

**A SUPERRESOLUTION TELESCOPE THAT USES ABERRATION EFFECTS
SUPPRESSION, DECONVOLUTION BY DIMENSIONAL REDUCTION,
OPTIMAL CONVEXITY AND CONVEXITY NORMALIZATION FOR IMAGE
SIZE AND DARK NOISE**

**David Maker
MA. University of Northern Colorado, 1979**

**A dissertation submitted to the faculty of the
Oregon Graduate Institute of Science & Technology
in partial fulfillment of the
requirements for the degree
Doctor of Philosophy
in
Applied Physics**

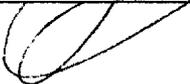
June 1997

The dissertation "A Superresolution Telescope that Uses Aberration Effects Suppression, Deconvolution By Dimensional Reduction and Optimal Convexity, and Convexity Normalization for Image Size and Dark Noise" by David Maker has been examined and approved by the following Examination Committee:

V. S. Rao Gudimetla, Thesis Advisor
Assistant Professor



J. Fred Holmes
Professor



James D. Parsons
Associate Professor

Martin E. Smithers
NASA Marshall Space Flight Center

TABLE OF CONTENTS

DEDICATION.....	vii
ACKNOWLEDGEMENTS.....	vii
LIST OF FIGURES.....	viii
ABSTRACT.....	xi
CHAPTER 1: Introduction.....	1
CHAPTER 2: Review of deconvolution algorithms.....	5
2.1 Review of Kirchoff integral.....	5
2.2 Incoherent sources.....	9
2.3 Noncoherent superresolution Techniques.....	11
2.4 Derivation of Least Squares and ML algorithm.....	13
2.5 Lucy Richardson algorithm.....	14
2.6 Van Cittert deconvolution.....	15
2.7 MAP and Maximum Likelihood Method.....	15
2.8 Microscope superresolution.....	16
2.9 Coherent superresolution.....	17
2.10 Gram-Schmidt orthogonalization.....	18
References.....	20
CHAPTER 3: Review of considerations concerning ambiguous image formation and SNR.....	22
3.1 Nyquist sampling theorem.....	23

3.2 Ambiguous image creation.....	23
3.3 Solution to the ambiguous image problem.....	24
3.4 Signal to noise ratio and the criteria for use of superresolution.....	30
3.5 Resolution dependence on SNR.....	31
3.6 Phase space representation of solution of ambiguous problem.....	32
References.....	34

**CHAPTER 4: A Superresolution telescope that uses aberration effects suppression
and deconvolution by dimensional reduction, optimal convexity
and convexity normalization for image size and dark noise.....**

4.1 PSF calculations.....	36
4.2 What is deconvolution.....	37
4.3 Solving matrix equation for k_i 's.....	38
4.4 Speedy calculation of k_i 's.....	38
4.5 Proof that (4.4.1) should be minimized.....	39
4.6 Illconditioned matrices and test sampling.....	40
4.7 Dealing with ambiguous image problem.....	42
4.8 Sampling of test data from isophote ridges.....	42
4.9 Sampling of test data for optimal second derivatives or optimal convexity....	48
4.10 PSF caused by annular apertures.....	50
4.11 Classification of aberrations.....	50
4.12 Astigmatism correction.....	51

4.13 Dark noise and shot noise.....	52
4.14 Convexity aberration caused by dark noise and shot noise and different image sizes.....	53
4.15 Parameter space.....	54
4.16 Sampling and large images.....	55
4.17 Experimental data.....	56
4.18 Simulations.....	72
4.19 CCD measurements.....	72
4.20 Astigmatism intrinsic to the CCD.....	73
4.21 Coma.....	74
4.22 Scattered light.....	74
4.23 Experimental considerations.....	74
4.24 Pinhole setup requirements.....	77
4.25 Telescopes.....	76
4.26 Comparison with other results.....	94
References.....	95
Appendix A: Deconvolution programs.....	97
Appendix B: Data transfer code.....	130
Appendix C: Planet finder program.....	133
Appendix D: Nyquist sampling program for know source numbers.....	140
Appendix E: Nyquist sampling with edge positions known.....	148

Appendix F: Other resolution techniques.....	152
F.1 PSF caused by aperture of variable opacity.....	152
F.2 Smoothing data by the method of fourth differences.....	153
References.....	155

ACKNOWLEDGMENT

A NASA fellowship with GSRP Grant Number NGT8-52803 helped me to complete this work. Work was done in the Astrionics Laboratory at the Marshall Space Flight Center under Robert Rood. Thanks to Rao Gudimetla, Martin Smithers, Howard Hall, Jim Parsons, and Fred Holmes and my wife Kathy and Allison my daughter. Also thanks to SBIG (Santa Barbara Instrument Group) for giving me valuable advice on CCD cameras.

LIST OF FIGURES

- Fig. 3-1 Three sources in a row.
- Fig. 3-2 Three sources that give the same image as figure 3-1.
- Fig. 3-3 Three sources that give the same image as figure 3-1.
- Fig. 3-4 Three sources that give the same image as figure 3-1.
- Fig. 3-5 Ambiguous image problem in two dimensions.
- Fig. 3-6 Graph of sum of k 's vs multidimensional phase space denoted by PX , PY and k
- Fig. 4-1 Ridge line for three point sources line up in a row.
- Fig. 4-2 Intensity for three point sources lined up in a row.
- Fig. 4-3 Ridge lines for three point sources forming a triangle.
- Fig. 4-4 Intensity vs X, Y for three point sources forming a triangle.
- Fig. 4-5 Second derivative of intensity as a function of radial distance.
- Fig. 4-6 Photo (X100) of three pinhole point sources.
- Fig. 4-7 Raw image from three point sources, $1/4$ Rayleigh Distance apart, $SNR \approx 60$.
- Fig. 4-8 Restoration of figure 4-7 achieved with deconvolution.
- Fig. 4-9 Photo (X100) of two pinhole point sources.
- Fig. 4-10 Raw image of two point sources placed $1/4$ Rayleigh distance apart, $SNR \approx 60$.
- Fig. 4-11 Restoration of figure 4-10 achieved with deconvolution.
- Fig. 4-12 Photo (X100) of three pinhole point sources in triangular configuration.
- Fig. 4-13 Raw image three point sources in triangular configuration about $1/4$
Rayleigh distance apart, $SNR \approx 60$.
- Fig. 4-14 Restoration of figure 4-13 achieved through deconvolution.
- Fig. 4-15 Photo (X100) four pinhole point sources.
- Fig. 4-16 Raw image of four point sources about $1/4$ Rayleigh distance apart, $SNR \approx 60$.
- Fig. 4-17 Restoration of figure 4-16 achieved through deconvolution.
- Fig. 4-18 Photo (X100) of two pinhole point sources.
- Fig. 4-19 Raw image of two point sources placed $1/10$ Rayleigh distance apart, $SNR \approx 60$.

- Fig. 4-20 Restoration of figure 4-19 achieved by deconvolution.
- Fig. 4-21 PC setup for pinhole experiment.
- Fig. 4-22 CCD setup for pinhole experiment.
- Fig. 4-23 Laser source, pinhole object and pinhole aperture.
- Fig. 4-24 Pinhole experimental setup.
- Fig. 4-25 Photo of superresolution telescope design in hallway.
- Fig. 4-26 Superresolution telescope CODE V ray trace.
- Fig. 4-27 Spot diagram for superresolution telescope.
- Fig. 4-28 Superresolution telescope design.
- Fig. 4-29 Actual PSF with aberrations in ordinary Newtonian telescope.
- Fig. 4-30 Raw image of two point sources about $1/4$ Rayleigh distance apart, $SNR \approx 30$.
- Fig. 4-31 Restoration of figure 4-29 achieved through deconvolution.
- Fig. 4-32 Raw image of three point sources about $1/6$ Rayleigh distance apart, $SNR \approx 30$,
forming a triangle.
- Fig. 4-33 Restoration of telescope image figure 4-31 achieved through deconvolution.
- Fig. 4-33 Raw image of four point sources about $1/6$ Rayleigh distance apart, $SNR \approx 30$.
- Fig. 4-34 Restoration of telescope image figure 4-32 achieved through deconvolution.

LIST OF TABLES

Table 2-1 Sample algorithms

ABSTRACT

A SUPERRESOLUTION TELESCOPE THAT USES ABERRATION EFFECTS SUPPRESSION, DECONVOLUTION BY DIMENSIONAL REDUCTION, OPTIMAL CONVEXITY AND CONVEXITY NORMALIZATION FOR IMAGE SIZE AND DARK NOISE

David Maker

Supervising Professor: V.S. Rao Gudimetla

In this dissertation we claim to have found the solution to the problem of resolving beyond the diffraction limit (superresolution). This problem is solved by dimensional reduction, convexity optimization, and convexity normalization for image size and dark noise. By dimensional reduction we mean deconvolution on isophote ridges, which are one dimensional, thus we have reduced the dimensionality of the problem from two to one. By optimizing convexity we mean that we choose points to test for image sources for which the second derivative (convexity) of the intensity along isophote ridges is the highest. By normalization of convexity for dark noise and image size we are making sure that our optimization of convexity is not biased by dark noise at different exposures or different background convexities for images of different sizes. This biasing would create artifacts.

We also invented ways to speed up our computation and overcome inverse matrix errors. For example we found a simple way to solve the illconditioned matrix problem so we could use the inverse matrix technique, and we are allowed here to replace explicit

least squares with the more convenient minimum of the sum of amplitudes squared. We use methods to overcome astigmatism and spherical aberration which are not new. With a narrow field of view we don't need to use the usual iterative stochastic methods (such as MAP). This is because smoothing is effective here since the scale of the PSF s (point spread functions) is much larger than the noise scale.

In this superresolution telescope we get a narrow field of view by a microscope-telescope combination. Pointing errors must be minimized to ensure that aberration effects are minimized, and astigmatism produced by air turbulence must be corrected for.

Experiments have produced repeatable 1/10 Rayleigh distance resolution for SNR =60 (with no prior knowledge of source configuration assumed). Through significant air turbulence over a 400 foot line of sight we get 1/6 Rayleigh resolution for 1.5 inch reflecting and refracting telescopes, about a factor of 12 better than you would expect.

CHAPTER 1

INTRODUCTION

Toraldi di Francia¹, in 1952, proposed that it must be possible to reconstruct details of an object smaller than the diffraction limit. We call this superresolution. Previous theories of resolution by Abbe² and Lord Rayleigh³ did not allow such resolution. But at least in optical microscopy superresolution has been achieved. What has been shown to be important is the information content of the optical signal, including some prior knowledge of the signal, but not arbitrary limits on resolution. The key thing in doing superresolution is to make use of this information in an intelligent fashion. For telescope superresolution localizing the testing of an individual PSF (point spread function) is important. In confocal microscopy the extent of the laser beam coverage is known. Many methods have been developed for achieving superresolution. For example the MAP (maximum a posteriori) method⁴ involves producing an iteration of an image function, maximizing the entropy of the image and satisfying the least squares criteria. However all these methods must be viewed in the context of Toraldo di Francia's concept of the ambiguous image⁵. The concept of an ambiguous image is that two or more different objects can give rise to the same image. Much has been written about how this ambiguous image problem will always provide a limit to superresolution.

The original contribution of this dissertation, as discussed in chapter 4, is the introduction of five new methods that, when used together, overcome this ambiguous image problem in the case of narrow field of view, high magnification optical telescopes. The five

new methods we have adopted are dimensional reduction, second derivative optimization, replacing the Nyquist sampling⁶ requirement by a requirement for dense symmetrical sampling, a method of using the inverse matrix technique without the usual large resulting errors and replacing the least squares with a minimum of the sum of amplitudes squared. Other more mundane innovations are methods for speedy numerical utilization of the Bessel functions. We use ways to overcome astigmatism and spherical aberration which are not new. As part of the telescope work we find new ways of overcoming aberrations caused by secondary mirror supports.

More specifically, the dimensional reduction involves testing points that are on isophote ridges. Tops of ridges form lines so we have reduced this from a two dimensional problem to a one dimensional problem, which is what we mean by dimensional reduction. Most of the computer algorithm is used to calculate the location of these ridge points.

We find points on these ridges that have maximum second spatial derivatives in intensity. These approximate source locations since the maximum of PDF s (probability distribution functions) are at second derivative maxima.

The Nyquist requirement of sampling of peaks and valleys of sine waves in all locations of the image is not used here except in the planet-finder simulation. The convexity test causes us to symmetrically sample on either side of a maximum. This also results in a large reduction in the illconditioned matrix problem so that the inverse matrix method can then be used. Smoothing is also very effective here since the scale of the noise is much smaller here than the scale of a PSF because of the narrow field of view. This eliminates the need for stochastic methods. The usual Gaussian elimination is used here in the inverse matrix calculation.

We sum the absolute powers of the square of the amplitudes, then test to see if the sum is a minimum and, when it is, conclude that particular amplitude and position applies to that source.

The calculation of the Bessel functions involves time consuming numerical integration. Here we use a representation that has no singularities and make a file of a table of numerical values these functions. This speeded up calculation by thousands of times.

Finally we solve the problem of convexity aberration due to dark noise and varying image size.

Pinhole experiments were done to check the theory. The pinhole experimental results imply that the technique is successful but difficult to use. At a SNR=30 (signal to noise ratio) we are able to resolve 2, 3 and 4 objects with a minimum separation of 1/10 of a Rayleigh distance using actual CCD data. For reflector and refractor telescopic superresolution point sources separated by 1/6 of a Rayleigh were resolved through turbulence, which is about a factor of 12 better than you could expect. This worked for several very different combinations of amplitudes and separations. Patterns such as triangles and worked with very different intensities for the sources.

In this thesis we derive our method theoretically and compare and contrast this method with other superresolution methods. For example in chapter 2 the standard derivation from Maxwell's equations of the Kirchoff integral are reviewed. From there we derive the PSF (point spread function) for a circular aperture. We then review the mathematics associated with noncoherent PSF's. In chapter 2, alternative superresolution methods are discussed such as the maximum entropy deconvolution. We also discuss microscope superresolution. We review some of the most recent developments, , namely the Gram-Schmidt orthogonalization procedure applied to superresolution. Chapter 3 reviews general considerations about superresolution such as ambiguous image and SNR considerations. It should be noted that our experimental results are within the theoretical bounds set there. Chapter 3 presents a mathematical model of the ambiguous image problem. In chapter 4 our method for doing superresolution is outlined. Here we include actual image restoration results. Also chapter 4 presents steps that must be taken to make this type of superresolution repeatable. In appendix A we include our superresolution code.. Appendix B includes our data transfer code. In appendix C we have our planet finder program, and in appendix D and E we include our Nyquist sampling programs. In appendix F we discuss various miscellaneous hardware techniques that might be helpful in getting superresolution such as apodization.

Thus we begin by noting that the electromagnetic theory of imaging is based on Maxwell's equations. Using Maxwell's equations we can find a wave equation for the electric (E) and magnetic (B) fields for free space propagation. These equations can be solved in the volume between a screen and aperture. This will give the formula for the amplitude function for the wave on the screen. Closely related to this amplitude function is the Kirchoff diffraction integral. Although not strictly a rigorous solution to this wave equation, it provides the basis for the theory of Fraunhofer and Fresnel diffraction. Here we will deal with Fraunhofer diffraction. We solve the Kirchoff diffraction integral for a circular aperture and distant point source and get the PSF (point spread function). Superposition of PSF's from many point sources can be used to construct images. The large degree of freedom for assignment of these PSF's to an arbitrary image is the basis of the ambiguous image problem.

References

- 1) G. Toraldo di Francia, Super Gain Antennas and Optical Resolving Power, *Nuovo Ciento*, 9 (3), 1952, p. 426.
- 2) E. Abbe, Beitrage zur Theorie des Mikroskops un der Mikroskopischen Wahrnehmung, *Archiv. Mikrosk., Anal*, 9 (27), 1873, p.413.
- 3) Lord Rayleigh, On the Theory of Optical Images, with Special Reference to the Microscope, *The London, Edinburgh and Dublin Philosophical Magazine and Journal*, 42 (5), 1896, p.167.
- 4) P. Sementelli, Analysis of the Limit of Superresolution, *J. Opt. Soc. of Am. A.*, 10 (11), 1993. p.2265.
- 5) G. Toraldo Di Francia, Resolving Power and Information, *J. Opt. Soc. of Am.*, 45 (7), 1955, p.497.
- 6) J. Lim, *Advanced Topics In Signal Processesing*, Prentice Hall, 1988, p.128.

CHAPTER 2

REVIEW OF DECONVOLUTION ALGORITHMS

This is a discussion of alternative superresolution methods, such as the maximum entropy deconvolution, and also a discussion of microscope superresolution. We review some of the most recent developments in superresolution, namely the Gram-Schmidt orthogonalization procedure. It is important to note that of the methods reviewed here almost all use standard Fourier transform image processing as in equation (2.3.7) and (2.3.10). The method we develop cannot use manipulation of sampling in frequency space (involving Fourier transforms) because of the need for ridge sampling in real space as will be seen in chapter 4. Also the real space methods introduced here all require Nyquist type sampling, whereas our method uses ridge and convexity sampling. Thus our method is very different from the standard ways of doing superresolution. Before introducing these standard methods, the development of the PSF (point spread function) for light passing through a circular aperture, must be reviewed.

2.1 REVIEW OF KIRCHOFF INTEGRAL

Recall Maxwell's equations in the Gaussian system¹ of units

$$\nabla \cdot \vec{D} = 4\pi\rho \quad (2.1.1)$$

$$\nabla \cdot \vec{B} = 0 \quad (2.1.2)$$

$$\nabla \times \vec{E} = -\frac{1}{c} \frac{\partial \vec{B}}{\partial t} \quad (2.1.3)$$

$$\nabla \times \vec{H} = \frac{4\pi}{c} \vec{J} + \frac{1}{c} \frac{\partial \vec{D}}{\partial t} \quad (2.1.4)$$

Recall that in free space²

$$\vec{B} = \vec{H} \quad (2.1.5)$$

and

$$\vec{D} = \vec{E} \quad (2.1.6)$$

Also equation (2.1.1) in free space becomes:

$$\nabla \cdot \vec{D} = 0 \quad (2.1.7)$$

so from equation (2.1.6):

$$\nabla \cdot \vec{E} = 0 \quad (2.1.8)$$

and also in free space:

$$\vec{j} = 0 \quad (2.1.9)$$

and from equation (2.1.5) and (2.1.2):

$$\nabla \cdot \vec{H} = 0 \quad (2.1.10)$$

Recall the vector identity

$$\nabla_X(\nabla_X \vec{A}) = \nabla(\nabla \cdot \vec{A}) - \nabla^2 \vec{A} \quad (2.1.11)$$

Next we take the curl (∇_X) of both sides of (2.1.3) and use (2.1.5) to get:

$$\nabla_X(\nabla_X \vec{E}) = \frac{1}{c} \frac{\partial}{\partial t} (\nabla_X \vec{B}) \quad (2.1.12)$$

Then we plug in for the left side³ of (2.1.12) into the identity given by (2.1.11). We also plug in on the right side of (2.1.12) equations (2.1.4), (2.1.5) and (2.1.9) to get:

$$\nabla(\nabla \cdot \vec{E}) - \nabla^2 \vec{E} = \frac{1}{c} \left(\frac{1}{c} \frac{\partial^2}{\partial t^2} \vec{D} \right) = \frac{1}{c} \left(\frac{1}{c} \frac{\partial^2}{\partial t^2} \vec{E} \right) \quad (2.1.13)$$

Next we plug in equation (2.1.8) into (2.1.13) and get:

$$\nabla^2 \vec{E} - \frac{1}{c^2} \frac{\partial^2}{\partial t^2} \vec{E} = 0 \quad (2.1.14)$$

This is the wave equation. Recall that for a plane wave in a vacuum that $\omega = kc$. We take a Fourier component of \vec{E} , which we write² as ψ , to be a plane wave of the form:

$$\psi(x,t) = A e^{i\omega t} \Psi(x) \quad (2.1.15)$$

We give a transverse component of \vec{E} the name $\psi(x)$. We then plug equation (2.1.15) into equation (2.1.14) and get a Helmholtz equation in $\Psi(x)$:

$$(\nabla^2 + k^2)\Psi(x) = 0. \quad (2.1.16)$$

Thus we can write the Green's function problem for this Helmholtz wave equation as:

$$(\nabla^2 + k^2)G(x, x') = \delta(x - x') \quad (2.1.17)$$

Using Green's theorem and equations (2.1.16) and (2.1.17) we get:

$$\Psi(x) = \oint_S [\Psi(x')n'\nabla'G(x, x') - G(x, x')n'\nabla'\Psi(x')]da' \quad (2.1.18)$$

Here n' is inwardly normal to the surface S which encloses a volume V . Equation (1.1.18) is nonzero only within this volume V . Next we take G to be the infinite space Green's function for outgoing waves, given by:

$$G(x, x') = \frac{e^{ikR}}{4\pi R} \quad (2.1.19)$$

Lets say the surface S consists of a screen and an aperture. If the screen is far away then we can say that the main contribution to the integral (2.1.18) comes from the integration over the aperture. Also the ψ satisfies the radiation condition for intensities that obey the inverse square law:

$$\Psi \rightarrow f(\theta, \phi) \frac{e^{ikr}}{r} \quad (2.1.20)$$

Thus equation (2.1.18) can be rewritten as:

$$\Psi(x) = -\frac{1}{4\pi} \int_S \frac{e^{ikR}}{4\pi R} n' \left[\nabla' \Psi + ik \left(1 + \frac{1}{kR} \right) \frac{\partial \Psi}{\partial R} \right] da' \quad (2.1.21)$$

We next make the assumption that Ψ and $\frac{\partial \Psi}{\partial n}$ vanish everywhere⁴ except in the openings and in the openings have the corresponding values of incoming plane wave in the free space. It can be shown that these boundary conditions should yield only trivial Ψ here. In any case for infinitely distant object giving waves that are plane waves impinging perpendicularly at the opening we have that:

$$\Psi(P) = \frac{k}{2\pi i} \int_{\text{opening}} \frac{e^{ikr}}{r} da' \quad (2.1.22)$$

Next we use (2.1.22) to derive our basic algorithm. So let x_i be a position in the pupil and

X, Y be a position on the screen. Let D be the axial distance from the pupil to the screen.

Then

$$r^2 = D^2 + (X - x)^2 + (Y - y)^2 \quad (2.1.23)$$

so that

$$r = \sqrt{D^2 + (X - x)^2 + (Y - y)^2} = D \left[1 + \frac{(x - X)^2}{2D^2} + \frac{(y - Y)^2}{2D^2} \right] \approx$$

$$D + \frac{x^2 + y^2}{2D} + \frac{X^2 + Y^2}{2D} + \frac{Xx + Yy}{D} \quad (2.1.24)$$

The phase factor is just a multiplicative constant and can be ignored.

In Fraunhofer diffraction $X^2 + Y^2 \ll xX + yY$ so we can write:

$$r \approx \frac{xX + yY}{D} = \frac{\vec{R} \cdot \vec{r}}{D} = \frac{R \cos \theta}{D} \quad (2.1.25)$$

Here α is the angle from the line drawn perpendicular and through the center of the aperture, λ is the wavelength of the light. Let "a" be the radius of the aperture, $b = \frac{2\pi}{\lambda} \sin \alpha$.

Thus we can write⁵ equation (2.1.22) as:

$$\Psi \sim \int_0^a \int_0^{2\pi} e^{ibr \cos \theta} r d\theta dr \quad (2.1.26)$$

$$\text{But } J_0(x) = \frac{1}{2\pi} \int_0^{2\pi} e^{ix \cos \theta} d\theta$$

so we write equation (2.1.26) as

$$\Psi \sim \int_0^a J_0(br) r dr = \frac{2\pi a^2}{ab} J_1(ab) \quad (2.1.27)$$

so that:

$$\Psi \sim \frac{J_1(R)}{R} = \text{Jinc}(R) \quad (2.1.28)$$

where $R \equiv \frac{2\pi a}{\lambda} \sin \alpha$. Here α is the angle from the line drawn perpendicular to the plane of and through the center of the aperture, λ is the wavelength of the light. Thus we have derived the PSF (point spread function) of a circular aperture from Maxwell's equations. The first null of the $J_1(R)$ is at the point

$$\frac{2\pi a}{\lambda} \sin \alpha = 3.8317 \quad (2.1.29)$$

So that we have

$$\sin \alpha = \frac{3.8317\lambda}{2\pi a} \approx \frac{1.22\lambda}{a} \approx \alpha \quad (2.1.30)$$

Translating the angle α to the dimensions of the screen we define the α to be the angular distance. If another PSF of an object of equal brightness was placed so that the center of that PSF was located at this angle α from the center of the first PSF, we say that the two PSF's are just resolvable by the Rayleigh criterion.

2.2 INCOHERENT SOURCES

Next we derive the equation for the intensity of the two dimensional pattern created by incoherent sources. Let us have two sources with fields E_1 and E_2 that differ by a random phase γ since the sources are incoherent. We then take the average of the square of the field E_T . Thus let

$$E_1 = E_{10} \cos \omega t \quad (2.2.1)$$

$$E_2 = E_{20} \cos(\omega t + \gamma) \quad (2.2.2)$$

so that

$$E_T = E_{10} \cos \omega t + E_{20} \cos(\omega t + \gamma) \quad (2.2.3)$$

Thus:

$$E_T^2 = (E_{10} \cos \omega t + E_{20} \cos(\omega t + \gamma))^2 =$$

$$=E_{10}^2(\cos \omega t)^2 + E_{20}^2(\cos(\omega t + \gamma))^2 + 2E_{10}\cos \omega t E_{20}\cos(\omega t + \gamma) \quad (2.2.4)$$

Taking the average $\langle \rangle$ of both sides of equation (2.2.4) we get for the intensity I:

$$\begin{aligned} I = \langle E_T^2 \rangle &= \langle (E_{10}\cos \omega t + E_{20}\cos(\omega t + \gamma))^2 \rangle = \\ &= \langle (E_{10}\cos \omega t)^2 \rangle + \langle (E_{20}\cos(\omega t + \gamma))^2 \rangle + \langle 2E_{10}\cos \omega t E_{20}\cos(\omega t + \gamma) \rangle \end{aligned} \quad (2.2.5)$$

We note here that for small random phase differences γ the last term is small. In our experiment such small phase differences existed close the middle of the optical axis of the system, but not at the edges. So even though the noncoherence may have been imperfect, we still saw phenomena that were close to what we would see for complete noncoherence. One experimental consequence of residual coherence is that off center images would not deconvolve nearly as well as the centered images due to the large phase difference γ that exists there.

For complete noncoherence the phase difference γ is random and the last term in equation (2.2.5) averages to zero. Thus the intensity "I" of the E field is given by:

$$\text{Intensity} = \langle E_T^2 \rangle = \langle E_1^2 \rangle + \langle E_2^2 \rangle \quad (2.2.6)$$

The E field on the screen created by a circular aperture is given by (1.1.28)

$$|\bar{E}| = \sqrt{k} \frac{J_1(R)}{R} \quad (2.2.7)$$

where k is the strength of the source and not wave number. Equation (2.2.6) then implies that the intensity on the screen is given by:

$$\text{Intensity} = \sum_i k_i \left(\frac{J_1(R_i)}{R_i} \right)^2 \equiv \sum_{i=1}^N k_i JINC^2(R_i) \quad (2.2.8)$$

This is the equation for the ideal noiseless pixel electron density in an image on a CCD caused by the "i" sources if imaged through a circular aperture. Equation (2.2.8) is the fundamental equation used in the inverse matrix part of our algorithm. In fact any image created by this aperture can be constructed for these PSF's (see 2.2.8):

$$\sum_{i=1}^N k_i J_1^2(R_i) = I(x, y) = \sum_{i=1}^N k_i \left(\frac{J_1(R_i)}{R_i} \right)^2 \quad (2.2.9)$$

where $R_i = \sqrt{((x_i - PX_i)^2 + (y_i - PY_i)^2)}$.

2.3. NONCOHERENT SUPERRESOLUTION TECHNIQUES

In this section we review some of the standard methods for doing superresolution. We will refer to later results in this paper to establish the nomenclature for this literature review.

All superresolution methods that I know of are based on a version of the inverse matrix method or a stochastic application of this method. The inverse matrix method itself gives poor superresolution and the other methods are modifications or corrections to it. We use as an example a circular aperture and begin with equation (2.2.8).

Let R_i be the distance between the measuring point (x, y) and the point where the source is located at (PX_i, PY_i) so that from the Pythagorean theorem:

$$R_i = \sqrt{((x - PX_i)^2 + (y - PY_i)^2)} \quad (2.3.1)$$

Thus from equation (2.2.8) we have for the intensity at point (x, y) :

$$I(x, y) = \sum_{i=1}^N k_i \left(\frac{J_1(R_i)}{R_i} \right)^2 \quad (2.3.2)$$

k_i is the intensity at point (PX_i, PY_i) , and $J_1(R)$ is the Bessel function of the first kind of order 1. We can then define for a circular aperture:

$$H(x - PX_i, y - PY_i) = \left(\frac{J_1(R_i)}{R_i} \right)^2 \quad (2.3.3)$$

In the limit as the sources become continuous:

$$k_i \rightarrow f(PX, PY)_i \quad (2.3.4)$$

which is our irradiance distribution. We define:

$$g(x,y) \equiv I(x,y) \quad (2.3.5)$$

Thus in the limit of a continuous summation in equation (2.3.2) we have:

$$\begin{aligned} g(x,y) &= \int_{PY1}^{PY2} \int_{PX1}^{PX2} f(PX,PY)H(x - PX,y - PY)d(PX)d(PY) \\ &\equiv h(x,y) \bullet f(x,y) \end{aligned} \quad (2.3.6)$$

Equation (2.3.6) is a convolution integral. $H(w_x, w_y)$ is our optical transfer function (OTF). Restricting our attention to spatially invariant systems, we can apply a Fourier signal processing technique which is a two dimensional convolution. Thus we do a Fourier transform of equation (2.3.6) and get:

$$G(w_x, w_y) = H(w_x, w_y)F(w_x, w_y) \quad (2.3.7)$$

Here G, H and F are Fourier transform pairs of the functions g, h and f respectively. Note that now our problem of determining f has been made into a simple algebra problem. Thus we solve for F and take the inverse Fourier transform. For example in the case of a circular aperture and a defocused lens we have that⁶:

$$H(w_x, w_y) = \frac{J_1(aR)}{aR} \text{ where } R = \sqrt{w_x^2 + w_y^2} \quad (2.3.8)$$

Another example is the optical transfer function (OTF) for atmospheric blur is given by:

$$H(w_x, w_y) = \exp\left(-\frac{(w_x^2 + w_y^2)}{2\sigma^2}\right) \quad (2.3.9)$$

and for uniform motion blur:

$$H(w_x, w_y) = \frac{\sin a\omega}{\omega} \text{ where } \omega = w_x \cos\theta + w_y \sin\theta \quad (2.3.10)$$

To use equation (2.3.7) on a computer the H, OPT, the F and the G become discrete matrices. All of the rest of the techniques below use (2.3.7). Thus a Fourier transform is assumed to have been used here. Including noise we can write:

$$g=[H]f+n \quad (2.3.11)$$

for a generalization of equation (2.3.7), where n is the noise contribution.

2.4 DERIVATION OF LEAST SQUARES ALGORITHM

Let $[\phi_a]$ be the standard deviation for the quantity "a". We now can use the Bayes estimation procedure² to derive the least squares algorithm. The Bayes estimation procedure applied to the problem given in equation (2.3.11) is that we maximize $p(f|g)$. Bayes law says that:

$$p(f|g) = \frac{p(g|f)p(f)}{p(g)} \quad (2.4.1)$$

We use the central limit theorem and say that the image noise is Gaussian and so we can write that:

$$p(n) = K_1 \exp \left\{ -\frac{1}{2} n'^{-1} [\phi_n]^{-1} n' \right\} \quad (2.4.2)$$

where $[\phi_n]$ is the standard deviation for the noise, $n' = n - \hat{n}$. The K_1 's here are normalization constants which can be ignored. Also we have that:

$$p(f) = K_2 \exp \{ -((f - \bar{f})') [\phi_f]^{-1} (f - \bar{f}) \} \quad (2.4.3)$$

$p(g)$ amounts to a scaling and can be left unspecified. We can plug in the noise in equation (2.4.3) into equation (2.4.2) and we get:

$$p(g|f) = K_3 \exp \left\{ -\frac{1}{2} (g - [H]^{-1} f)' [\phi_n]^{-1} (g - [H]^{-1} f) \right\} \quad (2.4.4)$$

Using equation (2.4.2) and equation (2.4.1) and taking the logarithm, we finally have:

$$\begin{aligned} \ln p(g|f) &= -\frac{1}{2} (g - [H]^{-1} f)' [\phi_n]^{-1} (g - [H]^{-1} f) - \frac{1}{2} (f - \bar{f})' [\phi_f]^{-1} (f - \bar{f}) + \\ &= \ln p(g) + K \end{aligned} \quad (2.4.5)$$

To maximize the $p(f|g)$ in equation (2.4.2), we take the derivative of equation (2.4.1) with respect to the individual f_i s and set it equal to zero. If the first and second terms are independent, then for the first term this condition means:

$$(g - [H]^{-1} f)' [\phi_n]^{-1} (g - [H]^{-1} f) = \eta^2 \quad (2.4.6)$$

where η is a constant. This is just the least squares criterion. If furthermore f is a nonrandom quantity then equation (2.4.7) is realised exactly and we have that:

$$f = ([H]^{-1})g \quad (2.4.7)$$

and this is the Maximum Likelihood (ML) estimate or inverse matrix method.

Given these definitions and equations we can now summarize several of the image restoration and superresolution techniques in the following table⁸.

Table 1: Sample algorithms

Name	Algorithm
Inverse	$f = ([H]^t [H])^{-1} [H]^t g$
ML(Maximum Likelihood)	$f = ([H]^{-1})g$
Constrained Least Squares	$f = ([H]^t [H] + \gamma([Q]^t [Q]))^{-1} [H]^t g$
MAP	$f_{n+1}(x) = f(x) \exp \left\{ \left[\frac{g(x)}{f_n(x)H(x)} - 1 \right] H(x) \right\}$

$[Q]$ is a linear operator matrix that differs for different models, $f(x)$ is the apriori characterization of the object photon emission rate, $f_n(x)$ is estimate of $f(x)$ at the n th iteration, $g(x)$ is image data, γ is a Lagrange multiplier that is chosen to include the fact that noise must be added to g in equation (2.3.11)

2.5 LUCY-RICHARDSON ALGORITHM

The Richardson-Lucy Algorithm⁹ works as follows. Let $\overline{f(x, y)}$ be a model estimate for an object feasibility filtering $\overline{g(x, y)}$ in this equation. Thus we rewrite (2.3.6) as

$$\overline{g(x,y)} = \int_{PY1}^{PY2} \int_{PX1}^{PX2} \overline{f(PX,PY)} H(x - PX, y - PY) d(PX) d(PY) \quad (2.5.1)$$

The Richardson-Lucy deconvolution method gets an estimate $f^{(r+1)}(PX,PY)$ at iterative step according to the algorithm

$$\overline{f^{r+1}(PX, PY)} = \overline{f^r(PX, PY)} \left[\int_{PY1}^{PY2} \int_{PX1}^{PX2} \frac{g(x, y)}{g(x, y)} H(x - PX, y - PY) d(PX) d(PY) \right] \quad (2.5.2)$$

$\overline{f^{r+1}}$ is calculated in equation (2.5.2) and the results put into equation (2.5.1). The resulting \overline{g} is calculated again and this result is put into (2.5.2). Note the sum of the g 's must be conserved in each iteration. The initial estimate f is usually a constant.

The Lucy Richardson method is made famous by its common use now in astronomy deconvolution software¹⁰.

2.6 VAN CITTERT DECONVOLUTION

Next we introduce the Van Cittert deconvolution¹¹ method. This method is based on the iterative equation

$$\overline{f^{r+1}(PX, PY)} = \overline{f^r(PX, PY)} + b \left(g(x, y) - \int_{PY1}^{PY2} \int_{PX1}^{PX2} \overline{f(PX,PY)} H(x - PX, y - PY) d(PX) d(PY) \right) \quad (2.6.1)$$

we set $g=f$ initially. b is a parameter that is set to get the best image fit.

2.7 MAP AND MAXIMUM LIKELIHOOD METHOD

We next just summarize derivations of 2 other standard superresolution methods.

For example to derive the MAP method¹² you maximize

$$p(f|g) = \prod_{x=1}^M \frac{[f(x)H(x)]^{g(x)} \exp[-H(x)f(x)]}{g(x)!} \quad (2.7.1)$$

where $p(f|g)$ is the Poisson density with parameter f and M is the number of pixels in the image plane. This is a stochastic method.

To derive the maximal likelihood we use the MAP method and assume that

$$p(f|g) = p(g|f) \quad (2.7.2)$$

which means that f is a nonrandom quantity.

Here we note that we sample in a Nyquist fashion equally over the whole image. But we note in chapter 4 that superresolution demands that we sample densely on either side of the maximum. Thus this "MAP" method can never achieve a great degree of superresolution. Also we note here that our method of doing superresolution involves only a few Airy disks in each CCD image. Thus the scale of the PSF's is much larger than the scale of the noise. Thus smoothing is a very effective way of minimizing the effect of noise with our method. Thus we do not have to rely on these type of stochastic methods to achieve superresolution.

