Submillimeter Array Technical Memorandum No. 116

HOLIS

A program package for calculating the antenna aperture field distribution from the measured far-field radiation patterns. Both the phase-coherent and phase-retrieval algorithms are implemented.

This package also performs preprocessing of the input far-field data, in order to vary the effective resolution length unit on the aperture plane, to suppress aliasing effect, and to change the phase reference plane where the aperture field is calculated.

A panel-fitting algorithm allows the determination of the corrective screw motions on each panel with four degrees-of-freedom, utilizing all the aperture data points which fall within the range of each panel as specified in an input nodes table.

An auxiliary calculation package performs various simulations to generate radiation patterns using the Fourier transform as well as the Kirchhoff's integral methods. An arbitrary panel on the SMA antenna can be displaced by an arbitrary amount, and the radiation field at an arbitrary near-field location is then calculated by either method. This allows the invertigation of the effect of aliasing, near-field, measurement errors and noise, etc. on the quality of the recovered aperture field. Several other auxiliary programs are also included.

Xiaolei Zhang
September 6, 1997

Contents

1 Introduction

HOLIS is a program package written in Fortran 77, which computes the aperture field of a microwave dish antenna from the measured far-field radiation patterns. The aperture-field calculation algorithms included in HOLIS include:

1. *The Standard Phase Coherent Fourier Transfer Technique*, which calculates the antenna aperture field from a direct Fourier transforms of the measured amplitude and phase beam patterns of the antenna.

2. *Misell Phase Retrieval Technique*, which makes use of two or more amplitude beam pattern maps measured at the different telescope focus settings, and employs an iterative approach to find the aperture amplitude and phase distributions.

3. *Global Fitting Technique*, which fits the aperture field pattern by minimizing the difference between the Fourier-transformed aperture field and the measured far-field. This last algorithm is yet to be implemented.

2 Structure of the Programs

The organization of the HOLIS package is as follows: under the "holis" main directory, there are four subdirectories, "preprocessing", "inversion", "panelfit" and "diffraction", each with the function of doing preprocessing on the far field data; calculating the aperture field from the far field patterns; fitting the panel corrective movement; as well as doing auxiliary calculations of patterns for simulation purposes, respectively.

Within each directory, there is always a makefile, which makes the executables of the programs. Type "make programname" will compile and link the necessary routines and make the executable called "programname". Also, typing "make clean" will remove all the object (the .o) files in the current directory.

In running the executable programs, it sometimes requires to first fill out a parameter file (examples of such a parameter file can be found in section 7). The parameters in the file can be changed and re-run the executable without the need to recompile the programs (the dimension of the matrices, however, has an upper limit, which is specified in an include file "holis.inc" existing in every subdirectory. A global declaration of implicit double precision is also made in "holis.inc".). Care must be taken to input real parameters with a decimal point, and integer parameters as integers, as shown in the default format of the parameter files. In the following, we give more detailed explanations of the functions of the various programs in each sub-directory.

3 Preprocessing

This is the first stage in data processing. Here the description pertains mainly to the phase-coherent approach. The preprocessing for the data taken for phase-retrieval approaches can also be done similarly by slightly modifying the code in this subdirectory.

For the phase coherent approach, the input data, in the form of two files containing the amplitude and phase distributions of the beam pattern, respectively, are assumed to have been taken on a grid of N by N in size, in the order of increasing azimuth for every (increasing) elevation. The preliminary calibration of the measured data to remove systematic drifts (boresight calibration) is assumed to have been done before sending data to this directory.

The routines in this subdirectory perform mainly three tasks:

1. Determining the phase-reference plane (which is usually chosen to be half-way into the depth of the main reflector to minimize diffraction effect) by performing the corrections to the measured phase described in Zhang (1995), section 5.

2. It has the option of increasing the resolution on the aperture plane by patching zeros in the area outside the measurement grid, to create effectively a larger-sized measurement grid for both the amplitude and phase files of the beam pattern.

3. The choice of weighting the far-field amplitude by a Gaussian to reduce the effect of aliasing.

Steps 2 and 3 are expected to be necessary when doing holography on a celestial source, where a smaller map is taken. Step 3 was also found to be useful at the start of the alignment of a dish antenna, when the panel errors are large and aliasing effect is strong (Zhang et al. 1996).

4 Calculating the Aperture Field

4.1 Phase-Coherent Algorithm

In the phase-coherent holography algorithm, the aperture field is obtained from the measured far-field pattern, usually obtained from the output in the *preprocessing* directory, through a direct Fourier transform (a forward FT in our convention). Specifically, the complex aperture field distribution $E(x, y)$ is calculated from the complex far-field pattern $A(a, b)$ through

$$\begin{aligned} E(x_m, y_n) &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} A(a_i, b_j) e^{-(x_m a_i + y_n b_j) \cdot 2\pi / N} \qquad (1) \\ &= DFT[A(a, b)], \end{aligned}$$

where N is the total number of samples in one dimension. $A(a_i, b_j)$ are sampled at interval λ/D' where D' is the extent of image we retrieve at the aperture plane. If we use D to denote the diameter of the antenna, D/D' is the Nyquist sampling rate for the holography data acquisition, and is a number generally less than 1. The spacing (linear resolution) on the aperture plane after the discrete Fourier transform is D'/N.

The sign convention of our program for forward and inverse DFT's is consistent with that used in Matlab, but opposite to that used in Morris (1985) and in Numerical Recipes in Fortran, p. 381.

One intricacy involved here is the issue of original shift, since normally an FFT routine will put the low-frequency components on the edges of the frequency domain. For the actual measured far-field pattern, which has a peak in the central region of the map, it turns out we need to do an origin shift both before and after the FFT, in order to recover the correct aperture distribution, i.e. that which has a platform near the central region of the map. If the far-field pattern used is from an (inverse) FFT followed by an origin shift, as is the case for our example below, which also peaks near the center, we only need to do an origin shift before the FFT to recover the correct aperture distribution in this second case, with no need to do an origin shift after the FFT.

The recovered aperture distribution is then masked for the subreflector blockage (possibly also the feedleg blockage in the future) and for the area outside the edge of the antenna dish. The aperture phase is adjusted to correct for the near-field effect (Zhang 1995, section 6). At this point we have obtained an aperture phase distribution ready to be fitted for large-scale errors.

In the current implementation of HOLIS we fit for 6 kinds of large-scale errors: (1). A constant term. (2). A linear gradient term in the x direction. (3). A linear gradient term in the y direction. (4). A defocus function (Ruze 1969). (5) A secondary-mirror diffraction pattern. (6) A second-order term in units of the distance on the aperture plane, $x^2 + y^2$. This last term is not suggested to be fitted if a reasonably-accurate measurement of the distance to the transmitter is obtained and near-field effect is subtracted out (by specifying the distance to the transmitter in the parameter file). This is because the second order term can be easily confused with the defocus term (Zhang 1995, Appendix II). The surface rms phase errors before and after the large scale fitting are calculated, and a residual phase map is written out to be copied to the *panelfit* directory to calculate the panel movement needed to compensate for the residual error.

