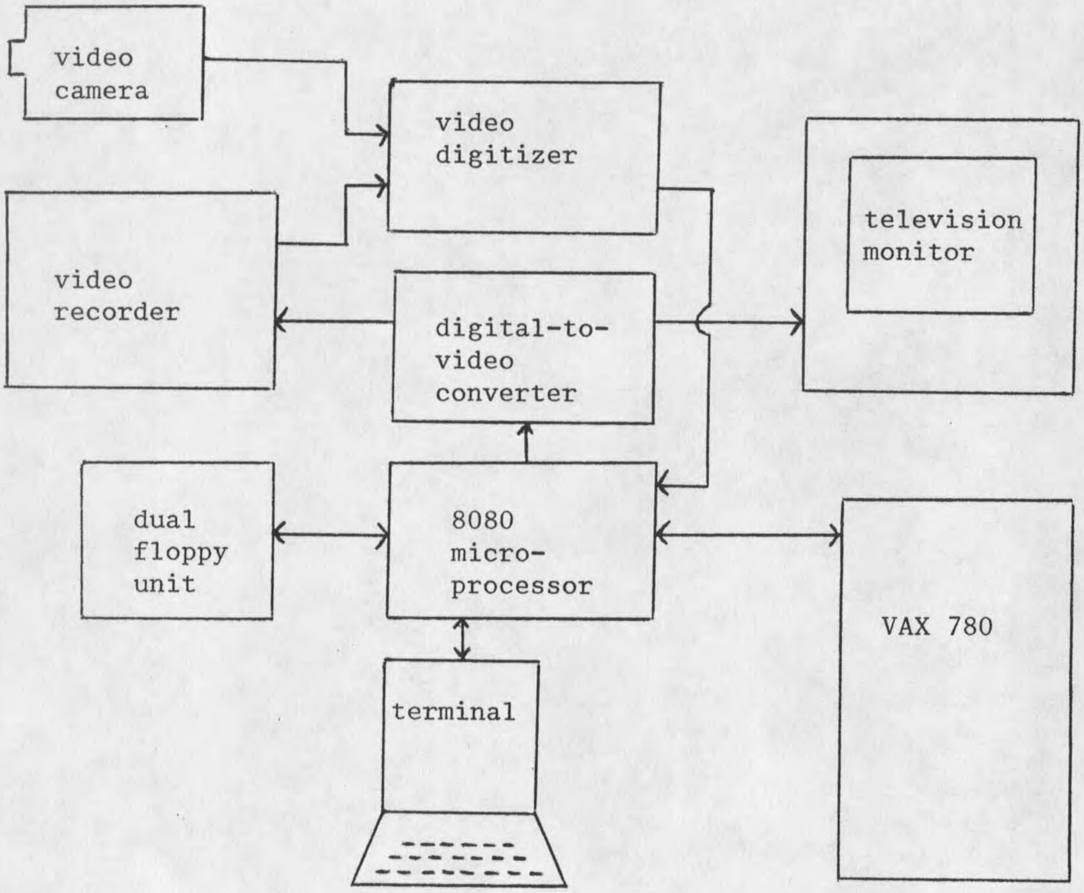


Figure 2. Diagram of the image system at Montana State University.

CHAPTER 2

IMAGING THROUGH THE ATMOSPHERE

Conventional stellar photography requires long exposure times to gather as much light as possible. A long exposure of a point star produces a severely blurred image that is much larger than the actual point star. The blurred region is often referred to as the seeing disk and is the result of the turbulent atmosphere. Images of multiple stars are so severely distorted that the actual stars cannot be resolved. However, it was discovered that short exposure photography (a few milliseconds) contains detail out to the diffraction limit of the telescope. So images that were once only resolvable through interferometry are now resolvable by taking a series of snapshots through a telescope with its aperture fully dialated. This chapter presents and discusses the major sources of distortion, namely the atmosphere and the telescope.

Fraunhofer Diffraction

Fraunhofer diffraction is a significant phenomenon in the field of astronomy. It describes the interfering effects of radiation waves as they penetrate an aperture, where the wave fronts are parallel to the plane of the aperture. Light radiation emitted from a point source, infinitely far from a circular aperture, penetrates the aperture and strikes a plate positioned at the focal point of the lens with an intensity distribution spread out in concentric rings with a bright central disk. The intensity of the diffraction pattern decreases as the distance from the center of the disk increases. The dotted line in figure 3 shows the nature of a typical diffraction pattern in one dimension.

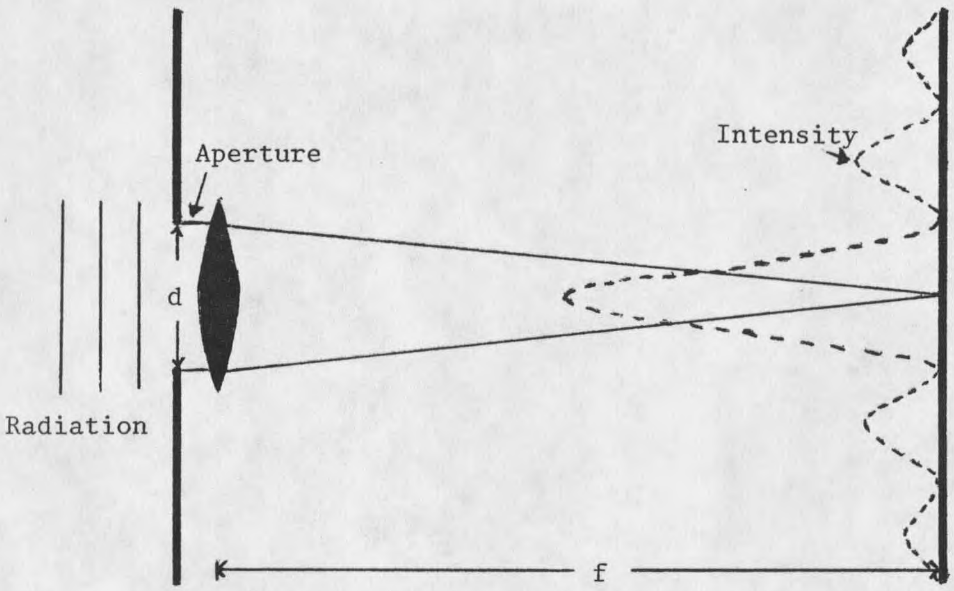


Figure 3. Fraunhofer diffraction.

More precisely, the intensity distribution is given by the formula,

$$I(\emptyset) = I(0)\left[\frac{2J_1(\pi d\emptyset/\lambda)}{\pi d\emptyset/\lambda}\right]^2 \tag{1}$$

where $\emptyset$ is the angle between the intensity line and the axis, d is the diameter of the aperture, λ is the radiation wavelength, and J_1 is the first order Bessel function given as

$$J_1(\pi d\emptyset/\lambda) = (1/2\pi)\int_0^{2\pi} \cos(\emptyset - (\pi d\emptyset/\lambda)\sin\emptyset)d\emptyset \tag{2}$$

The first dark ring is located at $\emptyset=1.22\lambda/d$, as shown in figure 4. The central disk, commonly referred to as the Airy disk, was first derived by Sir George Biddell Airy [18].

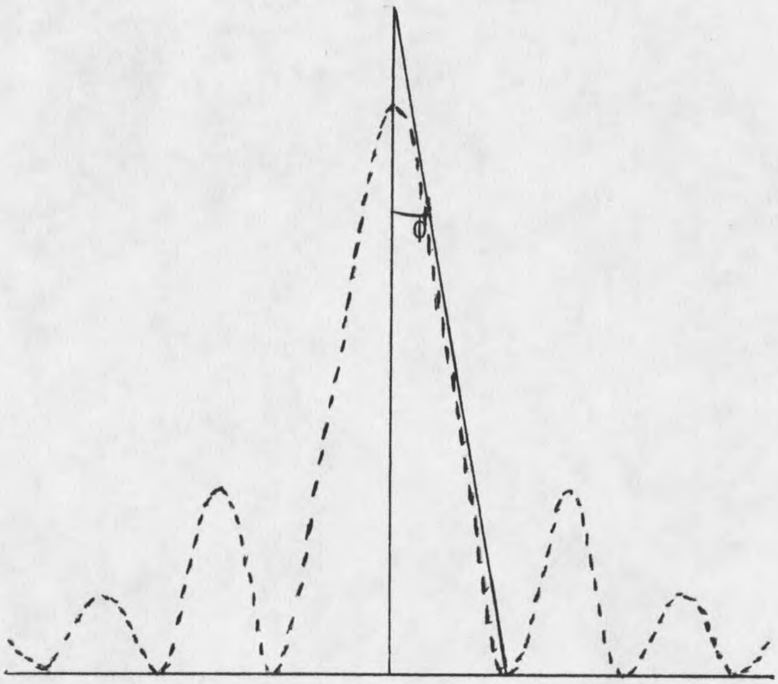


Figure 4. Airy disk.

Resolution

Now consider two point sources, a and b, that are emitting coherent radiation (in phase with each other). What would be displayed on the plate, in this case, is the overlay of one diffraction pattern on top of the other as shown in figure 5. As the points are drawn closer together, more overlap is observed in their diffraction patterns, making it harder to resolve the two points. Thus, two points are more easily resolved as the angular distance between them increases.

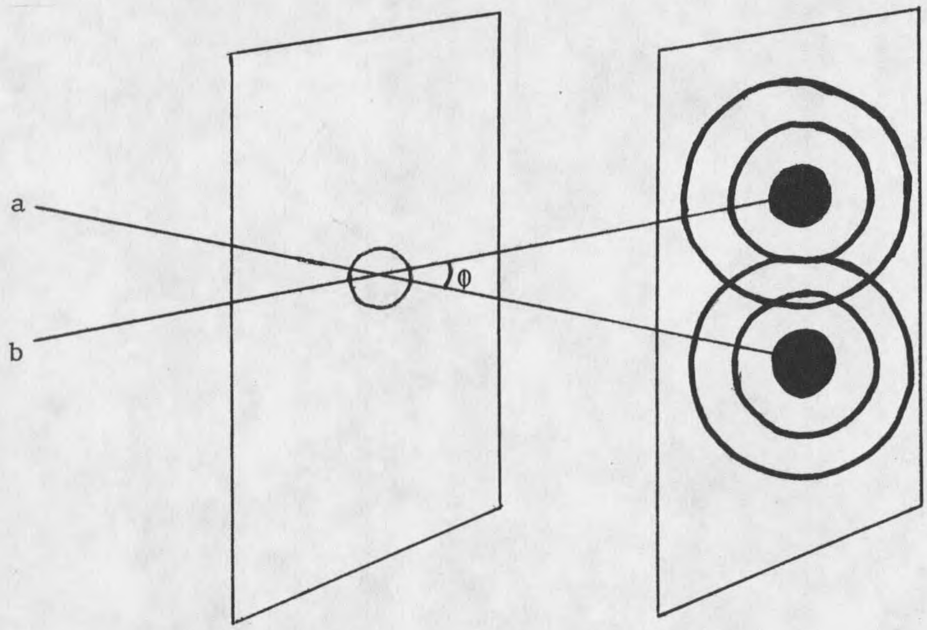


Figure 5. Two-point diffraction pattern.

If the center of one pattern is superimposed on the first dark ring of the other, as shown in figure 6, the angle between the two points, known to be equal to $1.22\lambda/d$, is called the Raleigh two-point resolution limit and is used as one way of measuring the resolution capability of lenses. The Raleigh limit is also important in the intuitive sense, stating that resolution can be increased by either increasing the aperture or by passing radiation of a higher wavelength. This is important in astronomical imaging, because not only does a large aperture gather more light (photons), it also increases the resolution, making it possible to observe finer details.

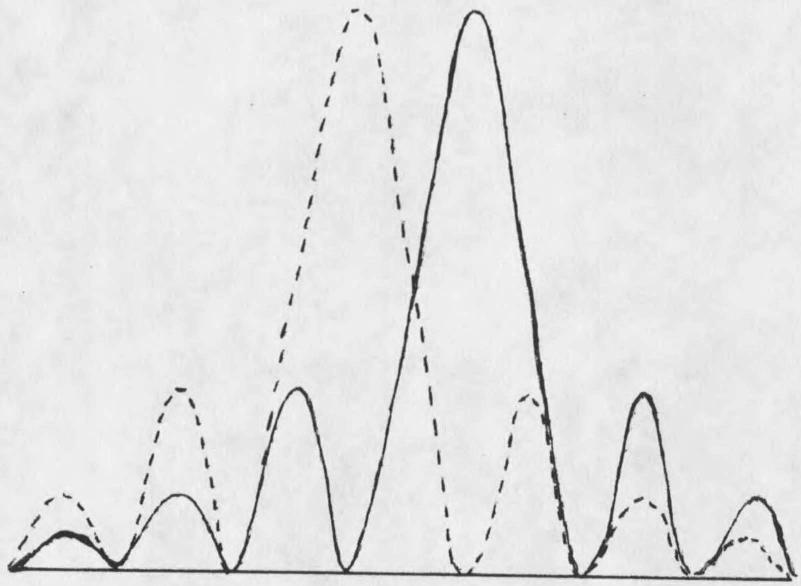


Figure 6. Raleigh two-point resolution limit.

Now consider an extended object of many points. Each point creates its own diffraction pattern yielding a blurred image of the object. The amount of blur is a function of a single point diffraction pattern more commonly referred to as the point spread function. Decreasing the aperture produces a larger Airy disk, thus causing more blur.

Convolution in the Spatial Domain

An image through a single lens is the result of the object intensity convolved with the point spread of the lens. If i(x,y) symbolizes the image, o(x,y) the true object, and h(x,y) the point spread function (psf), the convolution of o(x,y) with h(x,y) is defined by the integral,

$$i(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} o(x,y)h(x-m,y-n)dn\, dm \tag{3}$$

If $h(x,y)$ is known, the true object can be determined by deconvolving the image, $i(x,y)$, with the point spread function, which is not a trivial process in the object or spatial domain. Fortunately, a tool has been discovered that easily performs the operations of convolution and deconvolution , and that tool is the Fourier transform.

Fourier Transform

Fourier analysis shows that any periodic function may be represented by an infinite series of harmonic cosine and sine functions. Given that a function, $f(t)$, has a period of T seconds, the fundamental frequency, after decomposing $f(t)$ into its sinusoidal components, is 1/T hertz, and all other frequencies are integer multiples of 1/T hertz. Cosine and sine functions of the same frequency can be combined to yield a single cosine function of the same frequency but with an added phase angle.

The Fourier transform is an extension of Fourier analysis. Instead of representing the original time function, $f(t)$, as an infinite series of sinusoid functions with respect to time, the Fourier transform is a new function with respect to frequency. The Fourier transform of $f(t)$ is denoted as $F(j\omega)$ and each frequency has a magnitude component and a phase component. The combination of magnitude and phase denotes a cosine function with those attributes. Stripping the phase component from the transform yields only magnitude information which is often referred to as the Fourier modulus.