This concludes our derivations of standard telescope superresolution methods

2.8 MICROSCOPE SUPERRESOLUTION

Abbe's theory deals with the resolution of a grating with spacing d , angular radius of stop u and wavelength λ . If at least two diffraction orders are transmitted by the stop we say that is true resolution. The smallest grating for which this is the case is:

$$d = \frac{\lambda}{2} \sin(u) \quad (2.8.1)$$

We note that a hidden assumption in this theory is that the field of view of the microscope is unbounded. Because $\sin(u)$ cannot exceed 1, the smallest detail resolvable by a microscope has a size of about half a wavelength. The way to overcome the problem of limited resolution here is to use a "d" and a λ that in fact makes $\sin(u)$ more than 1. In that case only evanescent waves¹³ will be transmitted through the boundary. These waves have an amplitude that drops off exponentially with distance from the boundary. If a submicron fiberoptic is moved over the surface then details are detected that are much smaller than a half wavelength and we then have superresolution. Note that we have apriori knowledge of the object size (given by the tip diameter) and so Abbe's analysis doesn't apply

here. It is important to note that analysis of Abbe's theory in terms of the uncertainty principle is inherently flawed. Vigoureux and Courjon¹⁴ have shown that these evanescent waves have an impulse component in the object plane larger than h/λ so that the uncertainty principle is obeyed¹⁵ and the Abbe criterion for resolution is overcome.

Dislocation of wavefront methods are also used here to obtain superresolution.

Another method of obtaining superresolution is through phase microscopy.

The important thing in the above review for our purposes is that superresolution has been achieved in microscopy in a very significant way. Thus there is the strong implication that in telescopy the same kind of advance can be achieved as we claim we have done here.

2.9 COHERENT SUPERRESOLUTION

Here we review one recent method¹⁶ that achieves coherent superresolution.

Let there be the function:

$$g(\hat{x}) = \sum_{i=1}^m H(\hat{p}_i - \hat{x}) f_i(\hat{x}) \quad (2.9.1)$$

We note that the $f_i(x)$ is really composed of discrete values which we will call k_i

Let H be some basis function, we shall call $H(X)$ which is related to the (continuous) integral limit of (2.9.1), where X is a function of both the point source center location and the x position of the data point. Thus we will rewrite equation (2.9.1) as:

$$g(\hat{x}) = \sum_{i=1}^m H_i f_i(\hat{x}). \quad (2.9.2)$$

Using the least squares formalism that we derived and using equation (2.9.2), we have:

$$\eta^2 = \sum_{i=1}^n (g_i - \hat{g}_i)^2 = \sum_{i=1}^n \left(g_i - \sum_{i=1}^m H_i f_i(\hat{x}) \right)^2 \quad (2.9.3)$$

where η is a constant. We take the partial derivative with respect to H_i . We then get:

$$\left(\sum_{i=1}^n \left(g_i - \sum_{i=1}^m H_i f_i(\hat{x}) \right) \right) f(\hat{x}) = 0 \quad (2.9.4)$$

If the functions f are orthogonal¹⁶ then we can write:

$$\sum_{j=1}^m f_i(\hat{x}_j) f_k(\hat{x}_j) = 0 \text{ for } i \neq k \quad (2.9.5)$$

Thus equation (2.9.5) becomes:

$$H_i \left(\sum_{i=1}^n f_i^2(x_j) \right) H_i = \sum_{k=1}^n g_j f(x_j) \quad (2.9.6)$$

Thus we have that:

$$H_i = \frac{\sum_{j=1}^n g_j f_i(x_j)}{N_i} \quad (2.9.7)$$

where here:

$$N_i = \sum_{j=1}^n f_i^2(x_j) \quad (2.9.8)$$

Thus the amplitude coefficients of the image are easily found if the f functions are orthogonal. If the f functions are not orthogonal we can make them so by Gram-Schmidt orthogonalization.

2.10 GRAM SCHMIDT ORTHOGONALIZATION

Here we review the most important recent work on superresolution. We seek to orthogonalize the "f" function if it is not already orthogonal. We use the technique of Gram-Schmidt orthogonalization¹⁷. Let w be the weight function. Given

$$\Phi_0 = g_0 \quad (2.10.1)$$

with no previous f 's to worry about. We normalize thus

$$f_0 = \frac{\Phi_0}{\left(\int \Phi_0^2 w dx\right)^{\frac{1}{2}}} \quad (2.10.2)$$

For $n=1$, let

$$\Phi_1 = g_1 + b_{10}f_0 \quad (2.10.3)$$

We ask that Φ_1 be orthogonal to f_0 . Thus we have that:

$$\int \Phi_1 f_0 w(dx) = \int g_1 f_0 dx + b_{10} \int f_0^2 w dx = 0 \quad (2.10.4)$$

Since f_0 is normalized to 1,

$$b_{10} = -\int g_1 f_0 dx \quad (2.10.5)$$

This gives us the value of b_{10} . We will next normalize:

$$f_1 = \frac{\Phi_1}{\sqrt{\int \Phi_1^2 w dx}} \quad (2.10.6)$$

or in general we have that:

$$f_i = \frac{\Phi_i}{\sqrt{\int \Phi_i^2 w dx}} \quad (2.10.7)$$

in general:

$$\Phi_k = g_k + b_{k0}f_0 + b_{k1}f_1 + \dots + b_{k,k-1}f_{k-1} \quad (2.10.8)$$

Note that we:

$$b_{kj} = -\int g_k f_j w dx \quad (2.10.9)$$

This is a new procedure¹⁸ for doing superresolution in signal processing. We note that the $J_1(R)/R$ functions are not orthogonal as are the cosine and sine functions in signal processing. Thus this technique is not useful in doing the superresolution we need to do here. Also coherent superresolution is out of the question since in astronomy almost all sources are strongly incoherent. Furthermore in laboratory work the arbitrary phase differences require us to apply much more difficult algorithms and experimental procedures.

SUMMARY

In this chapter we developed, from first principles, the fundamental formula for the Fraunhofer image created by a circular aperture. This will be the core of our superresolution algorithm.

Then we reviewed various other superresolution methods. We note that these methods all use frequency sampling tricks (as in equation (2.7.10)) with Fourier deconvolution. Our method of finding ridges in chapter 4 cannot use Fourier transforms but nonetheless is required to overcome the ambiguous image problem. Thus this distinguishes our method of superresolution from the other methods.

References

- 1) J. Jackson, *Classical Electrodynamics*, 2nd ed., John Wiley and Sons, 1975, p.438.
- 2) J. Jackson, *Classical Electrodynamics*, 2nd ed., John Wiley and Sons, 1975, p.217.
- 3) E. Pugh, *Principles of Electricity and Optics*, 2nd ed., Addison-Wesley, 1973, p.419.
- 4) J. Jackson, *Classical Electrodynamics*, 2nd ed., John Wiley and Sons, 1975, p.423.
- 5) G. Arfken, *Mathematical Methods For Physicists*, 3rd ed., Academic Press Inc., 1985, p.580.
- 6) H. Andrews, *Digital Image Restoration*, Prentice Hall, 1988, p. 190.
- 7) H. Andrews, *Digital Image Restoration*, Prentice Hall, 1988, p. 190.
- 8) H. Andrews, *Digital Image Restoration*, Prentice Hall, 1988, p. 194.
- 9) H. Bi, When Does The Richardson-Lucy Deconvolution Converge, *Astronomy & Astrophysics. Supplement Series*, **108**, 1994, p.409.
- 10) R. Berry, Shooting the Sky With an Electronic Camera, *Astronomy Presents Observers Guide*, 1995, p.88
- 11) B. Crilly, A Quantitative Evaluation of Various Iterative Deconvolution Algorithms, *IEEE Transactions on Instrumentation and Measurement*, **40** (3), 1991, p.558.

- 12) P. Sementelli, Analysis of the Limit of Superresolution, *J. Opt. Soc. Am. A.*, **10** (11), 1993, p. 2267.
- 13) C. Pask, Simple Optical Theory of Super-resolution, *J. Opt. Soc. Am.*, **66** (1), 1976, p.68.
- 14) J. Vigoureux, Detection of Nonradiative Fields in Light of the Heisenberg Uncertainty Principle and the Rayleigh Criterion, *Applied Optics*, **31** (16), 1992, p.3170.
- 15) V. Tychinsky, On superresolution of Phase Objects, *Optics Communication*, **74** (41), 1991, p.7.
- 16) E. Volochkov, A Method of Two-dimensional Angular Superresolution of Coherent Radiation Sources Based on Estimating of One-dimensional Dumped Harmonic Signal Parameters, *Radiotekhnika i Electronika*, **38** (7), 1993, p.1291.
- 17) X. Lianggui, Sensitivity Analysis of Superresolution Algorithm Based on Gram-Schmidt Orthogonality Method, *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, **21** (3) 1993, p. 80.
- 18) G. Arfken, *Mathematical Methods for Physicists*, Academic Press Inc., 1985, p.517.
- 19) C. Lanczos, *Applied Analysis*, Prentice Hall, 1988, p.230.

CHAPTER 3

REVIEW OF CONSIDERATIONS CONCERNING AMBIGUOUS IMAGE FORMATION AND SNR

INTRODUCTION

As stated in the introduction, the purpose of this present work is to find a way to overcome the ambiguous image problem. In this chapter we describe what that problem is. In the next chapter we show how it is overcome.

The ambiguous image problem was first proposed by G. Toraldo Di Francia in 1952. The concept of an ambiguous image is that two or more different objects can give rise to the same image. This arises because many PSF (point spread function) source fits may give the same data function. We state the Nyquist sampling theorem and relate the ambiguous image problem to that sampling theorem. As an example of the ambiguous problem let us be given an image function data set of points. Let us say that we have a two dimensional imaging problem so that we use the Jinc squared function for our point PSF. It is the case that if the ML method is used to find an image then, in general, many different test fits of the Jinc squared functions over the whole X Y plane will give a very close fit to the same image intensity function. This is illustrated in the figures. Below this is illustrated with a one dimensional case as originally proposed by Di Francia. We have figures for one dimensional ambiguous image formation and two dimensional ambiguous image formation. The main thing to be gleaned from this chapter, though, is that our 1/10 Rayleigh resolution that we claimed is *within the bounds* set by theory.

Here we will also try to explain how it can be determined when a given SNR will allow you to have few ambiguous images. Thus we will set the bounds of the allowed resolution of our system.

3.1 NYQUIST SAMPLING THEOREM

If a continuous-time signal $x_c(t)$ has a bandlimited Fourier transform $X(\Omega)$ that satisfies the condition $|X_c(\Omega)| = 0$ for $\Omega > 2\pi F_c$ then $x_c(t)$ can be uniquely reconstructed without error, from equally spaced samples $x(n) = x_c(nT)$ at sampling rate F_s , $-\infty < n < \infty$, if $F_s \geq 2F_c$ where $F_c = 1/T$.

This is the Nyquist sampling theorem¹. F_c is the critical sampling frequency. Note that in practice we should use a sampling rate that is as close to the critical rate $2F_c$ as possible. Also $X(\Omega)$ is not in general zero.

3.2 AMBIGUOUS IMAGE CREATION

Now we follow Di Francia and will choose as an example the one dimensional image² produced by a microscope. The one dimensional diffraction pattern will have the

form $\frac{\sin\left(2\pi\alpha\frac{(x-s)}{\lambda}\right)}{\pi x}$. Thus the image will be a summation of these forms given by the convolution (recall (2.1.6) equation form):

$$a_i(x) = \int_{-\infty}^{\infty} a(s) \frac{A \sin\left(2\pi\alpha\frac{(x-s)}{\lambda}\right)}{\pi(x-s)} ds \quad (3.2.1)$$

But recall the identity:

$$\frac{\sin\left(2\pi\alpha\frac{x}{\lambda}\right)}{\pi x} = \int_{-\frac{\alpha}{\lambda}}^{\frac{\alpha}{\lambda}} \exp(2\pi ifx) df \quad (3.2.2)$$

so that no frequencies exist outside the range $-\frac{\alpha}{\lambda} < f < \frac{\alpha}{\lambda}$. This is especially obvious if

(3.2.2) is to be looked on as a Fourier transform type integral. Note that as a consequence of the Nyquist theorem that the image is completely determined by giving its complex

amplitudes at a square lattice of points spaced $\frac{\lambda}{2\alpha}$ apart. It can also be shown that the²

$$N = \frac{8\alpha^2}{\lambda^2} S \quad (3.2.3)$$

degrees of freedom. "S" equals the area of the image. Equation (3.2.3) gives the maximum amount of independent data that can be found in the image.

The image has only a finite number of degrees of freedom. But the degrees of freedom of the object is infinite. Thus a given image may correspond to a whole set of different objects. In other words, our image reconstruction procedure will generate "ambiguous images". Here we will illustrate this fact with the below figs 3-1 to 3-4.

3.3 SOLUTION TO THE AMBIGUOUS IMAGE PROBLEM

Chapter 4 explains how to solve this ambiguous image (artifact) problem. The solution leads to resolution beyond the diffraction limit (superresolution) using an optical telescope. Any solution must have the effect of disentangling all the possible intensity functions whose superposition gives the detected intensity function. The key is not to test for a fit everywhere in the two dimensional plane but to restrict our testing to those regions where it is most likely to get a *correct* least squares fit. We can reduce the dimensions of the problem to just those lines (one dimension) along which the peaks of the sources are most likely to be. To further restrict our choices we also look for regions that have high convexity. In other words our deconvolution is done by dimensional reduction, and convexity (second derivative of intensity) optimization. By dimensional reduction we mean deconvolution on isophote ridges, which are one dimensional, thus we have reduced the dimensionality of the problem from two to one. The deconvolution on ridges allows us to avoid testing where there are regions of the image that will give us artifact images allowed by the degrees of freedom given by equation (3.2.3). The convexity test caused us to only test approximately equal distances on either side of a local maxima. This even further restricts our test choices and makes it harder to choose the images for which the degrees of freedom in (3.2.3) allows us. Thus the Toraldo di Francia problem of the ambiguous image posed in this chapter is solved in the next chapter.

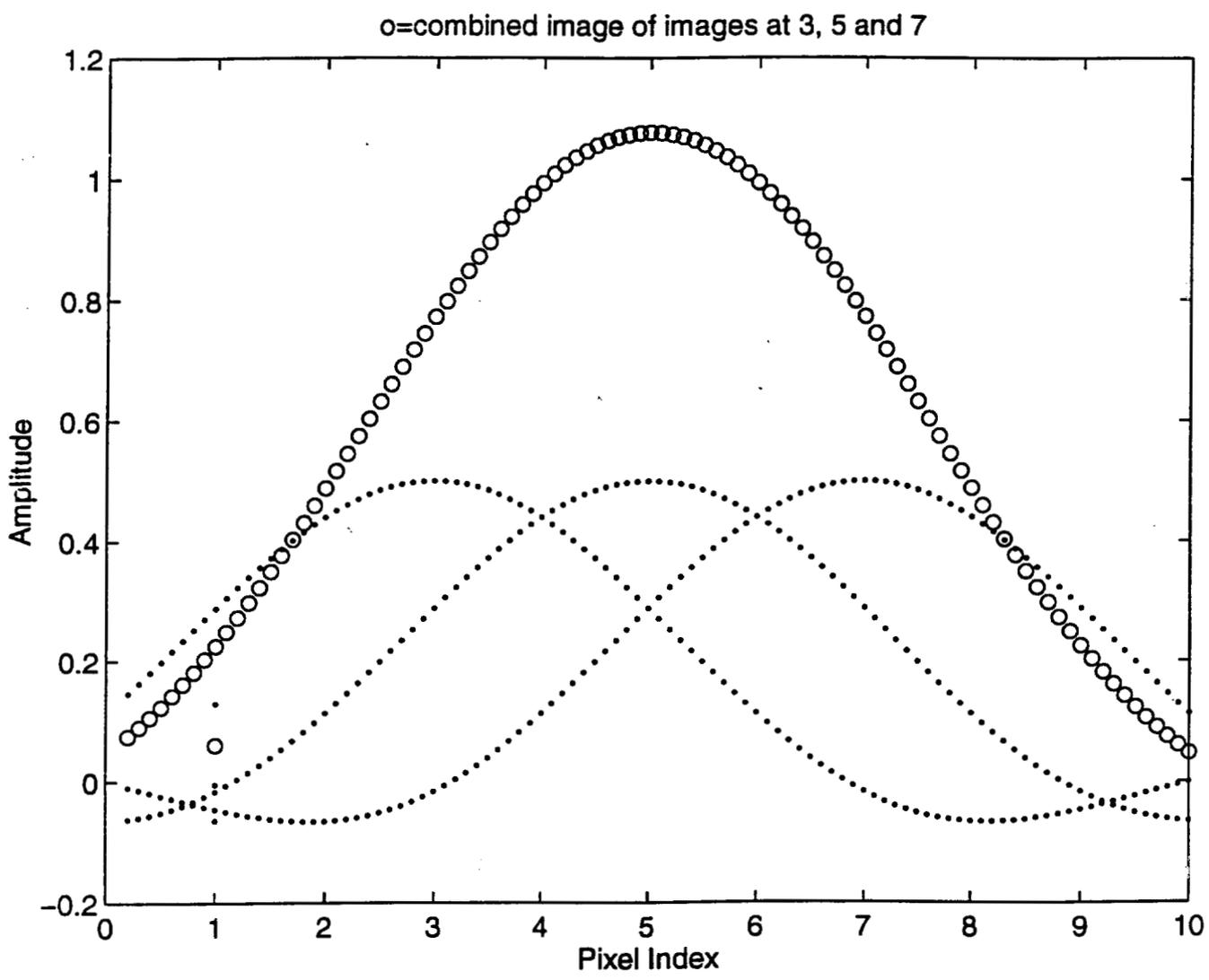


Fig. 3-1 Three sources in a row.

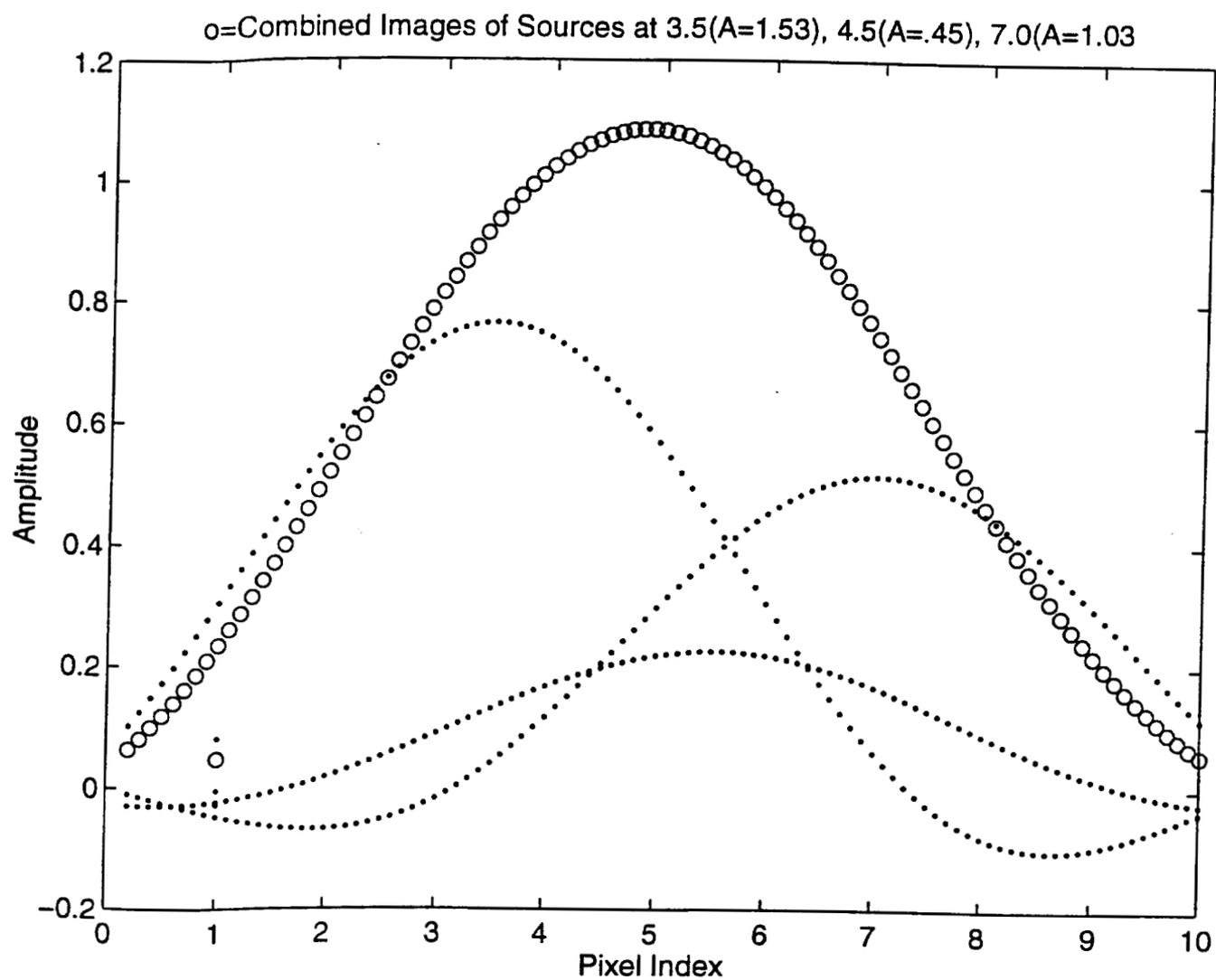


Fig. 3-2 Three sources that give the same image as Figure 3-1.

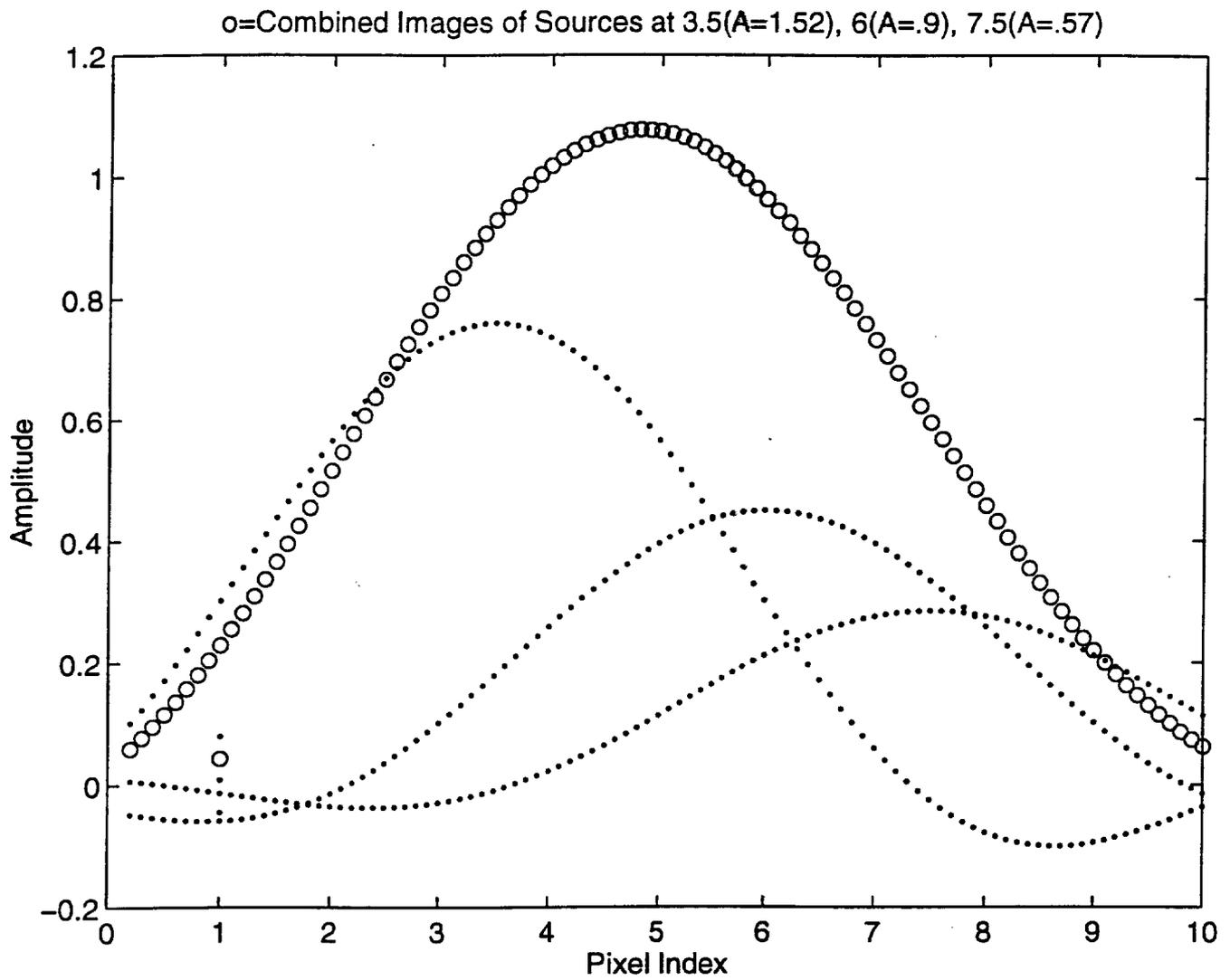


Fig. 3-3 Three sources that give the same image as figure 3-1.

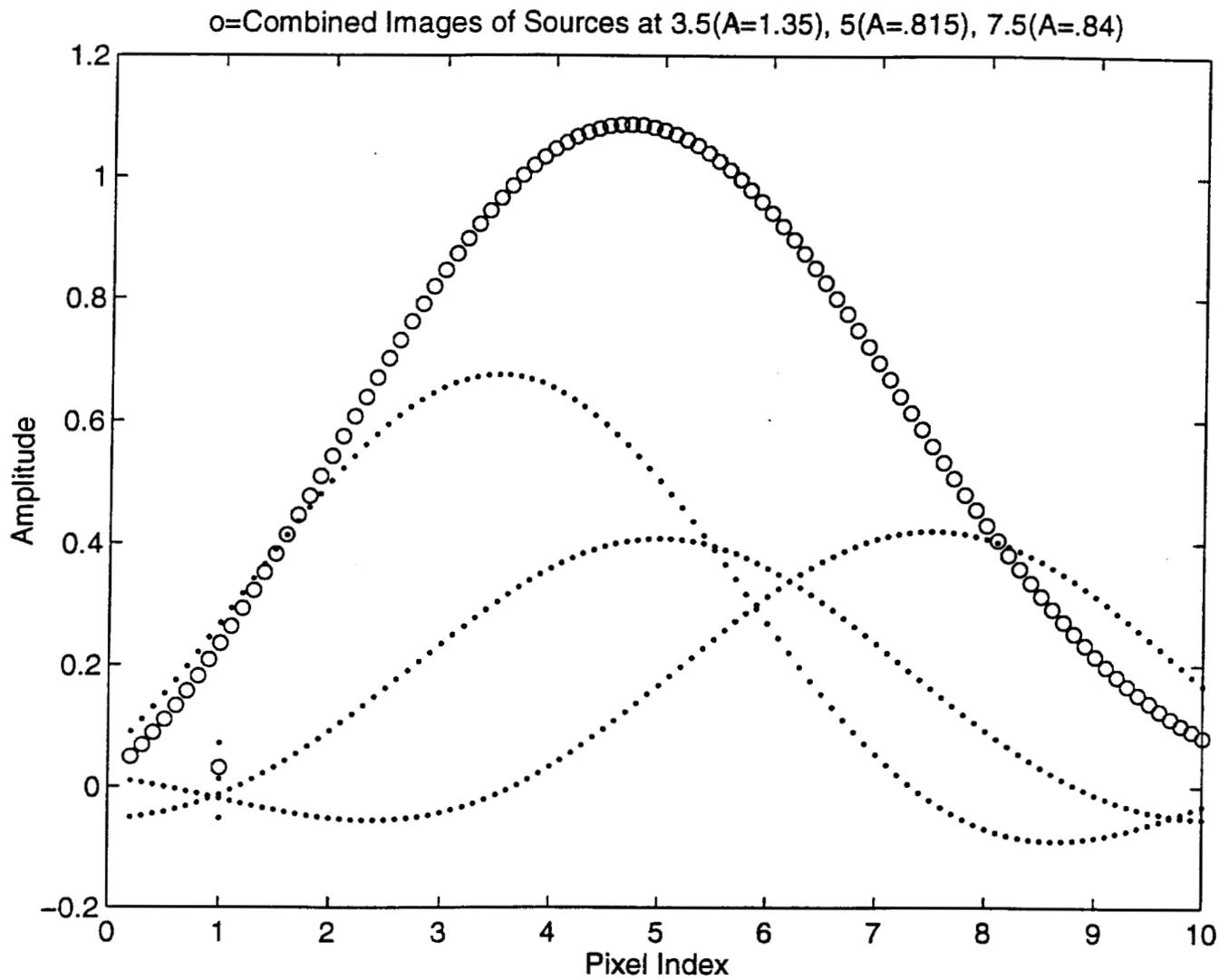


Fig. 3-4 Three sources that give the same image as figure 3-1.

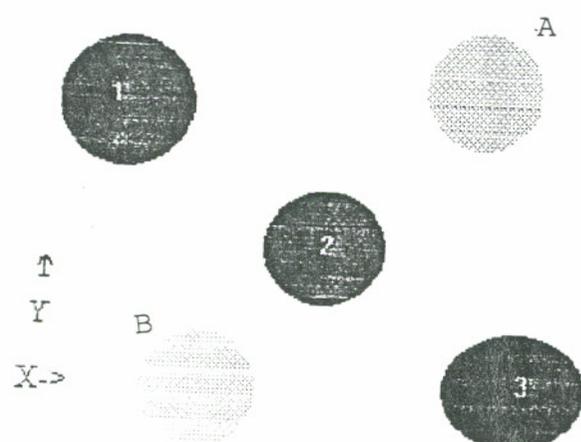


Fig. 3-5 Ambiguous image problem in two dimensions. Algorithm includes sources A and B as images even though they are not real sources.

3.4 SIGNAL TO NOISE RATIO AND THE CRITERIA FOR THE USE OF SUPERRESOLUTION

Here we review methods that determine the SNR bounds on superresolution³. Thus given a certain error tolerance, how much SNR do we need to get superresolution? To answer this let us define:

T = Tolerance in terms of spectrum magnitude of the second peak as a percent of the 0 frequency level (first peak) in the spectrum. This value the astronomer sets. Might be 5.

C_1 = the ratio of the peak of the sidelobe to the central peak for the null object which we take to be about 1/14 for a source image that occupies a small field of view.

ω_c = the cutoff frequency for the object

σ_n = noise power. For a good receiver system this is just shot noise.

χ = object extent.

ω_e = maximum extrapolation frequency. This is the largest spacial frequency that you can meaningfully extrapolate to.

Using methods discussed elsewhere³ that take into account the ambiguous image problem we can show that:

$$\omega_e - \omega_c = \frac{3}{2\chi} \left[\sin^{-1} C_1 - \sin^{-1} \left(\frac{C_1 T}{\sigma_n} \right) \right] \quad (3.3.1)$$

where the $(\omega_e - \omega_c)$ term on the right side represents the bandwidth extension. The larger it is, the better the superresolution. Note also the presence of the tolerance T function.

Also note though that if the noise σ_n is very small (long integration times) then the second sinc function is very nearly zero and the T tolerance plays no role. For example

if σ_n and $C_1 \approx \frac{1}{14}$ we get:

$$\omega_e - \omega_c = \frac{3}{2\chi} \left[\sin c^{-1} C_1 - \sin c^{-1} \left(\frac{C_1 T}{\sigma_n} \right) \right] = \frac{3}{2\chi} (6.79 - 2.22) = \frac{6.85}{\chi} \quad (3.3.2)$$

Note (given the properties of the sinc function) that for a given tolerance, if noise σ_n is very small (so the second term in (3.3.2) is replaced by 0), one should be able to get a bandwidth extension of $1.5 \cdot \frac{6.8}{\chi} = \frac{10}{\chi}$. If the cutoff frequency is of the order of $1/\chi$ which in our applications it probably is (i.e., the whole Rayleigh criterion region is in our field of view) then essentially the bandwidth extension is about $10\omega_c$ and basically you have about 10 times the Rayleigh criterion resolution. We note here that our claimed 1/10 Rayleigh resolution is within the limits of this rule. Also a four meter telescope would have the resolution usually stated for a 40 meter telescope!

Note that equation (3.3.2) is only valid if $T \leq \frac{\sigma_n}{C_1}$ and so:

$$\sigma_n < T \leq \frac{\sigma_n}{C_1} \quad (3.3.3)$$

With C_1 at about 1/4 for a small image and $T \approx 0.05$ we can specify the allowable noise in our system for meaningful superresolution. So $14 \cdot 0.05 = 0.7$ times the maximal signal power is our noise power. In our algorithm we find it convenient to put our own noise in the system using a random number generator to simulate an SNR. In a noisy system it may be possible to use the standard superresolution algorithms after we have applied our own. One limitation we must state here is that the instrument sampling rate ω_c must be at least twice the cutoff frequency ω_c for accurate extrapolation to be possible. So if the cutoff frequency is 200 then the number pixels along a line should be 200.

3.5 RESOLUTION DEPENDENCE ON SNR

For a one dimensional image we could consider the intensity

$$E = c^2 \left((d^2 - x^2) \exp\left(\frac{-x^2}{w^2}\right) \right) \quad (3.4.1)$$

c^2/E is like the signal to noise ratio since the Gaussian shape is caused by noise. Note this

SNR is not the astronomers $SNR = \sqrt{\text{electron.count.for.screen}}$

Note that this equation has two zeros at d and $-d$. "w" is the Gaussian beam width and is much larger than d . Equation (3.4.1) is the sum of Hermite functions of order 1 and 2 and a Gauss function. The zeros keep their relative location in propagating to the far field for these types of functions.

In the 1960 s it was shown⁴ that the calculation of SNR dependence of resolution can be done with prolate spheroidal functions since Sincs can be split into prolate spheroidal functions. If the total energy is kept constant then the field intensity at $x=0$ depends linearly on d^2 . Set $x=0$ in (3.4.1) and we note that:

$$\text{Resolution} \propto \sqrt{\frac{1}{SNR}} \quad (3.4.2)$$

3.6 PHASE SPACE REPRESENTATION OF SOLUTION OF AMBIGUOUS IMAGE PROBLEM

We recall that one of the new methods we use is to find points PX, PY and k that minimize the sum of k's. Let us graph our minimum of the sum of k s versus a "phase space" consisting of k and PX and PY.

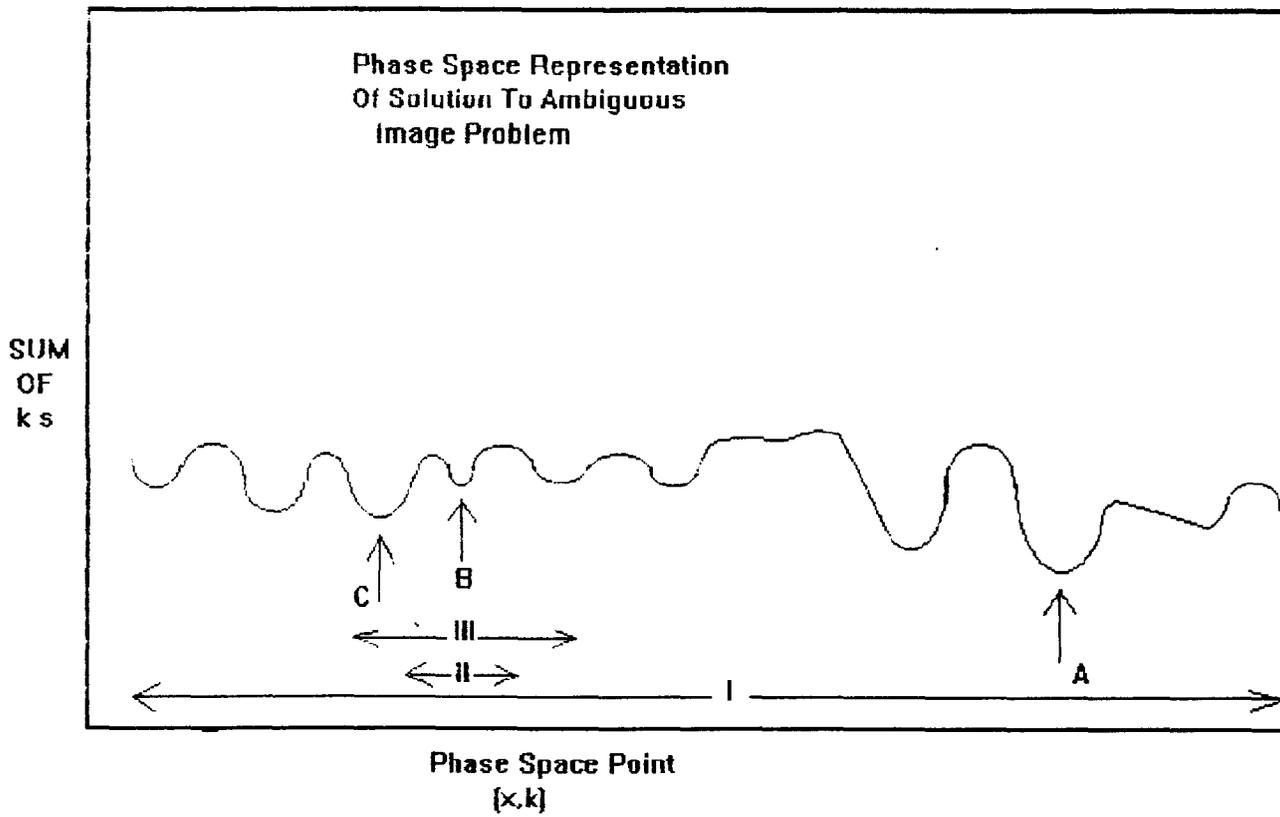


Fig. 3-6 Graph of sum of k s vs. projection of multidimensional phase space denoted by PX, PY and k.

Let point C be the correct image point. Because of the ambiguous image problem there are local minima in the sum of k s at point C (the actual image point), point B and point A, the global minima. The dimensional reduction makes it so that we need not test over all of region I, which would result in choosing the global minimum A, an ambiguous image artifact. It restricts our tests to let say region II. Because of the inaccuracy of the convexity test we introduce noise that causes us to test in the whole region III. Thus we can find the correct point C. This implies that a noiseless computer simulation actually will give us more artifacts than CCD images with significant noise! Also the restriction of the phase space test region has deep implications concerning the uncertainty principle as we see in the next section.