4.2 Misell Phase-Retrieval Algorithm

Misell phase retrieval algorithm (Morris 1985) uses two or more amplitude beam patterns measured at the different telescope focus settings, and employs an iterative approach between the aperture and far-field planes to calculate the aperture amplitude and phase distributions. So far, the convergence properties of the Misell algorithm was found to be not always good, especially in the presence of aliasing effect. If the constraints of the subreflector blockage and dish size, as well as the strats blockage pattern can be incorporated in the initial guess and in the processing of the result of each iteration, much better convergence can be obtained. An example of the use of Misell phase retrieval algorithm is given section 7.5.

5 Panel Movement Calculation

From the recovered aperture phase map $Ph(x, y)$, where x, y are the cartesian coordinates on the aperture plane, a surface error map can be derived (Zhang 1995, section 1). Specifically, the error map $\epsilon(x, y)$ is related to the phase map $Ph(x, y)$ through

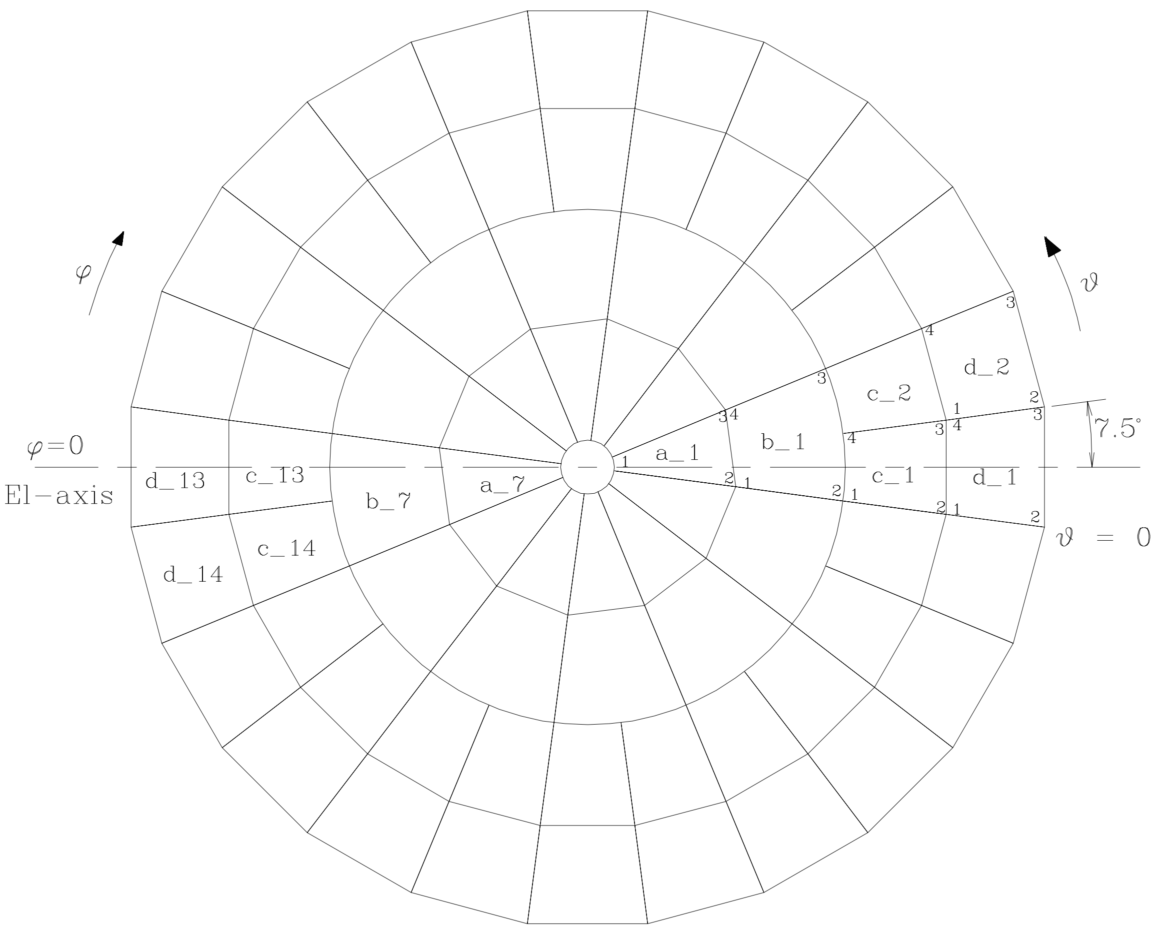

Figure 1: Frontview of the SMA antenna dish

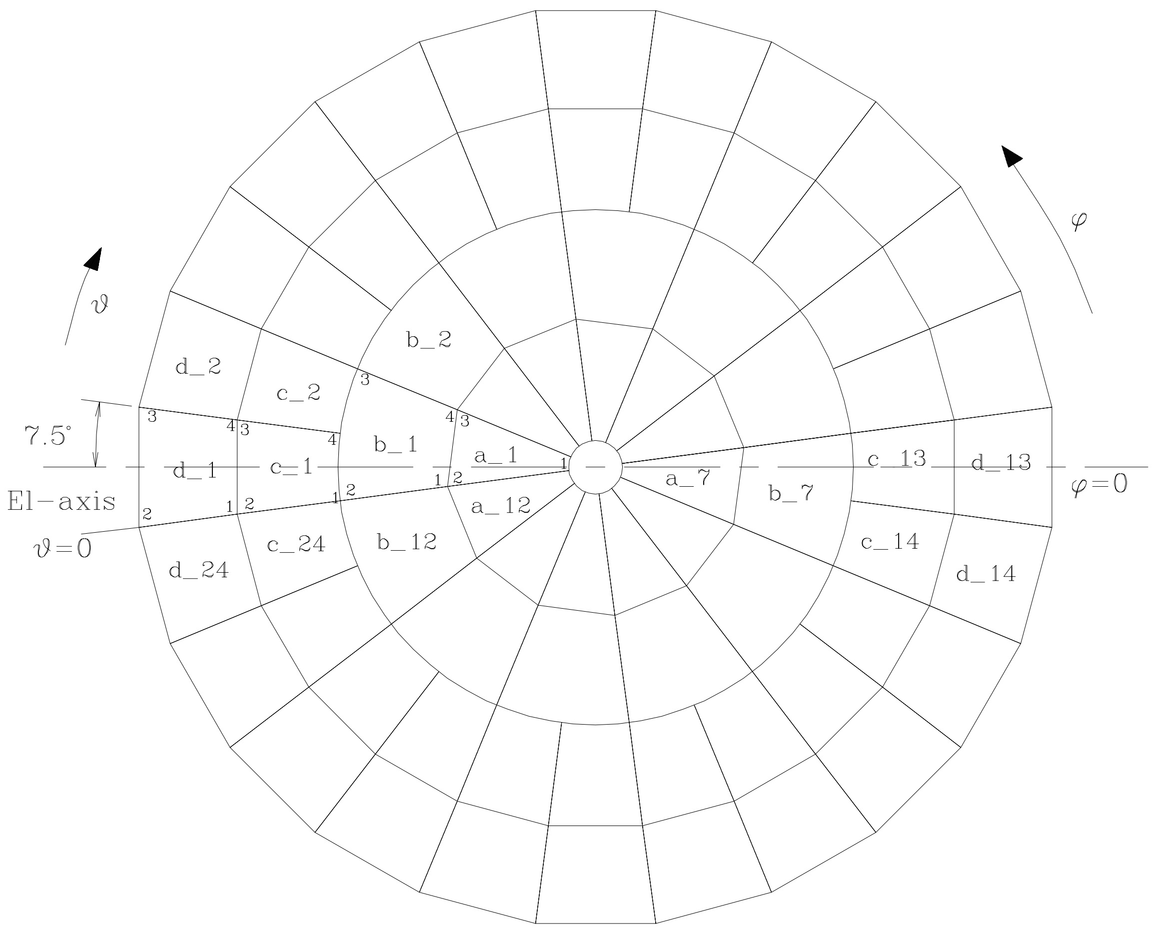

Figure 2: Backview of the SMA antenna dish

$$\epsilon(x,y) = \frac{\lambda}{4\pi}(1 + \frac{x^2+y^2}{4F^2})^{1/2} \cdot Ph(x,y) \tag{2}$$

where F is the primary focal length of the antenna, and where $\epsilon > 0$ means that the panel is moved outward from the correct location.

The actual panel screw movement needed to compensate for the aperture error distribution is calculated using the least-square fitting approach. The motions for all four screws (or all three screws for the first row) of a panel are calculated simultaneously using all the data points which fall within the range of the panel, with possible boundary masks.

In Figures 1 and 2, the front and back view of the SMA antenna panels, together with the naming convention for panels and for the screws on each panels. Note that there is a 7.5^o rotation in the arrangement of the panels, so the symmetry with respect to the elevation axis is lost.

The Fourier transform and the panel fitting programs refer to an aperture phase distribution which is effectively the backview of the antenna (Figure 2), whereas the actual setting of the screws is done from the front of the antenna dish. In order to minimize confusion our naming convention established an exact correspondence between the front and the back views. Specifically,

- In both the front and back views, the first row of panels is the innermost row, numbered a_1 through a_12, with the first panel starting from left (for front view) or right (for back view) side of the dish center, and its first radial edge 7.5^o below the elevation axis. The numbering then run counter-clockwise (for front view) or clockwise (for back view). Similarly for panels b_1 through b_12, c_1 through c_24 and d_1 through d_24. The first edge of the first panel in each row corresponds also to $\theta = 0$ in the panel fitting programs.

- The first row of the main reflector panels have three screws, and the rest of the panels have four screws. The numbering of the screws is such that for each panel, the first screw is the one which has the smallest r and θ value, and the second, third and fourth screws run counter-clockwise (for front view) and clockwise (for back view) from the first screw. Note that the angle θ also runs in different directions for the front and back views.

- In the panel fitting programs, a second angle, ϕ, is also used. $\phi = 0$ corresponds to the left side of the elevation axis and runs clockwise for the front view, and it corresponds to the right side of the elevation axis and runs counter-clockwise for the back view. The relation between the θ and ϕ coordinates are: $\phi = 180^o - \theta + 7.5^o$.