A significant property of the Fourier transform is its uniqueness. A function, f(t), with its period known, has a unique Fourier transform and vice-versa. Transformations from the time domain to the frequency domain and vice-versa are achieved by the following integrals,

$$f(t) = (1/2\pi) \int_{-\infty}^{\infty} F(j\omega)\exp(j\omega t)d\omega \tag{4}$$

and

$$F(j\omega) = \int_{-\infty}^{\infty} f(t)\exp(-j\omega t)dt \tag{5}$$

The Fourier transform has previously been defined for a function of time. But actually, Fourier analysis applies to any function of any unit. Images are functions of space rather than time. Thus, their Fourier transforms are functions of $j\omega$, where $j\omega$ has the unit cycles/space instead of cycles/time. Fourier transforms can also be applied to functions of multiple dimensions. A Fourier transform always possesses the same dimensionality as the original function.

Fourier Transform and Convolution

Now that the Fourier transform has been introduced, the question arises; how does it lend itself as a useful tool in digital image processing? The convolution of two functions was previously defined in the spatial domain by equation (3). Convolution in the spatial domain can be achieved by Fourier transforming the object and the point spread function, multiplying the two transforms together, and inverse Fourier transforming the result. The opposite is also true, in that convolution in the frequency domain can be achieved through simple multiplication in the spatial domain. Thus, it is advantageous to perform spatial convolution in the Fourier domain. Equations (6) and (7) describe the relationship between the spatial and Fourier domains, where * denotes convolution, small letters denote spatial functions, and capital letters denote their respective Fourier transforms. As was mentioned earlier, an image can be thought of as the object intensity convolved with the point spread function,

$$i(x,y) = o(x,y) * h(x,y) \tag{6}$$

But convolution in the spatial domain is more easily achieved by Fourier transforming the functions and multiplying them in the frequency domain,

$$I(u,v) = O(u,v)H(u,v) \tag{7}$$

For the rest of this paper small letters will denote object domain functions and capital letters will denote their respective Fourier transforms.

Deconvolution

Deconvolution is a simple process in the frequency domain. The object transform is easily computed by dividing the image transform by the transform of the point spread function, commonly referred to as the transfer function,

$$O(u,v) = I(u,v)/H(u,v) \tag{8}$$

The object can then be transformed to the spatial domain by performing the inverse Fourier transform on $O(u,v)$.

Many image processing techniques use the Fourier transform. Fast Fourier transform routines have been developed to speed up the process of transforming from one domain to the other. This is particularly useful for imaging algorithms that require Fourier transformations extensively.

The Turbulent Atmosphere

The total transfer function for an imaging system can be computed by multiplying the transfer functions of all the components, which includes all lenses and recording equipment. If the total transfer function is known, the process of determining the true object is a simple one, involving division in the frequency domain. The atmosphere is not homogeneous, and thus, does not exhibit a single diffraction index at any instant in time. Rather, the atmosphere can be thought of as being an ever changing collage of lenses, each with a different diffraction index [2].

A region in the atmosphere which is characteristic of having a single diffraction index is commonly referred to as a turbule. Turbules randomly change with time, primarily due to temperature changes [2]. Thus, resolution of very small objects (less than an arc second in diameter) is virtually impossible through long exposure photography. However, in 1970, A. Labeyrie [1] published a method of resolving star diameters by recording and averaging a series of short exposures, called speckle images. This marked the beginning of speckle interferometry. Scientists since then extended Labeyrie's ideas to not only resolve angular distances but to obtain reconstructions of objects, exhibiting detail out to the diffraction limit of the telescope.

Speckle Images

"Speckle" refers to the grainy pattern a coherent beam exhibits when it is reflected off a diffusing surface. A speckle pattern is also observed in a stellar photograph if it is taken under high magnification, with a short exposure time, and the received radiation has a narrow bandwidth [1,2]. Even though stars emit incoherent radiation, the resulting speckle pattern is a measure of the amount of coherency of the radiation received.

Due to the turbulent atmosphere, a long exposure photograph produces an image that is heavily blurred, where the blurred region is commonly referred to as the seeing disk. But a photograph taken with a short exposure time (short enough to immobilize the atmosphere) retains detail out to the diffraction limit of the telescope.

A speckle image of a single star has a speckly appearance. The speckles are dispersed in a pattern depending upon the refractive characteristic of the atmosphere at the time of the exposure. Each speckle has a size which is dependent on both the diffraction limit of the telescope and the radiation wavelength. If the bandwidth of the received radiation is narrow enough, the diffraction pattern may be characterized by the mean radiating wavelength and the radiation is considered quasimonochromatic. Quasimonochromatic radiation yields a clearer speckle pattern, which is why most speckle photographs are taken from filtered radiation.

Although a speckle image contains far more detail, the star(s) which produce the image still cannot be resolved from a single image unless the point spread function for that image is known. If the object is known to be a point source, the image simply represents the instantaneous point spread function. But in most cases, neither the object nor the point spread function is known. However, it was discovered that angular distances for simple star patterns can be resolved by averaging a series of independent speckle images in the frequency domain.

Speckles and the Point Spread Function

When diffraction was introduced, the point spread distribution was defined as being dependent on only the size of the telescope aperture. But because a typical telescope contains multiple lenses and recording equipment, the point spread broadens as the received radiation propagates through the imaging system. When such a system records radiation from a single point source, the result is a diffraction limited image of the point source. A diffraction limited image is one which contains detail out to the diffraction limit of the telescope.

Imaging through the atmosphere causes severe problems due to its non-homogeneous nature. Radiation emitting from a single point propagates erratically through the atmosphere resulting in a point spread distribution that looks like many stars, with the smallest speckle being a diffraction limited image of the actual point star itself [1]. So where the point spread function was once described as a single bright central disk surrounded by concentric rings, it is now described as many such structures spread out in a disordered pattern.

CHAPTER 3

SPECKLE INTERFEROMETRY

Much of the blurring observed in stellar photographs is caused by the atmosphere. If the atmosphere were to produce no distortion, increasing the aperture of a telescope would yield higher resolution and a sharper image. But because the atmosphere is a major source of distortion, there is a point where increasing the size of the aperture does not yield higher resolution.

Early stellar interferometry required small apertures to resolve angular distances for simple star patterns. A. Labeyrie discovered a method of resolving the diameters of stars by recording two sets of speckle images through a fully dialated aperture, one set taken of the object and the other of a known point source [1]. The object is then resolved by averaging the two sets of images, using the set of point source images to determine the average point spread.

Averaging Speckle Images

Initially one would think that the images can be averaged with respect to both phase and magnitude in the frequency domain. In this case the average image is described as the object convolved with the average point spread and is defined by the following Fourier transform pair,

$$\langle i(x,y)\rangle = o(x,y) * \langle h(x,y)\rangle \tag{9}$$

in the spatial domain, and

$$\langle I(u,v)\rangle = O(u,v)\langle H(u,v)\rangle \tag{10}$$

in the frequency domain, where $\langle\ \rangle$ denotes the average operator.

However, averaging $I(u,v)$ results in the loss of much spatial frequency information beyond the seeing disk due to the random nature of phases for the higher spatial frequencies [1,2]. Performing the inverse Fourier transform on $O(u,v)$ in this case yields an image that is severely blurred. The detail that was present in the individual speckle images is lost during the averaging process.

Retaining Diffraction Limited Information

However, $|I(u,v)|^2$, is a function, which when averaged, retains some information out to the diffraction limit. All phase information is lost, but the squared modulus is preserved. Thus Labeyrie based his method on the following equation,

$$\langle |I(u,v)|^2\rangle = |O(u,v)|^2 \langle |H(u,v)|^2\rangle \tag{11}$$

$\langle |I(u,v)|^2\rangle$ is computed by averaging the squared Fourier modulus of all the images. $\langle |O(u,v)|^2\rangle$ is the same as $|O(u,v)|^2$ which is the object to be resolved. If the average squared Fourier modulus of the transfer function, $\langle |H(u,v)|^2\rangle$, is known, $|O(u,v)|^2$ can be computed by dividing $\langle |I(u,v)|^2\rangle$ by $\langle |H(u,v)|^2\rangle$, which can be used to resolve angular distances of simple star patterns such as a single or binary star.

$<|H(u,v)|^2>$ is computed by averaging the squared Fourier modulus of recordings of a known point source that is located near the object. If the point source is located near the object and the point source exposures are taken shortly before or after the object exposures, the average transfer modulus for both sets are statistically identical [1].

Resolution from the Autocorrelation

Performing the inverse Fourier transform of $|O(u,v)|^2$ produces the autocorrelation of the object which is defined by the correlation integral,

$$o_{auto}(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} o(x,y)o(x+m,y+n)dn\ dm \qquad (12)$$

Cross-correlation is similar to convolution, except in convolution, one of the functions is folded about the dependent variable axis. If either of the two functions are even $(f(x) = f(-x))$, cross-correlation and convolution are the same.

The autocorrelation of an object can then be used to resolve simple angular distances such as diameters of stars or distances between binary stars. This is because it produces an image that is twice the size of the actual object. For example, suppose figure 7 represents an actual object. The autocorrelation of that object produces an image that is twice the width of the actual object as shown in figure 8. A two dimensional object, such as a star produces an autocorrelation that is twice the diameter of the actual star.

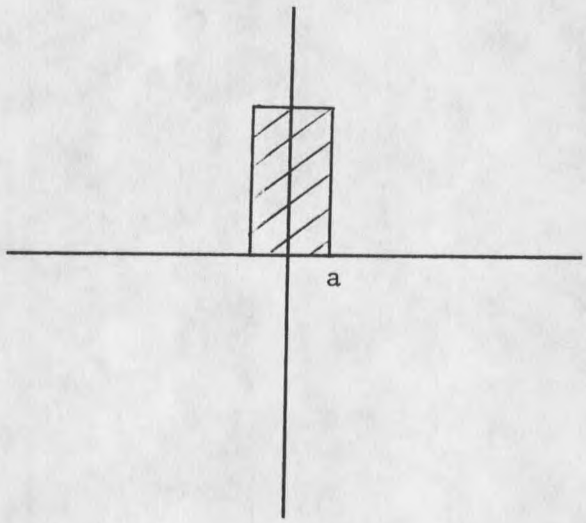


Figure 7. Actual object.

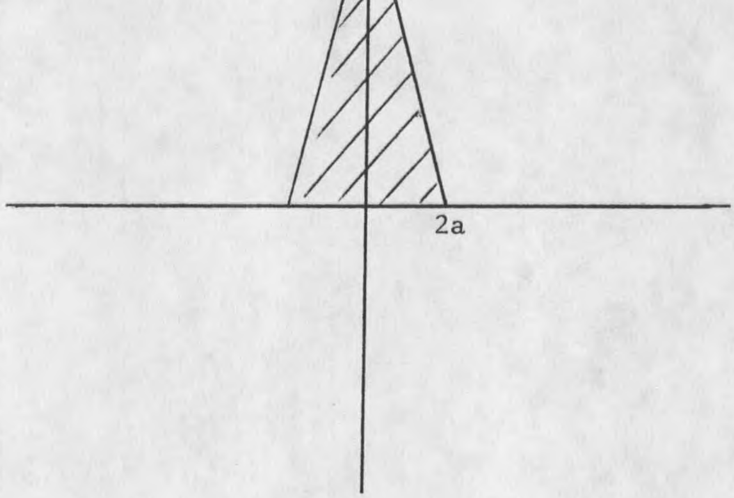


Figure 8. Autocorrelation of the actual object.

Resolving Versus Imaging

Although this method has been used successfully to resolve angular distances, it will not reconstruct the object because all phase information is lost. This brings up the difference between resolving and imaging. Objects whose diameters and distances from other objects can be determined are considered resolvable. If an image of an object is produced to yield an estimate of the true object, the object is considered imaged. Thus, an imaged object is resolvable but the reverse is not true [3]. If the phase for the object Fourier modulus can be determined, the object can be imaged as well as resolved. Since Labeyrie, many scientists have studied the problem of phase retrieval knowing only the Fourier modulus.

Procedure for Speckle Interferometry

1. Take a number of short exposures of the object (the more the better).

2. Take a number of short exposures of a known point source that is near the object, also.

3. Perform the Fourier transform on the object and point source images and record the squared Fourier modulus for all of them.

4. Compute the average squared Fourier modulus for the object, $<|I(u,v)|^2>$, and for the point source, $<|H(u,v)|^2>$.

5. Compute the squared Fourier modulus of the object by dividing

the average squared modulus of the image by the average squared modulus of the transfer function,

$$|O(u,v)|^2 = <|I(u,v)|^2>/<|H(u,v)|^2> \tag{13}$$

6. Finally, compute the autocorrelation of the object by performing the inverse Fourier transform on the squared modulus of the object. Angular distances may then be determined from the autocorrelation.

Quality of Speckle Images

The quality of the speckle images greatly influences the results obtained through speckle interferometry. Exposure time must be short enough to "freeze" the atmosphere and the time between exposures must be long enough to assure independence [2]. A typical exposure time is approximately 5 milliseconds [17].

The signal-to-noise ratio is a quality measurement of an image. A high ratio is desirable and indicates there is much more signal than noise. Noise is generally caused by aberrations in the measuring and recording equipment. Random noise is partially filtered by averaging many speckle images. Additional speckle images contribute more to the signal than to noise as long as the signal-to-noise ratio is not too low. Thus it is desirable to record as many speckle images as possible [1]. Noise that is well behaved can be dealt with by subtracting its effects from each image.

The rms signal-to-noise ratio is directly proportional to the number of detected photons per picture. Increasing the exposure time increases the number of detected photons. Therefore, the exposure time for each speckle image should be as long as possible and still immobilize the atmosphere [2].

Speckle interferometry has been used successfully to resolve the diameter of many stars. Its discovery is considered a major milestone in stellar image processing, and since then, many imaging methods have been developed as an extension of it.

CHAPTER 4

IMAGING

Speckle interferometry yields only an estimate of the Fourier modulus of the object intensity. Squaring the modulus and performing the inverse Fourier transform yields an estimated autocorrelation of the object, which by itself, can be used to resolve angular distances. Unfortunately, acquiring the Fourier modulus through speckle interferometry results in the loss of all phase information which is necessary, in most cases, to produce a true image of the object.

Many methods have been developed to estimate the object given a series of independent speckle images. Some methods are based on studies that show the uniqueness between the phase and modulus. Such methods require Fourier transformations extensively in order to determine the phase. Other methods are more intuitive and do not require Fourier transformations. It is not the intention of this chapter to rank each method with respect to the others, for each one has its merits. Instead, several methods will be discussed with respect to their theory and applications.