SUMMARY

In this chapter we reviewed various considerations concerning the ambiguous image problem and its relationship to the SNR. Thus we determined limits to superresolution.

We note that up to now this ambiguous image problem has not been solved. We illustrated a simple phase space method for solving the problem. In chapter 4 we will solve the ambiguous image problem.

References

- 1) J. Lim, *Advanced Topics in Signal Processing*, Prentice Hall, p.128.
- 2) G. Toraldo Di Francia, Resolving Power and Information, *J. Opt. Soc. of Am.*, **45** (7), 1955, p.497.
- 3) P. Sementelli, Analysis of the Limit of Superresolution, *J. Opt. Soc. of Am. A.*, **10** (11), 1993, p.2265.
- 4) D. Slepian, Prolate Spheroidal Wave Functions, Fourier Analysis and Uncertainty-I, *Bell Systems Technical Journal*, **43** (1), 1961, p.43.

CHAPTER 4

A SUPERRESOLUTION TELESCOPE THAT USES ABERRATION EFFECTS SUPPRESSION AND DECONVOLUTION BY DIMENSIONAL REDUCTION, OPTIMAL CONVEXITY AND CONVEXITY NORMALIZATION FOR IMAGE SIZE AND DARK NOISE

INTRODUCTION

This chapter provides an explanation of each of the contributions of this dissertation.

The original contribution of this dissertation is the introduction of five new methods that, when used together, overcome the ambiguous image problem in the case of narrow field of view, high magnification optical telescopes. The five new methods we have adopted for overcoming this problem are dimensional reduction, second derivative (convexity) optimization and normalization for image size and dark noise, replacing Nyquist sampling with symmetrical peak sampling, a method of using the inverse matrix technique without the usual large resulting errors and replacing the least squares with a minimum of the sum of amplitudes squared. Other more mundane innovations are methods for speedy numerical utilization of the Bessel functions. We use methods for overcoming astigmatism and spherical aberration which are not new. As part of the telescope work we find new ways of overcoming aberrations caused by secondary mirror supports. We also have ways for compensating for the effects of different levels of dark and shot noise at different exposure times and intensities.

Here we also discuss the practical knowledge needed to make our superresolution results repeatable. Care must be taken, especially on the experimental side, to ensure that various types of distortions do not occur. In the telescope work pointing errors must be minimized to ensure that aberration effects are minimized. Astigmatism must be corrected for and care must be used in sampling.

4.1 PSF Calculations

Here we explain the algorithm theory we used to achieve deconvolution by dimensional reduction. We note that PX_i, PY_i is the source position, (x,y) is the point at which we measure the intensity " $I(x,y)$ " on the screen and k is the amplitude of the source at PX, PY . The Pythagorean theorem tells us that the distance between these points is:

$$R_i = \sqrt{((x_i - PX_i)^2 + (y_i - PY_i)^2)} \quad (4.1.1)$$

and the theory of noncoherent imaging through a circular aperture says that⁶:

$$I(x, y) = \sum_{i=1}^N k_i \left(\frac{J_1(R_i)}{R_i} \right)^2 \quad (4.1.2)$$

$J_1(R)$ is the Bessel function of the first kind of order 1. N is the number of (possible) sources. Since we use the $J_1(R)/R$ PSF functions repeatedly it is necessary to have access to these functions instantaneously. So we make a table (computer file) of the values of $J_1(R)/R$. The computer algorithm must then calculate $J_1(R)/R$. One formula¹ we can use to calculate this is:

$$J_\nu(z) = \frac{\left(\frac{z}{2}\right)^\nu}{\Gamma\left(\nu + \frac{1}{2}\right)} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos(z \cos \theta) (\cos \theta)^{2\nu} d\theta \quad (4.1.3)$$

with $Re(\nu) > -\frac{1}{2}$

$\Gamma(z)$ is the gamma function, $J_\nu(z)$ is the Bessel function of the first kind of order ν .

For the case of $\nu = 1$ we note that this term has no singular points. Thus we have:

$$\frac{J_1(z)}{z} = \frac{1}{\sqrt{\pi}} \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos(z \cos \theta) (\cos \theta)^2 d\theta \quad (4.1.4)$$

This is what we used in our algorithm. To numerically integrate this equation most speedily we use Simpson's rule. See Bessel function subroutine near the end of appendix A. We note that series representations of $J_1(R)/R$ are not of much use here. The radii of

convergence for some series are not large enough or don't overlap. For alternating series that last term is very important if the series is taken very near the radii of convergence. Thus as many terms in the series are required as we used in the above numerical integration. Also some series (such as Chebyshev's) have singularity problems.

4.2 WHAT IS DECONVOLUTION?

In our title we noted that we are doing deconvolution. We need to explain what is meant by this. Thus we show here that equation (4.1.2) is a convolution integral in the limit of continuous sources. Thus in equation (4.1.2) let:

$$H(x - PX_i, y - PY_i) = \left(\frac{J_1(R_i)}{R_i} \right)^2 \quad (4.2.1)$$

Recall that the k_i 's were amplitudes of individual sources. Thus we go to the continuous source so that in (4.1.2):

$$k_i \rightarrow f(PX, PY)_i \quad (4.2.2)$$

which is our radiance distribution. Also in (4.1.2) we make the substitution:

$$g(x, y) = I(x, y) \quad (4.2.3)$$

Thus in the limit of a continuous summation (4.1.2) becomes:

$$\begin{aligned} g(x, y) &= \int_{PY1}^{PY2} \int_{PX1}^{PX2} f(PX, PY) H(x - PX, y - PY) d(PX) d(PY) \\ &= h(x, y) \bullet f(x, y) \end{aligned} \quad (4.2.4)$$

Note the subtractions in the arguments. This clearly has the form of a convolution integral. "Working backwards" to find $f(PX, PY)$ is what we call *deconvolving the convolution integral*. In chapter 2 we noted that most of the standard deconvolution algorithms use Fourier transform of equation (4.2.4). But Fourier transforms do sampling manipulations in frequency space. We show our dimensional reduction requires sampling in real space.

4.3 SOLVING MATRIX EQUATION FOR k_i 's

To be able to solve equation (4.1.2) on the computer we need to write it for individual points (x,y) in the image plane. Thus write equation (4.1.2) as a square matrix equation.

$$I_i(x, y) = \sum_{j=1}^N k_j \left(\frac{J_1(R_{ij})}{R_{ij}} \right)^2 \quad (4.3.1)$$

where i and j are the rows and columns respectively of the matrix and

$$R_{ij} = \sqrt{\left((x_i - PX_j)^2 + (y_i - PY_j)^2 \right)} \quad (4.3.2)$$

Choose a source point PX_j, PY_j at a point where we measure intensity at x_i, y_i . Thus

$$\left(\frac{J_1(R_{ij})}{R_{ij}} \right)^2 \text{ is determined in this step. } I_i(x,y) \text{ is determined since it measured intensity.}$$

Use Gaussian elimination to find k_j . Recall that Gaussian elimination is a standard elementary method of solving N linear equations for N unknowns. Then sum these k_j s. If

$$F = \sum_{j=1}^N k_j \quad (4.3.3)$$

is smaller than any previous $\sum_{j=1}^N k_j$ then record that set of k_j 's and (PX_j, PY_j) s.

See lines 220-222 in program 1 of appendix A. This method is roughly equivalent to least squares but is much faster and easier to implement. We have an upper limit of 9 for our N .

4.4 SPEEDY CALCULATION OF k_i s

We can speed up numerical calculations of (4.3.3) by thousands of times. One way to do this is to make a (Fortran) file table of $(J_1(R)/R)^2$ values which can be instantly referred to. This saves having to calculate each and every Bessel function in every calculation and in fact this is one of the main things that makes this method practical. See Bessel

function subroutine in program 1 of appendix A for the computer code that does this.

4.5 PROOF THAT (4.4.1) SHOULD BE MINIMIZED

Next mathematically prove equation (4.3.3). We sample data at as many points as there are sources. Thus (4.3.1) becomes a square matrix.

Let $\bar{I}(X_\alpha, Y_\alpha) = \sum_{j=1}^N k_j \left(\frac{J_1(R_{ij})}{R_{ij}} \right)^2$ where $\bar{I}(X_\alpha, Y_\alpha)$ is the measured data intensity at

the point (X_α, Y_α) . Write the square of the error η using equation (4.3.1) as

$$\sum_{\alpha=1}^N (I(X_\alpha, Y_\alpha) - \bar{I}(X_\alpha, Y_\alpha))^2 = \eta^2 \quad (4.5.1)$$

Then take the extrema of η^2 with respect to test choices of PX or PY in equation

(4.5.1). \bar{I} is not minimized with respect to PX and PY. Thus we have to take the derivative with respect to PX or PY or:

$$\frac{\partial}{\partial(PX)} \left(\sum_{\alpha=1}^N (I(X_\alpha, Y_\alpha) - \bar{I}(X_\alpha, Y_\alpha))^2 \right) = 0 =$$

$$\sum_{\alpha=1}^N 2 \left(\frac{\partial}{\partial(PX)} I(X_\alpha, Y_\alpha) \right) (I(X_\alpha, Y_\alpha) - \bar{I}(X_\alpha, Y_\alpha)) \quad (4.5.2)$$

and

$$\frac{\partial}{\partial(PY)} \left(\sum_{\alpha=1}^N (I(X_\alpha, Y_\alpha) - \bar{I}(X_\alpha, Y_\alpha))^2 \right) = 0 =$$

$$\left(\sum_{\alpha=1}^N 2 \left(\frac{\partial}{\partial(PY)} I(X_\alpha, Y_\alpha) \right) (I(X_\alpha, Y_\alpha) - \bar{I}(X_\alpha, Y_\alpha)) \right) = \quad (4.5.3)$$

Since $\bar{I}(X_\alpha, Y_\alpha)$ is not dependent on PX or PY $I(X_\alpha, Y_\alpha)$ is minimized with respect to PY and PX. If the X_α and Y_α are the same as the associated PX and PY, then

$\frac{J_1(R_i)^2}{R_i^2} = 1$ so that we have a square matrix of just k_i 's. So we can say that when the

function

$$\sum_{i=1}^N k_i \left(\frac{J_1(R_i)}{R_i} \right)^2 = I = \sum_{i=1}^N k_i(1) \quad (4.5.4)$$

is minimized with respect to the test PX and PY. Then

$$F = \sum_{i=1}^N k_i \quad (4.5.5)$$

is minimized and the PX s and PY s are the positions of the points of the objects. We have also determined the amplitude of the point source. Thus our method is a hybrid inverse matrix and the least squares. The inverse matrix technique was used in the inversion of the matrix to find the position and amplitude, and the least squares was used in equations (4.5.4) and (4.5.5). It may take many iterations before the function F is minimized.

4.6 ILL-CONDITIONED MATRICES AND TEST SAMPLING

If small changes in the data result in large changes in the solution then the problem is ill-conditioned. Here for example the illconditioned matrix problem can cause the inverse matrix calculations to be rendered useless. Thus this problem needs to be addresses since we use this method (our Gaussian elimination subroutine mentioned above). For example² let g be the data and f the actual source. As a review we can pose the problem as in (4.2.5). So let:

$$Hf = g \quad (4.6.1)$$

If "f" is changed by a small amount and the corresponding change in g is large the problem is ill-conditioned. If "f" is changed by a small amount and the corresponding g is then changed by a small amount we say the problem is not ill-conditioned.

We can define and illustrate the condition number by perturbing equation (4.6.2). Have δg be a perturbation of g and δf be a perturbation of f so that

$$H(f + \delta f) = g + \delta g \quad (4.6.2)$$

$$Hf + H\delta f = g + \delta g \quad (4.6.3)$$

Subtract equation (4.6.2) from equation (4.6.3) and multiply by H^{-1} to get:

$$\delta f = H^{-1}\delta g \quad (4.6.4)$$

We next take the norm of equation (4.6.1) and equation (4.6.4) and get the Cauchy-Schwarz inequalities:

$$\|H\| \|f\| \geq \|g\| \quad (4.6.5)$$

and

$$\|\delta f\| \leq \|H^{-1}\| \|\delta g\| \quad (4.6.6)$$

Then take the ratio of inequalities in (4.6.6) and (4.6.5) and we find that:

$$\frac{\|\delta f\|}{\|f\|} \leq \|H\| \|H^{-1}\| \frac{\|\delta g\|}{\|g\|} \quad (4.6.7)$$

The condition number $\kappa(H)$ of the matrix equation (4.6.7) is given by:

$$\kappa(H) \equiv \|H\| \|H^{-1}\| \quad (4.6.8)$$

$\kappa(H)$ clearly relates the relative uncertainty of the measured value g to the calculated value “ f ”. Note that $\kappa(H)$ is large if there are large relative differences in the sizes of the values of g in the PDF (probability distribution function).

The main observation made here is that the convexity optimization method used below usually picks both test and data points near the tops of the PDF so that $\kappa(H)$ is small and so we do not have an ill-conditioned matrix problem here. Thus one can use the inverse matrix technique described in chapter 2 without serious problems. Also for regionally normalized PDF's this method surprisingly doesn't cause any problems with discarding dim images.

4.7 DEALING WITH THE AMBIGUOUS IMAGE PROBLEM

Recall that ambiguous images are caused by combinations of functional PSF fits to the image intensity versus position function that give a least squares fit as good as the actual image function. If we could somehow restrict our choices of sampling to avoid the ones that give us ambiguous images then we would not get any ambiguous images. For example it is to be noted that PSF's have isophote ridges connecting them in this intensity space.

Also we note that the second space derivative of the intensity is largest at the top of PSF's. See figure 4-1. Thus if we could sample only places on isophote ridges where there were the largest second derivatives then we would be avoiding a lot of the ambiguous images.

4.8 SAMPLING OF TEST DATA FROM ISOPHOTE RIDGES

Recall that ambiguous images are caused by combinations of functional PSF fits to the image intensity versus position function that give a least squares fit as good as the

actual image function. In our case they give a minima in $\sum_{i=1}^N k_i$ that is as good as the cor-

rect images. It is also to be noted that these PSF's have isophote ridges connecting them in this intensity space. We illustrate this in figures 4-1 to 4-4 below. Also we note that the second space derivative of the intensity is largest at the top of PSF's. See figure 1.5.

Claims for using dimensional reduction in signal processing have been made by others⁷.

We first constrain x and y choices to constant R around a local maxima in intensity. Thus:

$R = \sqrt{x^2 + y^2}$ This is done in lines 27-28 of program1 in appendix A.

We increment in angle until we find a maximum of intensity (I) along circle of radius R

$I(s) = \text{Maximum}$ (4.8.1)

We then pick x, y and I(x,y) when I(s) is maximized. We then repeat at the next incremented R. The set of points x,y thus found will form a ridge.

Also we take intensity data that has been smoothed two times. This smoothing is

done in lines 10 through 11 of program 1 from appendix A. That this is an optimal method is seen only by trial and error. If there is no smoothing isolated noise jumps in intensity will be taken to be ridges. Too much smoothing and evidence of images within a small region will be erased.

If the intensity is larger than the intensity at the next angle and the previous angle we record the X,Y coordinate of that point. The points for which these maxima occur form isophote ridge lines. Next list the maxima that occur consequitively along a ridge. Then find the second spatial derivative of the intensity along a ridge.

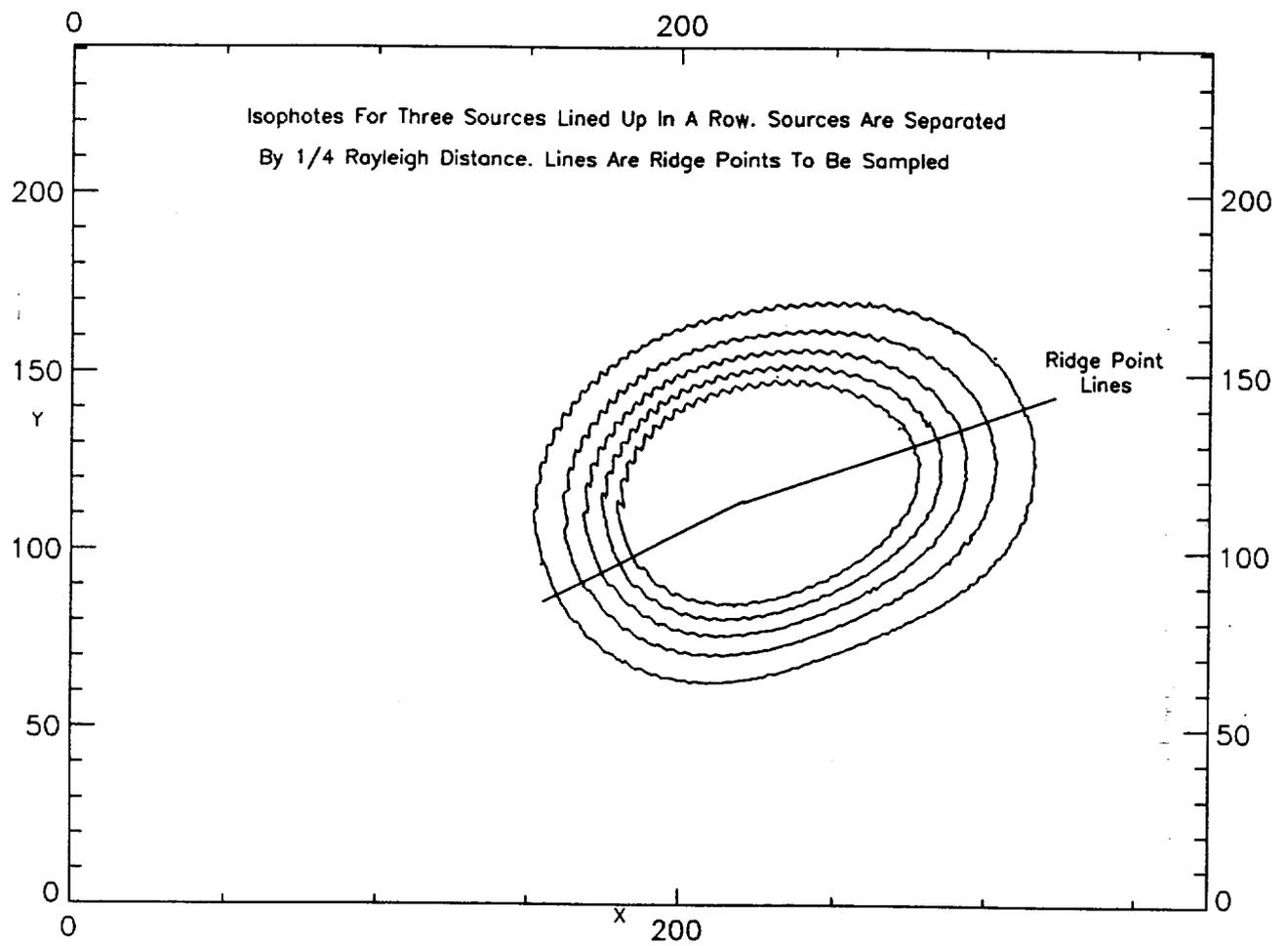


Fig. 4-1 Ridge line for three point sources line up in a row.

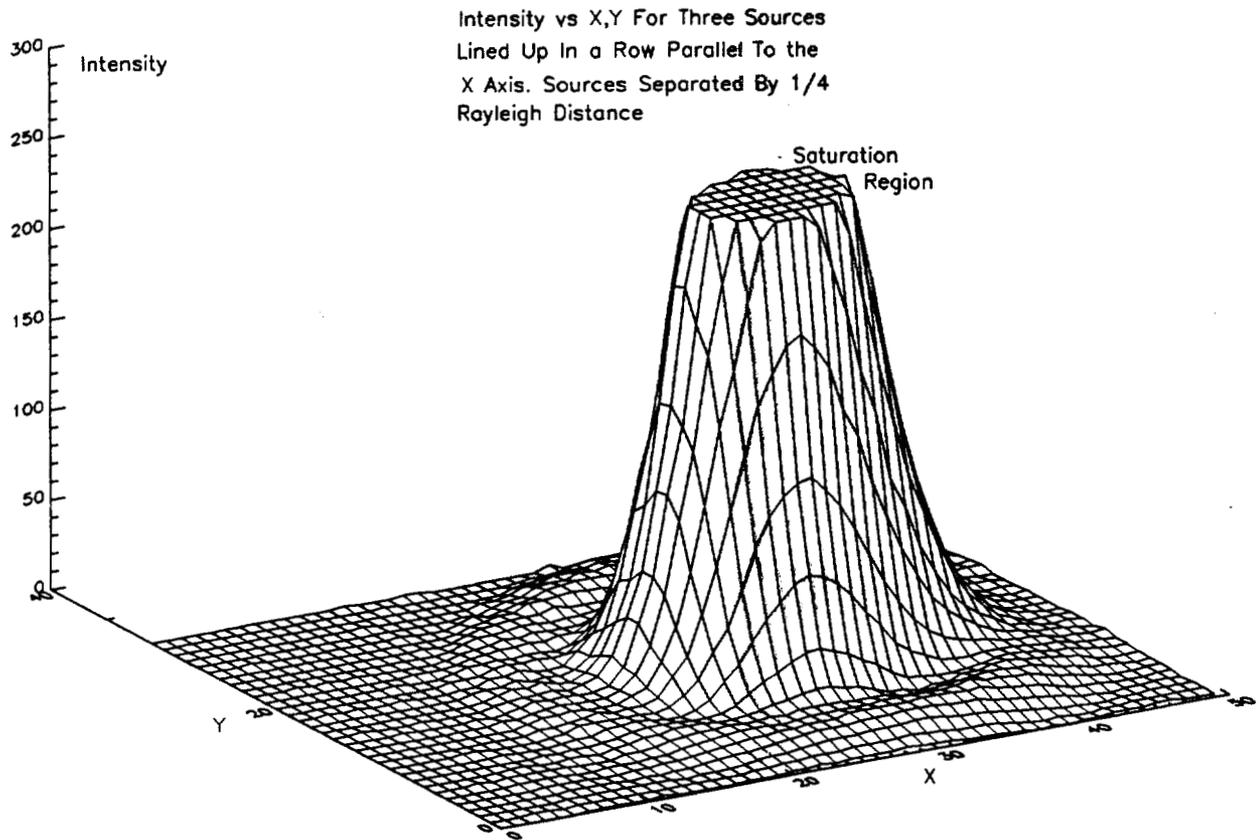


Fig. 4-2 Intensity for three point sources lined up in a row.

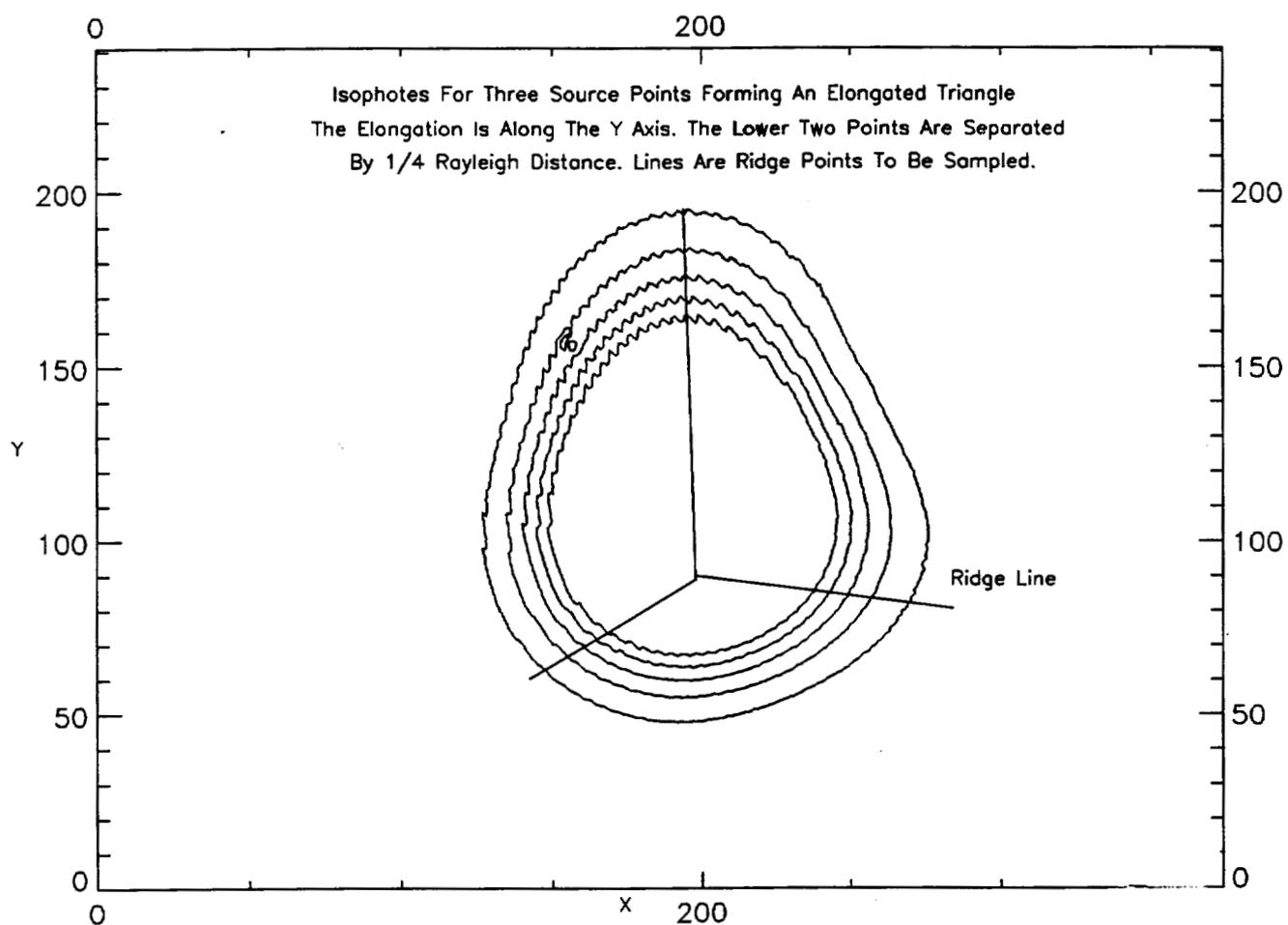


Fig 4-3 Ridge lines for three point sources forming a triangle.

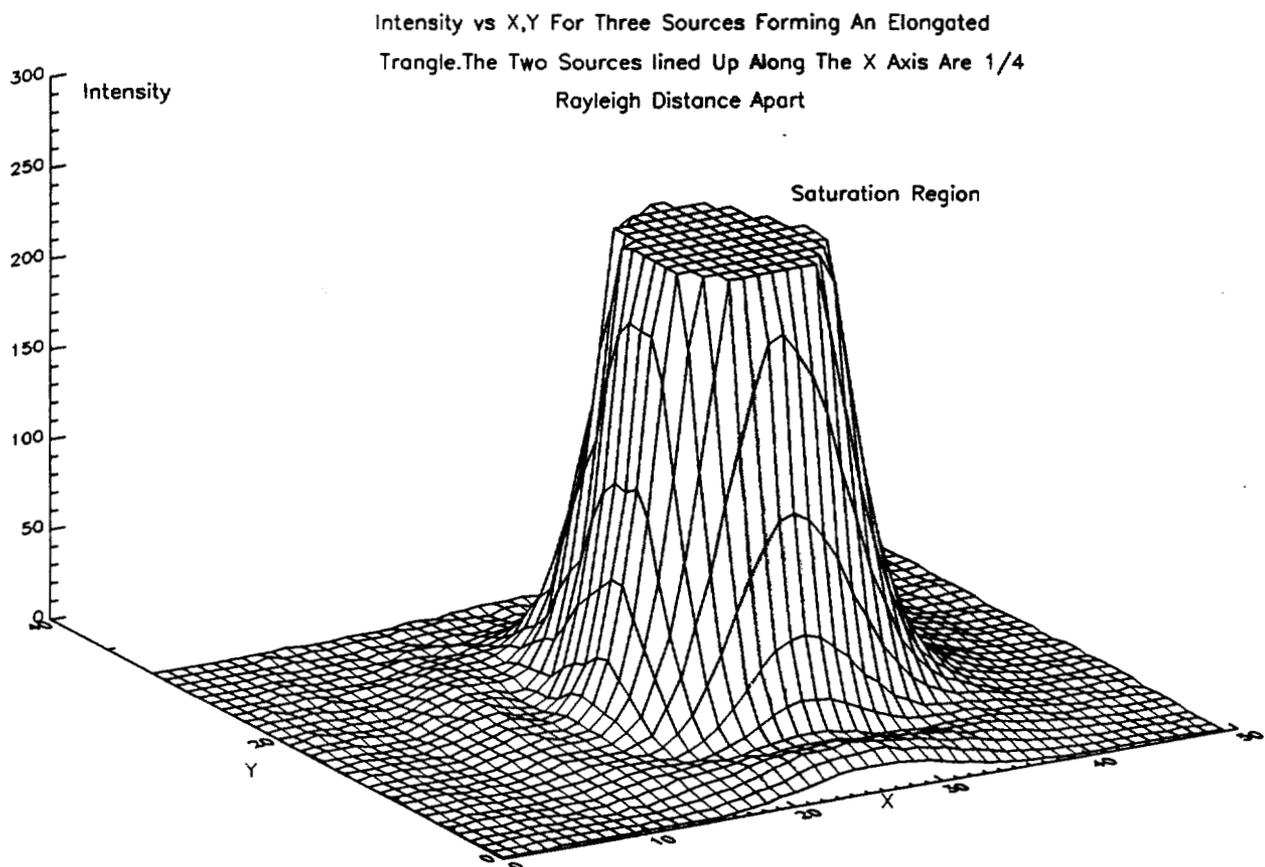


Fig.4-4 Intensity vs X,Y for three point sources forming a triangle.

4.9 SAMPLING TEST DATA FOR OPTIMAL SECOND DERIVATIVES OR OPTIMAL CONVEXITY

Let $I(x,y)$ be the intensity at the point (x,y) and "s" to be a displacement along an isophote ridge. The peaks of PSF's are obviously regions of highest second derivative. We plotted the magnitude of the second derivatives in figure 4-1 below. Note that the second derivatives are about 2X larger at the middle than at the edges. Then we sample regions of higher second derivatives in which:

$$\frac{\partial^2}{\partial s^2} I(x, y) = \text{LARGEST} \equiv \text{PSEC} \quad (4.8.1)$$

where "s" is along an isophote ridge. PSEC stands for second derivative parameter. We pick only the points where the intensity has largest second derivative. See lines 34-38 of program 1 in appendix A. This method of sampling seems best for images that are closer than about a third a Rayleigh separation. The strong negative value for the second derivative beyond that distance can actually cause our deconvolution method to fail. This further restricts the choices of x,y .

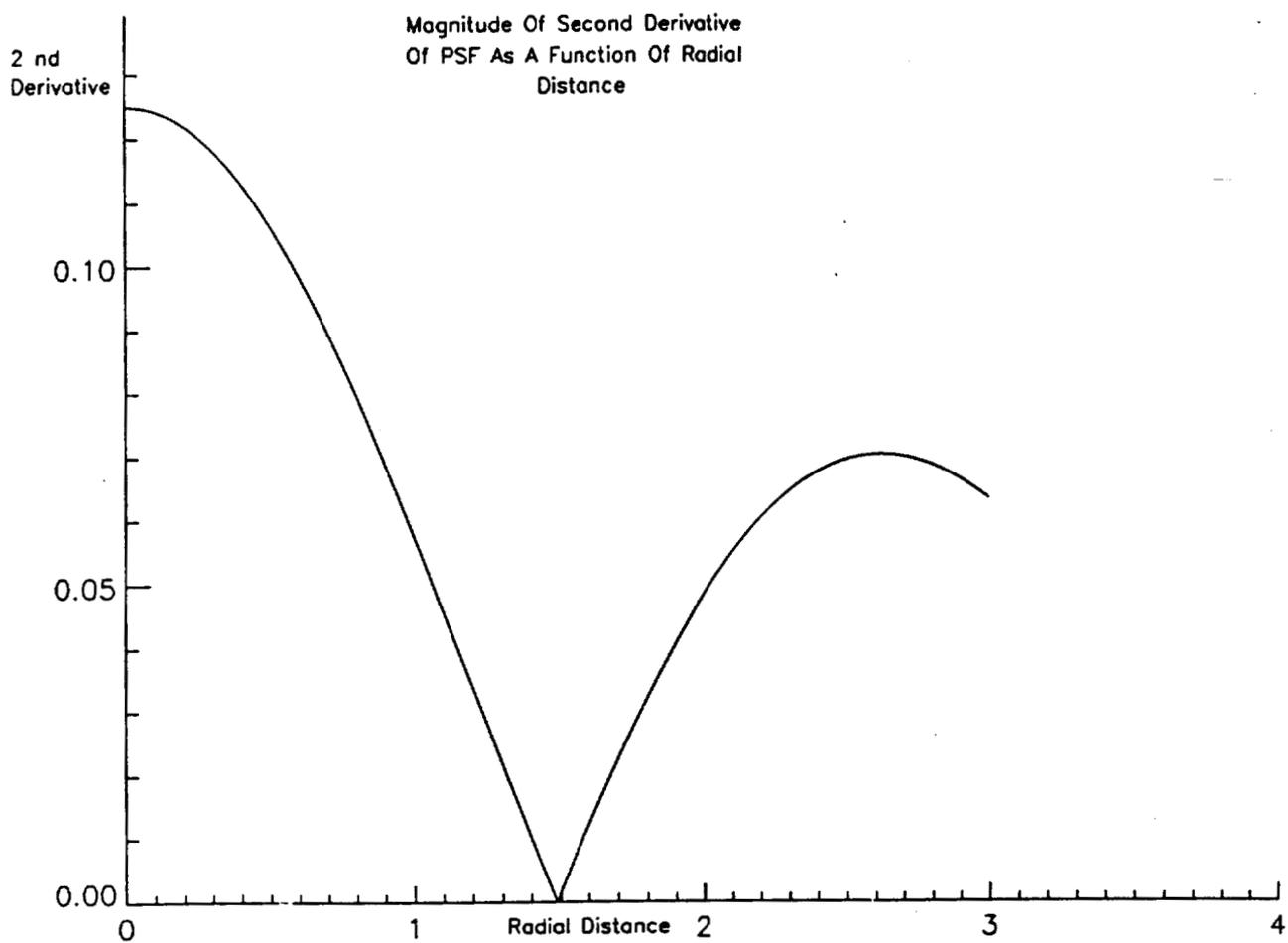


Fig. 4-5 Second derivative of intensity as a function of radial distance from the center of the PSF. The center of the PSF is at the origin.

4.10 PSF CAUSED BY ANNULAR APERTURES

Most types of reflecting telescopes have secondary mirrors located in the path of light that is incident on the primary mirror. We call these “annular” apertures. Here we outline a way to incorporate annular aperture³ PSF’s into our algorithm. The amplitude of the PSF given by one aperture for a point source is:

$$A(x, y) = k \frac{J_1(R)}{R} \quad (4.10.1)$$

where $R = \sqrt{((x - PX)^2 + (y - PY)^2)}$ in which (PX, PY) is the source location and (x,y) is an arbitrary location on the screen. Subtract the contribution of the annulus to the amplitude at the point x,y. Let ϵ = the ratio of the secondary mirror diameter to the primary mirror diameter. Thus we have for the amplitude:

$$A(x, y) = k \frac{J_1(R)}{R} - \epsilon^2 \left(k \frac{J_1(R)}{R} \right) \quad (4.10.2)$$

In our computer code for the PSF we replace $J_1(R)/R$ in equation (2.2.8) and (4.10.1) with (4.10.2). In this way we can incorporate optical systems with large secondary mirrors into the algorithm. This is used since the secondary mirror blocks paraxial light. If this method is ever to be used on a Hubble or NGST space telescope this apodization correction program will have to be used since these telescopes use large secondary mirrors.

4.11 CLASSIFICATION OF ABERRATIONS

Let $z=W(x,y)$ describe a two dimensional surface of height z at point x,y which here is a wavefront height⁴. Such a wavefront surface can be expressed in general form as:

$$W(x,y) = \sum_{n=0}^k \sum_{m=0}^n c_{mn} x^m y^{n-m} \quad (4.11.1)$$

Here k is the degree of the polynomial. We can express this wavefront in more general form by use of Kingslake formalism⁴:

$$W(x,y) = A(x^2 + y^2)^2 + By(x^2 + y^2) + C(x^2 + 3y^2) + D(x^2 + y^2) + Ey + Fx \quad (4.11.2)$$

where A=Spherical aberration coefficient,

B=coma coefficient,

C=astigmatism coefficient,

D=defocusing coefficient,

E=tilt about the x axis and

F=tilt about the y axis

Our choice of the PSF clearly only deals with spherical, defocus⁵ and astigmatic aberrations. Thus coma is not accounted for here. Only the A and D coefficients are associated with spherically symmetrical aberrations.

4.12 ASTIGMATISM CORRECTION

The surface- to-volume ratio of smaller turbulent cells is larger than that for the larger turbulent cells. The smaller turbulent cells lose their energy much more easily than the larger ones. Thus the energy carried by turbulence is mostly in the larger turbulent cells so that most aberrations are caused by the larger turbulent cells. These are displacement (tilt), and astigmatic aberrations. The spherical aberrations apparently do not cause many problems here in practice so we will discuss astigmatic aberrations. To take into account astigmatic aberrations we rotate the image (by angle *ast*) using standard vector rotation algorithms. We assume that we are doing this rotation about the highest intensity. We stretch the x coordinate using some magnitude of stretch function. For our CCD stretch aberration: stretch=.8 and *ast*=10 degrees. Therefore:

$$X_1 = (X \cos ast - Y \sin ast) stretch \quad (4.12.1)$$

$$Y_1 = X \sin(ast) + Y \cos(ast) \quad (4.12.2)$$

Then rotate back.

$$X = X_1 \cos(-ast) - Y_1 \sin(-ast) \quad (4.12.3)$$

$$Y = X_1 \sin(-ast) + Y_1 \cos(-ast) \quad (4.12.4)$$

Proceed as above to test to see if we have a minimum of $\sum abs(k)$ in (4.3.3). Note that if $\min \sum abs(k)$ isn't satisfied, then in almost all cases the resulting amplitudes

are extremely large and easily ignored. Thus we should be able to find the correct k 's and PX 's and PY 's in (4.3.1).