Using the above convention, therefore, the movement calculated for screw #1 on panel a_1, say, from the backview of the antenna structure can be directly used to set the screw #1 in panel a_1 from the front without the need for further conversion.

The general procedure for the panel fitting is as follows:

1. Before invoking the main program package *panelfit.com*, first we need to generate the coordinates of the nodes and screw positions for every panel, by running the program *nodes.f*. The coordinates are saved in two tables, *nodes.table* and *screws.table*. The following items then further describe the panel fitting procedure in the main program.

2. Pick a panel (the usual order for doing panel fitting is from a_1 through a_12, ... d_1 through d_24), read out its nodal positions from *nodes.table*, and its screw locations from *screws.table*.

3. Using the nodal coordinates information, calculate the radial and angular limits of the panel, which are then used by the searching program to find all the data points which fall within the range of the panel.

4. Using all the data points for the relevant panel, do a least square fit to calculate the coefficients of the error function (a general second order function).

5. Using the thus-determined error function, calculate the value of the error at the location of the screws. The negative of this value gives the motion of the screw needed to compensate for the error.

6. write out the screw motion into a table, for all the screws on every panel.

Note that since many panels of the SMA reflector have four screws, and since the panels themselves are not perfectly rigid, the general first order function $\epsilon(x, y) = ax + by + c$ is no longer adequate for determining the four degrees-of-freedom screw motions. We therefore choose to use a general second order function $\epsilon(x, y) = ax + by + c + dxy + ex^2 + fy^2$, which has six unknown coefficients. For 128 by 128 map we make, an average panel has from 60 to 115 data points fall within its range, even after masking out a certain amount of data near the edges of panels, which are more than enough to fit the six coefficients. After the error function is thus determined, we can calculate the corresponding error at (x_{screw}, y_{screw}) locations. Since there are at most four screws on each panel, only four independent degrees-of-freedom are actually utilized even with the six fitted coefficients.

6 Auxiliary Pattern and Diffraction Calculation Routines

This subdirectory contains some of the utility programs for generating various kinds of beam patterns for simulating purpose and for investigating the systematic effects. Here we outline the main programs contained in this directory.

The program "pattern.f" is used to generate far-field amplitude and phase patterns through the Fourier transform of the aperture field. The choices of operations performed on the aperture data include: (1) Masking the aperture data file (amplitude and phase) so that the data outside the main dish and inside the subreflector areas are put to zero. (2) Could including the effect of defocus on the aperture phase distribution. The amount of defocus is specified in the parameter file "pattern.prm" (3) It has an option to mask out the region also for strats and their shadow (implemented currently for SMA antennas). (4)It has the option of simulating the effect of the movement of each individual panel on SMA antenna (or any panel configuration specified by a nodes file "nodes.dat") by an specified amount (which is input by the user at run time).