Speckle Holography

Recall from speckle interferometry that the autocorrelation produces an image that is twice the size of the original object. If a known point source is located in the same isoplantic region (subjected to the same psf) as the object, the autocorrelation produces two structures that are true reproductions of the object and a third larger structure that is the autocorrelation of the object plus the autocorrelation of the reference point. This phenomenon is commonly referred to as speckle holography. An example of this is shown in figures 9 and 10 for a simple one dimensional case. Figure 9 contains a triangular structure, which in this case is the object, and a dirac function which denotes the reference point. The autocorrelation of the combined structures is shown in figure 10, where the two structures centered at a and -a are true images of the object, with the structure centered at -a being the mirror image of the structure centered at a. The large structure centered at the origin is the composite of the object autocorrelation and the reference point autocorrelation. Analytically, the resulting autocorrelation is defined by the following equation,

$$\mathrm{auto}(x') = \int_{-\infty}^{\infty} \delta(x)\delta(x+x')dx + \int_{-\infty}^{\infty} o(x)o(x+x')dx + o(x-a) + o(-x-a) \tag{14}$$

where $\delta(x)$ denotes the known point source and $o(x)$ the true object.

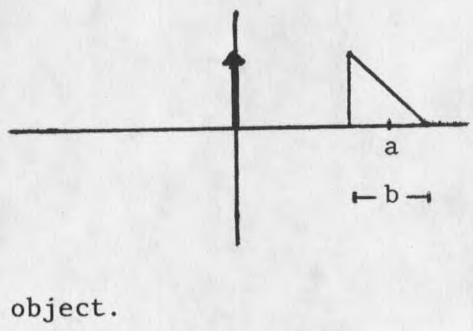


Figure 9. Original object.

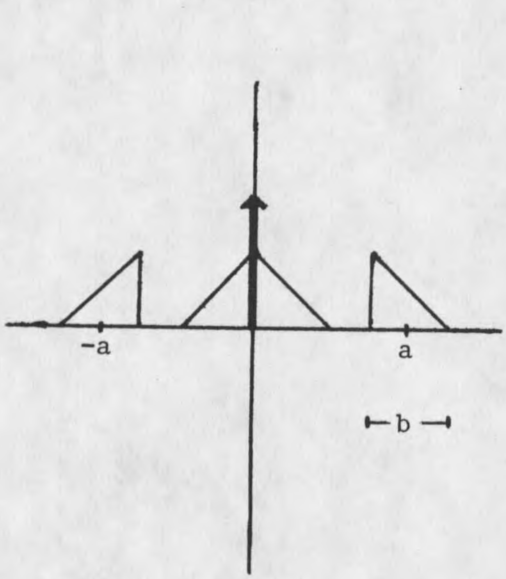


Figure 10. Autocorrelation.

Feasibility

Speckle holography has the advantage of not requiring additional speckle images of a known reference point as is required by other imaging techniques, and in some cases it can produce a true image of the object as part of the total autocorrelation distribution. However, its use is extremely limited for the following two reasons [2,3]:

1. There must exist a known point source in the same isoplantic region as the object.

2. The distance between the object and the point source must be large enough to produce an autocorrelation consisting of three non-intersecting regions. In other words , $a>=3b/2$ must be satisfied to ensure the separability of the object from the reference point.

Lynds-Harvey-Worden Method

Lynds, Harvey, and Worden [4] offered a more intuitive approach to obtain diffraction limited images. Based on the assumption that the brighter speckles, in a speckle image, are produced from the same object point, the image may be cross-correlated with an estimated point spread function to hopefully produce an accurate reconstruction of the true object. The estimated point spread function is determined by extracting a few of the brighter speckles in the original speckle image.

If $\Delta(x,y)$ denotes the centers of the speckles comprising the point spread function and their respective intensities, a typical image of a point source is described by the equation,

$$j(x,y) = \Delta(x,y) * f(x,y) + n(x,y) \tag{15}$$

where $f(x,y)$ is the ideal diffraction distribution of the telescope, and $n(x,y)$ denotes random noise. Similarly, an image of an extended object is described by the equation,

$$i(x,y) = \Delta(x,y) * f(x,y) * o(x,y) + n(x,y) \tag{16}$$

where $o(x,y)$ denotes the true object. In other words, the image is defined as the true object convolved with the combination of the psf due to the turbulent atmosphere and the psf due to the limited aperture of the telescope. An additional noise factor is introduced to account for any spurious images in the background.

Point Spread Extraction

The most difficult part of this method is the extraction of the dirac portion of the point spread function, $\Delta(x,y)$, from the original image. In the reconstruction of Alpha Orionis, Lynds et. al. [4] first computed the Fourier transform of the image and subjected the result to a low pass filter to minimize errors due to noise. After filtering, the inverse Fourier transform was performed to transform the image back to the spatial or object domain, and the centers of the speckles were then located and recorded with their respective intensities. Further processing was done to yield an estimate of the dirac point spread function (that portion of the total psf caused by the turbulent atmosphere), by eliminating those points with relatively small intensities. An estimate of Alpha Orionis was then determined by cross-correlating $\Delta(x,y)$ with the original image. Although, a more obvious approach would be to divide the Fourier transform of the original image by the Fourier transform of $\Delta(x,y)$ and then performing the inverse Fourier transform of the result, it was found that it produced false images in the background [4].

Feasibility

This method has the following two advantages:

1. It does not require a known point source to be included in the photographs with the object.

2. No additional images are needed of a known point source to estimate the point spread function.

However, this method is only good if the point spread function has been accurately estimated by extracting the brighter speckles. Determining the psf from images of extended objects with this method is virtually impossible unless the object is known to contain a single bright point that is much brighter than the others. This method has been successfully used on photographs of Alpha Orionis [4,5].

Shift-And-Add Method

Intuitively, one would think that the brightest point in each image is a distorted version of the object's brightest point [3,6,7], given that the signal-to-noise ratio is not too low. Furthermore, if the whole object lies within the same isoplantic region, each point of the object is subjected to the same point spread and the object points maintain the same position relative to each other [3,6,7]. Figures 11 thru 13 shows a graphic description of this. If figure 11 shows the point spread distribution of a single point (denoted by the asterisk), that same point spread induced on several object points produces a dirty image as shown in figure 12. Another image containing the same object points may display a different point spread distribution, but the object points maintain their relative distances, as shown in figure 13.

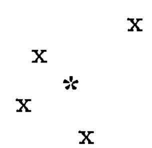


Figure 11. Point spread on a single point.

Figure 12. Same point spread on several points.

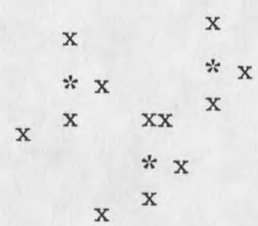


Figure 13. Different point spread on same points. Relative distances between object points are preserved.

Shift-and-add is a simple method. All images are aligned with respect to their brightest points, and added together, thus the name shift-and-add. The average is then computed by dividing the sum by the number of images to yield a diffraction limited image. The averaging process tends to retain object points and eliminate those points caused by noise and the point spread. This method is described by the following equation,

$$o_{sa}(x,y) = \langle s_m(x+x_m, y+y_m)\rangle \tag{17}$$

where s_m is the mth speckle image and (x_m, y_m) is the coordinate of the brightest point in the mth image.

Laboratory simulations [6] have shown that the brightness of the brightest points vary considerably which may produce a "dirty" picture. A "dirty" picture is one which contains an undesirable amount of fog in the background.

Adjusted-Shift-And-Add

The adjusted-shift-and-add method [3], a derivative of shift-and-add, was contrived to eliminate some of the background fog. Under this method, the speckle images are adjusted by multiplying each point by the brightest point in the same image. Bates claims the adjusted-shift-and-add method produces a truer image [3]. The adjusted method is described by the following equation,

$$o_{asa}(x,y) = \langle s_m(x_m, y_m) s_m(x+x_m, y+y_m)\rangle \tag{18}$$

Procedure

The shift-and-add procedure, whether it be the adjusted or the original version, is described as follows:

1. Take a number of short exposures of the object and digitize them. If the adjusted method is used, multiply each pixel in every image by the brightest pixel in the same image.

2. Add all the images together with respect to the brightest pixel in each image. In other words, align all the images so that the brightest pixel in each image is positioned at the same coordinate.

3. Divide each pixel, from step 2, by the number of images to yield an average, and hopefully a reasonable estimate of the true object.

Feasibility

These two methods have their advantages and disadvantages. If each image contains a pixel that is considerably brighter than the others, the shift-and-add or the adjusted-shift-and-add method may be used to produce a reasonably accurate diffraction limited image [3]. Their greatest advantage is their computational simplicity, for neither uses complex Fourier transformations.

Their obvious disadvantage is the constraint that the images must contain a pixel that is considerably brighter than the others, which may limit its use to the imaging of simple star patterns.

"Clean" --- Hogbom

"Clean" [8] is an iterative method to subtractively deconvolve an image knowing the point spread function. Irregularities in the baseline distribution, in the frequency domain, causes undesirable sidelobes in the spatial domain. A "dirty map" (DM) is a measured domain consisting of the convolution of the true signal with a distorting signal called the "dirty beam" (DB),

$$DM = (\text{true signal}) * DB \tag{19}$$

Hogbom first developed "clean" as a method of eliminating sidelobes in the "dirty beam", in radio interferometry, caused by aberrations in measuring equipment. However, "clean" also has a limited use in digital image processing as well. Irregularities in the baseline distribution, in the frequency domain, are caused by the limited aperture of the telescope, which causes the intensity distribution for a single point source, to spread. The degree of spread is determined by the diameter of the aperture, where increasing the aperture decreases the spread.

Given that a measured image is the "dirty map", the object is the true signal, and the point spread distribution is the "dirty beam", equation (19) becomes

$$i(x,y) = o(x,y) * h(x,y) \tag{20}$$

The object may then be determined by deconvolving the image with the point spread function. However, instead of deconvolving the image by division in the frequency domain, "clean" iteratively subtracts the point spread function from the image in the object domain. Cady and Bates [6,7] used this procedure to eliminate fog in the background that resulted from the shift-and-add procedure.

Procedure

"Clean" is an iterative deconvolution method and is described as follows [8]:

1. Record the image and point spread function.

2. Subtract, over the whole image, a normalized point spread function which is centered at the point where $i(x,y)$ has its maximum value, Im. The point spread function is normalized to αIm, where α is called the loop gain.

3. Repeat step 2, replacing the previous image with the newly computed one, until the current value of Im is no longer significant with respect to noise.

4. Return to the new image the dirac component of all the parts that were removed in step 2. The dirac component of a subtracted point spread function consists of a single point located at the center of the function with a value equal to the intensity of the function at its center point.

Phase Estimation -- Knox-Thompson Algorithm

Another imaging approach is to operate in the frequency domain by determining the phase for a measured Fourier modulus. The Knox-Thompson algorithm [9,10] is a non-iterative method to estimate the phase for a measured Fourier modulus. The phase is estimated by averaging phase differences between adjacent points over a series of speckle images. If Labeyrie's method is used to compute the Fourier modulus, the same series of images is used for the Knox-Thompson algorithm.

Recall that Labeyrie's method is based on the equation,

$$\langle|I(u,v)|^2\rangle = |O(u,v)|^2 \langle|H(u,v)|^2\rangle \tag{21}$$

$|O(u,v)|^2$ is computed by dividing $<|I(u,v)|^2>$ by $<|H(u,v)|^2>$. The Fourier modulus of the object is simply computed by taking the square root of $|O(u,v)|^2$ which can then be used as the measured modulus for the Knox-Thompson algorithm.

The algorithm is based on the following equation (in one dimension) [10],

$$<I(u)I^*(u+\Delta u)> = O(u)O^*(u+\Delta u)<H(u)H^*(u+\Delta u)> \tag{22}$$

where * denotes the complex conjugate, and Δu denotes a small change in frequency. If $\Delta u = 0$, equation (22) becomes

$$<I(u)I^*(u)> = O(u)O^*(u)<H(u)H^*(u)> \tag{23}$$

which is the same as

$$<|I(u)|^2> = |O(u)|^2 <|H(u)|^2> \tag{24}$$

which is the basis for Labeyrie's method.

Correctness of the Knox-Thompson Formula

To prove correctness of equation (22), it is true that

$$I(u) = O(u)H(u) \tag{25}$$

and similarly,

$$I(u+\Delta u) = O(u+\Delta u)H(u+\Delta u) \tag{26}$$

Taking the complex conjugate of both sides of equation (26) yields

$$I^*(u+\Delta u) = O^*(u+\Delta u)H^*(u+\Delta u) \tag{27}$$

which when multiplied by equation (26) yields

$$I(u)I^*(u+\Delta u) = O(u)O^*(u+\Delta u)H(u)H^*(u+\Delta u) \tag{28}$$

Now, applying the above equation to a series of images yields equation (22).

Computation of Phase Differences

Equation (22) provides a relationship between adjacent points in the frequency domain. From it, phase differences between adjacent points, may be computed by the following formula,

$$\begin{aligned} \text{phase}[\langle I(u)I^*(u+\Delta u)\rangle] &= [\text{phase}[O(u)] - \text{phase}[O^*(u+\Delta u)]] \\ &\quad + \text{phase}[\langle H(u)H^*(u+\Delta u)\rangle] \end{aligned} \tag{29}$$

Recall that when two complex numbers are multiplied, the phase of the result is equal to the sum of the component phases. Also recall that the phase of the conjugate is equal to the negative of the original.

To simplify equation (29) let

$$T(u) = \langle I(u)I^*(u+\Delta u)\rangle / \langle H(u)H^*(u+\Delta u)\rangle \tag{30}$$

which yields

$$\text{phase}[O(u+\Delta u)] = \text{phase}[O(u)] - \text{phase}[T(u)] \tag{31}$$

The phase of $T(u)$ may be computed from a series of short exposures of the object, for the numerator, and of a known point source, for the denominator. However, Knox [10] claims that $\langle H(u)H^*(u+\Delta u)\rangle$ has little effect on the estimation of phase differences. Therefore a good estimate of $T(u)$ is

$$T'(u) = \langle I(u)I^{*}(u+\Delta u)\rangle \tag{32}$$

and the phase differences may be computed, with little error, by the formula,

$$\text{phase}[O(u+\Delta u)] = \text{phase}[O(u)] - \text{phase}[T'(u)] \tag{33}$$

After the phase differences are computed for each image, they are averaged to yield an array of average phase differences which is then used to estimate the actual phase for the measured Fourier modulus. The actual phases for each point, in the frequency domain, are easily computed by summing the phase differences away from the origin. For example, the phase for $O(0,2)$ is computed by adding the phase difference between $O(0,0)$ and $O(0,1)$ with the phase difference between $O(0,1)$ and $O(0,2)$. If the phase of $O(0,0)$ is known, computing the phases becomes a trivial process. Fortunately, the phase of $O(0,0)$ is known to be zero because the function $o(x,y)$ is known to be real and non-negative.

After the phases are computed, they are combined with the measured Fourier modulus to yield an estimated Fourier transform of the true object. An image of the object is then obtained by performing the inverse Fourier transform.

Procedure Summary

The Knox-Thompson algorithm is easily applied to two dimensional objects and a proper procedure is summarized as follows;

1. Take a number of short exposures of the object and digitize them. If the Fourier modulus was computed using Labeyrie's method, the same images can be used for both the modulus estimation and phase estimation.