4.13 DARK NOISE AND SHOT NOISE

Dark noise in a CCD is caused by some electrons having high random kinetic energies due to thermal motion and being recorded on the CCD as if photons had given them that energy. You can assume that the thermal velocity v in the dark noise obeys a Maxwell-Boltzman distribution

$$f(v) = \frac{N h^3}{V (2\pi m k T)^{\frac{3}{2}}} \exp\left(\frac{-\frac{1}{2} m v^2}{k T}\right) \quad (4.13.1)$$

Here $f(v)$ is the velocity distribution in momentum space, " v " is the velocity, N is particle number, V is volume, T is absolute temperature, k is Boltzman's constant and m is mass. The longer the time of exposure the more electrons interact in this manner and give rise to false photon signals. Thus dark noise is clearly larger for longer time exposures. Also for small intensities the dark noise contribution is *relatively* large. These last two observations lead to the convexity problem we will discuss in the next section. Dark noise presents the biggest "noise" challenge to superresolution as we will see.

Shot noise is a result of quantum mechanical Fermi-Dirac distribution of the Fermi sea in the metal component of the CCD detector and is given by:

$$f(E) = \frac{2}{h^3} \left(\frac{1}{1 + B \exp\left(\frac{E}{kT}\right)} \right) \quad (4.13.2)$$

where E is the energy, B is a constant and is a function of the Fermi energy, $f(E)$ is the phase space density. This is a Fermi-Dirac distribution over whatever intensity is given. Thus shot noise is larger for higher intensities. We can minimize the effects of shot noise by normalizing the highest intensities to some fixed amount. That normalizes the shot noise component as well.

4.14 CONVEXITY ABERRATION CAUSED BY DARK NOISE AND DIFFERENT IMAGE SIZES

In our algorithm we normalize all of our maximum image intensities. This minimizes shot noise problems due to our superresolution. Also the convexity we are optimizing is:

$$\frac{\partial^2}{\partial s^2} I_i(x, y) = \frac{\partial^2}{\partial s^2} I(x, y) + \frac{\partial^2}{\partial s^2} I_B(x, y) \quad (4.14.1)$$

where $I(x, y)$ is the raw image intensity and $I_B(x, y)$ is the background intensity. Thus

$\frac{\partial^2}{\partial s^2} I(x, y)$ is the image convexity we need. But our convexity check thinks that

$\frac{\partial^2}{\partial s^2} I_i(x, y)$ is what we are trying to find. If the background convexity $\frac{\partial^2}{\partial s^2} I_B(x, y)$ for

some reason such as a background object or noise is large, then the $\frac{\partial^2}{\partial s^2} I(x, y)$ will

appear to be larger than it really is and spurious artifact images will appear. Dark noise and a small background image can cause this to occur. For images that are spread out over

most of the CCD the average convexity $\frac{\partial^2}{\partial s^2} I(x, y)$ of the intensity surfaces is clearly less

than it is for more localized surfaces. The test (4.14.1) is a test for convexity differences

$\frac{\partial^2}{\partial s^2} I(x, y)$. For localized surfaces the convexity due to a source plus the convexity due to

a nearby (perhaps smaller size) source (in 4.14.1) is larger than for a more spread out surface. Thus the convexity PSEC (see 4.8.1) setting in the program must be set smaller for more spread out images.

For longer exposures the contribution of dark noise to $\frac{\partial^2}{\partial s^2} I_B(x, y)$ "noise spikes"

becomes more important as discussed in section 15. These noise spikes give false second

derivative peaks since they make $\frac{\partial^2}{\partial s^2} I(x, y)$ appear larger than it is. The PSEC setting

also has to be a strong function of the duration of the exposure. This can be guaranteed by conditional statements in the program. Apparently the contribution of dark noise has a somewhat weaker dependence on image intensity. Thus one can also set PSEC according to the intensity. But for very short exposures (as in telescope work 1/100 sec. exposures through air turbulence) the dependence of convexity on intensity is negligible since the dark noise contribution is negligible. All of the above observations have been verified by experiments.

Thus calibration of the images is a combination of the stretch calibration and the PSEC calibration. The "stretch" calibration time can be shortened by mapping the width of the Airy disk (point source) at 45 degree increments. The PSEC, as the above illustrates, takes a lot more effort.

4.15 PARAMETER SPACE

We can plot the parameters **PSEC vs. stretch vs. useful airy disk diameter** in three dimensions. In the calibration stage we manipulate these three parameters until we find the location of the region for which a given calibration image takes on the appearance it is known to have. You do this for as many nominal calibration images as you can. Then find the region for which all of these regions in parameter space intersect and we set all of the above parameters for the centroid of this region. This is surprisingly easy for very high SNR, low aberration images. In the case of a low SNR and /or high aberration there may be no intersection region at all! In this case we probably should search for the source of the CCD aberration and/or noise, fix it, and try again. It is noted that in the case of laboratory pinhole aperture work that there are certain types of images that will never have an intersecting region in parameter space. On the ST-6 SBIG CCD it is noted that images that don't have thin film diffraction patterns overlaying them (and that also have about a 7 pixel thick hazy region around them) consistently *allowed* for a large parameter space intersection region. Thus they were good superresolution candidates. Images that appeared "milky" on the edges and had thin film diffraction patterns on them and that abruptly

terminated never allowed superresolution to work on them. All of the above statements have been verified experimentally.

Note also that only after the dimensional reduction, convexity optimization and convexity normalization for dark noise and astigmatic corrections does the program apply the inverse matrix algorithm of section 1. Thus the resulting restricted values of x,y are the choices of PX, PY that we will put in equation (4.1.1) (and therefore (4.1.2)) of section 1.

For example proceed as in section 1 to test to see if there is a minimum of $\sum k$ in (4.3.1). We note that if $\min \sum k$ isn't satisfied then in almost all cases the resulting amplitudes are extremely large and easily ignored. Thus we should be able to find the correct k s and PX s and PY s in (4.3.1). Note that in the high SNR cases with no air turbulence evident we need only one setting of AST and stretch and do not need to follow the above procedure of resetting the AST and stretch.

4.16 SAMPLING AND LARGE IMAGES

Doing superresolution on more than just a small diffraction ringed smudge is possible. We simply need to include at least four Rayleigh distances across the image. The only part of the result used is the inner 2 Rayleigh distances. Image displacement caused by air turbulence would be intolerable here so this would be a useful space telescope application only. We use our method of amplitude addition of overlapping sampled images to find the approximate location of an assumed point sources in each 2 Rayleigh distance region. Do this only for one individual PSF island at a time. We finally get a map of the location of point objects in the formerly hazy region. Also we sample at distances one Rayleigh distance in from the edges. Thus we make a mosaic of these inner regions to form a picture of our nebula. Nebulosities smaller than a Rayleigh distance (but much larger than a point) create a "buck shot" effect. Such a nebulosity appears composed of many dots. Care must be made to determine the actual resolution of the telescope (given the SNR and aperture) so that these "dots" have about the same size allowed by this resolution and thus are not really dots. This could also be useful for "resolving" details of the surface of a star.

4.17 EXPERIMENTAL DATA

We used an SBIG ST-6 CCD manufactured by Santa Barbara Instrument Company. The CCD receiver window has dimensions of 8 mm by 6 mm. It was thermoelectrically cooled. We used pinhole apertures in the 60 SNR experiments and in the rest we used 1.5 inch reflecting and refracting telescopes with a 400 foot line of sight. The reflecting telescope was off axis and had a 100X microscope attached to it so that the CCD window only covered an area of about three Airy disks. With this narrow field of view the scale of the noise was much smaller than the scale of the PSF's so that smoothing was very effective. But with the narrow field image movements (displacement) due to atmospheric turbulence appeared much larger than without the larger field of view. For example jerky image movements (about 1/3 second duration) with a standard deviation of about 1 mm were evident due to air turbulence. Yet we attained a superresolution of about 1/6 Rayleigh distance in this "accentuated" turbulence.

The results below in fig.4-5 to fig. 4-18 illustrate the outcome of one set of experiments. Included here is are actual photomicrographs of the pinholes, the raw image output from the CCD camera and the deconvoluted images. Also we have included a scale, in Rayleigh lengths (given the aperture), of our image sizes. Fig. 4-17 is the raw image of the pinholes seen in the photomicrograph in Fig. 4-5. The deconvoluted image is seen in Fig.4-18. Note the objects are about 1/10 of a Rayleigh distance apart. This is at least a factor of 5 better than the best results that have been achieved so far⁷. Note that Fig.4-21 is a deconvoluted image of two objects separated by 1/10 Rayleigh. Also in Fig. 4-17 the 4 source configuration is maintained even for a 1/4 Rayleigh separation.

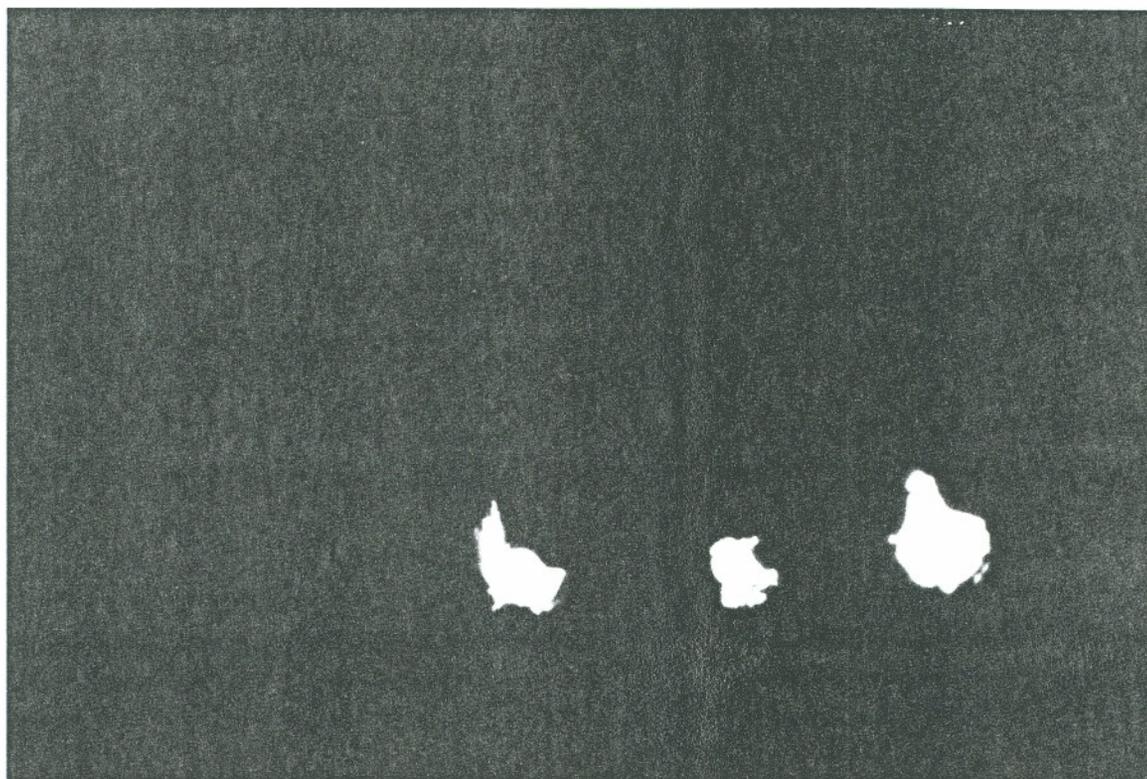
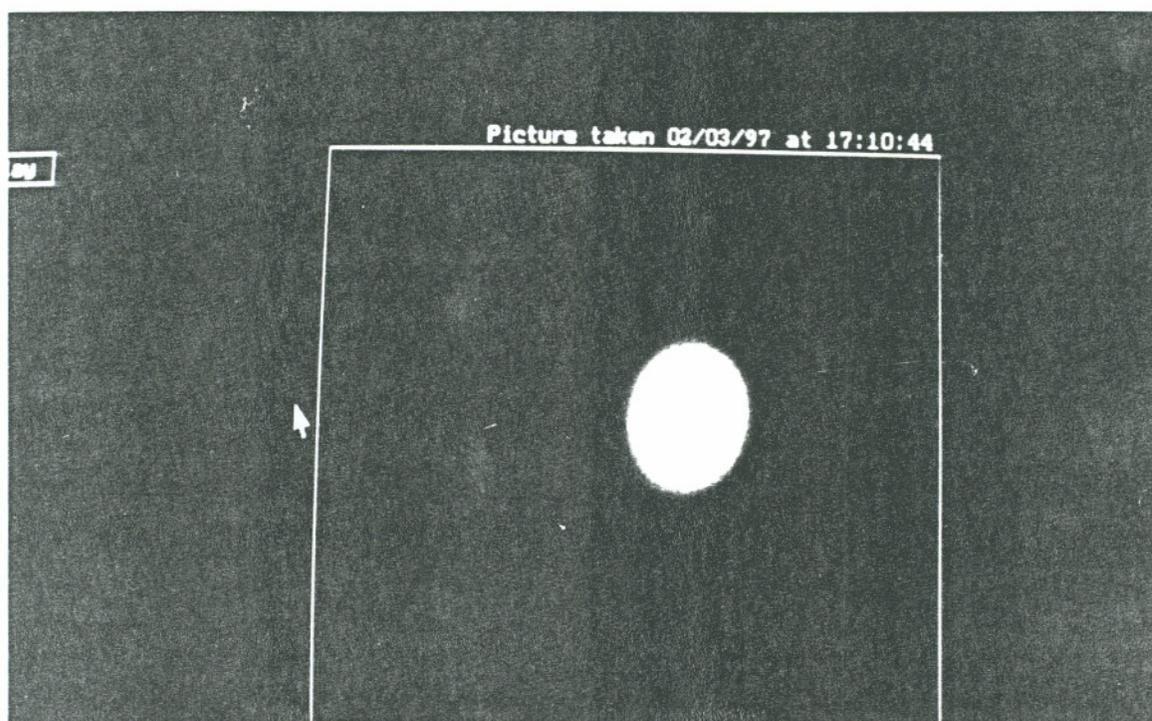


Fig. 4-6 Photo (X100) of three pinhole point sources.



← 1. Rayleigh →
Distance

Fig. 4-7 Raw image from three point sources, $1/4$ Rayleigh distance apart, $SNR \approx 60$.
See figure 4-6. No air turbulence evident.

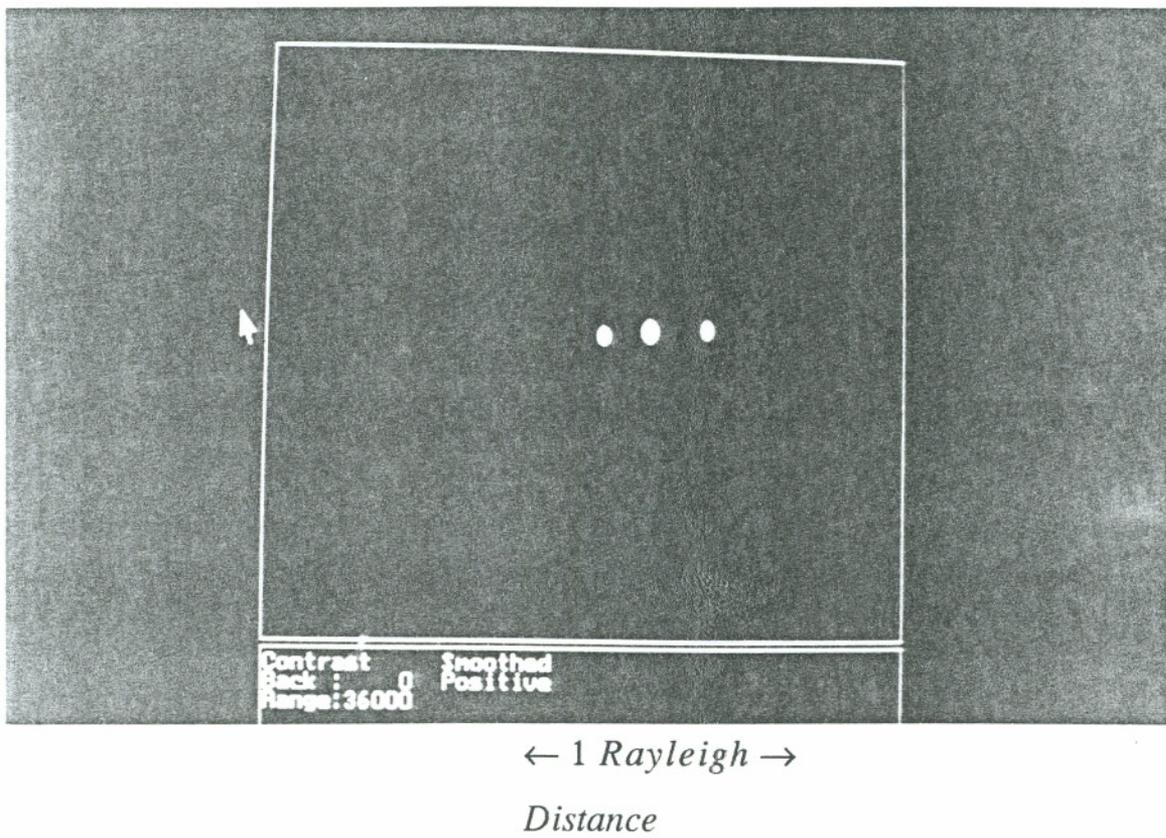


Fig. 4-8 Restoration of figure 4-7 achieved with deconvolution. See figure 4-6.

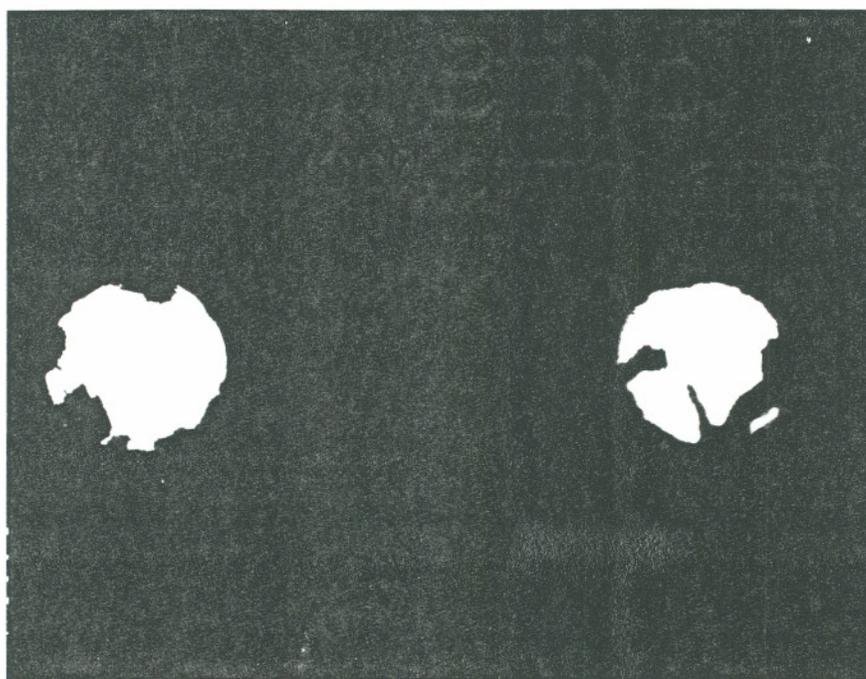
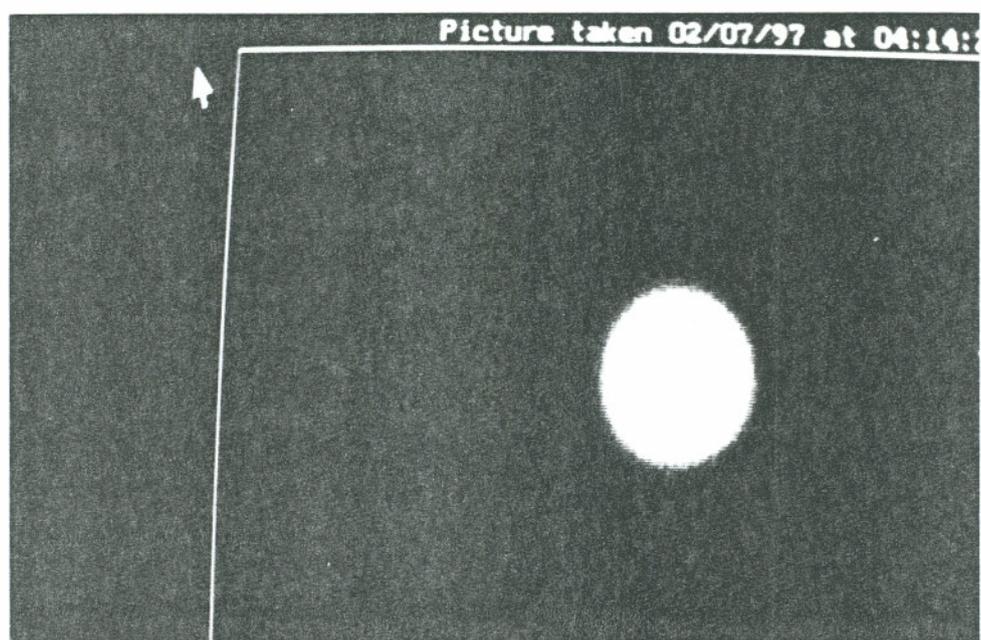
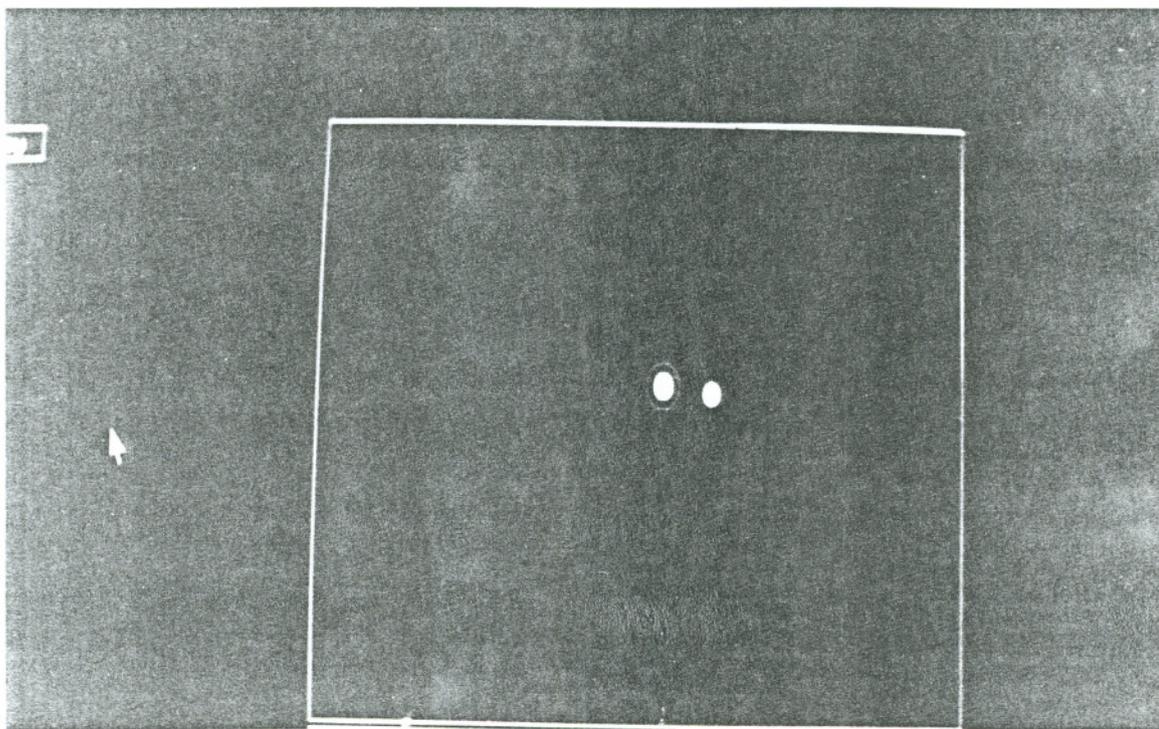


Fig. 4-9 Photo (X100) of two pinhole point sources.



← 1 Rayleigh →
Distance

Fig. 4-10 Raw image of two point sources placed $1/4$ Rayleigh distance apart, $SNR \approx 60$. See figure 4-9. No air turbulence evident.



← 1 Rayleigh →
Distance

Fig. 4-11 Restoration of figure 4-10 achieved with deconvolution. See figure 4-9.

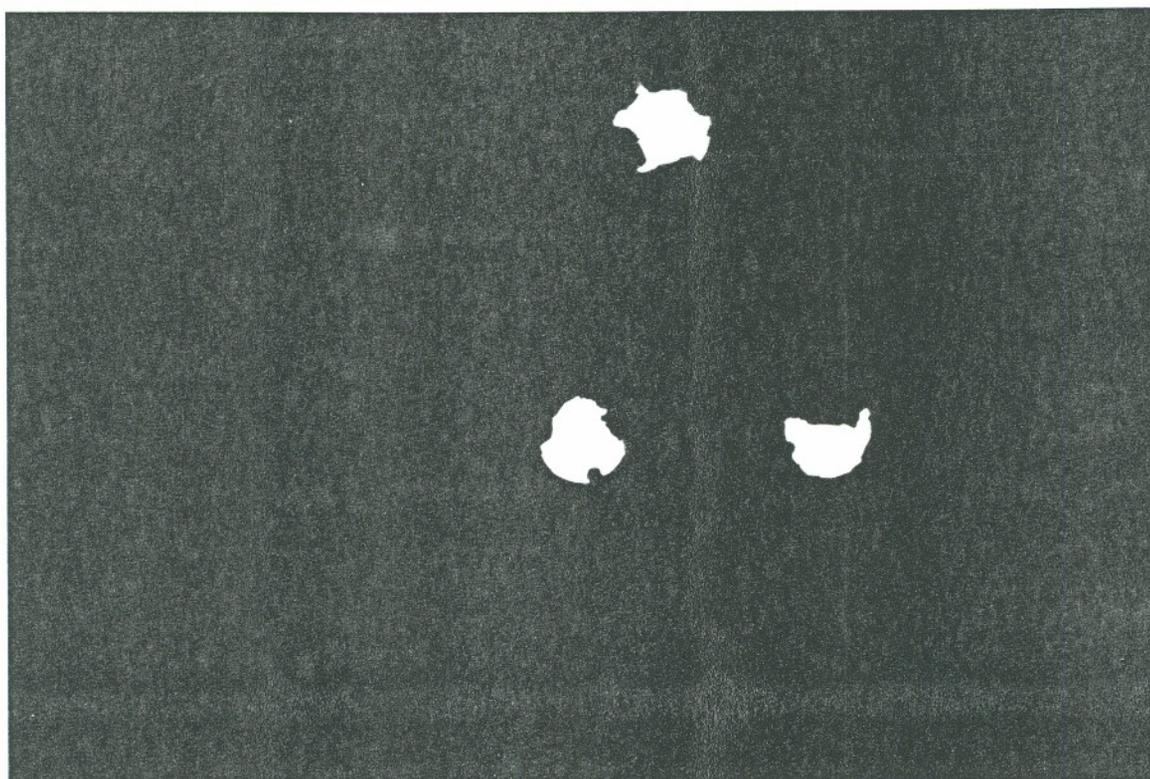


Fig 4-12 Photo (X100) of three pinhole point sources in triangular configuration.

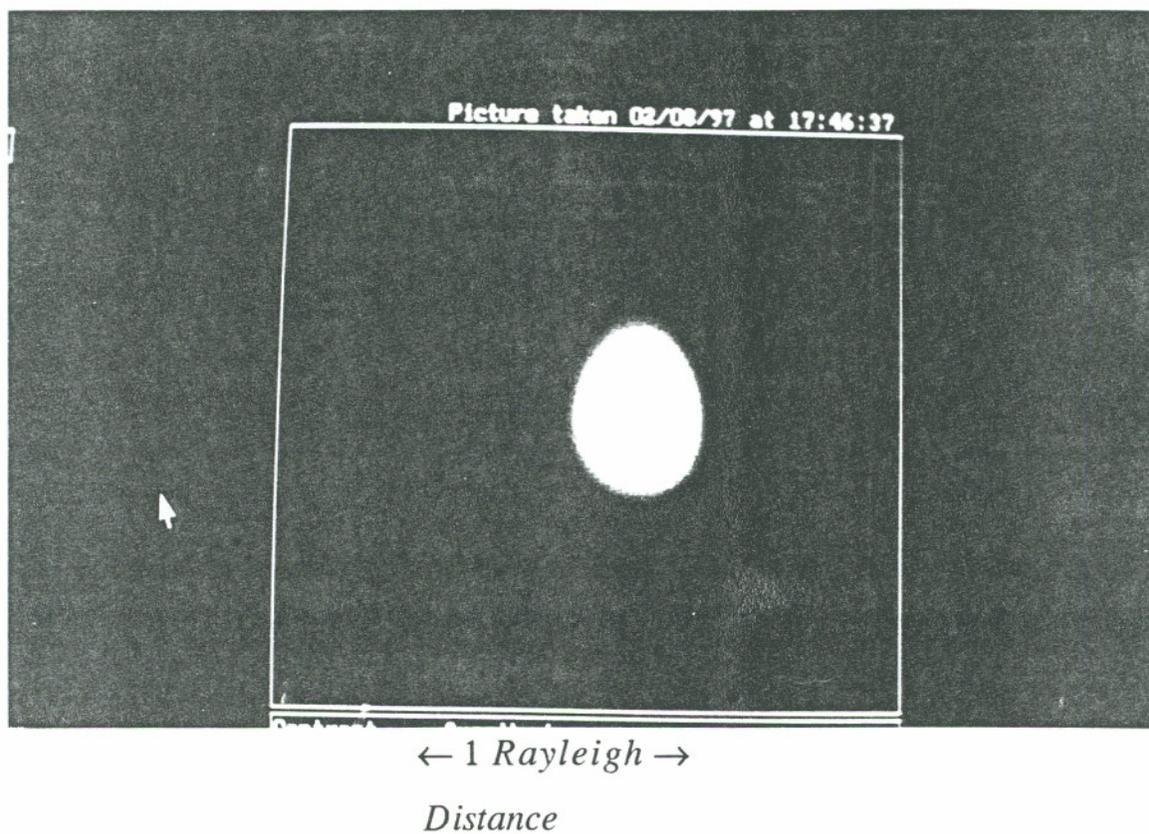
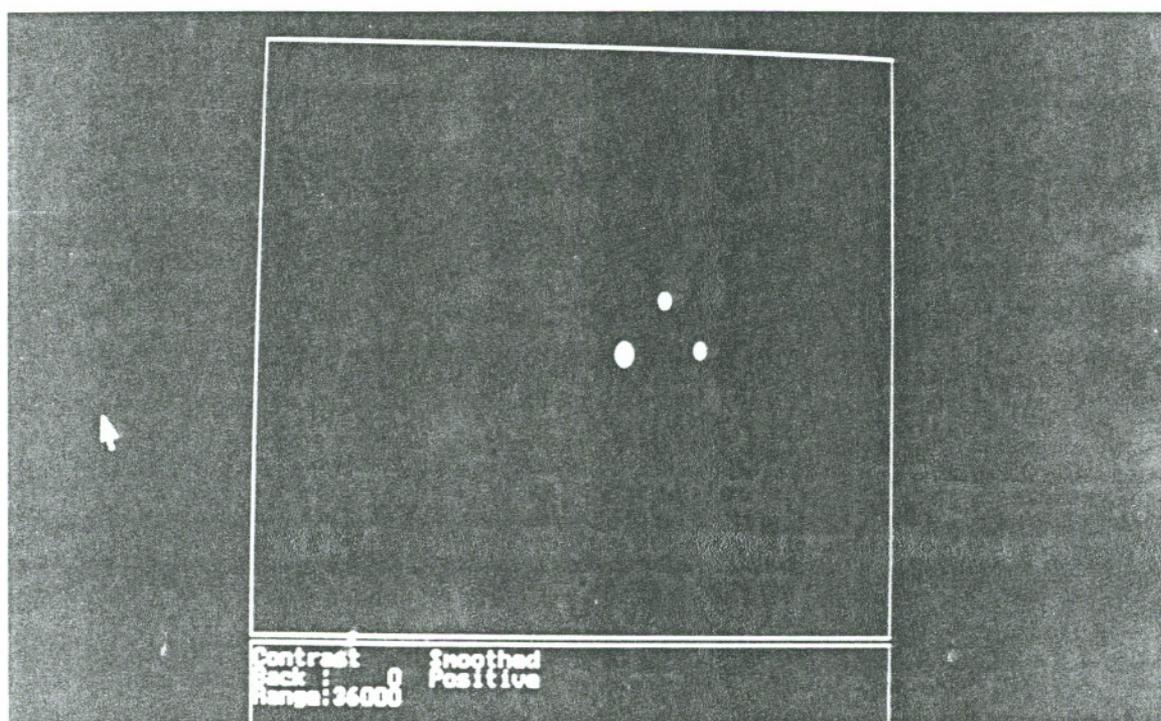


Fig 4-13 Raw image three point sources in triangular configuration about $1/4$ Rayleigh distance apart, $SNR \approx 60$. See Fig. 4-12. No air turbulence evident.



← 1 Rayleigh →

Distance

Fig. 4-14 Restoration of figure 4-13 achieved through deconvolution. See figure 4-12.

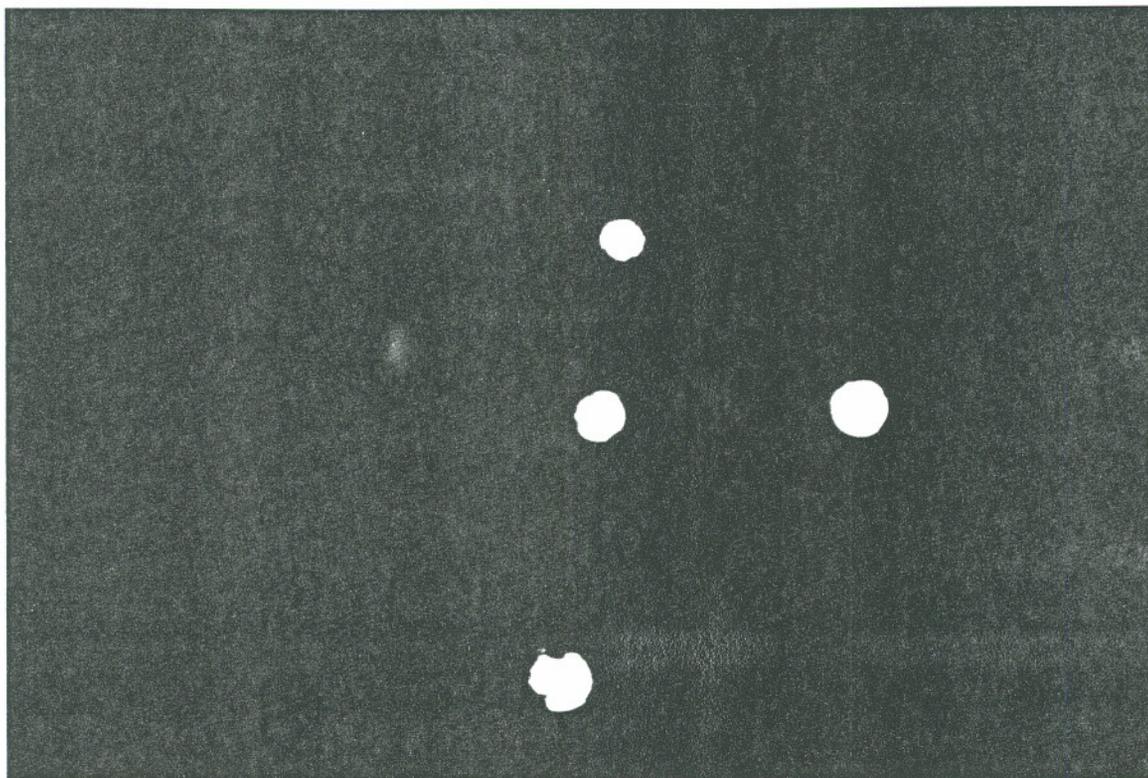
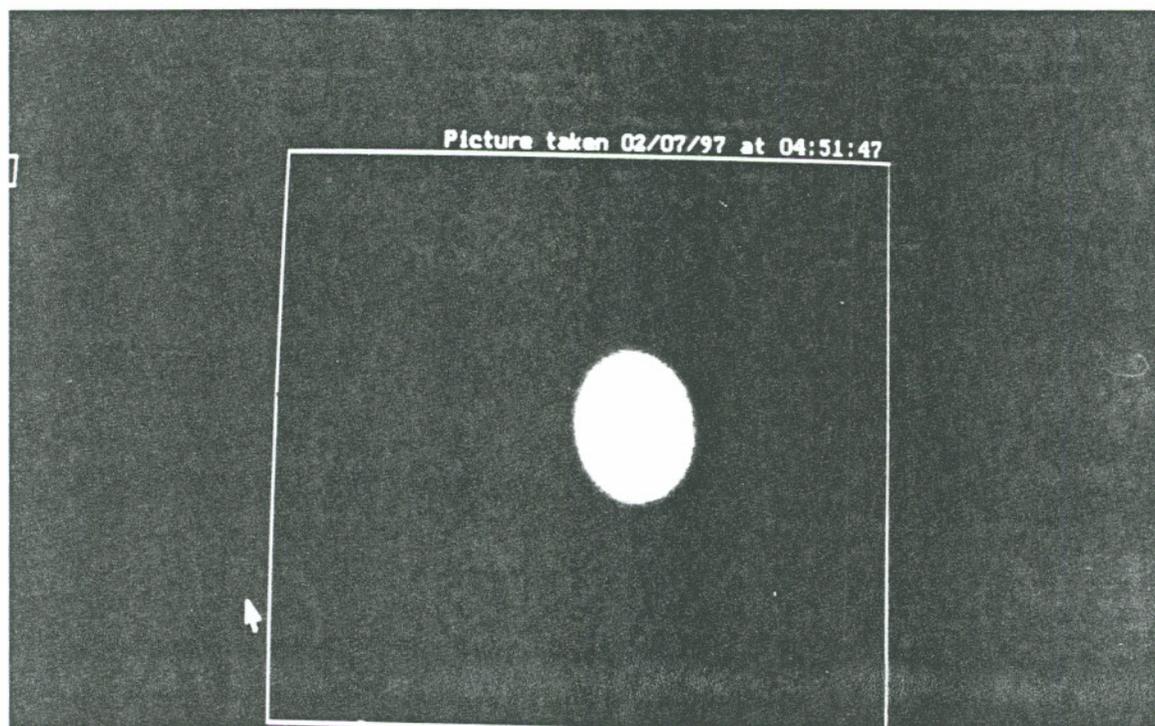
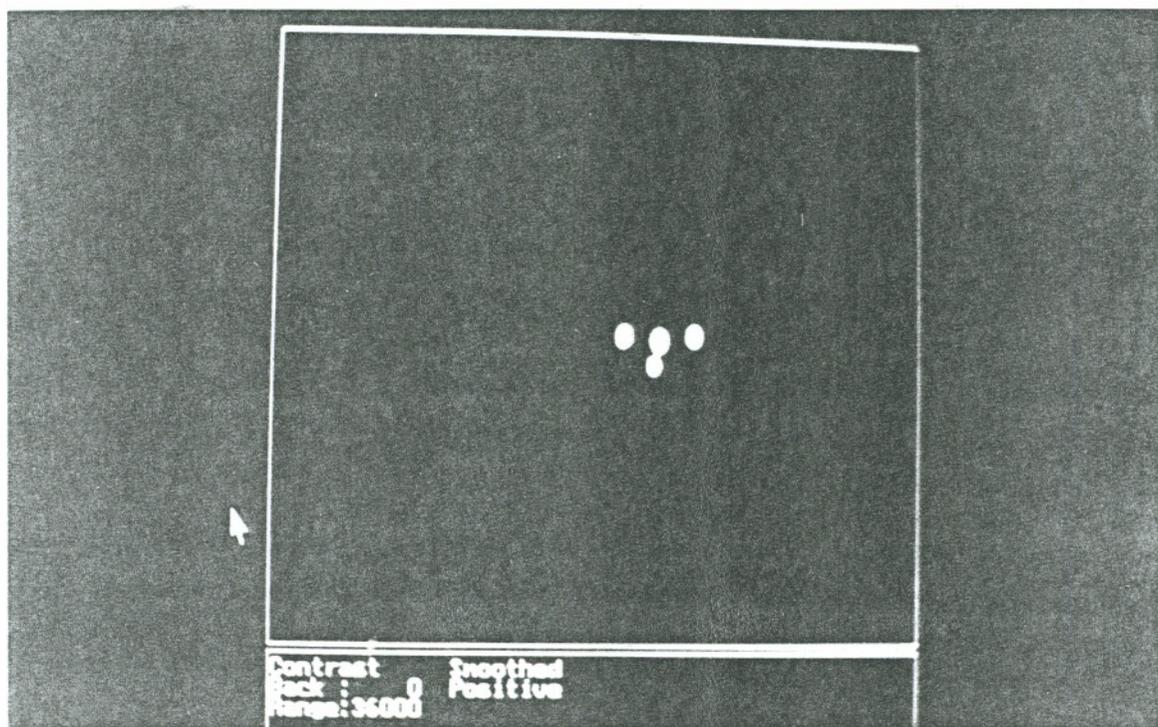


Fig. 4-15 Photo (X100) of four pinhole point sources.