The pair of programs "flatdiff1d.f" and "backtrans1d.f" calculates the one-dimensional diffraction pattern of any amplitude and phase on a flat surface at a given distance, and then use Fourier transform and near field correction to recover the aperture distribution. This pair of programs are used for investigating the aliasing effect of the various map sizes. The examples of such investigations can be found in Zhang (1995).

Program "flatdiff2d.f" calculates the two-dimensional diffraction pattern of a flat antenna aperture at another spherical surface a distance r away. The default is a constant amplitude and constant phase distribution inside the antenna aperture and zero outside. It has the option of having an initial defocused phase pattern on the antenna surface. Another option is to simulate raising a given panel by a given amount, both can be specified in the parameter file. The corresponding program for back-trasnform and near-field correction, etc. can of course be found in the "inversion" directory (it is the main "holis.f" series of programs).

Program "subdiff.f" calculates the scalar diffraction integral to obtain the secondary diffraction pattern on the primary illumination in a Cassegrain system. This program was provided by Richard Hills of MRAO. Since the result of subdiff.f is one-dimensional (on a radial line in the aperture), another program "diff2d.f" converts the output of "subdiff.f" into a two-dimensional diffraction map. An example of such a calculation for the SMA antennas is given in Zhang (1995).

Program "add_noise.f" adds varying amount of noise into the far-field beam pattern. It is used to investigate the effect of measurement error and noise on the quality of recovered aperture field pattern.

Program "spar" is a utility routine written by Colin Masson to calculate the power-blockage due to the presence of spars.

Program twod2oned.f converts a two-dimensional array into azimuthally-averaged one-dimensional array. This program is used to investigate the subreflector diffraction content in a measured (after the Fourier transform inversion) aperture pattern.

7 A Few Examples

7.1 Fourier Transform and Fitting of Large-Scale Errors

In the following, we present an example demonstrating the use of the routines in this directory to calculate the aperture distribution and to fit large-scale errors.

We use program "pattern.f" in the "diffraction" subdirectory to generate an aperture distribution which has a constant amplitude across the aperture except for the subreflector blockage, and which has a phase distribution corresponding to 0.32445 mm (= 0.1 λ at 92.4 GHz) defocus. The far-field amplitude and phase are obtained from the (inverse) FFT of the above aperture distribution.

We now input this far-field data file to our Holis package, and calculate the aperture distribution and fit for the large scale errors. Here is part of what the program returns:

Number of the large scale fitting parameter = 5

rms before the phasefit = 9.3655026470028D-02 radian
coef(1)= -1.6651458177358D-10 ! constant offset
coef(2)= 2.5987720932186D-15 ! tilt in x direction
coef(3)= 2.6029107270919D-15 ! tilt in y direction
coef(4)= 0.32445000339099 ! defocus in mm
coef(5)= -4.8622594431836D-10 ! diffraction term
coef(6)= 0. ! $x^2 + y^2$ *term*
chisq= 3.4790924433017D-17
rms after the phasefit = 6.9609861246177D-11 radian

We see that the program correctly calculated the aperture field, and the fitting routines attribute most of the phase distribution in the defocus term, and gives an estimated amount of defocus which is almost exactly equal to what we started with. Very little is attributed to the rest few large-scale error terms.

In this example, since the far field pattern input into the "holis.f" program is calculated from the Fourier transform of the aperture pattern (by "pattern.f"), we obtained a nearly perfect recovery of the aperture field in the inversion process. In section 7.3, we will see a case which corresponds to the real measuring situation, where there is aliasing effect due to the fact that there will inevitably some energy lost outside the finite-sized measured beam.

7.2 Panel Fitting

In the following, we give an example of the panel-fitting process. We start with an initial aperture phase distribution, shown in Figure 3, which corresponds to 0.1 λ defocus. We try to compensate this aperture phase by fitting and adjusting each individual panels, and then calculate the residual error distribution. Figure 4 shows the result of the residual error map after running the phase map in Figure 3 through our panel-fitting program. It is seen that the maximum residual phase error is at the sub-μm level, and even this residual error is mostly due to that the defocus error is of a different functional form than that due to the movement of the panels. The surface rms drops from 28.3 micron before the panel-fitting to $1.6 \cdot 10^{-2}$ micron after the panel-fitting.