2. Perform the Fourier transform on all the images to produce representations of the object images in the frequency domain.

3. Compute and save the phase differences between adjacent points for every image.

4. Average the phase differences from step 3.

5. Compute the phase for each point by summing up the phase differences, from step 4, away from the origin, and knowing that O(0,0) = 0.

6. Combine the phase estimate, from step 5, with a measured modulus.

7. Perform the inverse Fourier transform, on the result from step 6, to yield an estimate of the true object.

Feasibility

The Knox-Thompson algorithm has the disadvantage of being susceptible to noise. An erroneous phase difference has the ill effect of compounding the error for phase computations beyond the errored phase difference. Errors may be reduced by averaging phases from several paths [10]. For example, the phase of O(2,4), shown in figure 14, may

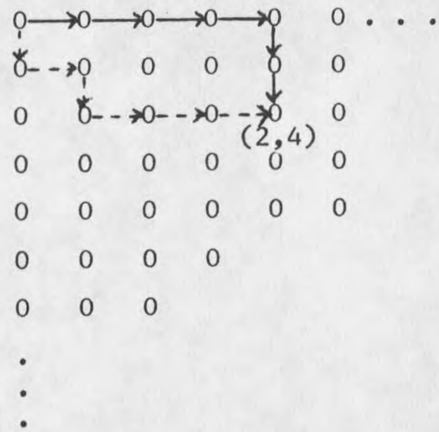


Figure 14. Phase difference matrix.

be computed by averaging the sum of phase differences over multiple paths. Figure 14 shows two such paths. Also, the telescope must have guided accurately on the object, in order to retrieve accurate phase information [2]. However, it does have the advantage in that it is well suited to run concurrently with Labeyrie's method.

Phase Estimation --- Error-Reduction Algorithm

Much has been written on phase retrieval given only the Fourier modulus. Given the phase and modulus, the object can be reconstructed by performing the inverse Fourier transform. The probability that a modulus has a unique phase, in the two dimensional case, is very high [11]. This uniqueness is the basis for the error-reduction algorithm.

The error-reduction algorithm [12,13] estimates the phase through an iterative process. Knowing the Fourier modulus and certain characteristics in the spatial domain, the phase is estimated by transforming between the spatial and frequency domains, and imposing constraints in each domain. The algorithm is monitored by computing the difference between the computed estimate and one which satisfies the given constraints. The difference, or error, can be computed in either the spatial or frequency domain, and is reduced after every iteration. The algorithm continues until the error is reduced to within a specified tolerance. The estimated phase from the final iteration is then combined with the given Fourier modulus to yield the Fourier transform of the object.

The original error-reduction algorithm was presented by Gerchberg and Saxton and was intended for phase retrieval given intensity measurements in both the spatial and frequency domains [13]. However, the original algorithm was modified to determine the phase knowing only the intensity in the frequency domain. A block diagram of this algorithm is shown in figure 15.

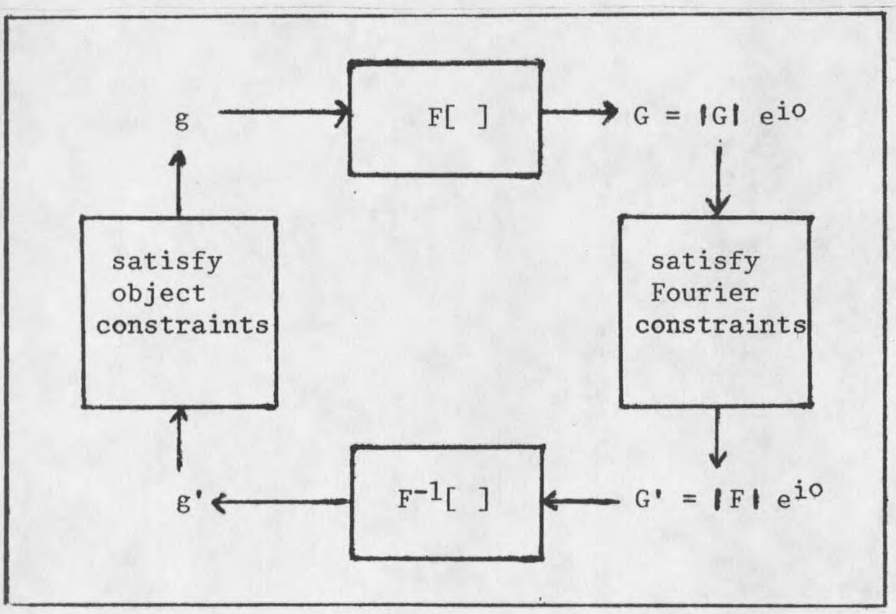


Figure 15. Block diagram of the error-reduction algorithm.

In figure 15, g and g' are spatial domain estimates and G and G' are their respective Fourier transforms. $|F|$ is the measured Fourier modulus, which is used as the frequency domain constraint. F[ ] is the Fourier transform operator and F^{-1}[ ] is the inverse Fourier transform operator. A single iteration of the algorithm involves the transformation from spatial domain to frequency domain and back to spatial domain. Constraints are imposed after every transformation to reduce the error and converge to a solution.

Although this algorithm is not guaranteed to converge, the worst it will do is maintain the same error that was computed in the initial object estimate. More a priori knowledge results in faster convergence, if the algorithm is to converge. Proof of convergence will be given later. A typical spatial domain constraint is that all intensities are positive. If the size of the object is known, it too can be used as a constraint, resulting in faster convergence.

Procedure Summary

Starting from the upper left of figure 15, the algorithm proceeds as follows:

1. Pick an initial image, g, and specify an error tolerance for the frequency domain and or spatial domain. The initial estimate may be attained randomly or intelligently. An intelligent estimate will result in faster convergence.

2. Perform the Fourier transform on g to yield a Fourier modulus, $|G|$, and phase estimate.

3. Compute the error between the measured modulus, $|F|$, and the computed modulus, $|G|$. If it is within the specified tolerance a phase solution has been found. The inverse Fourier transform is then performed to yield an estimate of the true object.

4. If the error is not within limits, impose frequency domain constraints by replacing the estimated modulus with the measured modulus, $|F|$.

5. Perform the inverse Fourier transform to yield an estimate of the object in the spatial domain.

6. Compute the error again, only this time in the spatial domain. If it is within the specified tolerance, the algorithm is terminated leaving g' as the final estimate of the true object.

7. If the error tolerance is not met, make minimal changes to g' and still satisfy spatial constraints. Usually, non-negativity is imposed as a constraint. If the size of the object is known, it can also be used by zeroing all values that fall beyond the size boundary. Repeat steps 2 thru 7.

Error Computation

Convergence of the algorithm is monitored by computing the squared error in the frequency domain and or spatial domain, where it is defined as [12]

$$E_F^2 = \frac{\sum_u \sum_v [|G(u,v)| - |F(u,v)|]^2}{\sum_u \sum_v |F(u,v)|^2} \tag{34}$$

in the frequency domain, and

$$E_0^2 = \frac{\sum_{x \epsilon \gamma} \sum_{y \epsilon \gamma} |g'(x,y)|^2}{\sum_x \sum_y |g'(x,y)|^2} \tag{35}$$

in the spatial domain, where γ includes all points which violate the spatial domain constraints. When the squared error, in either domain, is within its tolerance, a solution is found and the algorithm is terminated. Because of the uniqueness of Fourier transform pairs, when constraints are satisfied in one domain (before being imposed), they are satisfied in the other domain as well.

Proof of Convergence [13]

If the squared error, in the frequency domain, were defined as

$$E_{Fk}^2 = N^{-2} \sum_u |G_k(u) - G'_k(u)|^2 \tag{36}$$

by Parseval's theorem, it is true that

$$\begin{aligned} E_{Fk}^2 &= N^{-2} \sum_u |G_k(u) - G'_k(u)|^2 \\ &= \sum_x |g_k(x) - g'_k(x)|^2 \end{aligned} \tag{37}$$

at the kth iteration. Because minimal changes are made, $g_{k+1}(x)$ is closer to $g'_k(x)$ than $g_k(x)$. Thus it is true that

$$|g_{k+1}(x) - g'_k(x)| \leq |g_k(x) - g'_k(x)| \tag{38}$$

where $E_{0k}^2 = \sum_x |g_{k+1}(x) - g'_k(x)|$. Therefore, it is true that

$$E_{0k}^2 \leq E_{Fk}^2 \tag{39}$$

Similarly, by Parseval's theorem

$$E^2_{Ok} = \sum_x |g_{k+1}(x) - g'_k(x)|^2 \qquad (40)$$
$$= N^{-2} \sum_u |G_{k+1}(u) - G'_k(u)|^2$$

Because minimal changes are made in the frequency domain as well, it is true that $G'_{k+1}(u)$ is closer to $G_{k+1}(u)$ than $G'_k(u)$, and

$$|G_{k+1}(u) - G'_{k+1}(u)| \leq |G_{k+1}(u) - G'_k(u)| \qquad (41)$$

Therefore, it is true that

$$E^2_{Fk+1} \leq E^2_{Ok} \qquad (42)$$

and finally

$$E^2_{Fk+1} \leq E^2_{Ok} \leq E^2_{Fk} \qquad (43)$$

Thus the squared error is maintained or reduced after each iteration.

Feasibility

Fienup [13] concluded that the error-reduction algorithm usually converges rapidly for the first few iterations but very slowly for later iterations. It has been shown that convergence is sped up with a good initial estimate or more a priori knowledge [14]. Good initial estimates may be obtained through such methods as Lynds-Harvey-Worden or one of the shift-and-add methods.

Besides converging slowly, much CPU time is spent transforming between the spatial and frequency domains. However, the development of fast Fourier transform routines has made this algorithm more appealing.

Phase Estimation --- Input-Output Algorithm

This algorithm, a derivative of the error-reduction algorithm, was presented by Fienup as a solution to the convergence problem [12,13]. It is identical to the error-reduction algorithm in the frequency domain, differing only in the spatial domain. Instead of changing the object estimate so that it just satisfies the constraints, it is changed so as to force a solution. A block diagram of Fienup's Input-Output algorithm is shown in figure 16.

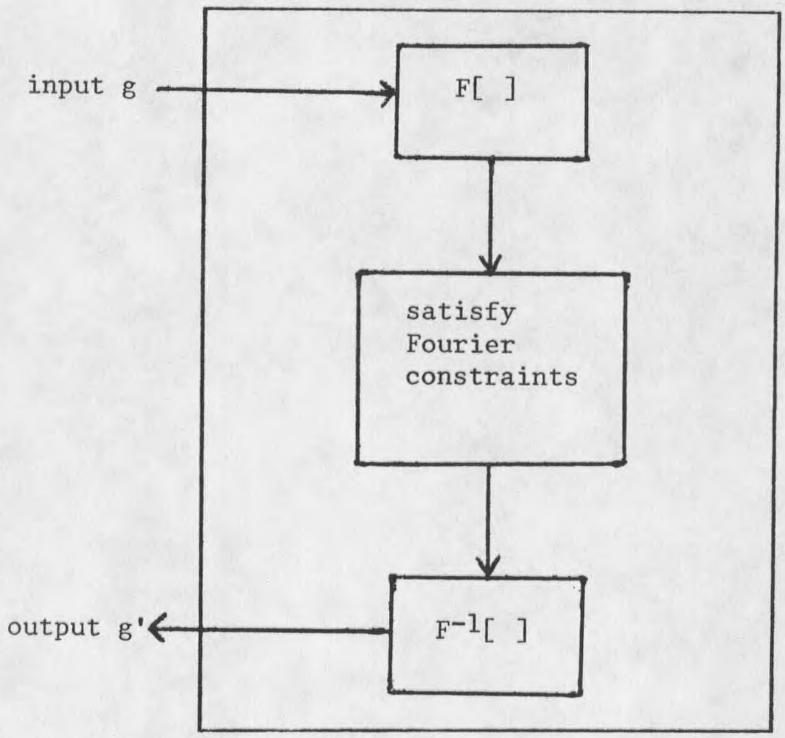


Figure 16. Block diagram of the Input-Output algorithm.

Note that the block diagram, shown in figure 16, omits spatial domain constraints, which does not mean that they are not required. It is omitted to stress that g is computed to force g' to satisfy the constraints even though g may not.

Even though a change in input does not necessarily lead to a proportional change in output, Fienup claims that the change in output will at least move in the same direction as the change in input. Ideally, a change in input, $\Delta g(x,y)$, will produce a change in output, $\alpha\Delta g(x,y)$, where α is a constant. Therefore, if a change in output, $g(x,y)$, is desired, the appropriate change in input is $\beta g(x,y)$, where $\beta=1/\alpha$. Even though β cannot actually be computed, a good choice of will produce an output that is closer to a solution.

Fienup [13] suggests three ways to compute g_{k+1}, the first being

$$g_{k+1}(x,y) = \begin{cases} g_k(x,y) & (x,y)\varepsilon\gamma \\ g_k(x,y) - \beta g'_k(x,y) & (x,y)\varepsilon\gamma \end{cases} \tag{44}$$

where γ includes all points that violate the spatial domain constraints. If a change in input produces a linear change in output, this is a logical choice. A perfectly chosen β would then force the function at that point to zero, during that iteration. Those points that satisfy the constraints retain the respective values from the input because there is no need to change a point that already satisfies the constraints.

The second and third suggested strategies are,

$$g_{k+1}(x,y) = \begin{cases} g'_k(x,y) & (x,y)\varepsilon\gamma \\ g'_k(x,y) - \beta g'_k(x,y) & (x,y)\varepsilon\gamma \end{cases} \tag{45}$$

and

$$g_{k+1}(x,y) = \begin{cases} g'_k(x,y) & (x,y)\varepsilon\gamma \\ g_k(x,y) - \beta g'_k(x,y) & (x,y)\varepsilon\gamma \end{cases} \tag{46}$$

which Fienup uses for what he calls the hybrid-input-output algorithm.

Because a resulting output may be driving to a solution, Fienup suggests strategy 2 as another choice in choosing the next input. If $\beta=1$ in this case, the algorithm simply becomes the error-reduction algorithm.

Finally, strategy 3 was presented as a combination of the previous two. Fienup [13] refers to this strategy as a hybrid strategy and claims it is the best strategy of the three.

Feasibility

The Input-Output algorithm converges faster than its ancestor, the error-reduction algorithm, which makes it more appealing as an imaging method. Areas which need further research are new strategies in selecting g_{k+1} and the combination of this algorithm with other methods. Bates combined a phase estimation method with the error-reduction algorithm and got favorable results [14]. Perhaps his phase estimation and Fienup's Input-Output algorithm would have produced even better results.

CHAPTER 5

A NEW IMAGING METHOD

Diffraction limited images have been attained by shifting and adding speckle images with respect to the brightest point in each image. However, using this method on objects that do not contain a single pixel that is brighter than the others, has produced spurious results. If the speckle images are optimally aligned, spurious results such as "ghost" stars are reduced. A new method, called correlate-shift-and-add, is simply a modified version of shift-and-add. But where shift-and-add correlates with respect to a single bright pixel, correlate-shift-and-add correlates with respect to the entire image.