← 1 Rayleigh →
Distance

Fig. 4-16 Raw image of four point sources about $1/4$ Rayleigh distance apart, $SNR \approx 60$. See figure 4-15. No air turbulence evident.



← 1 Rayleigh →
Distance

Fig 4-17 Restoration of figure 4-16 achieved through deconvolution. See figure 4-15.

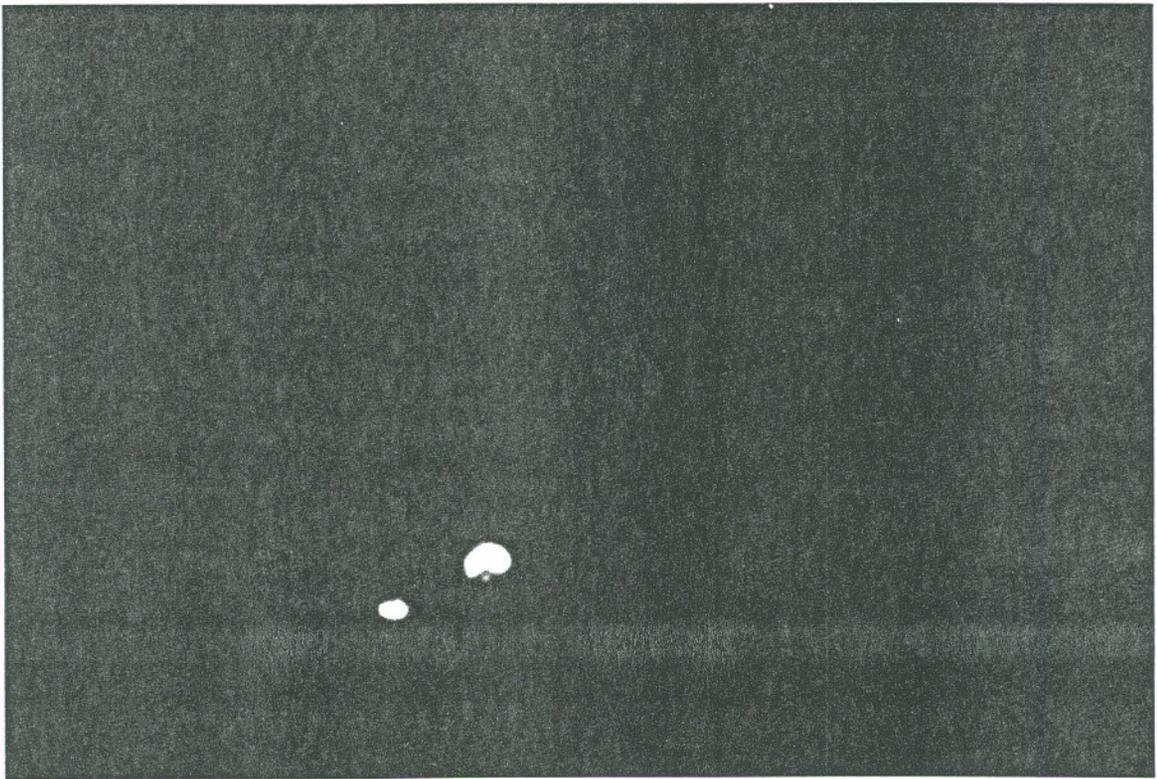


Fig. 4-18 Photo (X100) of two pinhole point sources.

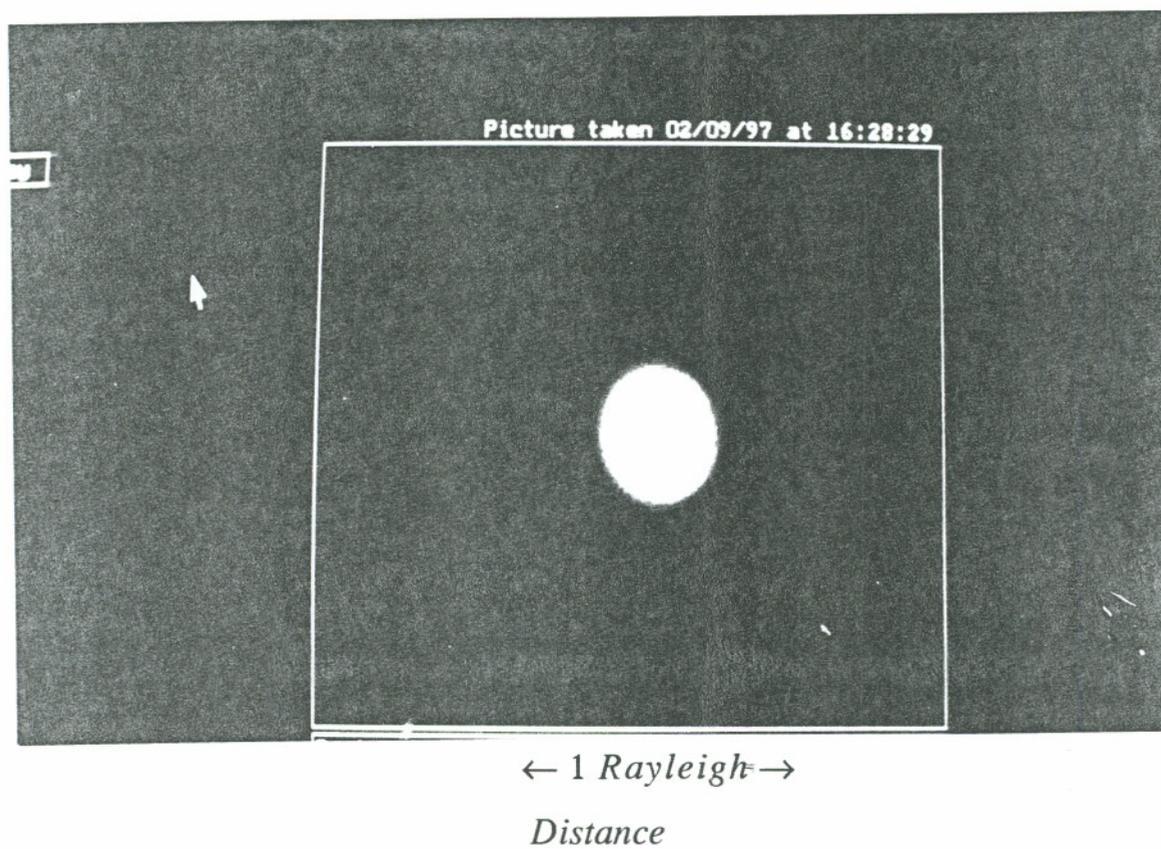


Fig. 4-19 Raw image of two point sources placed $1/10$ Rayleigh distance apart, $SNR \approx 60$. See figure 4-18. No air turbulence evident.

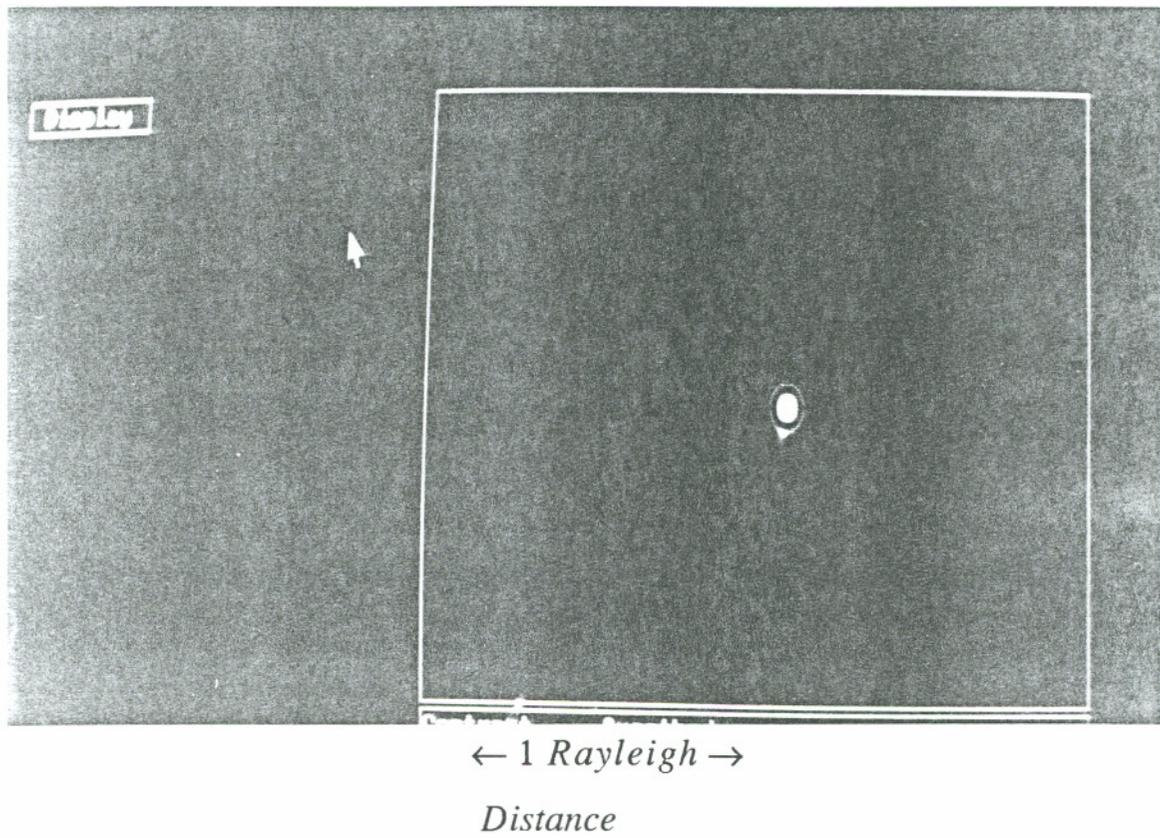


Fig. 4-20 Restoration of figure 4-19 achieved by deconvolution. See figure 4-18.

4.18 SIMULATIONS

There are three main simulations included here, a planet-finding program that sampled a grid of points and checked for a second or third object (Appendix B), programs that essentially Nyquist sampled one dimensional data (appendix C), and the simulations that ran as the program does in chapter 4 (appendix A). The methods of appendix B and C are discussed. The "planet-finder" program (appendix B) sampled the data in a grid pattern. The matrix equation (4.3.1) was solved for k_i and PY_i , PX_i , and the test given by minimizing (4.3.3) was used to see if the solution was acceptable. The center sun was given a coordinate so that only choices were made as to where the planets were located. A random number generator picks new values of k_i , PX_i , PY_i and if the equation (4.5.5) has a smaller value then these values are picked. It was noted that this method gave *very good contrast* as opposed to the method used in appendix A. This contrast was on the order of about 200 to one. The other method (appendix A) had better resolution but at most gave these results if there was better than 3 to 1 contrast. This planet-finder program is definitely worth doing continued research on. Unfortunately it was obvious from the outset the SNR and aberration effects were much too large to make this method practical using this particular experimental setup.

In the Nyquist sampling method (appendix C) we take two sets of points along a line and sampling the region outside the PDF in a Nyquist fashion. The inside region was sampled in a way that accentuated the bends in the data (i.e., high second derivatives). We inverted equation (4.3.1) again. Then the results from the first set of data points were compared with a least squares fit of the second set of data. The best fit choice of k_i , PX_i , PY_i was then chosen.

4.19 CCD MEASUREMENTS

We used an SBIG ST-6 CCD manufactured by Santa Barbara Instrument Group (SBIG), 1482 East Valley Rd., Santa Barbara California, 93108. The CCD receiver window has dimensions of 8 mm by 6 mm. It was thermoelectrically cooled so that much of the bulk of the CCD consisted of radiator metal sheets. Care must be taken in pointing the

CCD. Bright spots appear on the CCD image from diffraction of light through the baffles. They have the appearance of Arago bright spots because of the Babinet's principle. These bright spots must be made as sparse as possible by suitable baffling. Then the image region must be located in the region with the fewest Arago bright spots.

The CCD data calculation results were better in some ways than the simulations. Here we present the data. The data indicates that we achieved a 1/10 Rayleigh distance resolution for two objects.

4.20 ASTIGMATISM INTRINSIC TO THE CCD

There is a mystery in the pointing of the CCD itself. Apparently stronger-than-expected astigmatism results from pointing the CCD in certain directions. The image seems elongated in a diagonal fashion. This is tentatively caused by reflections in the CCD tube. The images gained from this kind of setup are useless. To fight this effect many trial and error eyepiece direction changes must be done with a point source PSF until the problem is taken care of. Even when this effect is corrected there is about a 20% level of astigmatism left. This leftover astigmatism must be corrected for on the computer. See lines 23-25 of program 1 in appendix A. To use these lines of code, CCD images must be taken of specific configurations of sources, such as the three-in-a-row source. Two tests should be given at 90 degrees apart for each test. This is much easier than testing using a PSF. Also the images should be taken on one section of the CCD screen with approximately the same set maximum number of electrons per pixel. Thus any corrections for astigmatism will be set for any picture of that intensity and position on the screen. We do this because the CCD seems to give very different aberrations for different screen positions and different intensities and different pointing directions of the CCD. On the CCD we were using the region in the upper left middle seemed to be the most aberration-free area on the screen. The stretch number is reset until all of the image satisfy our criteria that F (in 4.5.5) is minimized and then the points come out in the configuration given (such as the three points). This is done with several configurations of wide ranging PDF sizes (i.e., wider ranging stretch values). Once this is all done, then CCD images can be taken of new objects and the images will turn be correct. All this would not be necessary if one could

make a permanent setup that was mechanically stable, such as a space telescope.

4.21 COMA

In the telescope measurements pointing errors result in a strong Coma aberration. Also blooming caused by too high electron count per pixel can result in an electronic coma. Much trial and error pointing is needed so that coma is reduced as much as possible due to optical aberrations. For off axis telescopes it is best to have the eyepiece as close to the edge of the telescope tube as possible. This is because apparently in parabolic reflectors coma is almost impossible to eliminate especially off axis. Blooming in our CCD became a problem if too much dust accumulated on the CCD plate. Thus periodic dust removal is important. Blooming is also a problem if the CCD is left on too long. Thus the CCD must be shut down at frequent intervals to limit this effect.

4.22 SCATTERED LIGHT

The lenses and mirrors must be cleaned thoroughly. Scattered light from surfaces is a strong source of distortion of a PSF and creates an anomalously large Airy disk. It also cuts down on the resolution significantly. It must be noted that several practical steps must be done to guarantee a result with high SNR.

4.23 EXPERIMENTAL CONSIDERATIONS

In the pinhole experiment all surfaces must be painted flat black. Even the areas around the pinholes must be black. Light reflected off the CCD camera must be reflected away by glossy black paint so that it does reenter the CCD.

The CCD must have the glass covering removed. In the pinhole experiment, thin wedge type diffraction patterns caused by this plate makes our deconvolution algorithm useless. If these patterns are visible, then the deconvolution will not work.

To create an incoherent image the light coming through the object pinholes must be incoherent. Also ordinary incoherent sources such as incandescent light bulbs create many problems such as straylight and heating. They also require light filters to be used since our deconvolution algorithm requires monochromatic light. Reflecting light off rotating

disks and such will not achieve a high enough SNR. Putting in random motion a nonpolarizing transparent material close to the pinholes and in the beam path seems to be the best method for doing this. I did this by attaching thin sheets of translucent plastic to an electric motor rotor, with the motor causing the plastic to rotate in the path of the laser beam. To test for noncoherency one must observe at the aperture whether there are multiple slit diffraction patterns caused by multiple pinholes. If these patterns are observed than the light is not incoherent. The rotating plastic sheet must then be placed closer to the pinholes. Also exposure times must be such that the plastic sheets rotate at least once during an exposure. The plastic is a polarizer and so many orientations must be sampled for polarization effects to be averaged out. The rate of rotation must be large and the number of plastic sheets must also be large. In my opinion the uncertainty in the effect of choosing different types of plastic sheets here is the most fickle part of this work. If all this is correctly done the light that then enters the pinholes is certainly spacially incoherent and with long time exposures also gives the appearance of temporal incoherence. Thus we are allowed to use the assumption of incoherence that is made in equation (4.3.1).

D. Avoid Arago bright spots. Make sure the baffling is large enough so that only a small amount of light gets diffracted around the baffles. If the baffles are too small, a small bright spot appears on the PSF making it nearly impossible to use our deconvolution algorithm. The CCD surface that was used here had to be pointed away from true perpendicular at angle of about 5 degrees. This pointing step is probably the trickiest part of this procedure. The baffles must be placed so that the Arago bright spots are not close to the region of the CCD screen that has the least aberrations since that is where we will take our images.

Care must be taken in making the pinholes. The objective pinholes should be tested to see that the diffraction rings around the maxima are not splotchy for a beam of monochromatic light that has passed through the pinholes. Spinning the needles as they are stuck in thick aluminum foil seems to give very good pinholes. The pinhole aluminum must be painted flat black as well to avoid light being reflected back into the CCD and causing more spotting of the image.

If the CCD work is done using an SBIG CCD, which uses DOS, we must convert f

from DOS to Unix and from Unix to DOS if the Fortran analysis is done on a Unix system. Our algorithm code is nearly 2000 lines long and the runs are very time-consuming so the Unix system is needed. Special software must be written to convert from Unix to DOS and from DOS to Unix since the number formats are different. This software is included in the appendix.

Our pixel coverage is only a few rayleigh lengths in size. Thus there is a great deal of information to deal with in doing the deconvolution. The images for this kind of pixel coverage are very small at the telescope focal point so that we need to attach a powerful microscope to our telescope to get this pixel coverage on the CCD.

For telescope applications this means that the light intensity requirements are large. For the configuration we used we needed about -10 apparent magnitude to get an image that could be used for deconvolution with 1/100 second exposure. A dim image that normally would occupy one pixel is now spread out over about ten thousand pixels and must be about 30000 electrons per pixel in intensity at its brightest. For 100 electrons per pixel noise this is a factor of 300 times the minimum intensity observable. Thus we have about a 15 magnitude difference between the dimmest object a telescope can observe and objects that can use this deconvolution for a given exposure time. For optimization purposes this could be lowered to about a 10 magnitude difference. Also noise levels for advanced telescope applications seem to be a lot less than what we had in our experimental setup so this intensity requirement should be reduced another large factor (to an order of 8 apparent magnitude difference). Furthermore we did not make take advantage of the flat field capabilities of our software and further reduce our need for bright objects to get a high SNR. Thus we see here an eventual magnitude difference of about 7. Thus if a given space telescope has a limiting magnitude of about 25 at a given exposure than with that same exposure magnitude 18 objects could be superresolved, which would allow us to resolve millions of objects in the sky. Of course deep sky cosmological objects on the order of 25+ apparent magnitude are far beyond the reach of this apparatus.

Comparatively dim objects (but probably not planets) orbiting nearby stars may be observable with this device. To do this one must fix the number to be observed with the PSEC parameter to two. Then great care is needed to adjust for astigmatism in the manner

discussed in the above sections.

4.24 PINHOLE SETUP REQUIREMENTS

Here we illustrate how we can use a pinhole setup (figure 4-20 to 4-23) to test deconvolution. From equation (2.1.30) we have that:

$$\Theta \approx \frac{\text{image height}}{\text{image distance}} = \frac{1.22\lambda}{D} \quad (4.24.1)$$

We define image height here to be 1 Rayleigh distance. Note that a Rayleigh distance is dependent on the aperture diameter D . For $\lambda = 6338.3 \times 10^{-10} \text{ m}$, $D = .363 \text{ mm}$ and image distance = .94 m we have from (4.24.1) that the image height = 2 mm. Thus on the ST-6 CCD screen (which has dimensions 6.5 mm X 8.6 mm) we see that 1 Rayleigh distance = 2 mm. We will call the $D = .363 \text{ mm}$ distance the aperture diameter. If two pinholes are 1 mm apart .94 m in front of the aperture then we see from similar triangles that they will form images that are 1 mm apart at the CCD which is 1/2 a Rayleigh distance apart. Thus in this setup we know what fraction of a Rayleigh distance our images should be separated. Since the laser beams are on the order of .75 mm diameter we have a convenient testbed and we can very easily judge the resolution provided by our deconvolution algorithm.

4.25 TELESCOPES

Here we try to apply the methodology of section 4.24 to telescopes. We note that aberrations caused by pointing errors (see figure 4-28) motivates us to build an entirely new type of telescope just to do superresolution. We wish to determine the effective resolution of this telescope when our deconvolution algorithm is used to restore its images. Let

$$\Theta = \frac{1.22\lambda}{D} = \frac{h}{\text{Distance}} \quad (4.25.1)$$

where Distance equals distance to object from the telescope mirror, h is the height of the object and D is the aperture diameter (.0381 m), which in this case is a telescope instead of pinhole. Let $\lambda = 6338.3 \times 10^{-10} \text{ m}$, Distance = 116 m, $D = .0381 \text{ m}$. Then we can write:

$$\frac{1.22(6338.3 \times 10^{-10})}{0.381} = \frac{h}{116} \quad (4.25.2)$$

Thus $h \approx 2.35$ mm=pinhole separation for one Rayleigh length. We will next use the high radius of curvature mirror equation:

$$\frac{1}{f} = \frac{1}{p} + \frac{1}{q} \quad (4.25.3)$$

Here p is the above term called Distance. Let the focal length of the mirror be $f=83$ m, then

$$\frac{1}{1.98} = \frac{1}{116} + \frac{1}{q} \quad (4.25.4)$$

Thus $q=2.014$ m. So the difference from the normal focus is about $2.014-1.98=3.4$ cm.

Because of the closeness of the object, we must extend our CCD 3.4 cm beyond the normal focus.

$$M_1 = \frac{q}{p} = \frac{1.22m}{116} = 0.0105 \quad (4.25.5)$$

$$M_1 M_2 = \frac{\text{imageheight}}{\text{objectheight}} = \frac{0.002}{0.00235} = 0.85 \quad (4.25.6)$$

so that

$$M_2 = 81 \quad (4.25.7)$$

If the image is about 3.81 mm cm in front of the center of the eyepiece then

$$M_2 = \frac{q}{p} = \frac{\text{Length}}{0.0381} = 81 \quad (4.25.8)$$

Thus Length=3.1 m. The extension onto the eyepiece that attaches to the CCD is 3.1 m long. Note that M_2 is the magnification of the image at the eyepiece. The image at the focus of the primary mirror must be magnified 81 times. Thus we must attach a large microscope to our telescope to allow the telescope to do superresolution with our deconvolution algorithm.

It initially appeared that the reflecting telescope method would not work here as well as the refracting telescope method. Slight misalignments cause aberrations to be

amplified because of the magnification stage. Nonetheless we got it to work as well as the reflector. Note the “Spot” diagram of figure 4-26 for the off axis telescope we designed. This is to be contrasted with figure 4-28 in which there is secondary mirror in the beam path. Our telescope design has reduced the aberration effects to a minimum. Apparently the amount of aberrations is low enough here for this method to work. This is the motivation for the hardware design stage of our work. Also note the CODE V ray trace in figure 4-25. Note the actual pictures of the telescope in figure 4-23 and hallway “straylight chamber”.

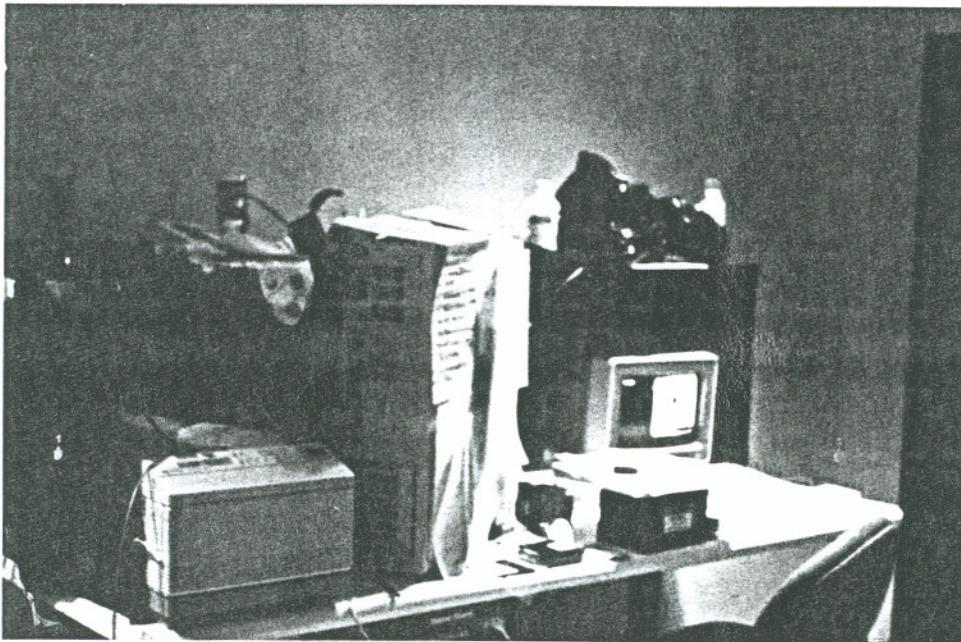


Fig. 4-21 PC setup for pinhole experiment.

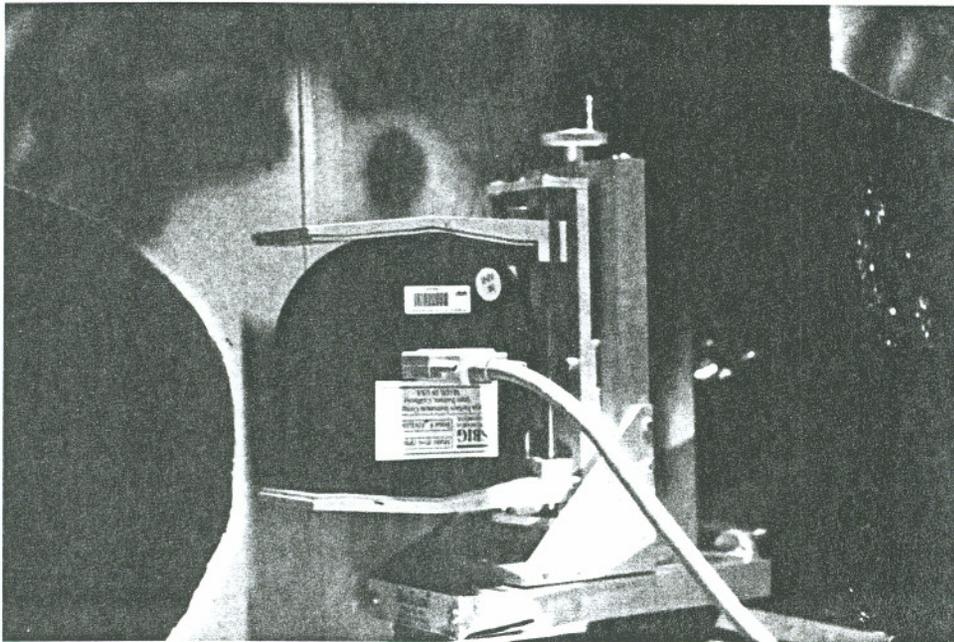


Fig. 4-22 CCD set up for pinhole experiment.

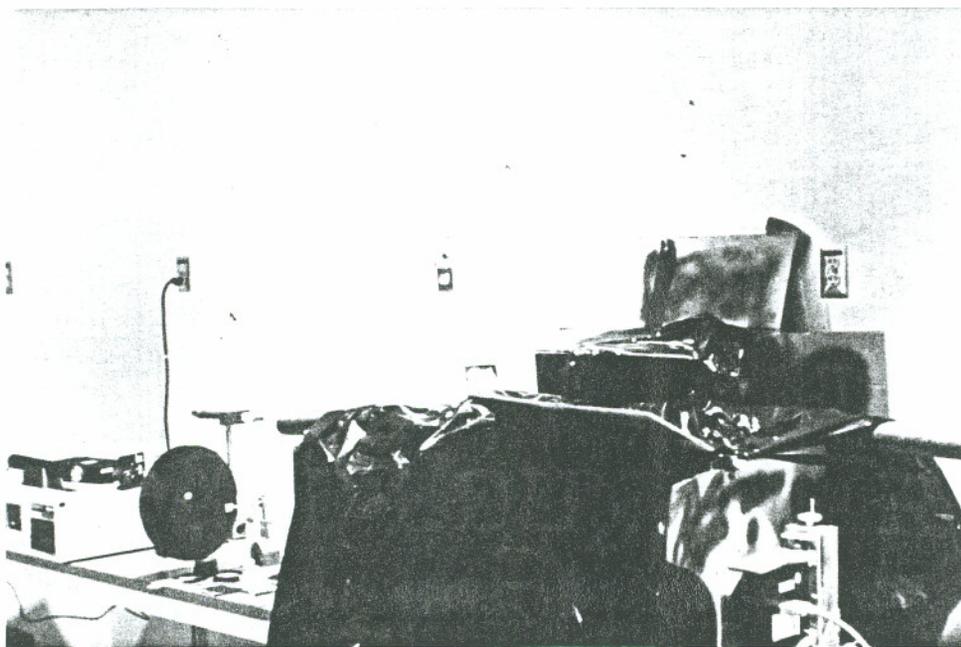


Fig. 4-23 Laser source, pinhole object and pinhole aperture.

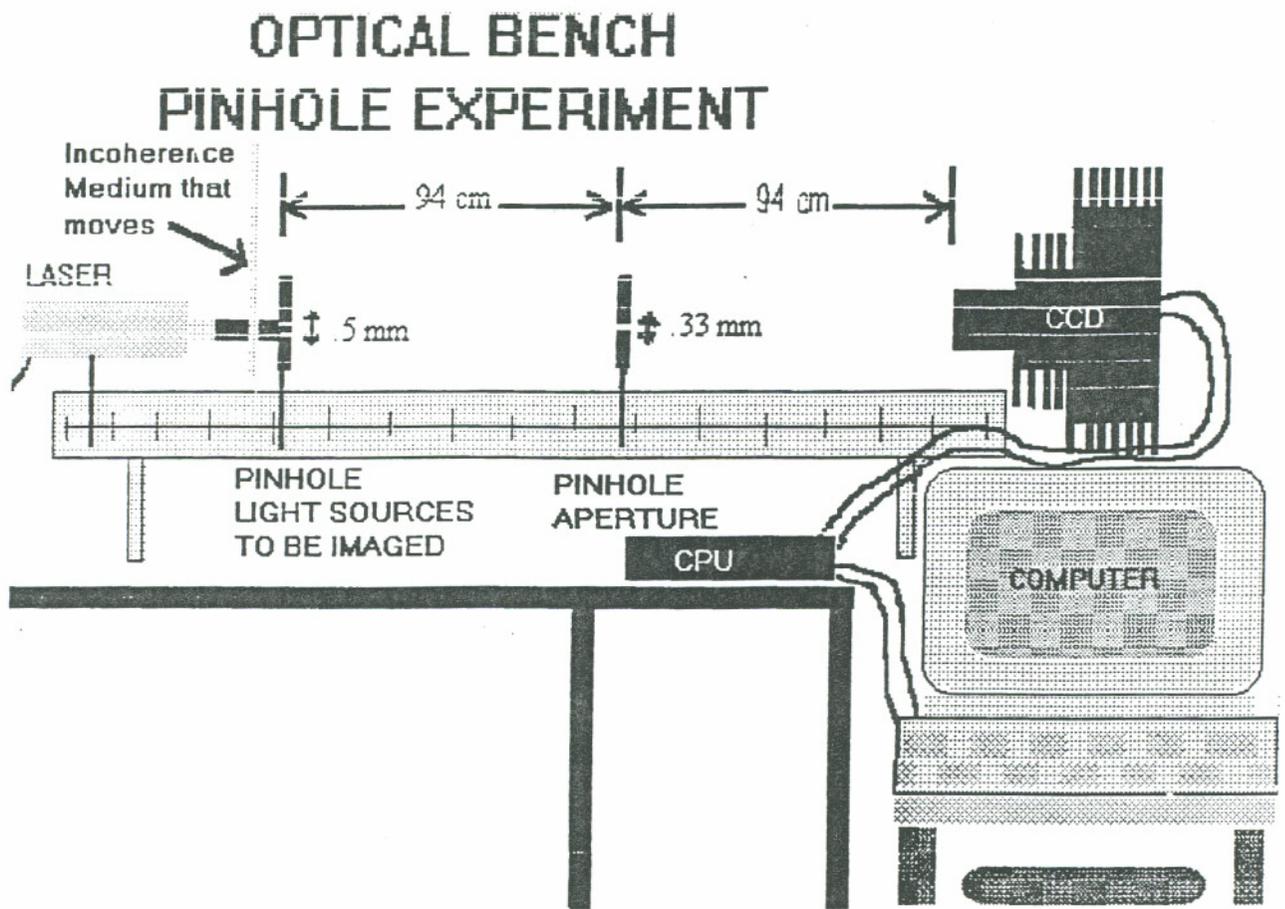


Fig.4-24 Pinhole experimental setup.