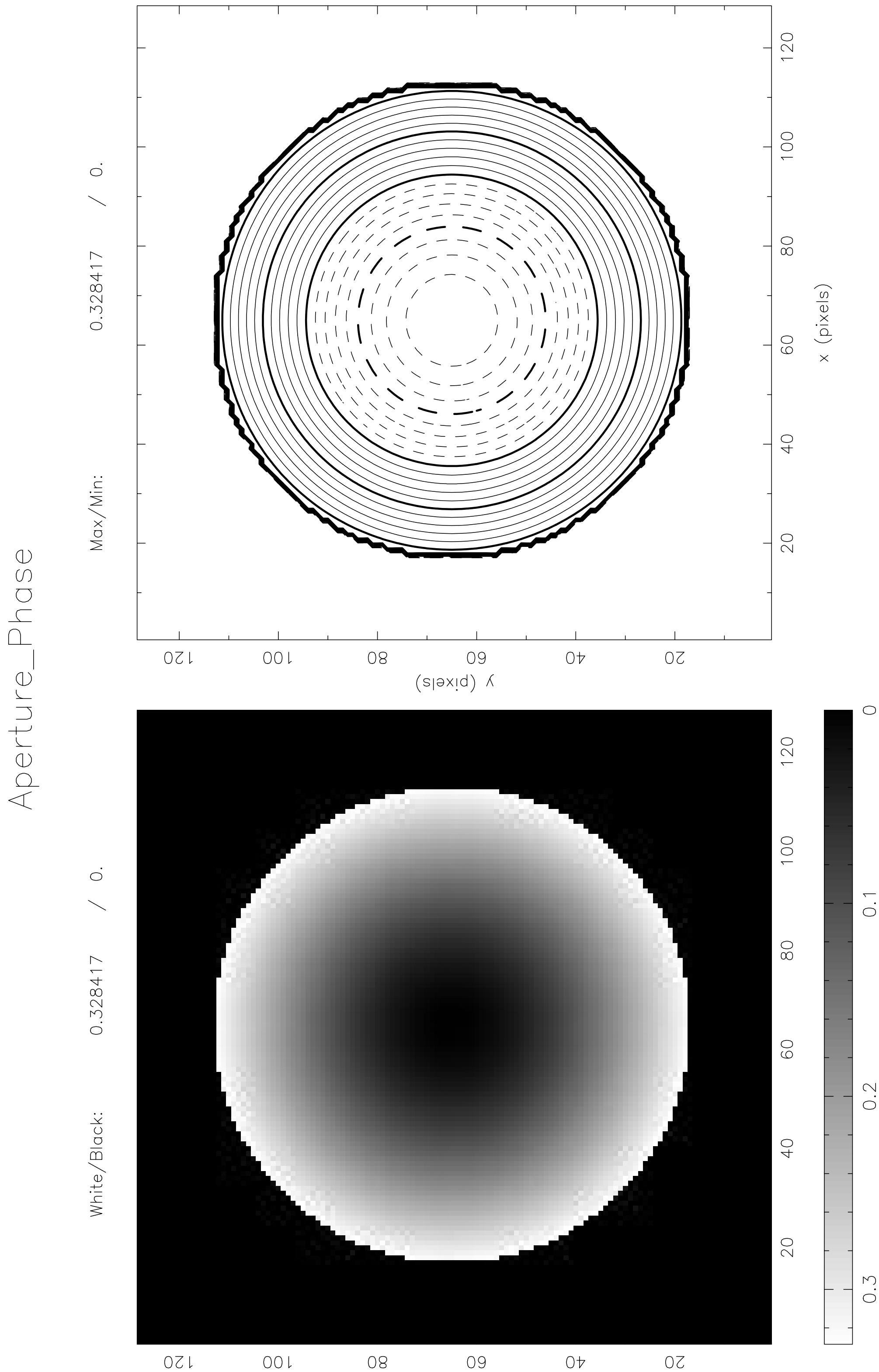

Figure 3: An aperture phase distribution (in radians) corresponds to 0.1 λ subreflector movement

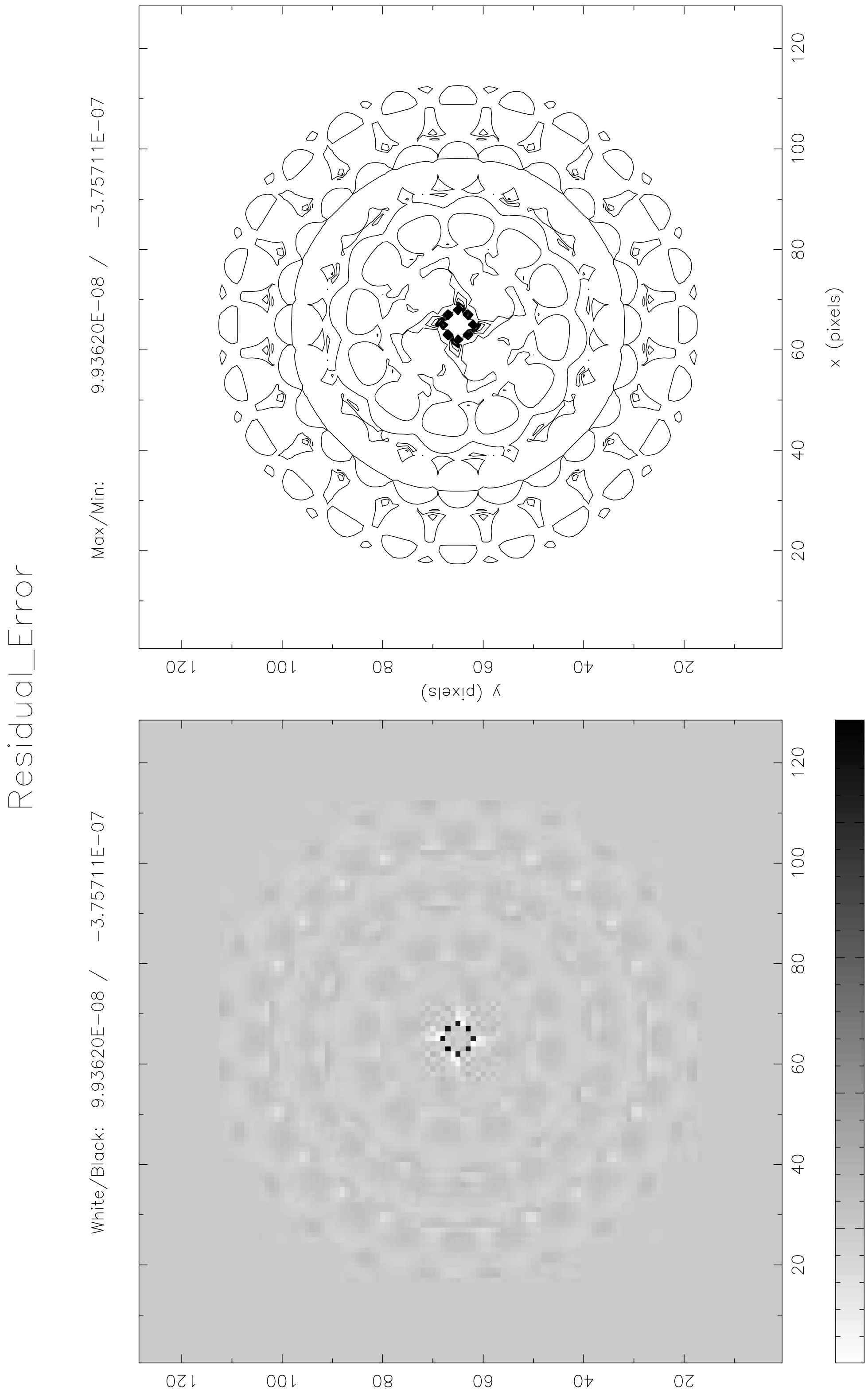

Figure 4: Residual surface error (in meters) after fitting and adjusting the panels