Theory Behind Correlate-Shift-And-Add

Like the shift-and-add methods, this method is based on the principle that the brighter pixels in every speckle image are distorted versions of the same object area. If the object contains a point that is much brighter than the other points, the shift-and-add principle works well to align and average the images. But when the object does not contain a single bright point, there is a decrease in the probability that speckle images all contain a pixel that is brighter than the other pixels and represents the same object point. Shift-and-add performed on such images sometimes produces "ghost" stars due to misalignment of the images.

However, correlate-shift-and-add correlates the images with respect to the entire image and is less likely to produce spurious results such as "ghost" stars. For example, suppose that images are taken of two stars of equal intensity. Due to random atmospheric turbulence, the dimmer star may appear brighter in some images. Correlate-shift-and-add shows maximum correlation between two images where both stars in both images are lined up regardless of the intensity difference between the stars.

Cross-correlation is similar to convolution and is described by the following integral in the spatial domain,

$$\mathrm{cross}(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} i1(x,y)\,i2(x+m,y+n)\,dn\,dm \qquad (47)$$

The only difference between the cross-correlation integral and the convolution integral is that neither function is folded about one of the axes in the cross-correlation integral. Cross-correlation in the spatial domain is an easy process in the frequency domain and is described by the equation,

$$\mathrm{CROSS}(u,v) = I1(u,v)\,I2^{*}(u,v) \qquad (48)$$

where * denotes conjugation.

Cross-correlating two images produces a third image with its brightest pixel positioned at the point where the two images have the highest correlation. The position and value of the brightest pixel indicates two properties: 1) the amount of shift required to align the second image with respect to the first image for maximum correlation, and 2) the degree to which they correlate or match. The first property is used to align the images and the second property is ignored to give equal weight to all the images.

Procedure

The correlation process can be achieved in either the spatial or frequency domain. Two single dimension arrays of N elements require N multiplications and N(N-1) additions in the spatial domain. Using the Cooley-Tukey formulation of the fast Fourier transform a transformation requires $(N/2)\log_2 N$ complex multiplications, $(N/2)\log_2 N$ subtractions, and $(N/2)\log_2 N$ additions [19]. Consider that one complex multiplication requires 4 real multiplications and 2 real additions, and a complex addition requires 2 real additions. The three transformations necessary to perform the cross-correlation operation executes a total of $12(N/2)\log_2 N+4N$ multiplications and $18(N/2)\log_2 N+2N$ additions. A spatial correlation between two 128 element arrays requires 16384 multiplications and 16256 additions. The same operation achieved the frequency domain requires 5632 multiplications and 8192 additions. Therefore, it is more efficient to perform spatial correlation in the frequency domain.

Each successive image is correlated with the running composite image, and the procedure is described as follows:

1. Read the first image to start the process.

2. Read the other images (one at a time) and correlate each image with the composite image as follows:

(a) Fourier transform the composite image.

(b) Fourier transform the new image and compute its conjugate. The conjugate of (a+jb) equals (a-jb).

(c) Multiply the two transforms together and perform the inverse Fourier transform to yield the cross-correlation of the two images.

(d) Locate the brightest pixel. Its position is used as a shift indicator.

(e) Shift and add the new image to the composite image, using the shift indicator.

Simulations

Binary Star Simulations

This method was tested with simulated data obtained by convolving a simulated binary star with 50 images of Alpha Corona Borealis (a known reference star). The two stars were separated by 30 pixels to simulate a separation of .2 arc seconds. A speckle image of the binary star is shown in figure 17.

In the first case, the two stars were made equal in intensity. Shift-and-add reproduced the binary star with the correct separation plus a third ("ghost") star. The center star was the brightest of the three, and the other star (in the binary system) was slightly more than half as intense. The "ghost" star was slightly less than half as intense as the center star. A photograph of this case is shown in figure 18.

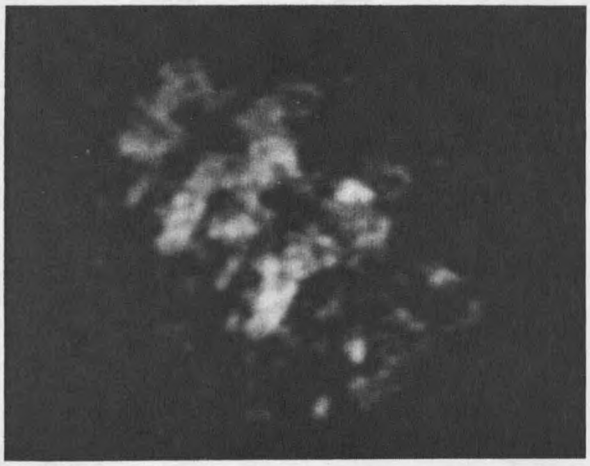


Figure 17. Simulated speckle image of a binary star.

Figure 18. Shift-and-add performed on simulated star images.

Correlate-shift-and-add, on the other hand, reproduced only the two original stars and displayed them with equal intensity. However, correlate-shift-and-add showed problems in locating the centers of the stars. Figure 19 shows that the stars are distinct but larger than the stars as a result of shift-and-add.

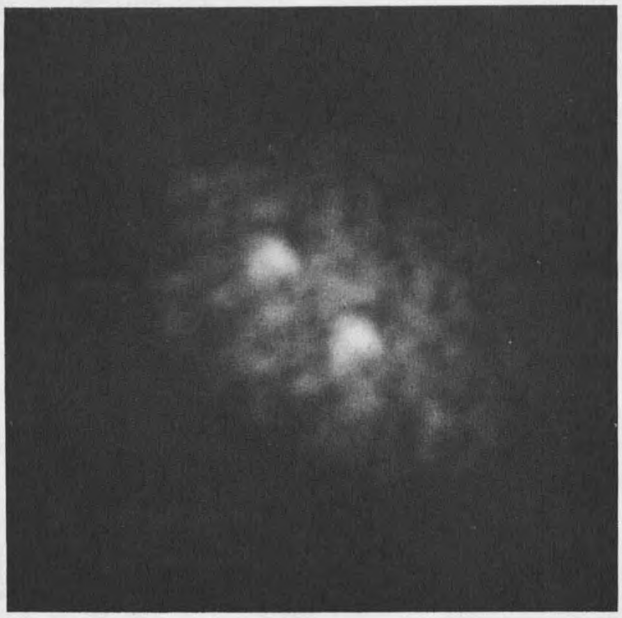


Figure 19. Correlate-shift-and-add performed on star images.

In the second case, one star was made only .7 as intense as the other. Again shift-and-add produced a "ghost" star and the correlate-shift-and-add method reproduced the binary star with a more accurate intensity ratio.

In the third case, the intensity ratio was 2 to 1 and produced better results with shift-and-add. The "ghost" star practically disappeared and the intensity ratio was more accurate than for correlate-shift-and-add (refer to figures 20 and 21).

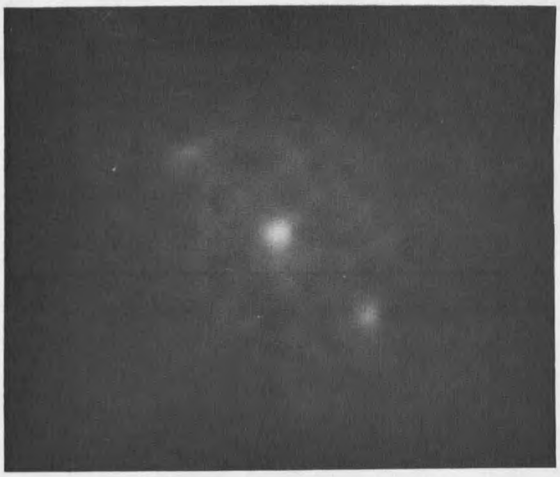


Figure 20. Shift-and-add performed with an intensity ratio of 2 to 1.

Figure 21. Correlate-shift-and-add performed with a 2 to 1 intensity ratio.

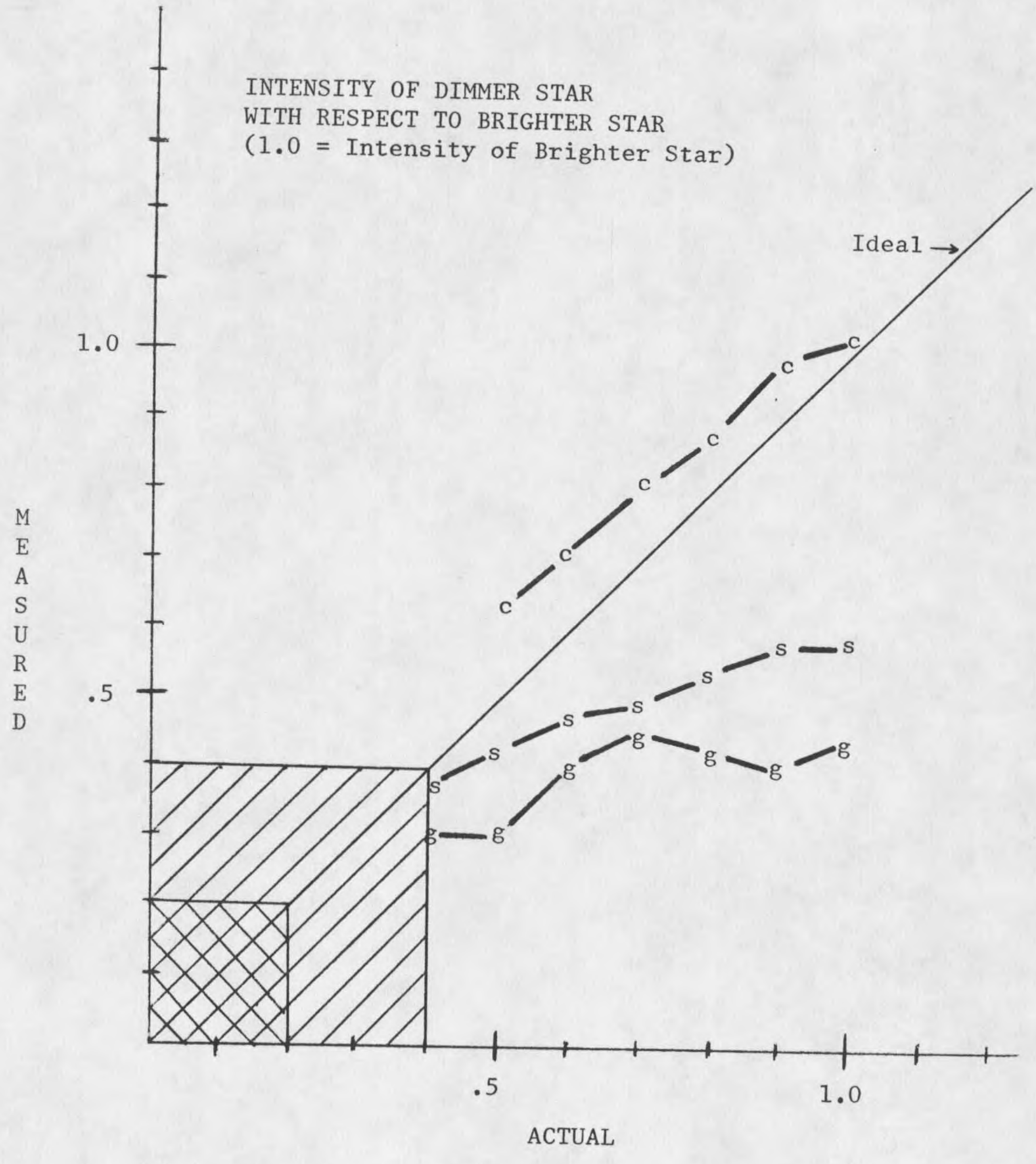
For the final binary star case, the intensity ratio was 5 to 1. In this case, neither method produced good results. The dimmer star disappeared using either method.

Table 1 shows specific results of the binary star simulations for both shift-and-add and correlate-shift-and-add. Figure 22 shows the measured intensity of the dimmer star with respect to the brighter star and figure 23 graphically shows the intensity error of the dimmer star for both methods. The error is computed as follows,

$$\text{Error} = \frac{\text{actual intensity} - \text{measured intensity} \times 100}{\text{actual intensity}} \tag{49}$$

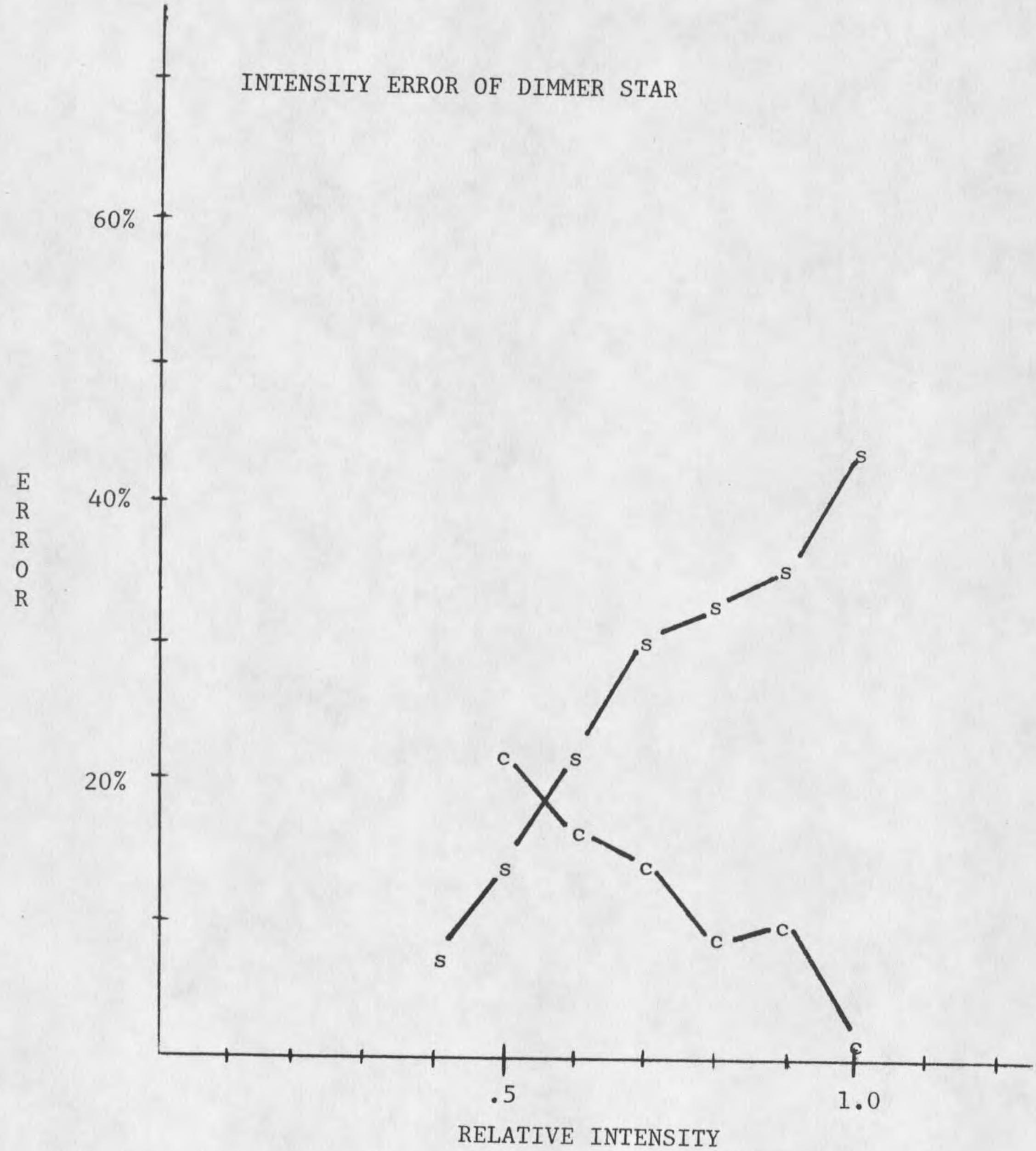
METHOD	ACTUAL INTENSITY RATIO	SIMULATED INTENSITY RATIO	"GHOST" STAR AND RATIO
SAA	1:1.0	1:0.57	Y,0.44
CSAA	1:1.0	1:1.02	N
SAA	1:0.9	1:0.58	Y,0.41
CSAA	1:0.9	1:0.99	N
SAA	1:0.8	1:0.54	Y,0.43
CSAA	1:0.8	1:0.87	N
SAA	1:0.7	1:0.49	Y,0.45
CSAA	1:0.7	1:0.80	N
SAA	1:0.6	1:0.47	Y,0.40
CSAA	1:0.6	1:0.70	N
SAA	1:0.5	1:0.43	Y,0.31
CSAA	1:0.5	1:0.61	N
SAA	1:0.4	1:0.37	Y,0.31
CSAA	1:0.4	DIMMER STAR DISAPPEARED	
SAA	1:0.2	DIMMER STAR DISAPPEARED	
CSAA	1:0.2	DIMMER STAR DISAPPEARED	