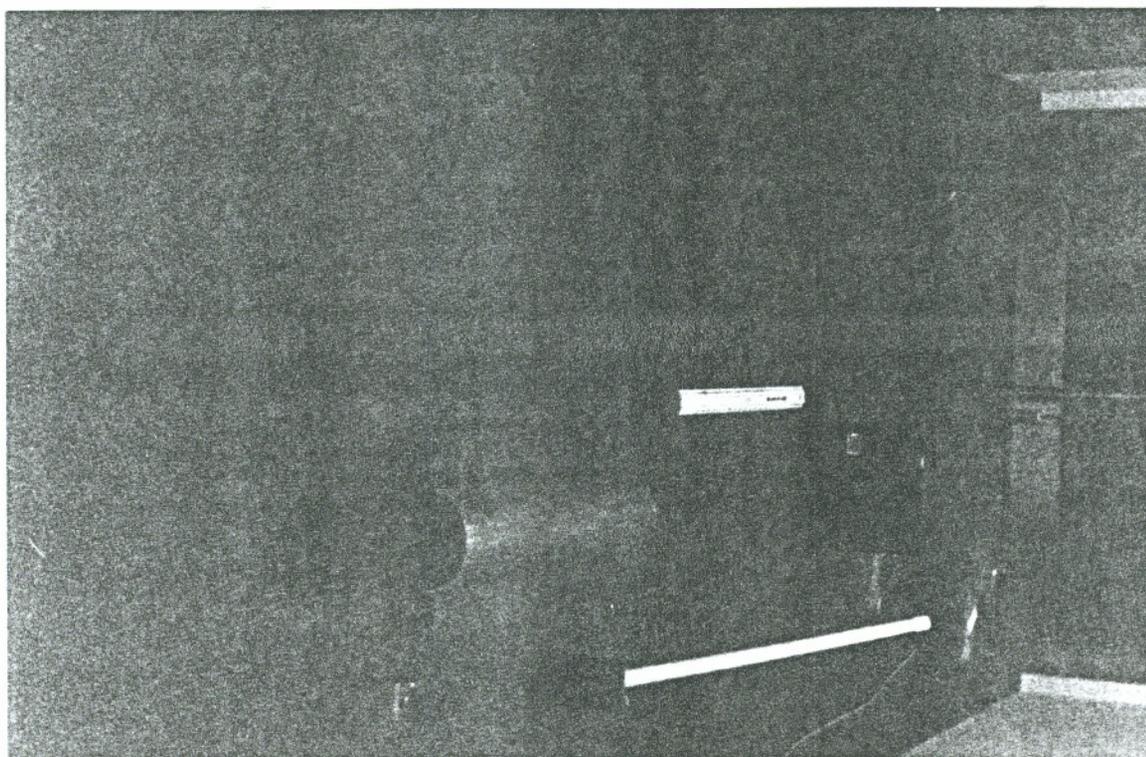


Fig. 4-25 Photo of superresolution telescope design in hallway.



DMAKER

23:01:38

Fig. 4-26 Superresolution telescope CODE V ray trace.

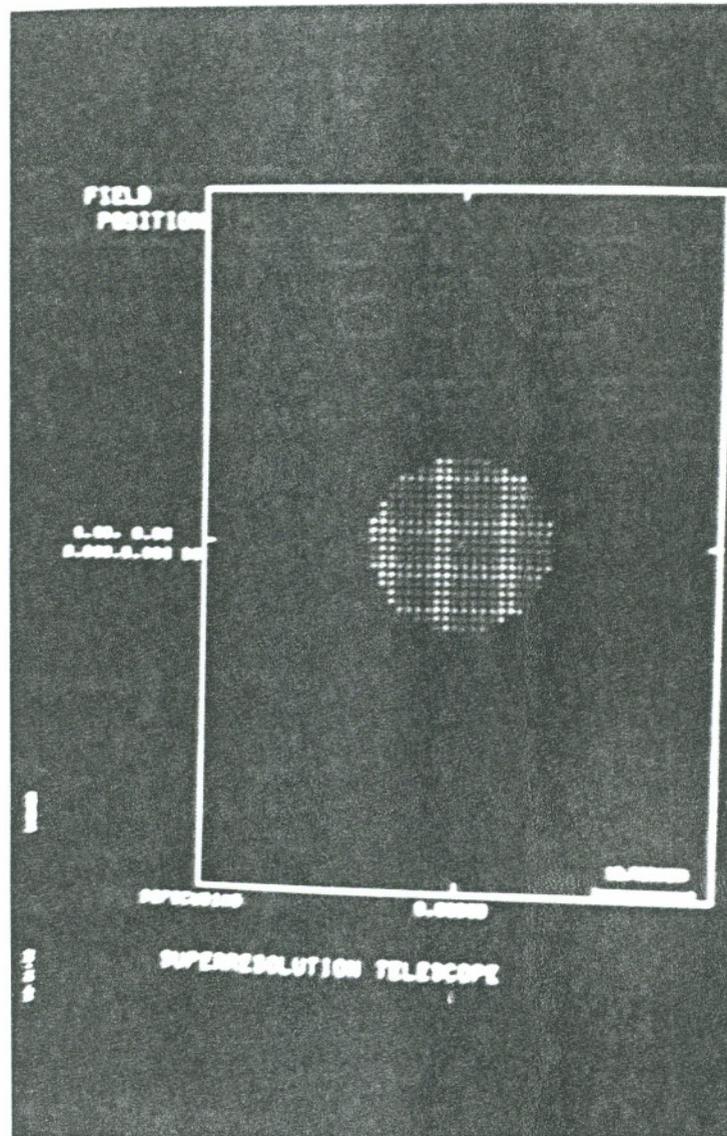


Fig. 4-27 Spot diagram for superresolution telescope.

Superresolution Telescope

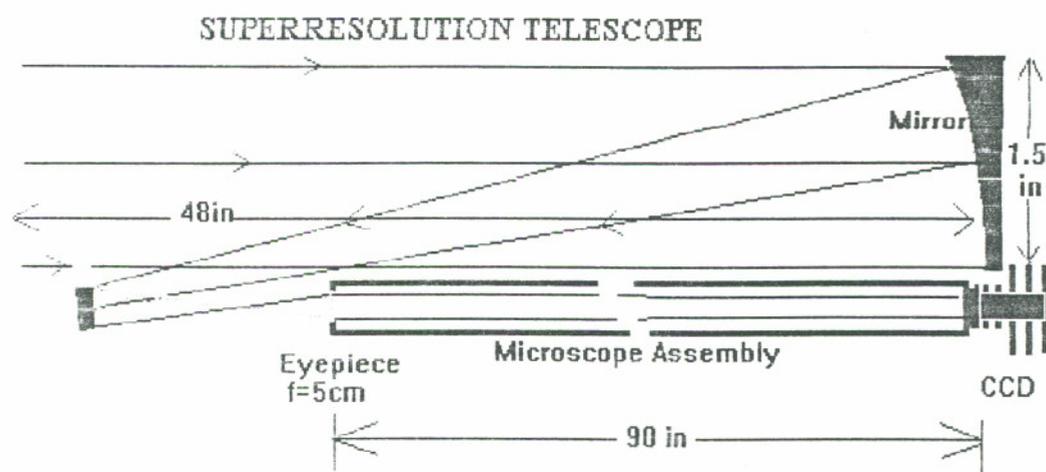
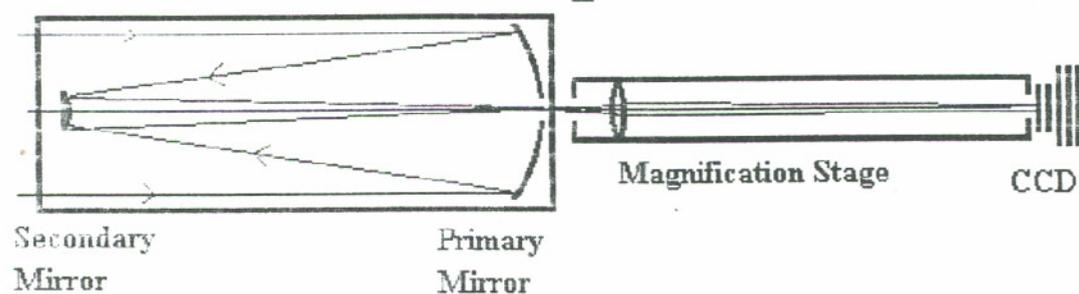


Fig.4-28 Superresolution telescope design.

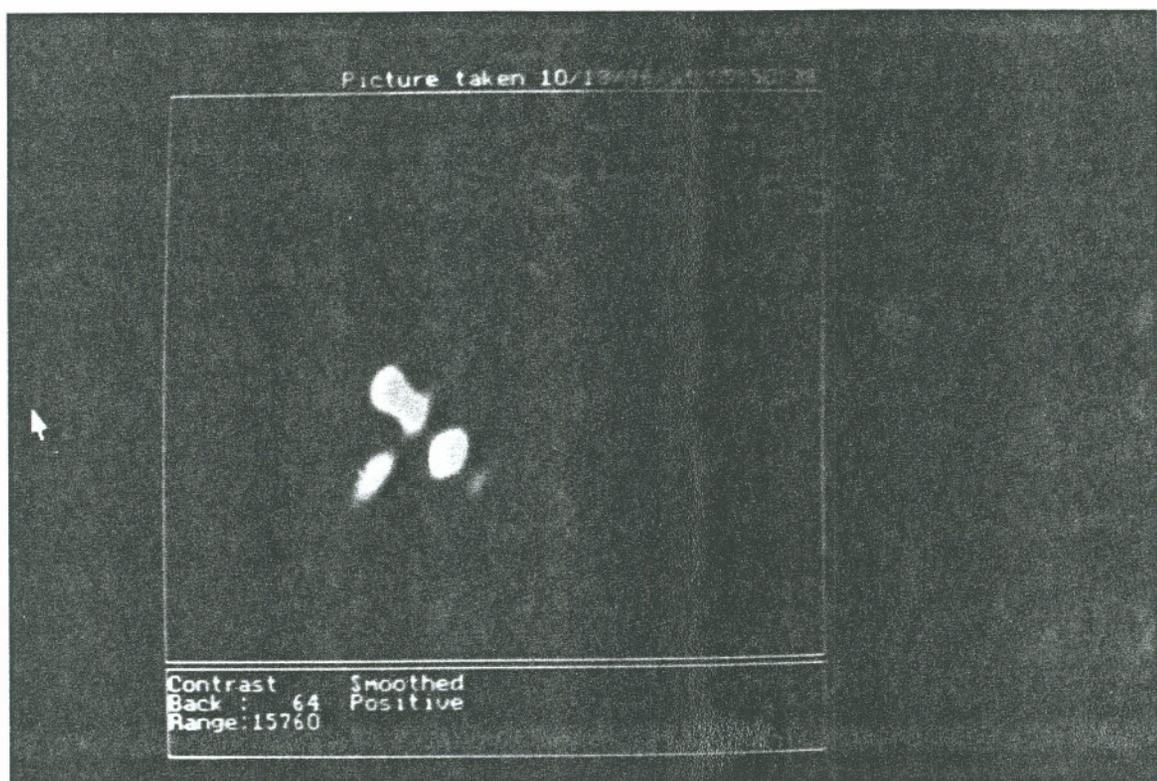
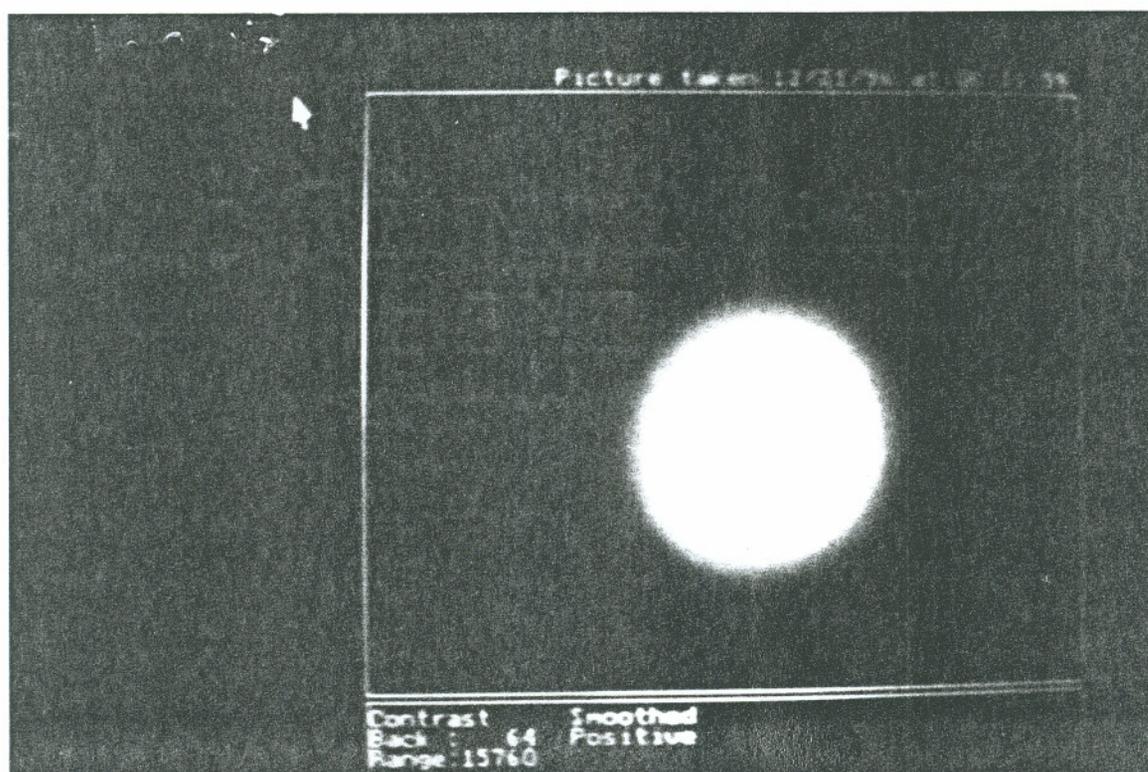


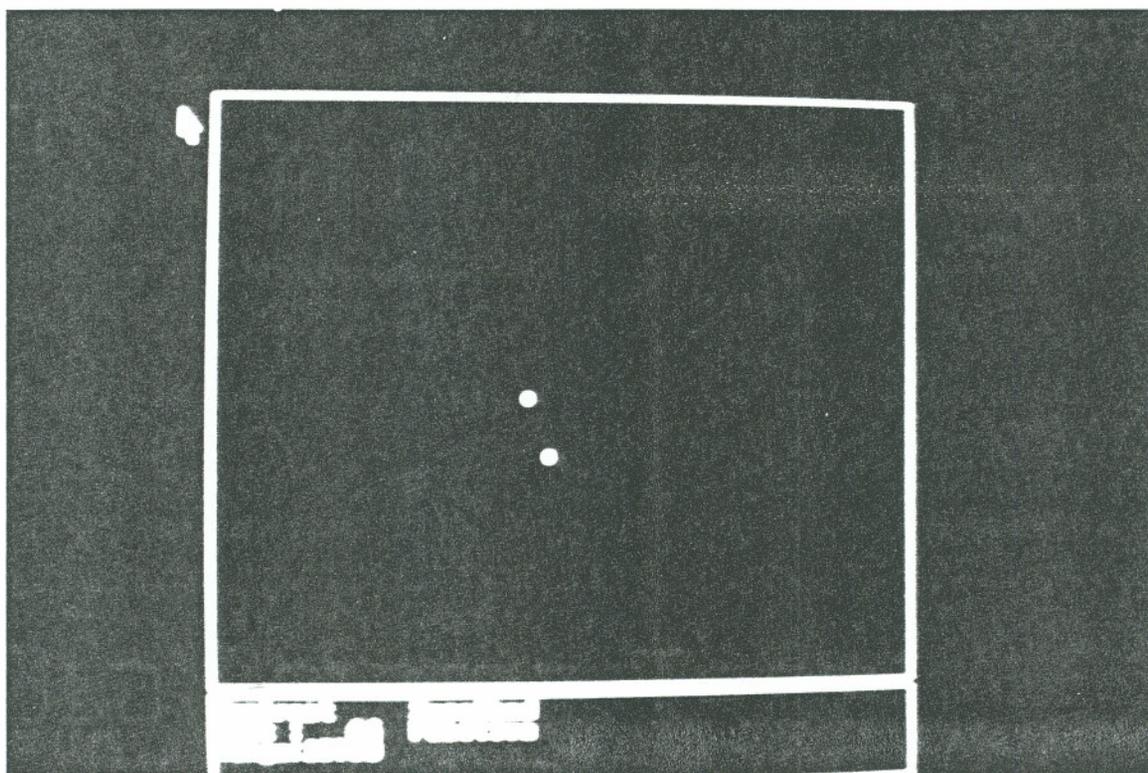
Fig. 4-29 Actual PSF with aberrations in ordinary Newtonian telescope. These PSF's are unusable for superresolution. A special superresolution telescope is needed.



← 1 Rayleigh →

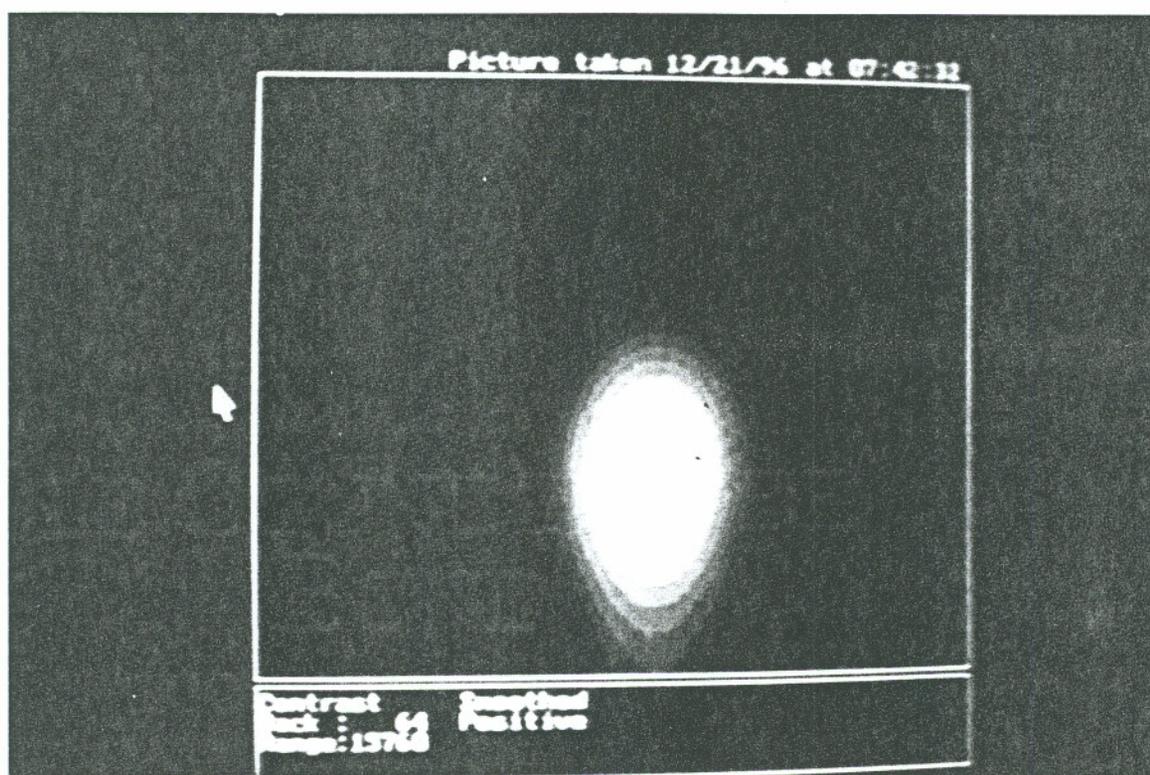
Distance

Fig. 4-30 Raw image of two point sources about $1/4$ Rayleigh distance apart, $SNR \approx 30$. Image taken reflecting telescope having 1.5 inch aperture in 130 meter line of sight. Air turbulence was evident. See figure 4-9.



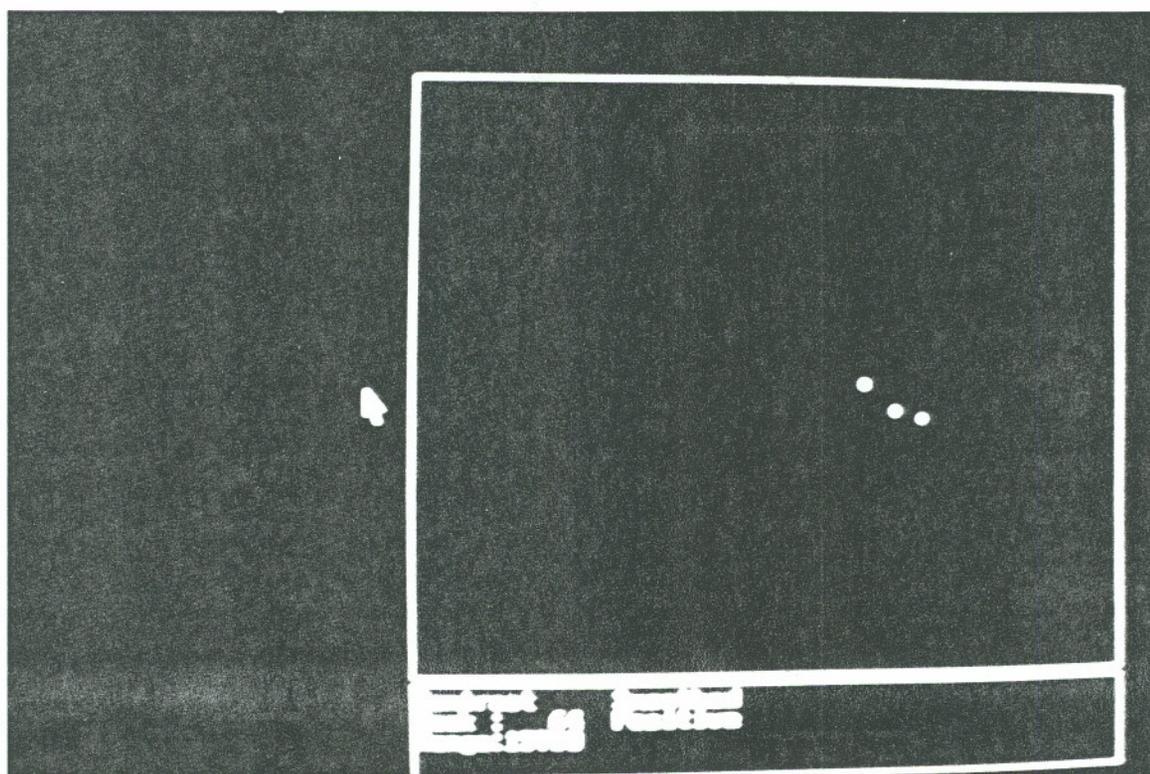
← 1 *Rayleigh* →
Distance

Fig 4-31 Restoration of figure 4-30 achieved through deconvolution. Air turbulence was evident. See figure 4-9.



← 1 Rayleigh →
Distance

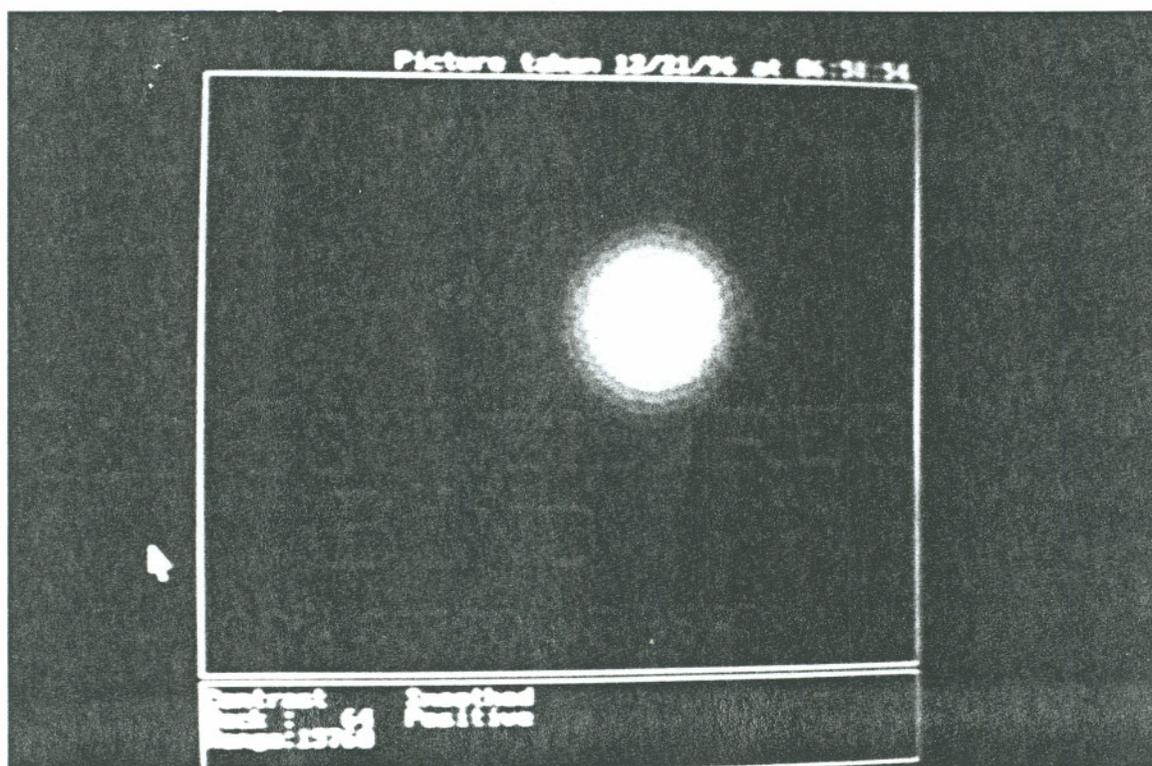
Fig. 4-32 Raw image of three point sources about $1/6$ Rayleigh distance apart, $SNR \approx 30$, forming a triangle (see figure 4-13 and 4-31). Image taken through a reflecting telescope of 1.5 inch aperture in 130 meter line of sight. Air turbulence was evident.



← 1 Rayleigh →

Distance

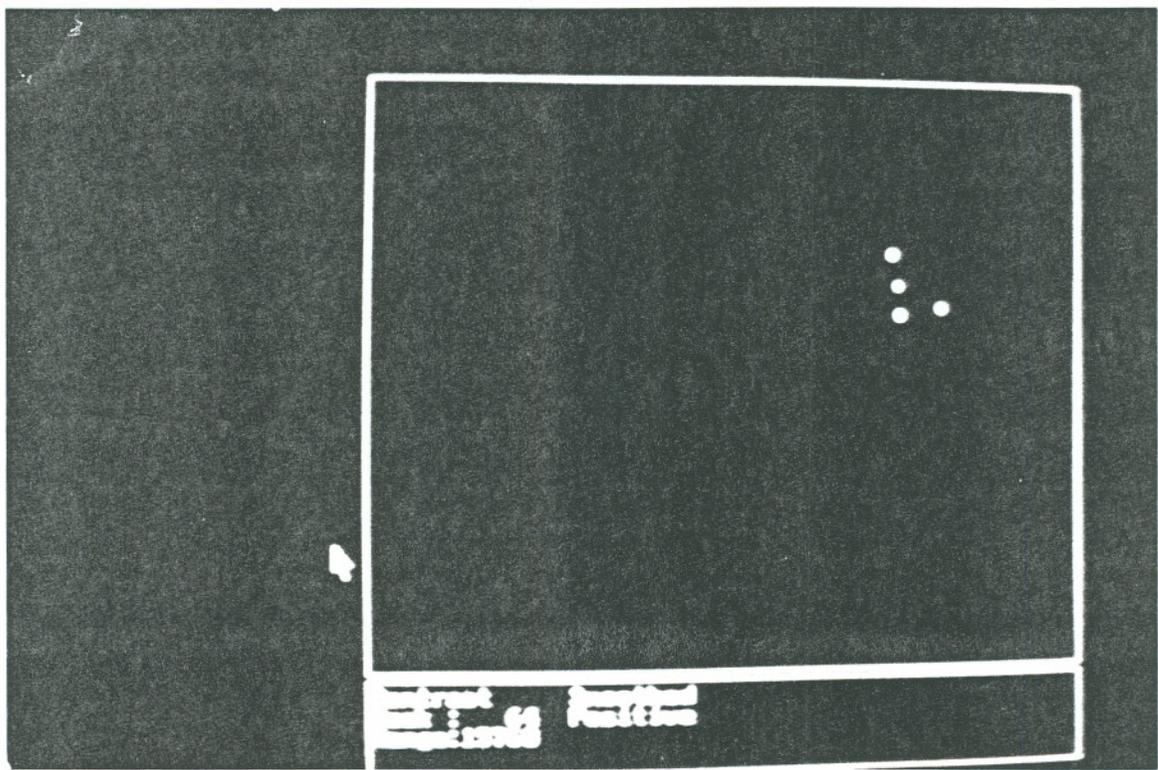
Fig 4-33 Restoration of telescope image figure 4-32 achieved through deconvolution. Triangular appearance is not evident but the number of objects is still counted correctly. Air turbulence was evident. See figure 4-13.



← 1 Rayleigh →

Distance

Fig. 4-34 Raw image of four point sources about $1/6$ Rayleigh distance apart, $SNR \approx 30$. See figure 4-15. Image taken through a reflecting telescope of 1.5 inch aperture in 130 meter line of sight. Air turbulence was evident.



← 1 Rayleigh →

Distance

Fig 4-35 Restoration of telescope image figure 4-34 achieved through deconvolution. Image taken through air turbulence. See figure 4-15.

4.26 COMPARISON WITH OTHER RESULTS

In our review of the literature in chapter 2 we tried to compare what was done with what we are trying to do. Here we review some of the remarks that were made there.

First we note that any amount of resolution is possible if there is complete knowledge of the intensities and numbers for let us say, two objects. For a high SNR image simple geometry will do a decent job in joining perpendicular bisecting sagittas to image points. Thus the type of superresolution we are talking about is for situations in which we neither know how many objects there are nor do we know the intensities. In this case no other method known can go much beyond about $1/2$ a Rayleigh without giving ambiguous artifacts.

For example we note that one samples in a Nyquist fashion *equally over the whole image* for the Lucy Richardson⁸, Van Cittert⁹ and MAP¹⁰ methods which are the best methods for doing superresolution. The CLEAN algorithm doesn't do this but it only achieves about $1/2$ Rayleigh superresolution. But we note in this chapter that unlimited superresolution demands that we sample densely on either side of the maximum and that we dimensionally reduce the problem while paying attention to convexity problems caused by dark noise and image size. Chapter 3 and 4 showed why image ambiguity results if this is not done. Thus these other methods can never achieve a great degree of superresolution much beyond about $1/2$ of a Rayleigh distance, and I have not seen claims to the contrary in the literature. Also note here that our method of doing superresolution involves only a few Airy disks in each CCD image. Thus the scale of the PSF's is much larger than the scale of the noise. Thus smoothing is a very effective way to minimize the effect of noise in our method. So we do not have to rely on stochastic methods such as the MAP method to achieve superresolution. Also Wavelet deconvolution is useless here because the scale of the Airy disk is so large relative to the noise. Thus using standard superresolution methods about $1/2$ Rayleigh distances are resolvable without prior knowledge of the distribution or intensities of the sources. Also in our experiments we noted that about 2 Rayleigh resolution is possible in the kind of atmospheric turbulence (again without prior knowledge of the object) we experienced. But we got about $1/10$ Rayleigh superresolution without air turbulence and about $1/6$ Rayleigh superresolution in significant air

turbulence.

SUMMARY

We developed a way of implementing superresolution that addresses the ambiguous image problem. Ours is an inverse matrix method which restricts testing to only those points on ridges of isophotes. Thus a two dimensional deconvolution problem has been reduced to a one dimensional problem: we are doing deconvolution by **dimensional reduction**. We have added second derivative (convexity) optimization and smoothing. This "dimensionally reduced" localization of the inverse matrix testing results in the elimination of the ambiguous images and minimization of the illconditioned matrix problem. The smoothing eliminates the need for stochastic methods since the scale of the noise is much smaller than the width of a PSF, which is about 150 pixels here. Using a pinhole "telescope" CCD experiment we show that this deconvolution technique results in a $1/10$ Rayleigh distance at $SNR \approx 30$, which is at least a factor of 5 better than the other known superresolution techniques allow for. We built a 1.5 inch superresolution telescope that incorporated optics with minimum aberration effects. Using a 1.5 inch superresolution reflecting telescope we showed that we could achieve about $1/6$ Rayleigh length resolution through air turbulence, which is on the order of a factor of 12 better than expected.

References

- 1) I. Gradshteyn, *Table of Integrals, Series, and Products*, 5th ed., Academic Press Inc., 1985, p.962.
- 2) I. Duff, *Direct Methods for Sparse Matrices*, Oxford Science Publications, 1986, p.79.
- 3) M. Born, *Principles of Optics*, 5th ed., Pergamum Press, 1975, p.416.
- 4) R. Kingslake, The Interferometer Patterns Due to Primary Aberrations, *Trans. Opt. Soc.*, 27 (94), p.1925.
- 5) D. Malacara, *Optical Shop Testing*, John Wiley and Sons, 1978, p.507.

- 6) J. Lee, *Statistical Thermodynamics*, 2nd ed., Addison Wesley, 1973, p.129.
- 7) P. Sementelli, Analysis of the Limit of Superresolution, *J. Opt. Soc. of Am. A.*, **10** (11), 1993, p.2266.
- 8) H. Bi, When Does The Richardson-Lucy Deconvolution Converge, *Astronomy & Astrophysics. Supplement Series*, **108**, 1994, p.409.
- 9) R. Berry, Shooting the Sky With an Electronic Camera, *Astronomy Presents Observers Guide*, 1995, p.88.
- 10) B. Crilly, A Quantitative-Evaluation of Various Iterative Deconvolution Algorithms, *IEEE Transactions on Instrumentation and Measurement*, **40** (3), 1991, p.558.
- 11) P. Sementelli, Analysis of the Limit of Superresolution, *J. Opt. Soc. Am. A.*, **10** (11), 1993, p.2265.