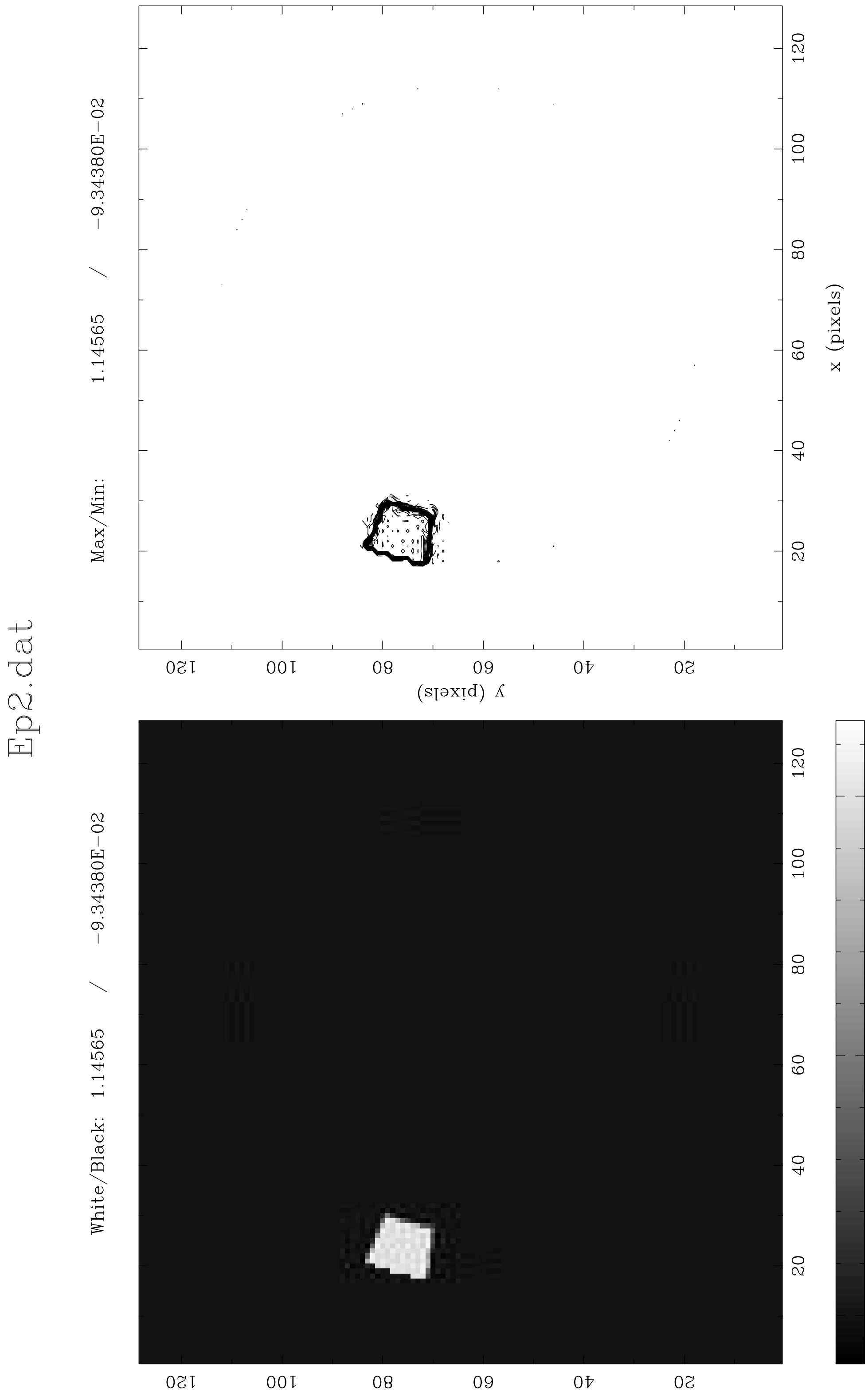

Figure 5: Aperture phase distribution (in radians) demonstrating the aliasing effect on a 128 by 128 grid

7.3 Investigation of the Effect of Near Field and Aliasing

In this example, we first create the beam patterns which correspond to that measured by a real antenna while steering in azimuth and elevation, with the transmitting source located at 250 m distance.

To do that, we make use of the program "flatdiff2d.f" in directory "diffraction", as follows:

!—— Input Parameter List for flatdiff2d.f ————-
!
! (Note: the input values always start at the 50th column)
! (Note: real numbers MUST be input with the decimal point in)
!
Filename for panel vertex coordinates... nodes.table
Filename for output amplitude map....... faramp.dat
Filename for output phase map........... farphase.dat
Dimension nbeam of the beam array....... 64
*! (Total beam array dimension = 2*nbeam.*
Dimension naper of the aperture array... 128
! (Total aperture array dimension =
*! 2 * naper + 1)*
!
!
!——————— parameters of the panels ————
!
Radius of the first ring of nodes(m).... .175
Radius of the second ring of nodes(m)... .974
Radius of the third ring of nodes(m).... 1.679
Radius of the fourth ring of nodes(m)... 2.356
Radius of the fifth ring of nodes(m).... 3.0
Number of panels in the first ring...... 12
Number of panels in the second ring..... 12
Number of panels in the third ring...... 24
Number of panels in the fourth ring..... 24
!
!——————- parameters of the telescope ———-
!
Focal length of the primary mirror(m).. 2.52
Diameter of the primary mirror (m)..... 6.0

Diameter of the primary mirror (m)..... 0.35
Nyquist sampling rate 0.75
Freq. of holography observation (GHz).. 92.4
Theta offsec from elevation axis (rad). -0.1309
Distance to the transmitter........... 250.
Telescope magnification 33.33
Amount of defocus (meter)............. 3.9e-2
!
!——————- extra features ——————————-
!
Add diffraction pattern (1) or not (0). 0
Filename of the diffraction pattern.... diff257.dat
!!! NOTE: THE DIMENSION OF THE DIFFRACTION FILE MUST BE
! COMPATIBLE WITH THE DIMENSION OF THE APERTURE ARRAY
! I.E. EQUAL TO (2 NAPER + 1)
!

where "nodes.table" is again generated by program "nodes.f" in directory "panelfit". Note that in this case the aperture grid is by convention a symmetric one around the center of the grid, i.e. its dimension is an odd number. In general, this number should not be equal to the far-field grid dimension, since if they are equal the aliasing effect may be hidden (in nature's own transform, the aperture is covered by infinitely fine grid points).

We then run the program "flatdiff2d.f", and at the run-time specify to move panel No. 50 on the aperture plane by a distance which corresponds to 1 radian of phase. The program then calculates the two-dimensional Kirchhoff's integral and evaluates the beam pattern. The result of this calculation, the amplitude and phase patterns "faramp.dat" and "farphase.dat", are then copies to directory "inversion", and after filling out the parameter file "withphase.prm", as follows,

!—— Input Parameter List for Withphase Algorithm ————-
! (Note: the input values always start at the 50th column)
! (Note: Real numbers must be written with a decimal point)
!
!———— parameters of the observations ————-
!
Far field amplitude file name .... faramp.dat
Far field Phase file name .... farphase.dat
Aperture field amplitude file name ..... Ea.dat
Aperture Field phase file name .... Ep.dat

Aperture Field residual phase file ..... Epr.dat
Size N of the N by N data file 128
Nyquist sampling rate 0.75
Range of the transmitter (meter) .... 250.
Amount of defocus (meter) 3.9e-2
Observing frequency (GHz) 92.4
Sampling interval (arcsec) 83.7
!
!—————— parameters of the telescope ————
!
Diameter of primary (m) 6.0
Diameter of secondary (m) 0.35
Focal length of primary (m) 2.520
Magnification of the Cassegrain system . 33.33
Diffraction file or iondish file diff128.dat
!
!————— parameters of the numerical scheme ————
!
Number of Large Scale Fitting Parameters 5
!

we run the program "holis.f", and obtain the aperture phase distribution (without taking out the large-scale error patterns) as shown in Figure 5. We see that the effect of aliasing (energy lost outside the measured beam) is such that it caused ripples in the recovered panel movement. The rms error is about 10% for a 128 by 128 grid (large beam of course will have smaller aliasing effect). The near-field correction, on the other hand, is found to work almost perfectly.