Table 1. CSAA vs SAA.



c——c——c = correlate-shift-and-add
s——s——s = shift-and-add (dimmer star)
g——g——g = shift-and-add ("ghost" star)
//// = dimmer star disappeared using CSAA
XXXX = dimmer star disappeared using SAA

Figure 22. Measured intensity of dimmer star.



c——c——c = correlate-shift-and-add
s——s——s = shift-and-add

Figure 23. Intensity error for CSAA and SAA.

Binary Star Simulation Conclusions

Correlate-shift-and-add produced promising results in the binary star simulations. Its strengths over shift-and-add are summarized as follows:

1. No "ghost" star is produced.

2. If the dimmer star is not less than 60% as intense as the other, correlate-shift-and-add yields a more accurate intensity ratio.

Its weaknesses are two fold:

1. When the dimmer star is less than half as intense as the other, correlate-shift-and-add causes the dimmer star to fade into the "foggy" background.

2. The stars are distinct but larger and more irregular than those produced by shift-and-add.

Extended Object Simulations

Correlate-shift-and-add was tested using simulated speckle images of an extended object. An object (shown in figure 24) was convolved with the same 50 Alpha Corona Borealis images used for the binary star simulations. This was done to simulate an extended object subjected to the same point spread functions. Figure 25 shows one of the speckle images of the object. Both shift-and-add and correlate-shift-and-add were performed and neither produced discernable results (refer to figures 26 and 27). At this point the value of correlate-shift-and-add on extended objects was questionable.

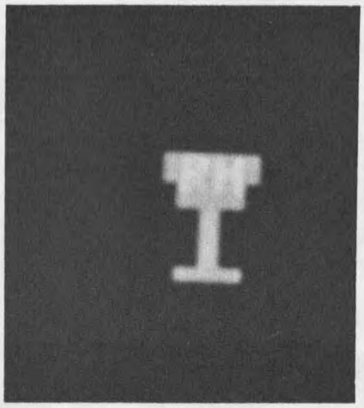


Figure 24. Actual Object.

Figure 25. Simulated speckle image of object.

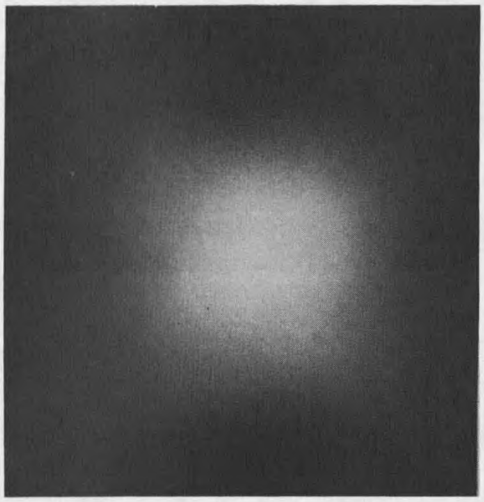


Figure 26. Shift-and-add on unrestricted extended object images.

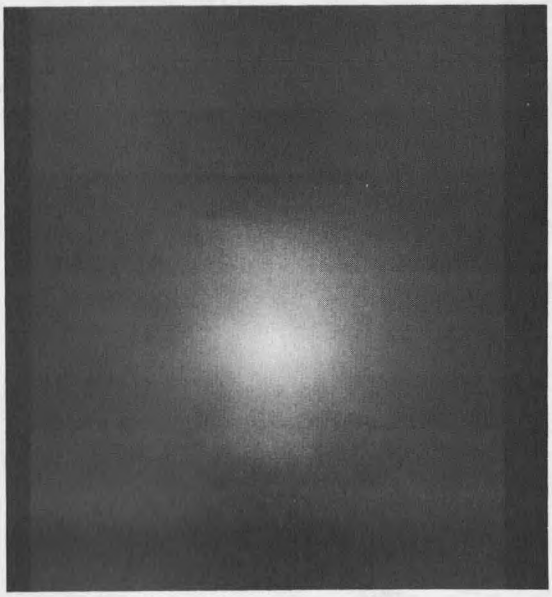


Figure 27. Correlate-shift-and-add on unrestricted images.

The Alpha Corona Borealis images produced point spread functions that were too complex for either of the methods to decipher the true object. Thus, the Alpha Corona Borealis images were modified to represent simpler point spread functions. Fifty percent of the brightest pixels were extracted from each image to produce delta functions. The extracted delta functions were then convolved with a point spread function (5 pixels in diameter) to simulate the effects of an aperture. The estimated psfs were then convolved with the extended object. One such image is shown in figure 28.

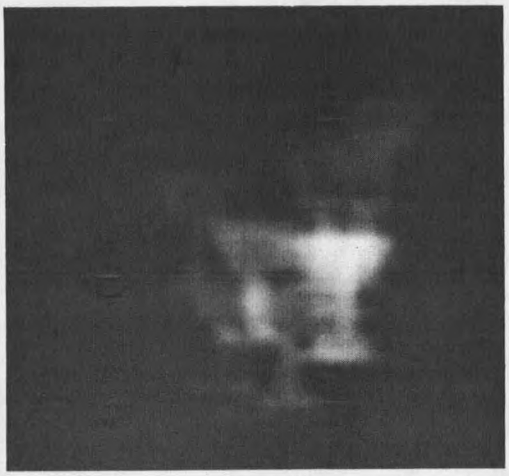


Figure 28. Object convolved with restricted point source images.

The images were then used to test shift-and-add, correlate-shift-and-add, and the Knox-Thompson algorithm. Shift-and-add (shown in figure 29) produced an image that contained a bright area where the object displayed brighter intensities but the outline of the object was totally lost.

Figure 29. Shift-and-add performed on restricted images.

The Knox-Thompson algorithm was applied using Labeyrie's method to estimate the average Fourier modulus. Its results (shown in figure 30) were more obscure than those obtained using shift-and-add.

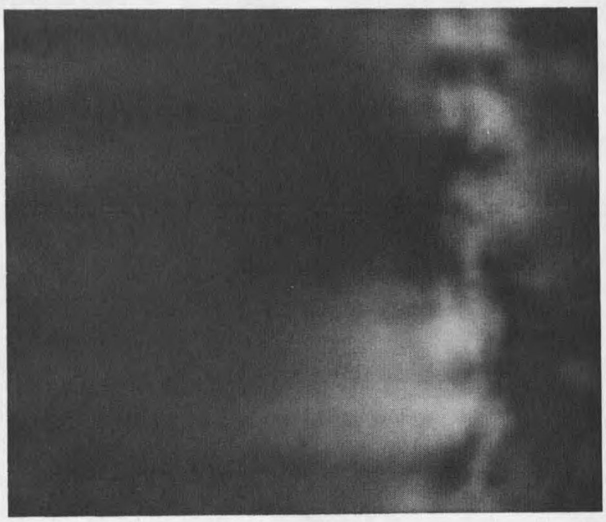


Figure 30. Knox-Thompson performed on restricted images.

Correlate-shift-and-add method, on the other hand, produced much better results in that the outline of the object was more recognizable (refer to figure 31).

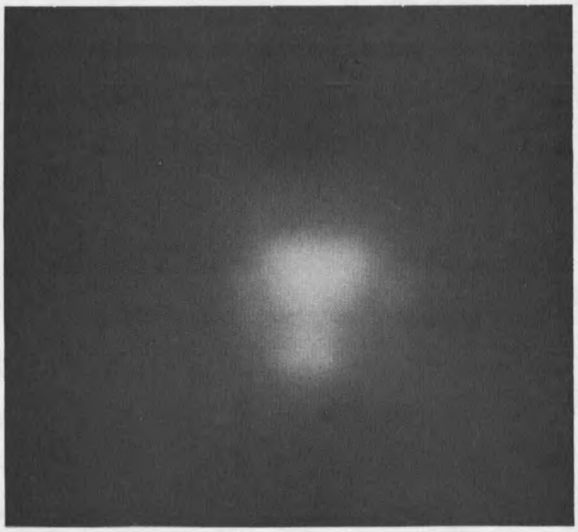


Figure 31. Correlate-shift-and-add performed on restricted images.

Extended Object Simulation Conclusions

Applying Knox-Thompson, shift-and-add, or correlate-shift-and-add to high resolution speckle images of an extended object where the psfs are complex produced unrecognizable results from 50 images. But when the psfs were "cleaned up" correlate-shift-and-add outperformed the other methods and produced an image that preserved the general shape of the object. It appears that correlate-shift-and-add may be useful on speckle images of extended objects where the psfs are not too complex.

CHAPTER 6

SUMMARY

Labeyrie introduced a way of resolving high resolution speckle images by averaging their Fourier moduli. The Fourier modulus turns out to be a function well suited for averaging since it is unaffected by shifts in the spatial domain. Information out to the diffraction limit of the measuring device can be obtained. Labeyrie discovered that star diameters can be resolved by averaging the squared Fourier moduli of many images and computing the average autocorrelation. The diameter of an object is easily obtained by dividing the diameter of the autocorrelation image by two. Simple angular distances can be resolved, but the object cannot be actually imaged because all phase information is lost.

Fienup proposed a method of obtaining the phase for a given Fourier modulus by iteratively transforming an image between the spatial and frequency domains. The algorithm is terminated when the Fourier modulus of the estimated image is equal to the measured modulus and all spatial domain constraints are met as well. Fienup concluded that the error-reduction algorithm converges rapidly for the first few iterations but slowly for later iterations. Thus, he introduced a modified algorithm called the Input-Output algorithm.

Another phase retrieval algorithm is the Knox-Thompson algorithm. Phase differences between adjacent points are not affected by spatial shifts. This algorithm estimates the phase by computing an average phase difference matrix and then summing the phase differences from the origin. The computed phase is then combined with a measured Fourier modulus (possibly obtained using Labeyrie's method).

An intuitive approach is offered by shift-and-add. If all points of an object are in the same isoplantic region, the object is preserved in all the images. If the object contains a point that is much brighter than the others, it is fair to assume that the brightest point in every image is a version of the same object point. The images are added together with respect to their brightest points to yield a diffraction limited image of the actual object. The adjusted-shift-and-add method was introduced to eliminate some of the background fog produced by shift-and-add. With the adjusted version, each image is multiplied by the value of the brightest point in the same image.

However, if the images are not properly aligned, spurious results such as "ghost" stars appear, which results when shift-and-add is performed on an object that does not contain a prominent point. Correlate-shift-and-add (CSAA) solves the alignment problem by correlating with respect to the entire image. Binary star simulations showed that CSAA produces a "ghostless" image, and if the dimmer star is more than 60% as bright as the other, a more accurate intensity ratio is yielded.

However, CSAA shows a weakness in aligning the centers of the stars. This may be due to correlating noise along with the object. Alignment errors may be reduced by masking out the region in the composite image that falls beyond the known extent of the object.

Masking the composite image may also improve CSAAs ability to resolve binary stars where one star is less than half as intense as the other. Averaging over more than 50 images may also produce clearer results.

CSAA produced favorable results on extended objects when the images were not too "messy". Better results may be achieved by performing CSAA on real images if only the "truer" images are selected. "Truer" images are those images which displays a fairly simple point spread function. Unfortunately, if the extent of the object were not known, intelligently selecting the better images may be virtually impossible. Masking the composite image may also improve CSAAs performance on extended objects.

Correlate-shift-and-add should be further researched using real and more than 50 images. Masking the composite image should also be seriously researched along with utilizing the degree of correlation to weigh each successive image. Recall that the cross-correlation operation indicates both the amount of shift and the degree of correlation. The degree of correlation may be used as a way to emphasize those images that correlate most with the composite image.

REFERENCES

REFERENCES

1. Labeyrie, A., "Attainment of Diffraction Limited Resolution in Large Telescopes by Fourier Analyzing Speckle Patterns in Star Images". Astron. and Astrophys. 6, pp. 85-87 (1970).

2. Dainty, J. C. (editor), Topics in Applied Physics, Vol. 9 Laser Speckle and Related Phenomena. New York:Springer-Verlag (1975).

3. Bates, R. H. T.,"Astronomical Speckle Imaging". Physics Reports 90, pp. 203-297 (1982).

4. Lynds, C. R., S. P. Worden, and J. W. Harvey, "Digital Image Reconstruction Applied to Alpha Orionis". The Astroph. J. 207, pp. 174-180 (1976).

5. Worden, S. P., C. R. Lynds, and J. W. Harvey, "reconstructed Images of Alpha Orionis Using Stellar Speckle Interferometry". J. Opt. Soc. Am. 66, No. 11, pp. 1243-1246 (1976).

6. Bates, R. H. T. and F. M. Cady, "Towards True Imaging by Wideband Speckle Interferometry". Optics Communications 32, No. 3, pp. 365-369 (1980).

7. Cady, F. M., Applications of Microcomputers in Interactive Image Processing. Thesis for Doctor of Phylosophy Degree at University of Canterbury, Christchurch, New Zealand (1980).