APPENDIX A

Deconvolution programs

```

PROGRAM SMOOTHPSFSANDRIDGE
INTEGER NNN,NN,N,SS,S,SU,ST,SV,NYL,NXL,XL,YL,III,II
INTEGER X(100000),Y(100000),GGGX(1000),GGGY(1000)
INTEGER GGY(1000,100),GGX(1000,100),SQFX(300)
INTEGER SQFY(300),SQX(300),SQY(300),SQQX(300)
INTEGER RH1,RL1,M1,M2,R1,RR1,XX,YY,XXX,DATUM
INTEGER SQQY(300),GX(500),GY(500),NNY,LNN,NYY
INTEGER QX,QY,RIN,MM,RI(3),NXL,NXH,NYL,NYH
INTEGER XXL(4),YYL(4),XXS(300),QQ,ENT,NON(20)
INTEGER nim,in,DA,U,SRCE,SRU,UUU,ABC,RRRRR
INTEGER IMM(100000),SMUTH,E1,MSEP,psfdata
INTEGER I,N1,N2,N3,N,S,NO(300),MINS,PIXPSF,finn
INTEGER NNO(9),N5,N6,N7,RUNS,RE,COH,REAP
INTEGER TELEAB,TURB,HEIGHT,WIDTH,INITIAL,hi,wi
INTEGER YYS(300),II,DA,HH,CAL,in,in1,fin,pin
INTEGER FINISH(10),POINTS,SOURCE,RAP,ANNULAR
DOUBLE PRECISION DY,DBY,B,AA,PSF(100000),AIR
DOUBLE PRECISION AB(2000),LLG,LG,LGGGGG,val
DOUBLE PRECISION GGG(20000),GGG1(1000),HS(100000)
DOUBLE PRECISION GG1(1000,100),GG(1000,100),AV,PP
DOUBLE PRECISION SQF(400),A1,B1,C1,ABCD,large
DOUBLE PRECISION X1,Y1,HSS(100000),yint,slope
DOUBLE PRECISION SQQ(300),SQ(300),LGG(10),astigmatism
DOUBLE PRECISION R,U1,B,E,E8,LINEX(300),LINEY(200)
DOUBLE PRECISION AN,ARG,ITG,PL,GFX(100),GFY(100)
DOUBLE PRECISION INT1,INT2,INT3,E2,PT2,PT7,TEL
DOUBLE PRECISION T(20,20),C(20),Z(20),S8,S9
DOUBLE PRECISION PX,PY,PPX,PPY,ABCDE,ALLISON,E55
DOUBLE PRECISION S3,S2,LGGG,DELX,DELY,LGT,R11
DOUBLE PRECISION MMM,BB,AIR,JR(20000),STT,kount
DOUBLE PRECISION CX(20),CY(20),YP(20),AT,BT,str
DOUBLE PRECISION R,ZZ(20),PT,BBB,SEP,LGT,G(1000)
DOUBLE PRECISION A(300),XP(20),SEPP,DD,TURX,par
DOUBLE PRECISION MMM,IM(100000),PSEC,radius,SMR(100)
DOUBLE PRECISION hir,wir,inr,finr,inr1,E8, SMMM
DOUBLE PRECISION stretch,rot1,rot2,ast,E11,E22,str
open(12,file='fort.312',status='old')
open(14,file='fort.82',status='old')

```

```

open(16,file='fort.81',status='old')
NAME=312
C Fast JINCS
C This program calculates the position
C and brightness of planets or
C brown dwarfs orbiting brighter stars
C The data must be in the form of a 100X100
C data array that appears as in the read
C statement just above line 4 .
C HA is the CCD electron number per pixel
C GX is the X coordinate, GY is the Y coordinate
C of the pixel measured in number of pixels
C N is the counter. This data is in a separate
C file we call fortran.57
GOTO 2
1 write(513,*) IMM(N)
GOTO 870
2 print *,'runs=1,fort.',NAME
C Experimental PSF image file=62
C Actual image file=61
C If apodizing, comment out GOTO 907
C in BES(FF,B) subroutine
C RIN=1 for tiny blobs
C RIN=2 for field filling nebula
C Set value of pp
C For planet finder set PP=0.0, PL=0.0
C (.0224 +DD)*A=argument of trig function
DD=0.0
C Set psfdata=1 if using psf from psf data file This
C data could be simulated or real data. Otherwise
C fort.82 J1(r)/r squared file and use psfdarta=0
psfdata=0
C If using experimental data set DATUM=1. For
C simulated data use DATUM=0
DATUM=1
C PIXPSF is rayleigh size in pixels, 38 is usual
C 76 is double. For data put in correct psf width.
PIXPSF=90
C Set number of smooths
SMUTH=2
C If coherent source set COH=1
COH=0
C Airy disk size as fraction of 38 in final image=AIR
AIR=.2
C f61:sep=8.0,pp=.000001

```

C f65:sep=9.0 ,pp=.0000001
 C f66:sep= ,pp=
 C Number of random runs on small number
 C of data points=RUNS
 RUNS=100
 SEPP=1.5
 PSEC=.00173
 C RUNS to minimize S3
 MSEP=10
 MINS=10
 astigmatism=1.0 !either 1.0 or 0.0
 str=0.006
 stretch=1.004 !Doesn't stretch if 1.0, this is
 rot1=1.0 !rotates individual sample,doesn't rotate
 rot2=1.005 !rotates whole thing, doesn't rotate if
 E55=1.0
 C Number of globs in CCD field that
 C are being deconvoluted
 SOURCE=1
 C Lower limit on source intensity
 C as fraction of max=PL
 PL=.3
 PT=0.0
 C If first run then set CAL=1.Comment out open(14
 CAL=0
 C If reverse apodization used, set RAP=1,
 C otherwise set RAP=0
 RAP=0
 C If annular apodization used then set
 C ANNULAR=1, otherwise set ANNULAR=0
 ANNULAR=0
 C Both ANNULAR and RAP cannot be 1 at the same time
 C Set TELEAB=1 if doing telescope abberations
 C Otherwise set TELEAB=0
 C Set telescope intensity PT2 at .25 Rayleigh and PT7
 C .75 Rayleigh using actual electron# ccd output
 C Set largest value in psf =LGT
 TELEAB=0
 LGT=25000.0
 PT2=20000.0
 PT7=1800.0
 C If doing gaussian turbulence set TURB=1
 C Otherwise TURB=0
 C Set TURX= to value of "r" at which there is
 C 1/2 maximum intensity. Use units of distance in

```

C   x=3.83 or 7.66 are used for rayleigh distance
    TURB=0
    TURX=2.0
C   Set CCD HEIGHT and WIDTH and INITIAL Pixels
    HEIGHT=242
    WIDTH=375
    INITIAL=2048
    fin=92798
    wi=WIDTH
    hi=HEIGHT
    pin=INITIAL
    wir=wi*1.0
    inr=pin*1.0
    fin=(wi*hi)+pin
    FINISH(1)=(wi*hi)+pin
    FINISH(2)=pin
    FINISH(3)=wi
    FINISH(4)=hi
    fin=92798
    finr=fin*1.0
    inl=pin+1
    inr1=inl*1.0
    in=pin
    STT=1.45*TURX*TURX
    RRRRR=1
    REAP=1
    print *, 'REAP=',REAP
C   *****
C   RE for writing in by hand. Set RE=1 and comment
C   to write in. Write in pixel coordinates and
C   as read on the PC screen
    RE=0
    GOTO 5
    GX(1)=117
    GY(1)=117
    A(1)=43700.0
    GX(2)=120
    GY(2)=120
    A(2)=44200
    GX(3)=142
    GY(3)=137
    A(3)=22524
    GX(4)=133
    GY(4)=133
    A(4)=34000

```

```

S=4
DO 3 N=1,S
IF(N.GT.1)THEN
M=N-1
PX=GX(N)-GX(M)
PY=GY(N)-GY(M)
P=(PY*PY)+(PX*PX)
P=P**.5
PT=P+PT
ENDIF
IF(A(N).GT.LG)THEN
LG=A(N)
XL=GX(N)
YL=GY(N)
ENDIF
3 CONTINUE
PT=PT/((S*1.0)+1.0)
DO 4 N=1,S
LINEX(N)=(GX(N)-XL+100)*.1
LINEY(N)=(GY(N)-YL+100)*.1
4 CONTINUE

```

C *****

```

5 RIN=1
FINISH(6)=SOURCE+1
PPPP=PSEC
MM=1
kount=1.0
FINISH(9)=1
FINISH(8)=SOURCE
U=9
UUU=U
BBB=38.3/(PIXPSF*1.0)
PT2=(PT2/LGT)*.25
PT7=(PT7/LGT)*.25
AT=(7.87*PT2)-(9.62*PT7)-1.4
BT=(-2.73*PT2)+(10.0*PT7)+3.66
AIR=1.0/AIR
IF(RE.EQ.1)THEN
GOTO 63
ENDIF
LG=0.0
DO 6 SU=1,90750

```

```

C  print *, 'SU=', SU
   read(12, *) DA
C  print *, 'SU=', SU, 'DA=', DA
   HS(SU)=DA*1.0
   IMM(N)=0
   IF(SU.LT.in1)THEN
   IMM(N)=DA
   ENDIF
   IM(N)=0.0
   IF(SU.LT.in1)THEN
   HS(SU)=0.0
   ENDIF
   IF(HS(SU).GT.LG)THEN
   LG=HS(SU)
   ENDIF
6  CONTINUE
   print *, 'DO 6 complete', 'first LG=', LG
   RI(1)=1
   LLG=LG
7  LG=0.0
   NNN=1
   DO 10 N=in1, fin
   NNY=(N-pin)/wi
   X(N)=N-(in+(NNY*wi))
   Y(N)=NNY+1
C  print *, 'X(N)=', X(N), 'Y(N)=', Y(N)
C  print *, 'NNY=', NNY, 'N=', N
   IF(Y(N).LT.60)THEN
   GOTO 10
   ENDIF
   II=hi-80
   IF(Y(N).GT.II)THEN
   GOTO 10
   ENDIF
   IF(X(N).LT.70)THEN
   GOTO 10
   ENDIF
   II=0
   II=wi-75
   IF(X(N).GT.II)THEN
   GOTO 10
   ENDIF
C  print *, 'Got to IF(HS(N)..', N=', N
   IF(HS(N).GT.(PL*LLG))THEN
   NNN=NNN+1

```

```

ENDIF
NON(1)=NNN
IF(HS(N).GT.LG)THEN
LG=HS(N)
XL=N-(pin+(NNY*wi))
YL=NNY+1
ENDIF
10 CONTINUE
XXL(1)=XL
YYL(1)=YL
LGG(1)=LG
IF(NNN.GT.20000)THEN
RI(1)=2
ENDIF
print *,'NNN=',NNN
C SMOOTHING
DO 13 NN=1,SMUTH
DO 11 N=in1,fin
NYH=N
NYL=N-(2*wi)
NMN=N-wi
NXH=NMN+1
NXL=NMN-1
SS=pin+(2*wi)
YYY=((N-pin)/wi)+1
XXX=N-(((N-pin)/wi)*wi)
IF(NN.EQ.1)THEN
PSF(NMN)=HS(NMN)
PSF(N)=HS(N)
PSF(NXH)=HS(NXH)
PSF(NYH)=HS(NYH)
PSF(NXL)=HS(NXL)
PSF(NYL)=HS(NYL)
ENDIF
IF(XXX.EQ.1)THEN
HSS(NMN)=HS(NMN)
GOTO 11
ENDIF
IF(XXX.EQ.wi)THEN
HSS(NMN)=HS(NMN)
GOTO 11
ENDIF
IF(YYY.LT.2)THEN
GOTO 11
ENDIF
ENDIF

```

```

IF(YYY.EQ.2)THEN
HSS(NMN)=HS(NMN)
GOTO 11
ENDIF
S2=PSF(NXL)+PSF(NXH)+PSF(NYL)
HSS(NMN)=(S2+PSF(NYH)+PSF(NMN))/5.0
PSF(NYL)=HSS(NYL)
11 CONTINUE
13 CONTINUE
print *,'DO 11 complete'
LLL=.05*LG
GOTO 16
print *,'LG=',LG,'XL=',XL,'YL=',YL
16 SRC=SOURCE+2
SRCE=0
LGGG=100.0
SMMM=0.0
print *,'SRC=',SRC,'LGGG=',LGGG
DO 20 MM=2,SRC
print *,'SRC=',SRC
NNN=1
RI(MM)=1
DO 18 N=2049,97298
C print *, 'Got to DO 18'
NNY=(N-pin)/wi
X(N)=(N-pin)-(NNY*wi)
Y(N)=NNY+1
NN=MM-1
PX=1.0*(X(N)-XXL(NN))
PY=1.0*(Y(N)-YYL(NN))
E=(PX*PX)+(PY*PY)
E=E**.5
IF(E.LT.76.0)THEN
SMMM=SMMM+HS(N)
C print *,'SMMM=',SMMM
ENDIF
IF(Y(N).LT.70)THEN
GOTO 18
ENDIF
II=hi-72
IF(Y(N).GT.II)THEN
GOTO 18
ENDIF
IF(X(N).LT.70)THEN
GOTO 18

```

```
ENDIF
II=0
II=wi-75
IF(X(N).GT.II)THEN
GOTO 18
ENDIF
M1=MM-1
DO 17 NN=1,M1
PX=1.0*(X(N)-XXL(NN))
PX=abs(PX)
PY=1.0*(Y(N)-YYL(NN))
PY=abs(PY)
IF(PX.LT.51.0)THEN
GOTO 18
ENDIF
IF(PY.LT.51.0)THEN
GOTO 18
ENDIF
IF(HS(N).LT.LLL)THEN
GOTO 18
ENDIF
17 CONTINUE
IF(HS(N).GT.(PL*LG))THEN
NNN=NNN+1
ENDIF
NON(MM)=NNN
IF(HS(N).GT.LGGG)THEN
LGG(MM)=HS(N)
LGGG=HS(N)
XXL(MM)=(N-pin)-(NNY*wi)
YYL(MM)=NNY+1
SMR(NN)=(SMMM/LGG(NN))/1225.5
print *,'SMR(NN)=',SMR(NN), 'NN=',NN
ENDIF
18 CONTINUE
SMR(NN)=(SMMM/LGG(NN))/1225.5
print *,'SMR(NN)=',SMR(NN), 'NN=',NN
IF(LGGG.LT.(.1*LG))THEN
GOTO 21
ENDIF
SRCE=SRCE+1
FINISH(7)=SRCE
IF(NNN.GT.20000)THEN
RI(MM)=2
ENDIF
```

```
20 CONTINUE
   print *,'SMR(1)=' ,SMR(1)
   print *,'SMR(2)=' , SMR(2)
21 SRCE=FINISH(7)+1
C STOP
C *****
C START LOOPS
  SRCE=SOURCE
  DO 700 MMMM=1,SRCE
    MM=(SRCE-MMMM)+1
    U=UUU
    XL=XXL(MM)
    YL=YYL(MM)
    RIN=RI(MM)
    PSEC=PPPP
    PP=PSEC
    S2=100000.0
    S3=10000000000000.0
    SEP=SEPP
23 DO 243 NTT=0,MSEP
C SEP=SEP-1.0
C ARG=.0224*A
  GGH=NTT
  DD=.000001*GGH
  GGGG=-1*INT((.1*GGH)+.5)
  PP=PSEC
  ALLISON=DD+.0224
  DO 242 NT=1,MINS
    IF(DATUM.EQ.1)THEN
      GOTO 24
    ENDIF
C Data angle sample settings for normal aperture
24 IF(PP.GT.1.0)THEN
  IF(S3.GT.1000000.0)THEN
    XP(1)=10.0
    YP(1)=10.0
    ZZ(1)=1.0
    IF(N.GT.1)THEN
      ZZ(N)=0.0
    ENDIF
  ENDIF
  GOTO 242
  PP=PSEC
  ENDIF
25 PP=PP*5.0
```

```

ENT=1
PT=0.0
S=1
RR1=1
DO 27 NN=1,1000
DO 26 N=1,100
GG1(NN,N)=0.0
GG(NN,N)=0.0
26 CONTINUE
27 CONTINUE
DO 38 R1=1,69,RIN
M=1
LN=1
IF(R1.EQ.1)THEN
RTT=90
ENDIF
IF(R1.EQ.2)THEN
RTT=26
ENDIF
IF(R1.EQ.3)THEN
RTT=18
ENDIF
R11=1.0/R1
E11=(1.0-ROT1)*MSEP
E22=ROT2
R11=E22*atan(R11)
R11=R11*180.0/3.14159
IF(R1.GT.3)THEN
RTT=INT(R11+.5)
ABC=ABC/3
ENDIF
DO 31 AN=1,360,RTT
ast=MSEP*.314*astigmatism
E=(NT-5)*(str)
X1=1.0*R1*cos((.017453*AN)+E11)
Y1=1.0*R1*sin((.017453*AN)+E11)
PX=(X1*cos(ast))-(Y1*sin(ast))
PY=((X1*E55*sin(ast))+(Y1*cos(ast)))*(stretch-E)
X1=(PX*cos(-ast))-(PY*sin(-ast))
Y1=(PX*sin(-ast))+(PY*cos(-ast))
X1=X1+.5
Y1=Y1+.5
XX1=INT(X1)
XX=XL+XX1
YY1=INT(Y1)

```

```

YY=YY1+YL
DA=(YY-1)*wi
N=((YY-1)*wi)+XX+pin
GGG(M)=HSS(N)
GGG1(M)=HS(N)
GGGX(M)=XX
GGGY(M)=YY
C print *,'Y(N)=',Y(N),'GGGY(M)=',GGGY(M),'M=',M
  IF(M.LT.3)THEN
    GOTO 30
  ENDIF
  M1=M-1
  M2=M-2
  IF(GGG(M).LT.GGG(M1))THEN
    GOTO 28
  ENDIF
  GOTO 30
28 IF(GGG(M2).LT.GGG(M1))THEN
  GG(RR1,LN)=GGG(M1)
  GGX(RR1,LN)=GGGX(M1)
  GGY(RR1,LN)=GGGY(M1)
  GG1(RR1,LN)=GGG1(M1)
C print
  LN=LN+1
  ENDIF
C print *,'LN=',LN
30 M=M+1
C print *,'Got past line 30','M=',M
31 CONTINUE
  LMM=LN
  NO(RR1)=LMM
C print *,'DO loop 31 complete','R1=',R1
32 RH1=RR1
  RM1=RR1-1
  RL1=RR1-2
  RR1=RR1+1
  IF(RR1.LT.4)THEN
    GOTO 38
  ENDIF
  DO 37 LNN=1,LMM
    PPP=1000.0
    NNN=NO(RM1)
    DO 33 N=1,NNN
      PX=(GGX(RM1,N)-GGX(RH1,LNN))*1.0
      PY=(GGY(RM1,N)-GGY(RH1,LNN))*1.0

```

```
P=(PX*PX)+(PY*PY)
P=P**.5
IF(P.LT.PPP)THEN
N1=N
PPP=P
P1=P
ENDIF
33 CONTINUE
dist=RIN*SEP
PPP=10000.0
NNN=NO(RL1)
DO 35 N=1,NNN
PX=(GGX(RL1,N)-GGX(RM1,N1))*1.0
PY=(GGY(RL1,N)-GGY(RM1,N1))*1.0
P=(PX*PX)+(PY*PY)
P=P**.5
IF(P.LT.PPP)THEN
N2=N
P2=P
PPP=P
ENDIF
35 CONTINUE
IF(P1.GT.dist)THEN
GOTO 37
ENDIF
IF(P2.GT.dist)THEN
GOTO 37
ENDIF
36 IF(GG(RL1,N2).LT..00000000001)THEN
GOTO 37
ENDIF
IF(GG(RH1,N1).LT..00000000001)THEN
GOTO 37
ENDIF
DIF=(P1+P2)/2.0
DIF=DIF+(.1*DIF*DIF)
AV=(GG(RH1,LNN)+GG(RL1,N2))/2.0
AV=AV+(PP*RIN*LG*DIF)
IF(GG(RM1,N1).LT.AV)THEN
GOTO 37
ENDIF
IF(GG(RM1,N1).GT.(.3*LG))THEN
G(S)=GG1(RM1,N1)
GX(S)=GGX(RM1,N1)
GY(S)=GGY(RM1,N1)
```

```
PT=PT+DIF
C print *,'G(S)='G(S), 'GY(S)='GY(S),'GX(S)='GX(S)
C print *,'S='S,' PP='PP
  S=S+1
  ENDIF
37 CONTINUE
C print *,'S='S,' PP='PP
38 CONTINUE
  print *,'S='S,' PP='PP','SEP='SEP
  IF(S.GT.43)THEN
    GOTO 25
  ENDIF
  IF(S.LT.2)THEN
    GOTO 242
  ENDIF
  IF(S.LT.27)THEN
    GOTO 42
  ENDIF
  GOTO 43
42 IF(S5.GT.26)THEN
  GOTO 242
  ENDIF
43 PT=PT/((S*1.0)+1.0)
  S5=S
  DO 45 N=1,S
    SQ(N)=G(N)
    SQX(N)=GX(N)
    SQY(N)=GY(N)
C print *,'SQY(N)='SQY(N),'SQX(N)='SQX(N)
45 CONTINUE
  DO 55 M=1,S
    LL=1000000000.0
    DO 50 N=1,S
      IF(SQ(N).LT.LL)THEN
        SQQ(M)=SQ(N)
        SQQX(M)=SQX(N)
        SQQY(M)=SQY(N)
C print *,'SQQX(M)='SQQX(M),'SQQY(M)='SQQY(M)
        LL=SQ(N)
        NN=N
      ENDIF
50 CONTINUE
    SQ(NN)=1000000000.0
55 CONTINUE
  M=1
```

```
DO 60 N=1,S,2
IF(SQQ(N).LT..01)THEN
GOTO 60
ENDIF
SQF(M)=SQQ(N)
SQFX(M)=SQQX(N)
SQFY(M)=SQQY(N)
A(M)=SQF(M)
XXS(M)=SQFX(M)-XL+100
YYS(M)=SQFY(M)-YL+100
LINEX(M)=XXS(M)*.1
LINEY(M)=YYS(M)*.1
M=M+1
60 CONTINUE
XXS(M)=100
YYS(M)=100
LGGGGG=LG+.5
A(M)=INT(LGGGGG)
IF(A(M).LT.1)THEN
GOTO 61
ENDIF
LINEX(M)=10.0
LINEY(M)=10.0
M=M+1
61 DO 62 N=2,S,2
IF(SQQ(N).LT..001)THEN
GOTO 62
ENDIF
SQF(M)=SQQ(N)
SQFX(M)=SQQX(N)
SQFY(M)=SQQY(N)
XXS(M)=SQFX(M)-XL+100
YYS(M)=SQFY(M)-YL+100
A(M)=SQF(M)
LINEX(M)=XXS(M)*.1
LINEY(M)=YYS(M)*.1
M=M+1
62 CONTINUE
S=M-1
63 DO 64 SS=1,S
print *,'LINEX(M)=' ,LINEX(SS),'LINEY(M)=' ,LINEY(SS)
print *,'A(M)=' ,A(SS),'Allison=' ,ALLISON
print *,'SMR(MMMM)=' ,SMR(MMMMMM),'MMMM=' ,MMMM
64 CONTINUE
in=pin
```

```

print *,'S=Number of points on line=',S
print *,'S=limit of DO 5 read statement in ridge2'
in=S/9
IF(in.GT.5)THEN
in=5
ENDIF
NN=1
nim=1
DO 70 SS=1,10
DO 69 ST=1,2
NO(NN)=nim+(ST-2)
IF(SS.EQ.1)THEN
GOTO 65
ENDIF
GOTO 66
65 IF(ST.EQ.1)THEN
GOTO 69
ENDIF
66 IF(SS.EQ.10)THEN
GOTO 67
ENDIF
GOTO 68
67 IF(ST.EQ.2)THEN
GOTO 71
ENDIF
68 print *,'NO(NN)=',NO(NN)
SU=NO(NN)
NN=NN+1
69 CONTINUE
nim=nim+in
70 CONTINUE
S=SU
71 SU=1
C Put in for spherical sampling
C keep tests along lines
C DO 75 NN=1,8
C E=(2*3.14159)/8.0
C U1=20.0*cos(E*NN)
C XX=INT(U1+.5)
C CX(NN)=(XX*1.0)+(XL*1.0)
C CY(NN)=((20*20)-(XX*XX))*1.0
C 75 CONTINUE
C C(9)=LG
C CX(9)=LX
C CY(9)=LY

```

```
print *,'in=',in,'Nim=',nim
in=pin
IF(psfdata.EQ.1)THEN
GOTO 84
ENDIF
IF(MMMM.GT.1)THEN
GOTO 143
ENDIF
IF(RRRRR.GT.1)THEN
GOTO 143
ENDIF
GOTO 143
C Comment out next line if still using simulation psf
C Change E to DA when using real data
C DA=0
84 N=1
85 E=0.0
IF(RRRRR.GT.1)THEN
GOTO 89
ENDIF
read(16,*) E
PSF(N)=E*1.0
89 N=N+1
IF(N.GT.92798)THEN
GOTO 90
ENDIF
GOTO 85
RRRRR=RRRRR+1
90 LG=0.0
RRRRR=RRRRR+1
DO 100 N=in1,fin
NNY=(N-pin)/wi
X(N)=N-((NNY*wi)+in)
Y(N)=NNY+1
E2=PSF(N)
IF(Y(N).LT.60)THEN
GOTO 100
ENDIF
II=hi-60
IF(Y(N).GT.II)THEN
GOTO 100
ENDIF
IF(X(N).LT.122)THEN
GOTO 100
ENDIF
ENDIF
```

```

II=0
II=wi-135
IF(X(N).GT.II)THEN
GOTO 100
ENDIF
IF(E2.GT.LG)THEN
LG=E2
C print *,'LG=',LG,'X(N)=' ,X(N)
XL=N-(in+(NNY*wi))
YL=NNY+1
ENDIF
E2=0.0
100 CONTINUE
print *,'LG=',LG,'XL=',XL,'YL=',YL
C If not using linear interpolation remove
C "C" in next line
C GOTO 110
SU=1
107 DO 109 XX=1,200
N=((YL-1)*wi)+XL+XX+in
AB(XX)=PSF(N)
NNY=(N-pin)/wi
XXX=XX-1
slope=1.0*(AB(XX)-AB(XXX))
yint=((AB(XX)*1.0)-(slope*XX))*1.0
IF(XX.GT.120)THEN
AB(XX)=0.0
ENDIF
DO 108 NN=1,100
PX=(XX*1.0)+(NN*.01)
E=(slope*PX)+yint
GGG(SU)=0.0
GGG(SU)=(E/LG)*.25
SV=SU-2
ST=SU-1
IF(SU.GT.3)THEN
JR(ST)=(GGG(SV)+GGG(ST)+GGG(SU))/3.0
ENDIF
SU=SU+1
C print *,'ST=',ST,'JR(ST)=' ,JR(ST)
108 CONTINUE
C print *,'AB(XX)=' ,AB(XX),'LG=',LG
109 CONTINUE
JR(1)=(AB(1)/LG)*.25
JR(2)=(AB(1)/LG)*.25

```

```

C STOP
GOTO 143
110 DO 111 XX=1,200
N=((YL-1)*wi)+XL+XX+pin
AB(XX)=PSF(N)
111 CONTINUE
NNN=1
DO 120 N=3,197
SS=N+2
ST=N+1
SU=N-1
SV=N-2
DY=AB(SV)-(4*AB(SU))+(6*AB(N))
DY=DY-(4*AB(ST))+AB(SS)
A1=AB(N)-((3.0/35.0)*DY)
DBY=(-2*AB(SV))-AB(SU)+AB(ST)
B1=(DBY+(2*AB(SS)))/10.0
CCCCY=AB(SS)+AB(ST)+AB(N)+AB(SU)+AB(SV)
C1=(CCCCY-(5*A1))/10.0
DO 115 NN=-50,49
X1=NN*.01
NNN=(N*100)+NN
E=A1+(B1*X1)+(C1*X1*X1)
GGG(NNN)=0.0
GGG(NNN)=(E/LG)*.25
SS=NNN-2
ST=NNN-1
IF(NNN.GT.3)THEN
JR(ST)=(GGG(SS)+GGG(ST)+GGG(NNN))/3.0
ENDIF
E1=(E/LG)*.25
C write(82,*) E1
C print *,'NNN=',NNN,' JR(NNN)=' ,JR(NNN)
115 CONTINUE
120 CONTINUE
JR(1)=(AB(1)/LG)*.25
JR(2)=(AB(1)/LG)*.25
JR(3)=(AB(1)/LG)*.25
print *,'PSF done'
C stop
143 RRRRR=RRRRR+1
IF(psfdata.EQ.1)THEN
GOTO 203
ENDIF
IF(ENT.GT.1)THEN

```

```
GOTO 203
ENDIF
IF(REAP.GT.1)THEN
GOTO 203
ENDIF
DO 202 III=1,20000
RRRRR=RRRRR+1
C Get rid of next 2 lines when
C using instrument PSF. Add read(16,*) line
IF(CAL.EQ.0)THEN
GOTO 150
ENDIF
CALL BES(R,III)
write(82,*) R
C print *,'III=',III,' BES=',R
IF(psfdata.EQ.1)THEN
BBB=1.0
ENDIF
150 II=INT((III*BBB)+.5)
C Comment out next line
C II=III
IF(II.GT.12000)THEN
JR(II)=0.0
GOTO 202
ENDIF
IF(CAL.EQ.1)THEN
GOTO 155
ENDIF
156 IF(REAP.GT.1)THEN
GOTO 160
ENDIF
read(14,*) R
155 JR(II)=R
160 TTTTTT=1
202 CONTINUE
C stop
print *,'REAP=',REAP
203 REAP=REAP+1
print *,'S3=',S3,'S2=',S2,'NNN=',NON(MM)
C S2=10000.0
C S3=10000000000000000.0
IF(S.LT.2)THEN
GOTO 204
ENDIF
GOTO 207
```

```

204 fin=FINISH(1)
    DO 206 N=in1,fin
        IMM(N)=0.0
206 CONTINUE
    N=(wi*(YL-1))+XL+pin
    IMM(N)=LG
    GOTO 720
    REAP=REAP+1
207 IF(S.LT.9)THEN
    U=S
    GOTO 210
    ENDIF
    N1B=NO(1)
    N1E=NO(2)
    N2B=NO(3)
    N2E=NO(4)
    N3B=NO(5)
    N3E=NO(6)
    N4B=NO(7)
    N4E=NO(8)
    N5B=NO(9)
    N5E=NO(10)
    N6B=NO(11)
    N6E=NO(12)
    N7B=NO(13)
    N7E=NO(14)
    N8B=NO(15)
    N8E=NO(16)
    N9B=NO(17)
    N9E=NO(18)
C Our set of simultaneous equations
C is solved similar to that below for
C K1 and K2 given C(XI)
C
C           N       N
C           |       |
C           V       V
C C(X1) =  K1F(X1-P1) + K2F(X1-P2)  <----I
C C(X2) =  K1F(X2-P1) + K2F(X2-P2)
    DO 239 N9=N9B,N9E
    DO 238 N8=N8B,N8E
    DO 237 N7=N7B,N7E
    DO 236 N6=N6B,N6E
    DO 235 N5=N5B,N5E
    DO 234 N4=N4B,N4E

```

```

DO 233 N3=N3B,N3E
DO 232 N2=N2B,N2E
DO 231 N1=N1B,N1E
NNO(1)=N1
NNO(2)=N2
NNO(3)=N3
NNO(4)=N4
NNO(5)=N5
NNO(6)=N6
NNO(7)=N7
NNO(8)=N8
NNO(9)=N9
C For planet finder
C LINEX(1)=10.0
C LINEY(1)=10.0
C A(1)=LGG(MM)
210 POINTS=9
ST=1
IF(nim.GT.26)THEN
RUNS=1
ST=0
ENDIF
IF(S.LT.9)THEN
POINTS=S
ENDIF
FINISH(5)=POINTS
DO 228 NPP=1,RUNS
DO 215 I=1,POINTS
DO 214 N=1,POINTS
NN=NNO(N)
II=NNO(I)
IF(S.LT.9)THEN
NN=N
II=I
ENDIF
C(I)=A(II)/LG
X1=30054432160.3245345123*II*NN*N+(NPP*NPP*5.16432*I)
X1=X1+129383.3482372*NPP*NPP+1.234*NPP+I*4.2
QX1=cos(X1)
QX2=abs(QX1)
RANDX=0.0
RANDX=abs(QX2)*QX1/(QX2+.0000000001)
PX=0.0
DELX=0.0
DELX=0.16*RANDX*ST*2*((RUNS-1)/(RUNS*1.0))

```

```

DELX=(.1*INT((DELX*10.0)+.5))+.00000001
SQ(NN)=LINEX(NN)+DELX
PX=SQ(NN)
Y1=9067475667050.32423432*PI*I*N*NPP+(NPP*2.16932*NN)
Y1=Y1+43029.3213*NPP*NPP+N*234.2
QY1=cos(Y1)
QY2=abs(QY1)
RANDY=0.0
RANDY=abs(QY2)*QY1/(QY2+.0000000001)
PY=0.0
DELY=0.0
DELY=0.16*RANDY*ST*2*((RUNS-1)/(RUNS*1.0))
DELY=(.1*INT((DELY*10.0)+.5))+.00000001
SQQ(NN)=LINEY(NN)+DELY
PY=SQQ(NN)
CX(I)=LINEX(II)+DELX*(1-RE)
CY(I)=LINEY(II)+DELY*(1-RE)
JJ=((YL-101)+(CY(I)*10))*wi
JJ=JJ+((CX(I)*10)+(XL-100))+in
C(I)=(A(II)*RE/LG)+(HSS(JJ)*(1-RE)/LG)
C C(I)=A(II)/LG
C PX=LINEX(NN)
C PY=LINEY(NN)
RRR=0.0
RRR=(CY(I)-PY)*(CY(I)-PY)
C print *,'RRR=',RRR
T(I,N)=0.0
RRR=((CX(I)-PX)*(CX(I)-PX))+RRR
RRR=RRR**.5
radius=RRR
RRR=(999.0*RRR)+1.001
SS=0
SS=INT(RRR)
E=JR(SS)
IF(COH.EQ.1)THEN
A1=C(I)
A1=A1**.5
E=E**.5
C(I)=A1
ENDIF
C print *,'JR(SS)=' ,JR(SS)
TEL=(AT*radius)+(BT*radius*radius)
IF(TELEAB.EQ.0)THEN
TEL=1.0
ENDIF

```

```

E=E*TEL
IF(TURB.EQ.1)THEN
E=.25*exp(-radius*radius/STT)
ENDIF
T(I,N)=E
C IF(SEP.GT.6.0)THEN
C print *,'JR(SS)=' ,JR(SS),'RRR=' ,RRR,'SS=' ,SS
C print *,'I=' ,I,'C(I)=' ,C(I)
C print *,'I=' ,I,'CX(I)=' ,CX(I),'CY(I)=' ,CY(I)
C print *,'N=' ,N,'PX=' ,PX,'PY=' ,PY
C print *,'T(I,N)=' ,T(I,N)
C ENDF
214 CONTINUE
215 CONTINUE
C stop
Z(1)=0.0
Z(2)=0.0
Z(3)=0.0
Z(4)=0.0
Z(5)=0.0
Z(6)=0.0
Z(7)=0.0
Z(8)=0.0
Z(9)=0.0
220 CALL GAUSS(T,C,U)
CALL BSOLVE(T,Z,C,U)
S2=0.0
val=0.0
DO 222 N=1,POINTS
S2=S2+abs(Z(N))
val=val+Z(N)
C print *,'Z(N)=' ,Z(N)
222 CONTINUE
E=SMR(MM)*2.0
C print *,'SMR(MM)=' ,SMR(MM)
IF(val.LT.E)THEN
IF(ENT.GT.5)THEN
GOTO 245
ENDIF
ENDIF
C IF(SEP.GT.6.0)THEN
C print *,'S3=' ,S3,'S2=' ,S2,'SEP=' ,SEP,'PP=' ,PP
C ENDF
C print *,'S2=' ,S2,'S3=' ,S3
223 DO 224 N=1,9

```

```
IF(abs(Z(N)).GT.200.0)THEN
GOTO 228
ENDIF
IF(Z(N).LT.-20.0)THEN
GOTO 228
ENDIF
IF(S2.GT..0001)THEN
GOTO 224
ENDIF
GOTO 228
224 CONTINUE
S4=S3
IF(S2.GT.S3)THEN
GOTO 228
ENDIF
S3=S2
DO 225 N=1,POINTS
NN=NNO(N)
IF(S.LT.9)THEN
NN=N
ENDIF
XP(N)=SQ(NN)
YP(N)=SQQ(NN)
ZZ(N)=Z(N)
IF(COH.EQ.1)THEN
ZZ(N)=ZZ(N)*ZZ(N)
ENDIF
GFX(N)=100*(((XL-100)*.1)+XP(N))
GFY(N)=100*(((YL-100)*.1)+YP(N))
PPX=GFX(N)*.01
PPY=GFY(N)*.01
print *, 'S3=',S3, ' SMR(MM)=' ,SMR(MM), 'MM=' ,MM
225 CONTINUE
print *, 'NAME=' ,NAME
228 CONTINUE
ENT=ENT+1
IF(S.LT.9)THEN
GOTO 242
ENDIF
231 CONTINUE
232 CONTINUE
233 CONTINUE
234 CONTINUE
235 CONTINUE
236 CONTINUE
```

```
237 CONTINUE
238 CONTINUE
239 CONTINUE
242 CONTINUE
243 CONTINUE
245 S=S
C DO 246 N=1,92798
C IM(N)=0.0
C print *,'IM(N)=',IM(N)
C246 CONTINUE
    DO 246 SU=1,finn
        IM(SU)=0.0
246 CONTINUE
    M=1
    large=0.0
    DO 248 N=1,POINTS
        PPX=GFX(N)*.01
        PPY=GFY(N)*.01
        IF(S2.LT..00001)THEN
            XP(1)=10.0
            YP(1)=10.0
            ZZ(1)=1.0
            S3=S4
            ENDIF
            IF(SEP.GT.9.0)THEN
                XP(1)=10.0+XXL(MM)
                YP(1)=10.0+YYL(MM)
                ZZ(1)=1.0
            IF(N.GT.1)THEN
                ZZ(N)=0.0
            ENDIF
            GOTO 247
            ENDIF
            IF(S3.GT.1000000.0)THEN
                XP(1)=10.0
                YP(1)=10.0
                ZZ(1)=1.0
            IF(N.GT.1)THEN
                ZZ(N)=0.0
            ENDIF
            ENDIF
            IF(S3.GT.100000.0)THEN
                IF(PP.GT..5)THEN
                    GOTO 247
                ENDIF
```

```

GOTO 251
ENDIF
IF(S3.GT.100000.0)THEN
M=1
IF(PP.GT..5)THEN
GOTO 247
ENDIF
GOTO 251
ENDIF
247 IF(ZZ(N).GT.large)THEN
large=ZZ(N)
X1=PPX
Y1=PPY
ENDIF
write(507,*) 'X=',XP(N),'Y=',YP(N),'K',N,'=',ZZ(N)
C print *,'X=',PPX,'Y=',PPY,'K',N,'=',ZZ(N)
E=((YYL(MM)-101)*1.0)+(YP(N)*10))*375.0
E=E+(XP(N)*10)+((XXL(MM)-100)*1.0)+2048.0
NN=INT(E+.5)
finn=fin
IM(NN)=ZZ(N)*LGG(MM)
G(N)=ZZ(N)*LGG(MM)
NNY=(NN-pin)/wi
GX(N)=NN-((NNY*wi)+pin)
GY(N)=NNY+1
C print *,'G(N)=',G(N)
C write(507,*) E
C print *,'NN=',NN,'IM(NN)=',IM(NN),'N=',N
248 CONTINUE
X1=XP(1)
Y1=XP(1)
DO 250 N=1,POINTS
IF(SMR(MM).LT.1.0)THEN
ZZ(1)=large
ZZ(2)=0.0
ZZ(3)=0.0
ZZ(4)=0.0
ZZ(5)=0.0
ZZ(6)=0.0
ZZ(7)=0.0
ZZ(8)=0.0
ZZ(9)=0.0
ENDIF
E=SMR(MM)

```

```

IF(E.LT.1.1)THEN
XP(1)=10.0
YP(1)=10.0
ENDIF
IF(S3.GT.1000000.0)THEN
XP(1)=10.0
YP(1)=10.0
ZZ(1)=1.0
IF(N.GT.1)THEN
ZZ(N)=0.0
ENDIF
ENDIF
G(N)=(ZZ(N)/large)*LGG(MM)
249 E=(((YYL(MM)-101)*1.0)+(YP(N)*10))*375.0
E=E+(XP(N)*10)+((XXL(MM)-100)*1.0)+2048.0
NN=INT(E+.5)
IM(NN)=(ZZ(N)/large)*LGG(MM)*30000/LGG(1)
NNY=(NN-2048)/375
PY=.1*(NNY+1)
PX=.1*(NN-(((NN-2048)/375)*375)+2048))
print *,'X=',PX,'Y=',PY,'K',N,'=',ZZ(N)
print *,' IM=',IM(NN)
250 CONTINUE
M=M-1
GOTO 252
251 SEP=SEP+.5
PSEC=PSEC
print *,'S3=',S3,'PSEC=',PSEC
GOTO 24
252 E=SOURCE*.9
C kount=1
E=SOURCE*.9
SOURCE=FINISH(8)
IF(kount.GT.E)THEN
IF(S3.GT.1000000.0)THEN
DO 698 N=1,POINTS
XP(1)=10.0
YP(1)=10.0
ZZ(1)=1.0
IF(N.GT.1)THEN
ZZ(N)=0.0
ENDIF
GOTO 248
698 CONTINUE
ENDIF