In the above case, the aliasing effect is compounded by the fact that the raised panel is located on the edge of dish, so the aliasing ripples on the dish edge also add into the aliasing effect of the panel edge. For most of the inner panels, since the aliasing ripples are mostly symmetrically distributed around the mean, and are also localized near the edge, their effect to the actual derived panel screw motion is much smaller than the maximum level of the ripples. It is found that after blanking out the edge data (5mm width), the fitted panel movement (by running the "panelfit" program) is about 3due to aliasing is obviously going to decrease as the surface gets better aligned, and the edge effect becomes less significant.

7.4 Investigation of the Effect of Measurement Error and Noise

We use the far-field patterns of a flat aperture distribution to investigate the influence of random measurement errors on the aperture-field-recovering quality. Care must be taken, however, to add noise into the far field pattern of the "raised panel" experiment. Experience shows that, the artificially-generated noise can often create false features in the recovered aperture field. This was not found to be a problem in the actual holography experiment (Zhang et al. 1996).

The program used is the "add_noise.f" in directory "diffraction". We found that if a Gaussian random noise of 1is added to the "measured" beam pattern, the recovered aperture phase distribution has a maximum phase error about 1.1 degree. For the SMA holography system (Zhang et al. 1995), the short-term instrumental gain and phase stabilities are on that order (the long-time temperature drift will be taken out by boresight calibration), so they do not seem to be an obstacle to the achievement of the accuracy of the measurement.

7.5 Testing Misell Phase-Retrieval Algorithm

In this example, we illustrate the use of Misell phase-retrieval algorithm to recover the aperture field from two amplitude beam patterns measured at two different focus settings.

We use "pattern.f" in the "diffraction" directory to create two amplitude patterns "faramp1.dat" and "faramp2.dat", each defocused by plus and minus 1 wavelength (± 3.2445 mm at 92.4 GHz). After copying these files to the "inversion" directory, and filling out a parameter file, as follows,

!—— Input Parameter List for Misell Algorithm ——————
! (Note: the input values always start at the 50th column)
! (Note: real numbers must be written with the decimal point)
!
!—————— parameters of the observations ——————
!
! Can have up to 4 defocused maps, but the minimum is 2
!
Number of maps used 2
First Far field amplitude filename faramp1.dat

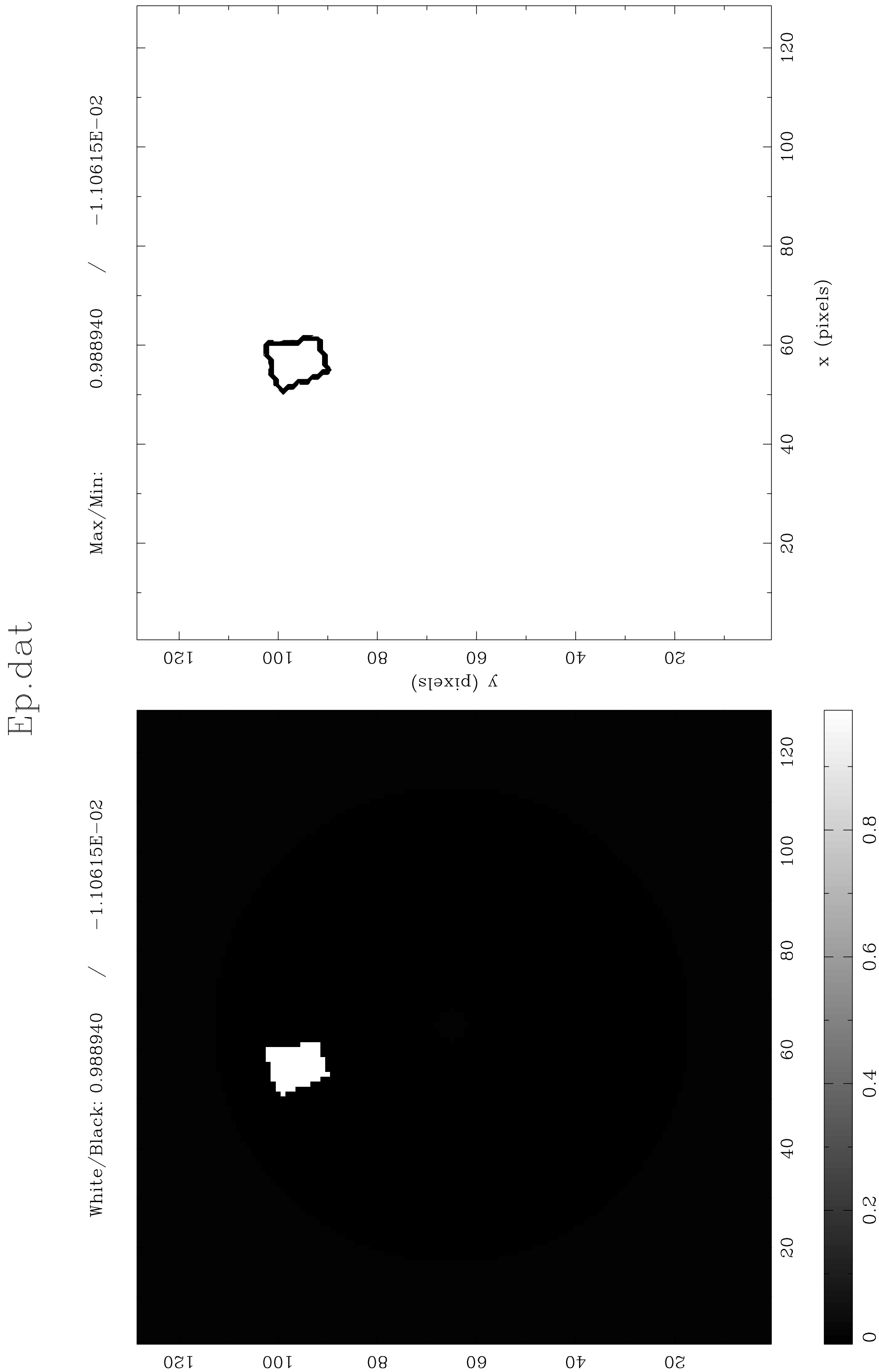

Figure 6: Aperture phase recovered using the Misell algorithm. Two defocused far-field patterns were used as input.