8. Hogbom, "Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines". Astron. Astrophys. Suppl. 15, pp. 417-426 (1974).

9. Knox, Keith T. and Brian J. Thompson, "Recovery of Images from Atmospherically Degraded Short-Exposure Photographs". The Astrophysical Journal 193, pp. L45-L48 (1974).

10. Knox, Keith T., "Image Retrieval from Astronomical Speckle Patterns". J. Opt. Soc. Am. 66, No. 11, pp. 1236-1239 (1976).

11. Bruck, Yu. M. and L. G. Sodin, "On the Ambiguity of the Image Reconstruction Problem". Optics Communications 30, No.3, pp. 204-308 (1979).

12. Fienup, J. R., "Space Object Imaging Through the Turbulent Atmosphere". Optical Engineering 18, No. 5, pp. 529-534 (1979).

REFERENCES--Continued

13. Fienup, J. R., "Phase Retrieval Algorithms: A Comparison". Applied Optics 21, No. 15, pp. 2758-2769 (1982).

14. Bates, R. H. T. and W. R. Fright, "Composite Two-Dimensional Phase-Restoration Procedure". J. Opt. Soc. Am. 73, No. 3, pp. 358-365 (1983).

15. Green, William B., Digital Image Processing, A Systems Approach. New York:Van Nostrand Reinhold Company Inc. (1983).

16. Castleman, Kenneth R., Digital Image Processing. Englewood Cliffs, N.J.:Prentice-Hall Inc. (1979).

17. Kuiper, G. P. and B. M. Middlehurst (editors), Telescopes. Chicago: The University of Chicago Press (1960).

18. Meyer-Arendt, J. R., Introduction to Classical and Modern Optics. Englewood Cliffs, N.J.:Prentice-Hall Inc. (1984).

19. Bergland, G. D., "A Guided Tour of the Fast Fourier Transform". IEEE Spectrum 6, No. 7, pp. 41-52 (1969).

APPENDIX

SOFTWARE ROUTINES

Overview

Various programs have been written to analyze and process speckle images. Some programs are simple and process only one image, whereas others are more complex and require multiple images to compute averages in both the spatial and frequency domains.

There are many techniques that have been developed to process speckle images. These programs have been designed to be used interdependently. In other words, a user may execute any number of these programs in a single command file to perform one of the many speckle interferometric techniques. Each program is described in this appendix, and examples are given as to their use.

Data Conversion Routines

An image file created on the micro-based image processing system (MIPS) is in a suitable form for processing on the MIPS but is not in a form suitable for processing on the VAX 780. Fourier transform programs, on the VAX, transform an image between the spatial and frequency domains. An image in the frequency domain consists of modulus and phase information and require that image files in the frequency domain consist of complex numbers, where a complex number consists of a real component and an imaginary component. Because much of the image processing is performed in the frequency domain, all image files that are to be processed on the VAX, whether they require complex numbers or not, must be in "complex" form.

Two programs have been written to make the image data compatible for either the VAX 780 or the micro image processing system. Data that is compatible for the micro system is in "video" form, and data that is in a form compatible for the VAX 780 processing routines is in "complex" form.

Video-To-Complex Routine --- "VITOCO"

This program converts a file of "video" form to one of "complex" form. This conversion is necessary to process the data using the Fourier transform and phase estimating programs on the VAX 780.

The logical unit name for the input file (in "video" form) is "INFILE", and the logical unit name for the output file (in "complex" form) is "OUTFILE".

The first record of both files contains the row dimension of the square matrix. The total number of pixels must be a multiple of 16 in the range 16 thru 65536 (where the matrix can be as small as 4 by 4 and as large as 256 by 256). The example below shows the "video" format for an 8 by 8 matrix.

The data records for the input file must contain a space followed by 16 ASCII representations of one byte hexadecimal numbers. Each number is represented by 2 bytes of ASCII hexadecimal code.

The data records for the output file contain 32 (16 complex pairs) decimal numbers (in E14.7 FORTRAN format) with a space before every number. The first number of a complex pair is the real component and the second number is the imaginary component.

Example

For example, suppose one wishes to convert the "video" formatted file, "VIDEO1.DAT;1", to "complex" form and store the "complex" version in "COMPLEX1.DAT;1". To do this, the following assignments and commands must be executed:

```
$ ASSIGN VIDEO1.DAT;1 INFILE
$ ASSIGN COMPLEX1.DAT;1 OUTFILE
$ RUN VITOCO
```

If "VIDEO1.DAT;1" contained

```
100   08
200    0001020304050607080908070605040З
300    10111213141516171819181716151413
400    20212223242526272829282726252423
500    30313233343536373839383736353433
```

"COMPLEX1.DAT;1" would contain

```
100   00008
200    0.0000000E+00  0.0000000E+00 ...
  .
  .
  .
500    0.4800000E+02  0.5100000E+02 ...
```

Complex-To-Video Routine --- "COTOVI"

This program converts a file in "complex" form to one in "video" form. This conversion is necessary for the processing of image data on the MIPS.

The logical unit name for the input file (in "complex" form) is "INFILE", and the logical unit name for the output file (in "VIDEO" form) is "OUTFILE".

Example

For example, suppose one wishes to convert the "complex" file, "COMPLEX1.DAT;1", to "video" form and store the "video" version in "VIDEO1.DAT;1". To do this, the following commands must be executed:

```
$ ASSIGN COMPLEX1.DAT;1 INFILE
$ ASSIGN VIDEO1.DAT;1 OUTFILE
$ RUN COTOVI
```

Fourier Transform Routines

Many imaging techniques are performed in the frequency domain. Therefore there must be a way of transforming images between the object and frequency domains. The VAX 780 contains Fast Fourier Transform software routines in the IMSL library that are used by the next two programs.

Fourier Transform Routine --- "FTRAN"

This program takes a "complex" matrix and performs a Fourier transform on it which transforms the image from its spatial domain representation to its frequency domain representation.

The logical unit name for the input file (in the spatial domain) is "INFILE", and the logical unit name for the output file (in the frequency domain) is "OUTFILE".

Both files are of the same dimension and in "complex" form as described in the video-to-complex routine documentation.

Example

For example, if one wishes to transform "OBJ1.DAT;1" to the frequency domain and store the results in "FREQ1.DAT;1", the following commands must be issued.

```
$ ASSIGN OBJ1.DAT;1 INFILE
$ ASSIGN FREQ1.DAT;1 OUTFILE
$ RUN FTRAN
```

Inverse Fourier Transform Routine --- "ITRAN"

This program takes a "complex" matrix and performs an inverse Fourier transform on it. The inverse Fourier transform transforms the image from its frequency domain representation to its object or spatial domain representation.

The logical unit name for the input file (in the frequency domain) is "INFILE", and the logical unit name for the output file (in the spatial domain) is "OUTFILE".

Like the Fourier transform program, both input and output files are in "complex" form and their matricies are of the same dimension.

Example

For example, if one wishes to transform "FREQ1.DAT;1" to the object domain and store the results in "OBJ1.DAT;1", the following commands must be issued:

```
$ ASSIGN OBJ1.DAT;1 INFILE
$ ASSIGN FREQ1.DAT;1 OUTFILE
$ RUN ITRAN
```

Intensity and Magnitude Routines

Some image processing methods use the Fourier modulus. These programs were written to compute the modulus and intensity of a given Fourier transform. They are also useful tools in analyzing Fourier transforms where the modulus or intensity can be computed and displayed on the video monitor.

Before executing the desired program, "INFILE" and "OUTFILE" must be assigned to the desired file names.

```
$ ASSIGN file-id. INFILE
$ ASSIGN file-id. OUTFILE
$ RUN program-id.
```

Fourier Modulus Routine --- "MAGNITUDE"

This program computes the magnitude (modulus) of the input file (in "complex" form) and outputs the results to another file (in "complex" form). The magnitude is stored in the real part and the imaginary part is zeroed for all complex numbers in the output file. The magnitude of $(a+jb)=\sqrt{a^2+b^2}$.

The logical unit name for the input file (or Fourier transform) is "INFILE", and the logical unit name for the output file (or Fourier modulus) is "OUTFILE".

Intensity Routine --- "INTENSITY"

This program computes the intensity of the input file (in "complex" form) and outputs the results to another file (also in "complex" form). The intensity (modulus squared) is stored in the real part of a complex pair and the imaginary part is zeroed. The intensity of $(a+jb)=a^2+b^2$.

The logical unit name for the input file (or Fourier transform) is "INFILE", and the logical unit name for the output file (or intensity) is "OUTFILE".

Amplitude Routine --- "AMPLITUDE"

This program computes the magnitude given the intensity. The intensity is assumed to be stored in the real part of the input file. The magnitude (or the square root of the intensity) is stored in the real part and the imaginary part is zeroed. "INFILE" is the logical unit name assigned to the intensity file and "OUTFILE" is assigned to the resulting amplitude file.

Conjugate Routine --- "CONJUGATE"

This program computes the conjugate of a given Fourier transform file, where the conjugate of (a+bj) is (a-bj). "INFILE" is assigned to the original Fourier transform file and "OUTFILE" is assigned to the resulting conjugate file.

Arithmetic Routines

These routines perform "complex" arithmetic operations between files (image matricies). Two matricies may be added or multiplied, or a matrix may be subtracted from or divided by another matrix. All arithmetic operations are performed between corresponding elements. For example, the addition of two matricies results in the following:

$$\begin{matrix} M1(1,1)+M2(1,1) & M1(1,2)+M2(1,2) & \cdots & M1(1,m)+M2(1,m) \\ \vdots & & & \\ M1(n,1)+M2(n,1) & M1(n,2)+M2(n,2) & \cdots & M1(n,m)+M2(n,m) \end{matrix}$$

The arithmetic programs require the assignment of three different files, two input matrix files and an output matrix file (where all three must be unique). Here is a list of the commands necessary to perform one of the arithmetic programs:

```
$ ASSIGN file-id. INFILE
$ ASSIGN file-id. INFILE2
$ ASSIGN file-id. OUTFILE
$ RUN program-id.
```

Addition Routine --- "ADD"

This program performs "complex" addition. The contents of "INFILE" are added with the contents of "INFILE2" and the result is stored in "OUTFILE". Complex addition operates as follows:

$$(a+jb)+(c+jd) = ([a+c]+j[b+d])$$

Subtraction Routine --- "SUBTRACT"

This program performs "complex" subtraction. The contents of "INFILE2" are subtracted from "INFILE" and the result is stored in "OUTFILE". Complex subtraction operates as follows:

$$(a+jb)-(c+jd) = ([a-c]+j[b-d])$$

Multiplication Routine --- "MULTIPLY"

This program performs "complex" multiplication. The contents of "INFILE" are multiplied by the contents of "INFILE2" and the result is stored in "OUTFILE". Complex multiplication operates as follows:

$$(a+jb)(c+jd) = ([ac-bd]+j[ad+bc])$$

Division Routine --- "DIVIDE"

This program performs "complex" division. The contents of "INFILE" are divided by the contents of "INFILE2" and the quotient is stored in "OUTFILE". Complex division operates as follows:

(a+jb)/(c+jd) = ([ac-bd]+j[bc-ad])

Initializing Routines

Fienup's algorithm to determine the phase for a given Fourier modulus requires an initial image to start the program. These programs can be executed to provide such an initial image.

Random Initializer --- "RANDOM"

This program randomly initializes a complex matrix (image file). The real parts are randomly initialized with real numbers between 0 and 255, and the imaginary parts are all zeroed.

"OUTFILE" is assigned to the file that is to be initialized. After assigning the output file and beginning execution, the program asks for the dimension of the matrix. One then enters the number of elements per side (the matrix is assumed to be square), and the program begins to initialize.

```
$ ASSIGN file-id. OUTFILE
$ RUN RANDOM
```

Specific Initializer --- "INITIAL"

This program is similar to the random initializer except the matrix is initialized with specific real and imaginary values. Once invoked, the program asks for the dimension of the matrix and for the real and imaginary values to initialize with. One then enters the size of the matrix, followed by the real value, followed by the imaginary value. The program then initializes the whole matrix with those values.

Predisplay Routines

There are 64 gray levels (where a gray level is designated in one byte) that can be recognized and processed by the MIPS. Therefore, before an image is converted to "video" format and displayed on the video monitor, its real part must not contain values greater than 255 or less than 0. Out of range values must be clipped or the whole file must be scaled.

Clipping-And-Scaling Routine --- "CLIPSCALE"

This program first clips and then scales the real part of a given image file. The program is interactive. It displays the maximum and minimum values and then requests the following values:

1. High clipping value
2. Low clipping value
3. Desired maximum value

"INFILE" is assigned to the file that is to be altered and "OUTFILE" is assigned to the clipped and scaled file. The following commands must be executed to perform this function.

```
$ ASSIGN file-id. INFILE
$ ASSIGN file-id. OUTFILE
$ RUN CLIPSCALE
```

Quadrant Swapping Routine --- "SWAPQUAD"

The Fourier transform routines treat the image files as periodic functions where the origin is located at the first element instead of at the center. In other words, when an image is Fourier transformed and the resulting Fourier modulus is displayed on the video monitor, the maximums are displayed at the corners instead of in the middle. Having the origin in the middle makes the modulus more recognizable. This program swaps quadrant 1 with 3 and 2 with 4 to relocate the origin at the center.

"INFILE" is assigned to the file (in "complex" form) that is to be altered and "OUTFILE" is the logical unit name assigned to the resulting file with the quadrants swapped.

Retrieve Image Routine --- "RETIMAGE"

This program simply retrieves an image from a file containing multiple images (all in "video" form). The input file can be created by by merging image files using the "APPEND" command in VMS. The averaging programs require that all images be contained in one file.

The program asks for the image index number, where "1" represents the first image. The image is then retrieved and stored in another file. "INFILE" is assigned to the file of multiple images and "OUTFILE" is assigned to the file which is to contain the desired image.

Shift-And-Add Routines

Shift-and-add is a method of obtaining a diffraction limited image by averaging a series of speckle images in the spatial domain. This method assumes that the brightest point in each image is a distorted version of the same object point. Images are shifted (with no rotation) to align their brightest points and then added together. The average is then computed by dividing the sum by the total number of images.

Original Shift-And-Add Routine --- "SHIFTADD"

This program shifts all images with respect to the first image in the input file. The program first asks for the number of images in the input file and then reads the first one to be used as a reference. The coordinate of the brightest pixel, in the first image, is displayed on the CRT, and the program asks if the coordinate is to be changed. At this point, the user enters the desired coordinate (y,x). A zero can be entered for a coordinate element that is to be unchanged. The first image is then shifted so that its brightest pixel is positioned at the specified coordinate. The other images are then read and added (while being shifted with respect to the brightest point in the first image).

The input file, assigned to the logical unit name "INFILE", must contain all the speckle images, and each image must be in "video" form (refer to the video-to-complex routine documentation). The result is stored (in "complex" form) in the file assigned to "OUTFILE". The result must be converted to "video" form before it can be transported to the MIPS.