```

```

print *, 'This is the output from fort.', NAME
IF(S3.GT.100000.0)THEN
E=((YYL(MM)-101)*1.0)+(YP(N)*10))*375.0
E=E+(XP(N)*10)+((XXL(MM)-100)*1.0)+2048.0
NN=INT(E+.5)
IM(NN)=(ZZ(N)/large)*LGG(MM)*30000/LGG(1)
NNY=(NN-2048)/375
PY=.1*(NNY+1)
PX=.1*(NN-(((NN-2048)/375)*375)+2048))
print *, 'X=', PX, 'Y=', PY, 'K', N, '=', ZZ(N)
ENDIF
GOTO 710
ENDIF
kount=kount+1.0
C FINISH(9)=kount
print *, 'This is the output from fort.', NAME
IF(S3.GT.100000.0)THEN
E=((YYL(MM)-101)*1.0)+(YP(N)*10))*375.0
E=E+(XP(N)*10)+((XXL(MM)-100)*1.0)+2048.0
NN=INT(E+.5)
IM(NN)=(ZZ(N)/large)*LGG(MM)*30000/LGG(1)
NNY=(NN-2048)/375
PY=.1*(NNY+1)
PX=.1*(NN-(((NN-2048)/375)*375)+2048))
print *, 'X=', PX, 'Y=', PY, 'K', N, '=', ZZ(N)
ENDIF
IF(S3.GT.1000000.0)THEN
DO 699 N=1, POINTS
XP(1)=10.0
YP(1)=10.0
ZZ(1)=1.0
IF(N.GT.1)THEN
ZZ(N)=0.0
ENDIF
GOTO 248
699 CONTINUE
ENDIF
IF(RE.EQ.1)THEN
GOTO 710
ENDIF
700 CONTINUE
710 M=1
DO 720 N=1, 92798
E=IM(N)+.5
IF(E.GT..6)THEN

```

```

G(M)=IM(N)
NNY=(N-pin)/wi
GX(M)=N-((NNY*wi)+pin)
GY(M)=NNY+1
C print *,'G(M)=',G(M)
M=M+1
C write(507,*) E
ENDIF
720 CONTINUE
N3=M-1
M=0
721 print *,'POINTS=',POINTS
N1=FINISH(4)
N2=FINISH(3)
NN=in1
DO 850 NYL=1,242
DO 840 NXL=1,375
X1=((NXL-NYL)*.1)+16.9
Y1=NYL*.1
E=0.0
DO 820 N=1,N3
PX=GX(N)*.1
PY=GY(N)*.1
AA=0.0
AA=G(N)
C print *,'AA=',AA,'PX=',PX,'PY=',PY,'POINTS=',POINTS
R=(X1-PX)*(X1-PX)
R=((Y1-PY)*(Y1-PY))+R
R=(R**.5)*.5*15.0
R=(999.0*R)+1.001
SS=0
SS=INT(R)
E8=JR(SS)*AA
C print *,'E8=',E8,'SS=',SS,'JR(SS)=' ,JR(SS),'AA=',AA
IF(SS.GT.20000)THEN
E8=0.0
ENDIF
E=E+E8
820 CONTINUE
E=E+.5
NNY=NYL-1
NN=(NNY*375)+NXL+2048
IMM(NN)=INT(E)
C print *,'PX=',PX,'PY=',PY,'IMM(NN)=' ,IMM(NN)
840 CONTINUE

```

```

850 CONTINUE
    N1=FINISH(1)
    DO 870 N=1,92798
        NNY=(N-2048)/375
        PX=1.0*(N-((NNY*375)+2048))
        PY=1.0*(NNY+1)
    C   print *,'PX=',PX,'PY=',PY,'IMM(N)=' ,IMM(N)
        GOTO 1
870 CONTINUE
900 STOP
    END
    SUBROUTINE BES(FF,B)
    INTEGER o,B,UU
    DOUBLE PRECISION ARG,AN
    DOUBLE PRECISION INT1,INT2,INT3,ITG
    DOUBLE PRECISION J1,k1,k2,k3,RR,FF
    DOUBLE PRECISION ua,u2,u4,u6,u8,ku
    C   If apodizing, comment out next line
        GOTO 907
        UU=INT(B+.5)
        ua=.001*B
        u2=ua*ua
        u4=ua*ua*ua*ua
        u6=ua*ua*ua*ua*ua*ua
        u8=u4*u4
        k1=(sin(ua)/ua)*((2.0/u4)-(36.0/u6))
        k2=(cos(ua)/u2)*((14.0/u4)-(36.0/u6))
        k3=(.5/u4)+(4.0/u6)+(36.0/u8)
        ku=288.0*(k1+k2+k3)
        IF(UU.LT.49)THEN
            ku=1.0
        ENDIF
    C   JR(UU)=ku*ku
        FF=ku*ku*.25
    C   print *,'u=',u,'ku2=',ku
905 CONTINUE
        GOTO 911
907 RR=B*.001
        J1=0.0
        DO 910 o=-50,50
            AN=o*.0314159
            ARG=RR*sin(AN)
            INT1=cos(ARG)*cos(AN)*cos(AN)
            AN=AN+.015708
            ARG=RR*sin(AN)

```

```

INT2=cos(ARG)*cos(AN)*cos(AN)*4.0
AN=AN+.015708
ARG=RR*sin(AN)
INT3=cos(ARG)*cos(AN)*cos(AN)
ITG=INT1+INT2+INT3
J1=ITG+J1
910 CONTINUE
FF=J1*J1/360000.0
911 FF=FF
RETURN
END
SUBROUTINE BSOLVE(V,HI,D,Q)
INTEGER I,J,Q
DOUBLE PRECISION V(20,20), HI(20), D(20),SUM
DO 200 I=Q, 1,-1
SUM =D(I)
DO 100 J=I+1,Q
SUM=SUM-V(I,J)*HI(J)
100 CONTINUE
HI(I)=SUM/(V(I,I)+.00000000001)
C print *,'HI=',HI,'V(I,I)=' ,V(I,I)
200 CONTINUE
RETURN
END
SUBROUTINE GAUSS(L,M,Q)
INTEGER J,K,PIVOT,I,INDEX,Q
DOUBLE PRECISION TEMP,L(20,20),M(20),RATIO,ABS
DO 100 I=1,Q
INDEX=I
PIVOT=INT(ABS(L(I,I)))
DO 200 J=I+1,Q
IF (ABS(L(J,I)) .GT.PIVOT) THEN
PIVOT=INT(ABS(L(J,I)))
INDEX=J
ENDIF
200 CONTINUE
IF (INDEX.GT.I) THEN
DO 400 K=I,Q
TEMP=L(I,K)
L(I,K)=L(INDEX,K)
L(INDEX,K)=TEMP
400 CONTINUE
TEMP=M(I)
M(I)=M(INDEX)
M(INDEX)=TEMP

```

```
ENDIF
DO 300 J=I+1,Q
RATIO=L(J,I)/(L(I,I)+.00000000001)
DO 500 K=I+1,Q
L(J,K)=L(J,K)-L(I,K)*RATIO
500 CONTINUE
M(J)=M(J)-M(I)*RATIO
300 CONTINUE
100 CONTINUE
RETURN
END
```

APPENDIX B

Data transfer code

```

C  In this appendix we convert from dos data to
C  unix data file.
C  Thus the images taken on the ccd using the PC will be
C  transfered to the sun unix system.
character cdata(250000)*1  ! raw data
character cheader(50)*42      !
C  file header lines
character*40 fname
character charin, CR, LF
integer fgetc, status
integer*2 ilow, ihigh
integer mdata(150000),ccdata(1000000)  ! data file
write(*,*) "Enter name of IMAGE file to process"
read(*,*) fname
open(unit=9, file=fname, form='unformatted')
CR = CHAR(13)
LF = CHAR(10)
nline = 1
np = 1
do i=1,2048
status = fgetc(9, charin)
if (cheader(nline)(1:3) .ne. 'End' ) then
if (charin .ne. CR .and. charin .ne. LF ) then
cheader(nline)(np:np) = charin
np = np + 1
elseif (charin .eq. CR ) then
cheader(nline)(np:np) = '\n'
nline = nline + 1
np = 1
endif
endif
end do
write(*,*) (cheader(j),j=1,nline)
C  now get the integer data
idata_no = 0
C  do i = 2049, 400000
do while ( status .eq. 0)
status = fgetc(9, charin)
idata_no = idata_no + 1
cdata(idata_no) = charin

```

```

end do
idata_no = idata_no -1    ! account for reading
C EOF
write(*,('No of bytes read is ', 2i10)) idata_no
if ( idata_no mod 2 .ne. 0 ) then
write(*,*) "Error in no. of data points odd no read
C BYE"
stop
endif
C now convert data
k = 0
do i = 1, idata_no, 2
C do i = 1, 100, 2
k = k + 1
ilow = cdata(i)
if ( ilow .lt. 0 ) then
ilow = ilow + 256
endif
if ( ihigh .lt. 0 ) then
ihigh = ihigh + 256
endif
ihigh = cdata(i+1)
mdata(k) = ilow + 256 * ihigh
C write(*,('2z4.3, 3i6'))
C 1 cdata(i), cdata(i+1), ilow, ihigh, mdata(k)
end do
DO 5 N=1,90750
M=N+2048
ccdata(M)=mdata(N)
5 CONTINUE
DO 6 N=1,2048
ccdata(N)=0
6 CONTINUE
DO 10 N=1,92798
write(315,*) ccdata(N)
10 CONTINUE
C write(*,('No of data pts is ', i5)) k
stop
end
Program Change
character cdata(250000)*1
character charin    ! raw data
character*40 fname1
character*40 fname2
integer fputc, status

```

```
integer*2 ilow, ihigh
integer mdata(150000)
write(*,*) "Enter name of input file"
read(*,*) fname1
print *,'92798=data point number'
write(*,*)"Enter number of data points"
read(*,*) idata_no
write(*,*)"Enter name of output file"
read(*,*) fname2
open(unit=8,file=fname1)
open(unit=9,file=fname2,form='unformatted')
do k=1,idata_no
read(8,*) mdata(k)
end do
i=0
DO k=1,idata_no
ihigh=mdata(k)/256
ilow=mdata(k)-256*ihigh
IF(ilow.GT.128)THEN
ilow=ilow-256
ENDIF
IF(ihigh.GT.128)THEN
ihigh=ihigh-256
ENDIF
i=i+1
cdata(i)=char(ilow)
i=i+1
cdata(i)=char(ihigh)
END DO
DO i=1,2*idata_no
charin=cdata(i)
status=fputc(9,charin)
END DO
stop
end
```

APPENDIX C

PLANET FINDER PROGRAM

```

C Planets is the name of this program
  INTEGER W,II,S,QQ,width,QQQ,III,NNNN,XX,YY
  INTEGER SS,ST,SU,SV,WW,N,NO(10000),Q
  DOUBLE PRECISION T(300,300),Z(300),R,HX,HY,NN
  DOUBLE PRECISION E,C(300),PP,SUM,SUM3,GG
  DOUBLE PRECISION NX(10000),NY(10000),RR,RAND
  DOUBLE PRECISION X(10000),Y(10000),A(10000),PX,PY
  DOUBLE PRECISION NNY(10000),NNX(10000),HH(10000)
  DOUBLE PRECISION JJ(20000),RAND,NNNX(10000)
  DOUBLE PRECISION NNNY(10000),abs,cos,GX(10000)
  DOUBLE PRECISION GY(10000),GA(10000),k,FO
  DOUBLE PRECISION E1,E2,E3,E4,HA(10000),FH
  open(12,file='fort.57',status='old')
C Fast JINCs
C This program calculates the position
C and brightness of planets or
C brown dwarfs orbiting brighter stars
C The data must be in the form of a 100X100
C data array that appears as in the read
C statement just above line 4 .
C HA is the CCD electron number per pixel
C GX is the X coordinate, GY is the Y coordinate
C of the pixel measured in number of pixels
C N is the counter. This data is in a separate
C file we call fortran.57
  DO 3 S=1,20000
  CALL BES(E,S)
  JJ(S)=E
3 CONTINUE
  print *,'BESSEL DONE'
  width=100
  FH=0.0
  FO=0.0
C write to write(51,*) X,A
  DO 4 N=1,10000
C *****
  read(12,*) N,GX(N),GY(N),HA(N)
4 CONTINUE
C *****

```

```

print *,'HA(1375)=' ,HA(1375)
DO 5 N=1,10000
II=N/width
III=N-(II*width)
IF(III.LT.2)THEN
GA(N)=HA(N)
GOTO 5
ENDIF
SS=N-width
IF(SS.LT.2)THEN
GA(N)=HA(N)
GOTO 5
ENDIF
ST=N+width
IF(ST.GT.9999)THEN
GA(N)=HA(N)
GOTO 5
ENDIF
SU=N+1
SV=N-1
E1=HA(SS)+HA(ST)+HA(SU)+HA(SV)
GA(N)=((E1+(2*HA(N)))/6.0)+(FH*GA(N))
GA(N)=GA(N)+(((GA(N)*FO)+1)*FH)
5 CONTINUE
S=1
C Density of Data
DO 7 YY=5,95,15
DO 6 XX=5,95,15
NNNN=((width*(YY-1))+XX)
NO(S)=NNNN
S=S+1
NNN=NO(S)
6 CONTINUE
7 CONTINUE
Q=S-1
S=Q
DO 9 N=1,10000,10
II=N/width
III=N-(II*width)
IF(III.LT.2)THEN
GOTO 9
ENDIF
SS=N-width
ST=N+width
SU=N+1

```



```

R=0.0
RR=0.0
HX=X(II)
HY=Y(II)
DO 16 III=1,QQ
E1=abs(NX(III)-5.0)
E2=abs(NY(III)-5.0)
GG=1.0
IF(E1.LT..001)THEN
IF(E2.LT..001)THEN
GG=.01
ENDIF
ENDIF
RAND=3207604736.9830*II*NN+III*III
RAND=cos(RAND)
RAND=(3.14159/2.0)*RAND
RAND=abs(RAND)
RAND=cos(RAND)-.5
C Set Random Number size here
PX=NX(III)+(RAND*1.5*GG)
NNX(III)=PX
RAND=9337470372.3207*NN*III+II*II
RAND=cos(RAND)
RAND=(3.14159/2.0)*RAND
RAND=abs(RAND)
RAND=cos(RAND)-.5
PY=NY(III)+(RAND*1.5*GG)
GG=1.0
NNX(III)=PX
NNY(III)=PY
R=(HX-PX)*(HX-PX)
R=R+((HY-PY)*(HY-PY))
R=R**.5
RR=(999.0*R)+1.001
S=0
S=INT(RR)
T(W,III)=0.0
E=JJ(S)
T(W,III)=E
16 CONTINUE
W=W+1
18 CONTINUE
20 CALL GAUSS(T,C,QQ)
CALL BSOLVE(T,Z,C,QQ)
SUM=0.0

```

```

DO 24 N=1,QQ
SUM=SUM+abs(Z(N))
24 CONTINUE
IF (SUM.GT.SUM3)THEN
GOTO 30
ENDIF
DO 25 N=1,QQ
IF(abs(Z(N)).GT.SUM3)THEN
GOTO 30
ENDIF
25 CONTINUE
SUM3=SUM
print *,'SUM3=',SUM3
DO 26 N=1,QQ
HH(N)=Z(N)
NNNX(N)=NNX(N)
NNNY(N)=NNY(N)
26 CONTINUE
DO 27 N=1,QQ
PRINT *,'K=',Z(N),'PX1=',NNNX(N)
print *,'PY1=',NNNY(N)
print *,' '
27 CONTINUE
30 CONTINUE
DO 40 N=1,QQ
k=HH(N)
Write(53,*)NNNX(N),NNNY(N),k
40 CONTINUE
STOP
END
SUBROUTINE BES(FF,B)
INTEGER B,o
DOUBLE PRECISION RRR,FF,WW
DOUBLE PRECISION J1,ITG,ARG,AN
DOUBLE PRECISION INT1,INT2,INT3
WW=1.0
RRR=B*0.001*WW
J1=0.0
DO 2 o=-50,50
AN=o*.0314159
ARG=RRR*sin(AN)
INT1=cos(ARG)*cos(AN)*cos(AN)
AN=AN+.015708
ARG=RRR*sin(AN)
INT2=cos(ARG)*cos(AN)*cos(AN)*4.0

```

```

AN=AN+.015708
ARG=RRR*sin(AN)
INT3=cos(ARG)*cos(AN)*cos(AN)
ITG=INT1+INT2+INT3
J1=ITG+J1
2 CONTINUE
FF=(J1*.00166667)*(J1*.00166667)
RETURN
END
SUBROUTINE BSOLVE(V,HI,D,QQQQ)
INTEGER I,J,EE,QQQQ
DOUBLE PRECISION V(300,300),HI(300),D(300),SUM
EE=QQQQ
DO 200 I=EE,1,-1
SUM=D(I)
DO 100 J=I+1,EE
SUM=SUM-V(I,J)*HI(J)
100 CONTINUE
HI(I)=SUM/V(I,I)
200 CONTINUE
RETURN
END
SUBROUTINE GAUSS(L,M,QQQ)
INTEGER J,K,I,INDEX,PIVOT,FF,QQQ
DOUBLE PRECISION TEMP,L(300,300),M(300),RATIO,ABS
FF=QQQ
DO 100 I=1,FF
INDEX=I
PIVOT=ABS(L(I,I))
DO 200 J=I+1,FF
IF (ABS(L(J,I)).GT.PIVOT)THEN
PIVOT=ABS(L(J,I))
INDEX=J
ENDIF
200 CONTINUE
IF(INDEX.GT.I)THEN
DO 400 K=I,FF
TEMP=L(I,K)
L(I,K)=L(INDEX,K)
L(INDEX,K)=TEMP
400 CONTINUE
TEMP=M(I)
M(I)=M(INDEX)
M(INDEX)=TEMP
ENDIF

```

```
DO 300 J=I+1,FF
RATIO=L(J,I)/L(I,I)
DO 500 K=I+1,FF
L(J,K)=L(J,K)-L(I,K)*RATIO
500 CONTINUE
M(J)=M(J)-M(I)*RATIO
300 CONTINUE
100 CONTINUE
RETURN
END
```

APPENDIX D

NYQUIST SAMPLING PROGRAM FOR KNOWN SOURCE NUMBERS

```

PROGRAM RESOLUTION
INTEGER I,U,G,N1,N2,N3,N,S
DOUBLE PRECISION T(20,20),C(20),Z(20),X(20)
DOUBLE PRECISION E,Y,K1,K2,K3,P,P1,P2,P3
DOUBLE PRECISION H(20),F(20),SQ,PP,X1,X2,X3
DOUBLE PRECISION S1,S2,S3,A1,A2,A3,PP,SNR
DOUBLE PRECISION JR(20000),R,SUMM,S4,S2
DO 3 I=1,20000
CALL BES(R,I)
JR(I)=R
3 CONTINUE
U=3
C Use Bessel Function 0 extrapolation
C from the amplitude graphs to
C set do loop limits. Find that one can
C estimate the positions of the outer two
C objects . Thus that is how the outer two limits are
C set in DO 30 and DO 28
S4=1000000000000.0
S3=1000000000.0
S1=10000.0
DO 30 N1=-210,-195
DO 29 N2=-194,194
DO 28 N3=195,205
C X=-2,1,2
C A= 1,.3,2
H(1)=-8.6
F(1)=-.0337
H(2)=-6.8
F(2)=.005569
H(3)=-4.8
F(3)=.111
H(4)=-3.8
F(4)=.1972
H(5)=-1.8
F(5)=.5481
H(6)=.8
H(6)=1.126
H(7)=1.8
F(7)=1.137

```

H(8)=2.6
 F(8)=1.0068
 H(9)=4.8
 F(9)=.284
 H(10)=6.0
 F(10)=-0.02335
 H(11)=7.2
 F(11)=-.12
 H(12)=10.0
 F(12)=.048216
 X(1)=-3.8
 C(1)=.1972

- C Keep H s and X s approximately same for
- C accurate sampling for least squares
- C algorithm to work properly

X(2)= 1.4
 C(2)=1.1554
 X(3)=3.8
 C(3)=.6364
 DO 15 I=1,U
 PP=N1*0.01002
 P=PP
 T(I,1)=0.0
 RRR=(X(I)-P)*(X(I)-P)
 RRR=RRR**.5
 RRR=(999.0*RRR)+1.001
 S=0
 S=INT(RRR)
 E=JR(S)
 T(I,1)=E
 PP=N2*0.01001
 P=PP
 T(I,2)=0.0
 RRR=(X(I)-P)*(X(I)-P)
 RRR=RRR**.5
 RRR=(999.0*RRR)+1.001
 S=0
 S=INT(RRR)
 E=JR(S)
 T(I,2)=E
 PP=0.01006*N3
 P=PP
 T(I,3)=0.0
 RRR=(X(I)-P)*(X(I)-P)
 RRR=RRR**.5

```
RRR=(999.0*RRR)+1.001
S=0
S=INT(RRR)
E=JR(S)
T(I,3)=E
15 CONTINUE
Z(1)=0.0
Z(2)=0.0
Z(3)=0.0
20 CALL GAUSS(T,C,U)
CALL BSOLVE(T,Z,C,U)
SOM=0.0
SQ=0.0
DO 27 G=1,10
P=N1*.01002
RRR=(H(G)-P)*(H(G)-P)
RRR=RRR**.5
RRR=(999.0*RRR)+1.001
S=0
S=INT(RRR)
E=JR(S)
A1=E*Z(1)
PP=N2*0.01001
P=PP
RRR=(H(G)-P)*(H(G)-P)
RRR=RRR**.5
RRR=(999.0*RRR)+1.001
S=0
S=INT(RRR)
E=JR(S)
A2=(E*Z(2))+A1
PP=N3*0.01006
P=PP
RRR=(H(G)-P)*(H(G)-P)
RRR=RRR**.5
RRR=(999.0*RRR)+1.001
S=0
S=INT(RRR)
E=JR(S)
A3=(E*Z(3))+A2
SOM=(A3-F(G))*(A3-F(G))
SQ=SOM+SQ
IF(abs(SQ).LT.10.0)THEN
GOTO 23
ENDIF
```

```

GOTO 27
23 DO 24 N=1,3
  IF(abs(Z(N)).GT.10.0)THEN
    GOTO 27
  ENDIF
  IF(Z(N).LT.0.0)THEN
    GOTO 27
  ENDIF
24 CONTINUE
C  print *,'P1=',N1*.1002,'P2=',N2*.1001,'P3=',N3*.1006,'SQ=',SQ
C  print *,'S3=',S3,'S4=',S4
  S1=1.0*SQ
  SQ=0.0
  IF(S1.GT.S3)THEN
    GOTO 27
  ENDIF
  S2=abs(Z(1))+abs(Z(2))+abs(Z(3))
C  IF(S2.LT.2.998)THEN
C  GOTO 27
C  ENDIF
  S3=S1
  S4=S2
  K1=Z(1)
  K2=Z(2)
  K3=Z(3)
  X1=N1*0.01002
  X2=N2*0.01001
  X3=N3*0.01006
  print *,'P1=',X1,'P2=',X2,'P3=',X3
  print *,'S3=',S3,'S4=',S4
  print *,'K1=',K1,'K2=',K2,'K3=',K3
27 CONTINUE
28 CONTINUE
29 CONTINUE
30 CONTINUE
  write(52,*) 'k1=',K1,', 'P1=',X1,'K2=',K2,', 'P2=',X2
  write(52,*) 'K3=',K3,'P3=',X3
  write(52,*) 'X=-2,0,2 Equal Amplitude,SNR>1000'
  PRINT *,'K1=',K1,'P1=',X1,', 'K2=',K2,'P2=',X2
  print *,'K3=',K3,'P3=',X3,"
  STOP
  END
  SUBROUTINE BES(FF,B)
  INTEGER o,B
  DOUBLE PRECISION J1,ARG,AN

```

```

DOUBLE PRECISION RR,INT1,INT2,INT3
DOUBLE PRECISION ITG,FF
RR=B*.001
J1=0.0
DO 2 o=-50,50
AN=o*.0314159
ARG=RR*sin(AN)
INT1=cos(ARG)*cos(AN)*cos(AN)
AN=AN+.015708
ARG=RR*sin(AN)
INT2=cos(ARG)*cos(AN)*cos(AN)*4.0
AN=AN+.015708
ARG=RR*sin(AN)
INT3=cos(ARG)*cos(AN)*cos(AN)
ITG=INT1+INT2+INT3
J1=ITG+J1
2 CONTINUE
FF=J1*.00166667
RETURN
END
SUBROUTINE BSOLVE(V,HI,D,Q)
INTEGER I,J,Q
DOUBLE PRECISION V(20,20), HI(20), D(20),SUM
DO 200 I=Q, 1,-1
SUM =D(I)
DO 100 J=I+1,Q
SUM=SUM-V(I,J)*HI(J)
100 CONTINUE
HI(I)=SUM/(V(I,I)+.00000000001)
C print *,'HI=',HI,'V(I,I)=',V(I,I)
200 CONTINUE
RETURN
END
SUBROUTINE GAUSS(L,M,Q)
INTEGER J,K,PIVOT,I,INDEX,Q
DOUBLE PRECISION TEMP,L(20,20),M(20),RATIO,ABS
DO 100 I=1,Q
INDEX=I
PIVOT=INT(ABS(L(I,I)))
DO 200 J=I+1,Q
IF (ABS(L(J,I)) .GT.PIVOT) THEN
PIVOT=INT(ABS(L(J,I)))
INDEX=J
ENDIF
200 CONTINUE

```

```
IF (INDEX.GT.I) THEN
DO 400 K=I,Q
TEMP=L(I,K)
L(I,K)=L(INDEX,K)
L(INDEX,K)=TEMP
400 CONTINUE
TEMP=M(I)
M(I)=M(INDEX)
M(INDEX)=TEMP
ENDIF
DO 300 J=I+1,Q
RATIO=L(J,I)/(L(I,I)+.00000000001)
DO 500 K=I+1,Q
L(J,K)=L(J,K)-L(I,K)*RATIO
500 CONTINUE
M(J)=M(J)-M(I)*RATIO
300 CONTINUE
100 CONTINUE
RETURN
END
```

APPENDIX E

NYQUIST SAMPLING WITH EDGE POSITIONS KNOWN

```

PROGRAM RESOLUTION
INTEGER I,U,G,N1,N2,N3,N,S
DOUBLE PRECISION T(20,20),C(20),Z(20),X(20)
DOUBLE PRECISION E,Y,K1,K2,K3,P,P1,P2,P3
DOUBLE PRECISION H(20),F(20),SQ,PP,X1,X2,X3
DOUBLE PRECISION S1,S2,S3,A1,A2,A3,PP,SNR
DOUBLE PRECISION JR(20000),R,SUMM,S4,S2
DO 3 I=1,20000
CALL BES(R,I)
JR(I)=R
3 CONTINUE
U=3
C Use Bessel Function 0 extrapolation
C from the amplitude graphs to
C set do loop limits. Find that one can
C estimate the positions of the outer two
C objects . Thus that is how the outer two limits are
C set in DO 30 and DO 28
S4=1000000000000.0
S3=10000000000.0
S1=10000.0
DO 30 N1=-210,-195
DO 29 N2=-194,194
DO 28 N3=195,205
C X=-2,0,2
C A= 1,2,3
H(1)=-7.8
F(1)=.02642
H(2)=-5.4
F(2)=-.0307
H(3)=-4.0
F(3)=.117
H(4)=-1.8
F(4)=1.1538
H(5)=-.2
F(5)=2.0762
H(6)=1.8
H(6)=2.14
H(7)=3.0
F(7)=1.481
H(8)=4.8

```

```

F(8)=.305
H(9)=6.8
F(9)=-.1757
H(10)=10.0
F(10)=.078
H(11)=-10.0
F(11)=-.017836
H(12)=1.2
F(12)=2.2953
X(1)=-3.2
C(1)=.3806
C Keep H s and X s approximately same for
C accurate sampling for least squares
C algorithm to work properly
X(2)= .8
C(2)=2.314
X(3)=3.8
C(3)=.9223
DO 15 I=1,U
PP=N1*0.01002
P=PP
T(I,1)=0.0
RRR=(X(I)-P)*(X(I)-P)
RRR=RRR**.5
RRR=(999.0*RRR)+1.001
S=0
S=INT(RRR)
E=JR(S)
T(I,1)=E
PP=N2*0.01001
P=PP
T(I,2)=0.0
RRR=(X(I)-P)*(X(I)-P)
RRR=RRR**.5
RRR=(999.0*RRR)+1.001
S=0
S=INT(RRR)
E=JR(S)
T(I,2)=E
PP=0.01006*N3
P=PP
T(I,3)=0.0
RRR=(X(I)-P)*(X(I)-P)
RRR=RRR**.5
RRR=(999.0*RRR)+1.001

```

```
S=0
S=INT(RRR)
E=JR(S)
T(I,3)=E
15 CONTINUE
Z(1)=0.0
Z(2)=0.0
Z(3)=0.0
20 CALL GAUSS(T,C,U)
CALL BSOLVE(T,Z,C,U)
SOM=0.0
SQ=0.0
DO 27 G=1,10
P=N1*.01002
RRR=(H(G)-P)*(H(G)-P)
RRR=RRR**.5
RRR=(999.0*RRR)+1.001
S=0
S=INT(RRR)
E=JR(S)
A1=E*Z(1)
PP=N2*0.01001
P=PP
RRR=(H(G)-P)*(H(G)-P)
RRR=RRR**.5
RRR=(999.0*RRR)+1.001
S=0
S=INT(RRR)
E=JR(S)
A2=(E*Z(2))+A1
PP=N3*0.01006
P=PP
RRR=(H(G)-P)*(H(G)-P)
RRR=RRR**.5
RRR=(999.0*RRR)+1.001
S=0
S=INT(RRR)
E=JR(S)
A3=(E*Z(3))+A2
SOM=(A3-F(G))*(A3-F(G))
SQ=SOM+SQ
IF(abs(SQ).LT.10.0)THEN
GOTO 23
ENDIF
GOTO 27
```

```

23 DO 24 N=1,3
    IF(abs(Z(N)).GT.10.0)THEN
        GOTO 27
    ENDIF
    IF(Z(N).LT.0.0)THEN
        GOTO 27
    ENDIF
24 CONTINUE
C   print
C   print *, 'S3=',S3,'S4=',S4
    S1=1.0*SQ
    SQ=0.0
    IF(S1.GT.S3)THEN
        GOTO 27
    ENDIF
    S2=abs(Z(1))+abs(Z(2))+abs(Z(3))
C   IF(S2.LT.2.998)THEN
C   GOTO 27
C   ENDIF
    S3=S1
    S4=S2
    K1=Z(1)
    K2=Z(2)
    K3=Z(3)
    X1=N1*0.01002
    X2=N2*0.01001
    X3=N3*0.01006
    print *, 'P1=',X1,'P2=',X2,'P3=',X3
    print *, 'S3=',S3,'S4=',S4
    print * , 'K1=',K1,'K2=',K2,'K3=',K3
27 CONTINUE
28 CONTINUE
29 CONTINUE
30 CONTINUE
    write(52,*) 'k1=',K1,'P1=',X1,'K2=',K2,'P2=',X2
    write(52,*) 'K3=',K3,'P3=',X3
    write(52,*) 'X=-2,0,2 Equal Amplitude,SNR>1000'
    PRINT *, 'K1=',K1,'P1=',X1,' ', 'K2=',K2,'P2=',X2
    PRINT *, 'K3=',K3,'P3=',X3,"
    STOP
    END
    SUBROUTINE BES(FF,B)
    INTEGER o,B
    DOUBLE PRECISION J1,ARG,AN
    DOUBLE PRECISION RR,INT1,INT2,INT3

```

```

DOUBLE PRECISION ITG,FF
RR=B*.001
J1=0.0
DO 2 o=-50,50
AN=o*.0314159
ARG=RR*sin(AN)
INT1=cos(ARG)*cos(AN)*cos(AN)
AN=AN+.015708
ARG=RR*sin(AN)
INT2=cos(ARG)*cos(AN)*cos(AN)*4.0
AN=AN+.015708
ARG=RR*sin(AN)
INT3=cos(ARG)*cos(AN)*cos(AN)
ITG=INT1+INT2+INT3
J1=ITG+J1
2 CONTINUE
FF=J1*.00166667
RETURN
END
SUBROUTINE BSOLVE(V,HI,D,Q)
INTEGER I,J,Q
DOUBLE PRECISION V(20,20), HI(20), D(20),SUM
DO 200 I=Q, 1,-1
SUM =D(I)
DO 100 J=I+1,Q
SUM=SUM-V(I,J)*HI(J)
100 CONTINUE
HI(I)=SUM/(V(I,I)+.00000000001)
C print *,'HI=',HI,'V(I,I)=' ,V(I,I)
200 CONTINUE
RETURN
END
SUBROUTINE GAUSS(L,M,Q)
INTEGER J,K,PIVOT,I,INDEX,Q
DOUBLE PRECISION TEMP,L(20,20),M(20),RATIO,ABS
DO 100 I=1,Q
INDEX=I
PIVOT=INT(ABS(L(I,I)))
DO 200 J=I+1,Q
IF (ABS(L(J,I)) .GT.PIVOT) THEN
PIVOT=INT(ABS(L(J,I)))
INDEX=J
ENDIF
200 CONTINUE
IF (INDEX.GT.I) THEN

```

```
DO 400 K=I,Q
TEMP=L(I,K)
L(I,K)=L(INDEX,K)
L(INDEX,K)=TEMP
400 CONTINUE
TEMP=M(I)
M(I)=M(INDEX)
M(INDEX)=TEMP
ENDIF
DO 300 J=I+1,Q
RATIO=L(J,I)/(L(I,I)+.000000000001)
DO 500 K=I+1,Q
L(J,K)=L(J,K)-L(I,K)*RATIO
500 CONTINUE
M(J)=M(J)-M(I)*RATIO
300 CONTINUE
100 CONTINUE
RETURN
```

APPENDIX F

OTHER SUPERESOLUTION TECHNIQUES

INTRODUCTION

Here we introduce miscellaneous techniques that are related to the issues addressed in the previous chapters. Apodization is introduced as a possible method of increasing superresolution. Recall the second derivative technique from the introduction. The non PSF ring images would tend to add substantial second derivatives to the intensity and thus be counted as sources. But the method of apodization here eliminates these rings around the PSF s and so you would think that the second derivative technique would be enhanced.

The magnification used here is rather large since one PSF fills the whole field of view which is about 242 by 375 pixels. Recall that the scale of the noise is on the order of the scale of a pixel so the noise can be easily suppressed by smoothing without greatly distorting the PDF. Thus we review the method of fourth differences as a way to smooth the data. Thus we do not need the iterative stochastic techniques that dominate optical deconvolution these days. See chapter 2 for a discussion of these techniques.

F.1 PSF CAUSED BY APERTURE OF VARIABLE OPACITY

Here we discuss the situation in which there is a varying opacity¹.

We pick as the normalized pupil function

$$f(r) = \frac{1}{8}[(1-r^2)^2(8+10r^2-3r^4)] + \frac{3}{8}(8r^2-4r^2+r^2)\log\left(\frac{1+(1-r^2)^{\frac{1}{2}}}{r}\right) \quad (\text{F.1.1})$$

enough in space so that the second derivative doesn't change very much over a few measurements. We next assume that over 5 measurements that the points fit on a parabola of the form

$$y = a + bx + cx^2 \quad (\text{F.2.1})$$

Since we are dealing with 5 data (not 3) we must here use the principle of the least squares. Thus we write:

$$\sum_{k=-2}^{+2} (y - y_k)^2 = \sigma^2 \quad (\text{F.2.2})$$

We minimize this quantity with respect to the a,b,c coefficients in equation (F.2.1). The data points here are $x=-2,-1,0,1,2$. These we simply plug into equation (F.2.2) and get:

$$\begin{aligned} \sum_{k=-2}^{+2} (y - y_k)^2 &= (a - 2b + 4c - y_{-2})^2 + (a - b + c - y_{-1})^2 + (a - y_0)^2 \\ &+ (a - b + c - y_1)^2 + (a + 2b + 4c - y_2)^2 \end{aligned} \quad (\text{F.2.3})$$

We take the derivative of equation (F.2.3) with respect to c and equate it to zero to get the condition of a minimum of equation (F.2.3) with respect to c. This condition is then:

$$10a + 34c - \sum_{k=-2}^2 k^2 y_k = 0 \quad (\text{F.2.4})$$

We next take the derivative of equation (F.2.2) with respect to a and get:

$$5a + 10c - \sum_{k=-2}^2 y_k = 0 \quad (\text{F.2.5})$$

So we solve equation (F.2.4) and (F.2.5) for a and get:

$$70a = -6y_{-2} + 24y_{-1} + 34y_0 + 24y_1 - 6y_2 \quad (\text{F.2.6})$$

$$a = y_0 + \frac{3}{35}(y_{-2} - 4y_{-1} + 6y_0 - 4y_1 + y_2) \quad (\text{F.2.7})$$

We next plug in this and solve for c in equation (F.2.5) and get:

$$c = \frac{1}{10} \sum_{k=-2}^2 y_k - \frac{1}{2} \left(y_0 - \frac{3}{35}(y_{-2} - 4y_{-1} + 6y_0 - 4y_1 + y_2) \right) \quad (\text{F.2.8})$$

Next we take the derivative of equation (F.2.8) with respect to c and set that equal to zero.

The result for minimum gives us that

$$10b = -2y_{-2} - y_{-1} + y_1 + 2y_2 \quad (\text{F.2.9})$$

or

$$b = \frac{-2y_{-2} - y_{-1} + y_1 + 2y_2}{10} \quad (\text{F.2.10})$$

Thus the key is to smoothing the PSF's is to plug in the results of equations (F.2.7), (F.2.4) and (F.2.10) into equation (F.2.1) at each data point. Then we fill in the 50 points on each side of the data PSF to create the PSF table. Recall that we needed this table to get computer speed fast enough to solve our coherent superresolution problem. This also allows us to take into account some optical system aberrations as well. This method was *not* used here. Discontinuities occur at the end of each polynomial evaluation which cause false ridges to appear in the data. Here we just connect the data points by straight lines instead of using the above technique

SUMMARY

Here we analysed some techniques for doing superresolution using apodization. We found in the simulations that they don't work. We also tried to smooth our data using the method of fourth differences. We found in our simulations that didn't work also.

References

- 1) F. Leaver, Apodization to Produce Monotonically-Decreasing, Radially Symmetrical Point Spread Function, *Optics Communications*, 15 (3), 1975, p.374.
- 2) C.Lanczos, *Applied Analysis*, Dover Publications, 1956, p.316.