Example

For example, suppose speckle images are stored in separate files (files with the file name but with different version numbers). The following commands can then be executed to perform the shift-and-add method.

```
$ !
$ !     PLACE ALL IMAGES IN ONE FILE
$ !
$ APPEND IMAGE.DAT;2,IMAGE.DAT;3,...  IMAGE.DAT;1
$ ASSIGN IMAGEOUT.DAT;1 OUTFILE
$ ASSIGN IMAGE.DAT;1 INFILE
$ !
$ !     PERFORM THE SHIFT-AND-ADD PROCEDURE
$ !
$ RUN SHIFTADD
```

At this point, the user interacts with the program by entering the number of images and the desired coordinates of the brightest pixel. The following commands are then executed to convert the "complex" file to "video" form to be transported to the MIPS.

```
$ ASSIGN IMAGEOUT.DAT;1 INFILE
$ ASSIGN IMAGEOUT.VID;1 OUTFILE
$ !
$ !     CONVERT TO "VIDEO" FORM FOR MICRO SYSTEM
$ !
$ RUN COTOVI
```

The "APPEND" command appends all images to "IMAGE.DAT;1" and "COTOVI" converts the resulting file from "complex" form to "video" form. If desired, the result can be clipped and scaled before converted to "video" format (refer to the scaling routine documentation).

Adjusted-Shift-And-Add Routine --- "ADJSHIFT"

This program is identical to the original shift-and-add program except that each speckle image is modified before being shifted and added. Each pixel, in an image, is multiplied by the brightest pixel of the same image. Bates claims that this modified version produces a better image.

Correlate-Shift-And-Add Routine --- "CORRSHIFT"

This program is also identical to the original shift-and-add program, except that all images are correlated with the running composite image to determine the amount of shift.

Frequency Domain Averaging

Speckle interferometry averages speckle images in the frequency or Fourier domain. If a series of speckle images are taken of an object and another series of a known point source, the squared Fourier modulus of the object can be obtained by dividing the average squared Fourier modulus of the object speckles by the average squared Fourier modulus of the point source speckles. The autocorrelation can then be obtained by performing the inverse Fourier transform on the squared modulus and can be used to resolve angular distances of simple objects, because the autocorrelation produces an image that is twice the diameter of the actual object.

Average-Squared-Fourier-Modulus Routine --- "MAGAVG"

This program computes the average squared modulus for a series of speckle images. Like the shift-and-add routines, the input file must contain all the images in "video" form. The program first asks for the number of speckle images then the following is performed for each image.

1. An image is retrieved from the input file in "video" form.

2. The image is converted to "complex" form for processing.

3. The "complex" version is then Fourier transformed and the squared Fourier modulus is computed.

4. The squared modulus is then added with the squared moduli of the other images.

After all the images have been processed the resulting squared modulus is divided by the total number of images to obtain the average squared modulus.

The input file, assigned to "INFILE", must contain all the speckle images. The resulting squared modulus is stored in the file assigned to "OUTFILE". The modulus is stored in "complex" form with the squared modulus contained in the real part and the imaginary part is zeroed.

Example

This is an example of how this program can be used to obtain the autocorrelation of an object given files containing the object images and files containing images of a known point source located near the object in question.

```
$ !
$ !     PLACE OBJECT IMAGES IN ONE FILE AND THE POINT
$ !     IMAGES IN ANOTHER FILE
$ !
$ APPEND OBJ.DAT;2,OBJ.DAT;3,...  OBJ.DAT;1
$ APPEND PNT.DAT;2,PNT.DAT;3,...  PNT.DAT;1
$ ASSIGN OBJ.DAT;1 INFILE
$ ASSIGN OBJ.SQU;1 OUTFILE
```

```
$ !
$ !     COMPUTE THE AVERAGE SQUARED MODULUS OF
$ !     THE UNRESOLVED OBJECT
$ !
$ RUN MAGAVG
```

At this point, the user interacts with the program by entering the number of images in the input file. The program then proceeds to compute the average squared modulus of the object images. The same procedure is followed to compute the average squared modulus of the known point source.

```
$ ASSIGN PNT.DAT;1 INFILE
$ ASSIGN PNT.SQU;1 OUTFILE
$ !
$ !     COMPUTE THE AVERAGE SQUARED MODULUS OF POINT
$ !     SOURCE
$ !
$ RUN MAGAVG
```

Like before, the user is requested to enter the number of images to average, and the average squared modulus of the point source is computed. The squared modulus of the object is then computed by dividing the average squared modulus of the object images by the average squared modulus of the point source.

```
$ ASSIGN OBJ.SQU;1 INFILE
$ ASSIGN PNT.SQU;1 INFILE2
$ ASSIGN COBJ.SQU;1 OUTFILE
$ !
$ !     COMPUTE SQUARED MODULUS OF RESOLVED OBJECT
$ !
$ RUN DIVIDE
$ ASSIGN COBJ.SQU;1 INFILE
$ ASSIGN AUTO.CMP;1 OUTFILE
$ !
$ !     PERFORM INVERSE FOURIER TRANSFORM TO COMPUTE
$ !     THE AUTOCORRELATION
```

```
$ !
$ RUN ITRAN
$ ASSIGN AUTO.CMP;1 INFILE
$ ASSIGN AUTO.VID;1 OUTFILE
$ !
$ !     CONVERT TO "VIDEO" FORM FOR THE MICRO SYSTEM
$ !
$ RUN COTOVI
```

The image, after having been transformed back to the spatial domain (by executing "ITRAN"), can be clipped and scaled before being converted to "video" format. Clipping and scaling can be used to enhance high intensities and or suppress lower intensities. The resulting file, "AUTO.VID;1", contains the estimated autocorrelation of the true object and can be transported to the MIPS to be displayed on the video monitor.

Phase Estimation Routines

If the phase for a given Fourier modulus can be determined, the object can be reconstructed. One method of attaining the phase is to iteratively convert the image from the object domain to the frequency domain and back to the object domain, imposing constraints in each domain. A phase solution is found when the error is reduced to within a specified tolerance.

Another method, given a series of speckle images, computes the average phase difference between adjacent points in the frequency domain. This method is known as the Knox-Thompson algorithm.

Knox-Thompson Algorithm --- "XKNOX" and "YKNOX"

These programs estimate the phase by computing the average phase difference between adjacent points. Like the shift-and-add programs, all images must be contained in one file and in "video" form. Also, the program starts off by requesting the number of speckle images to average.

The phase is computed by first computing the average phase differences between adjacent points and then summing the phase differences outward from the origin (where the phase at the origin is known to be 0).

"INFILE" is assigned to the file of images and "OUTFILE" is assigned to the resulting file that is to contain the phases in "complex" form. The output can then be multiplied by the modulus file to hopefully yield the Fourier transform of the true object. A reconstruction of the object in the spatial domain can be computed by performing the inverse Fourier transform on the modulus + phase file.

Example

This is an example of how these programs can be used to obtain an image given a series of speckle images (IMAGE.DAT) and the Fourier modulus (MOD.DAT;1). The Fourier modulus can be obtained thru speckle interferometric techniques.

```
$ !
$ !     PLACE ALL IMAGES IN ONE FILE
$ !
$ APPEND IMAGE.DAT;2,IMAGE.DAT;3,...  IMAGE.DAT;1
$ ASSIGN IMAGE.DAT;1 INFILE
$ ASSIGN PHASE.DAT;1 OUTFILE
$ !
$ !     PERFORM THE KNOX-THOMPSON ALGORITHM
$ !
$ RUN XKNOX
```

At this point, the user interacts with the program by entering the number of images in the input file, and the program proceeds to estimate the phase.

```
$ ASSIGN MOD.DAT;1 INFILE
$ ASSIGN PHASE.DAT;1 INFILE2
$ ASSIGN FOURIER.DAT;1 OUTFILE
$ !
$ !     COMBINE THE PHASE WITH THE MODULUS
$ !
$ RUN MULTIPLY
$ ASSIGN FOURIER.DAT;1 INFILE
$ ASSIGN RECON.DAT;1 OUTFILE
$ !
$ !     PERFORM INVERSE FOURIER TRANSFORM
$ !
$ RUN ITRAN
$ ASSIGN RECON.DAT;1 INFILE
$ ASSIGN RECON.VID;1 OUTFILE
$ !
$ !     CONVERT TO "VIDEO" FORM FOR MICRO SYSTEM
$ !
$ RUN COTOVI
```

"RECON.VID;1" can then be transported to the micro-based image processing system for processing and or displaying. "XKNOX" computes the phase differences between adjacent points in the same row, whereas "YKNOX" computes the phase differences between adjacent points in the same column.

Phase Averaging Routine --- "PHASAVG"

This program is designed to compliment the Knox-Thompson routines. The two Knox-Thompson programs compute the phase through different paths. This program can be executed to average the resulting phases and produce a better image. "INFILE" and "INFILE2" are assigned to the phase files, resulting from the two Knox-Thompson programs, and

"OUTFILE" is assigned to the file which is to contain the average phase information.

Input-Output Algorithm --- "FIENUP"

This program estimates the phase for a given Fourier modulus through an iterative procedure. Given an initial image (in the Fourier domain) and a Fourier modulus, this program transforms the image between the spatial and Fourier domains, imposing constraints in both domains. The Fourier domain constraint is the Fourier modulus and the spatial domain constraint is that all values are positive.

"INFILE" is the logical unit name assigned to the file which contains the Fourier modulus (in "complex" form). "INFILE2" is assigned to the file containing the initial image (also in "complex" form) to kick off the algorithm. "OUTFILE" is assigned to the file, which when after the algorithm has completed, contains the last image (in "complex" form). The last image may require clipping and scaling before being converted to "video" form and transported to the MIPS. "OUTFILE2" and "OUTFILE3" are assigned to files used for post analysis.

"OUTFILE2" is a file for storing iteration numbers and their respective squared error computations (in the spatial domain). The contents of this file may be plotted on the Versatec plotter by calling "PLOT" on the VAX 780 system. If this file is enabled and meant to be plotted on the plotter, the first two records must be previously set up with the first record containing the plot heading and the second record containing the dependent and independent variable names. Iteration numbers and error computations are simply appended to the file.

"OUTFILE3" is a file, which when enabled, is used for storing spatial domain images for specified iterations. Images are stored in consecutive versions of the assigned file name in "complex" form. Before an image can be transported to the MIPS, it must be clipped and scaled and then converted to "video" form, which can be accomplished by executing the appropriate programs on the VAX 780 system.

Example

For example, suppose one wished to determine the phase for the Fourier modulus in "MOD.DAT;1" and store the resulting Fourier transform in "TRAN.DAT;1". If "INIT.DAT;1" contains the initial image estimate and neither "OUTFILE2" nor "OUTFILE3" are to be enabled for post analysis, the following commands can be executed to perform Fienup's iterative algorithm.

```
$ ASSIGN MOD.DAT;1 INFILE
$ ASSIGN TRAN.DAT;1 OUTFILE
$ ASSIGN INIT.DAT;1 INFILE2
$ RUN FIENUP
```

At this point, the user must interact with the program by entering the squared-error limit. After the program accepts the error limit it requests the maximum iteration count. If no ceiling is desired the user enters "0". The error plot increment is then requested, which is used as a step counter for the iteration numbers and error computations. For this example, this file is not to be enabled, and so "0" is entered. But if it were enabled (by entering a non-zero integer) "OUTFILE2" must be previously assigned.

The next thing the program asks for is the strategy information for selecting the next input. The list of strategies are:

Strategy 1

$$\begin{array}{lll} Gk{+}1(x,y)=Gk'(x,y) & \text{for } Gk'(x,y) \text{ not in } \gamma & (50) \\ Gk{+}1(x,y)=Gk(x,y)-\beta Gk'(x,y) & \text{for } Gk'(x,y) \text{ in} & \end{array}$$

Strategy 2

$$\begin{array}{lll} Gk{+}1(x,y)=Gk(x,y) & \text{for } Gk'(x,y) \text{ not in } \gamma & (51) \\ Gk{+}1(x,y)=Gk(x,y)-\beta Gk'(x,y) & \text{for } Gk'(x,y) \text{ in } \gamma & \end{array}$$

Strategy 3

$$\begin{array}{lll} Gk{+}1(x,y)=Gk'(x,y) & \text{for } Gk'(x,y) \text{ not in } \gamma & (52) \\ Gk{+}1(x,y)=Gk'(x,y)-\beta Gk'(x,y) & \text{for } Gk'(x,y) \text{ in } \gamma & \end{array}$$

Strategy 4

$$\begin{array}{lll} Gk{+}1(x,y)=Gk(x,y) & \text{for } Gk'(x,y) \text{ not in } \gamma & (53) \\ Gk{+}1(x,y)=Gk'(x,y)-\beta Gk'(x,y) & \text{for } Gk'(x,y) \text{ in } \gamma & \end{array}$$

where $Gk(x,y)$ denotes the input for the kth iteration, $Gk'(x,y)$ its respective output, and γ is the set of all points that violates the non-negativity constraint. β is a constant that is specified by the user and effects the rate of convergence.

Strategy 3, with β=1.0, is the error-reduction algorithm strategy. Strategy 2 is used in Fienup's original Input-Output algorithm, and strategy 1 is used in Fienup's hybrid-Input-Output algorithm. It was concluded by Fienup that the hybrid strategy was the best of the three. However, this implementation allows the user to select any of the 4 strategies.

A user may even enter a series of strategies. In this case, the program cycles through a strategy list and invokes a strategy N number of times before invoking the next strategy in the list. N and β are specified by the user for each strategy. A strategy can be used more than once in a list, but the maximum number of strategy entries is 20.

The program first asks for the number of strategies to be used. The user must then enter that number and wait for the program to respond by asking for individual strategy information. The information must then be entered by typing in the strategy number, number of iterations per set (N), and the convergence constant (β). The program will accept as many strategies as specified.

The program then requests the number of iteration results. This option is used to analyze the convergence of the algorithm. If a number of iteration results (Gk'(y,x)) are to be saved for post analysis, an integer greater than 0 but not exceeding 20 must be entered. A "0" indicates no images are to be saved. If a non-zero number is entered, the iteration numbers for their respective outputs must then be entered. Results are stored in files with the same file name but with different version numbers. "OUTFILE3" must be previously assigned if this option is to be enabled.

The program then proceeds with the algorithm and monitors the squared error in the spatial domain by the formula,

$$\text{ERROR} = \frac{\sum_{x \varepsilon \gamma} \sum_{y \varepsilon \gamma} Gk'(x,y)^2}{\sum_{x} \sum_{y} Gk'(x,y)^2} \qquad (54)$$

where γ includes all those points that violates the non-negativity constraint. The program terminates and stores the last results if either the error is within the specified tolerance or the maximum iteration count is reached. The last iteration count and squared error computation is finally displayed on the CRT.

The output image must be clipped and scaled before it can be converted to "video" form and transported to the MIPS because it may contain negative values. Once clipped and converted, it can be transported to the MIPS and be displayed on the video monitor.