Second Far field amplitude filename faramp2.dat
Third Far field amplitude filename .....
Fourth Far field amplitude filename ....
First defocus value (m) 0.0032445
Second defocus value (m) .... -0.0032445
Third defocus value (m) ....
Fourth defocus value (m) ....
Aperture field amplitude file name ..... Ea.dat
Aperture Field phase file name .... Ep.dat
Aperture Field residual phase file ..... Epr.dat
Size N of the N by N data file 128
Nyquist sampling rate 0.75
Distance to the transmitter (meters) ... 250.e10
Observing frequency (GHz) 92.4
Sampling interval (arcsecs) 83.7
!
!———————— parameters of the telescope ——————
!
Diameter of primary (m) 6.0
Diameter of secondary (m) 0.35
Focal length of primary (m) 2.520
Magnification of the Cassegrain system . 33.33
Diffraction file or iondish file diff128.dat
!
!————— parameters of the numerical scheme ——————
!
Number of Large Scale Fitting Parameters 5
Maximum number of iterations 200
!
!————— parameters of the initial guess aperture ———
!
Aperture edge taper in dB .... 0.
Random phase magnitude (radian) 0.2
Number of maps with different seeds ... 1
The first initial seed 999
!

we run "holis" with the second option for Misell phase retrieval. After 200 iterations, we obtain the aperture phase distribution as shown in Figure 6. It can be seen that the convergence in this case, where the far-field patterns are used and where there is no aliasing error, is quite excellent. This is partly the result of our enforcing the dish and subreflector mask at each iteration step.

Without such extra constraints the convergence was not nearly as good (there was about 10% error in the recovered panel movement).

When the aliasing effect is present (as is the case in practice), i.e. if we obtain the two amplitude beam patterns rather through the Kirchhoff's integral, the Misell algorithm has a much harder time to converge to the correct solution, especially if we still use the Fourier transform when iterating between the aperture and far-field planes – in effect, we are trying to force the aperature and far-field solutions to be that related by the wrong transform pair and still have the correct form, which of course is hard to obtain except in very few lucky situations (mostly when the panel errors are small and the antenna surface is already more or less aligned). Therefore, it appears that some of the difficulty in the convergence properties of Misell algirithm observed in practice is due to compounding with the aliasing effect, and is not necessarily intrinsic to the Miseel algorithm itself, especially if we apply the correct masking constraint at each iteration, as our above example has shown.

Appendix. Programs, Subroutines and Functions

Routines Used in Directory "Preprocessing"

The directory "**preprocessing**" contains the following files:

- makefile

Fortran routines

- PROGRAM preprocess.f
  PERFORMS THE FOLLOWING FUNCTIONS: (1) CORRECTS FOR WAVEFRONT CURVATURE TO DETERMINE THE EQUIVALENT REFERENCE PLANE (NORMALLY CHOSEN TO BE HALF-WAY INTO THE DEPTH OF THE DISH). (2) ENLARGES THE DATA ARRAY SIZE TO INCREASE RESOLUTION AT THE APERTURE PLANE (NORMALLY THE INCOMING 128 BY 128 DATA SET PROVIDES SUFFICIENT RESOLUTION SO THIS IS NOT DONE). (3) WEIGHTING OF THE FAR-FIELD DATA ARRAY FOR REDUCING ALIASING.

- SUBROUTINE read_data.f
  READS IN RAW AMP AND PHASE

- SUBROUTINE readprm_pre.f
  READS PARAMETERS IN WITHPHASE CALCULATION

- SUBROUTINE reference.f
  DETERMINES THE PHASE REFERENCE PLANE

- SUBROUTINE sizeconv.f
  CONVERTS THE SMALL APERTURE ARRAY TO A LARGER ONE BY PADDING ZEROS IN THE OUTER PART, TO INCREASE APERTURE PLANE RESOLUTION

- smooth.f
  SMOOTHS THE AMPLITUDE FAR FIELD ARRAY

- write_data.f
  WRITES OUTPUT AMPLITUDE AND PHASE TO DATA FILES

Include files

- holis.inc

Parameter files

- preprocess.prm

Routines Used in Directory "Inversion"

The directory "**inversion**" contains the following files:

- makefile

Fortran routines

- PROGRAM holis.f
  CALCULATES THE APERTURE FIELD OF AN ANTENNA FROM THE MEASURED RADIATION PATTERNS, USING DIFFERENT TYPES OF "HOLOGRAPHY" TECHNIQUES.

  THREE TYPES OF ALGORITHMS ARE IMPLEMENTED. THESE ARE:

  (1) STANDARD PHASE-COHERENT HOLOGRAPHY, WHICH REQUIRES 1 FAR-FIELD AMPLITUDE MAP AND 1 FAR-FIELD PHASE MAP. (2) MISELL'S PHASE-RETRIEVAL ALGORITHM, WHICH REQUIRES 2 FAR-FIELD AMPLITUDE MAPS OBTAINED AT DIFFERENT FOCUS SETTINGS. (3) GLOBAL CHI-SQUARED FITTING APPROACH. IT CAN TAKE N FAR FIELD AMPLITUDE MAPS, AS WELL AS N PHASE MAPS, OBTAINED AT DIFFERENT FOCUS SETTINGS. WEIGHT CAN BE ADJUSTED BETWEEN THE AMPLITUDE AND PHASE DATA USED. IN PARTICULAR, ZERO WEIGHT TO THE PHASE CORRESPONDS TO USING THE DEFOCUSED AMPLITUDE MAPS ONLY FOR THE FITTING. (THIS LAST ALGORITHM IS YET TO BE IMPLEMENTED)

- SUBROUTINE apply_mask.f
  MASKS THE COMPLEX APERTURE DATA FILE WITH THE DATA POINTS OUTSIDE THE MAIN REFLECTOR AND INSIDE THE SUBREFLECTOR PUT TO ZERO

- SUBROUTINE apply_mask1.f
  MASKS THE APERTURE PHASE FILE WITH THE DATA POINTS OUTSIDE THE MAIN REFLECTOR AND INSIDE THE SUBREFLECTOR PUT TO ZERO

- SUBROUTINE apply_mask2.f
  MASKS THE COMPLEX APERTURE DATA FILE WITH THE DATA POINTS OUTSIDE THE MAIN REFLECTOR AND INSIDE THE SUBREFLECTOR PUT TO ZERO (ONE LESS RING OF DATA MASKED THAN apply_mask.f)

- SUBROUTINE apply_mask3.f
  MASKING THE APERTURE DATA FILE WITH THE DATA POINTS OUTSIDE THE MAIN REFLECTOR AND INSIDE THE SUBREFLECTOR PUT TO ZERO. IT ALSO INCLUDES STRUTS,SHADOW,AND SUBREFLECTOR SUPPORT STRUCTURE MASKS FOR THE SMA ANTENNAS

- SUBROUTINE c_pass.f
  PASSES DATA BETWEEN TWO COMPLEX ARRAIES

- SUBROUTINE change_amplitude.f
  SETS THE COMPLEX ARRAY VALUE TO ZERO FOR ZERO AMPLITUDE POINTS

- SUBROUTINE check_converge.f
  CHECKS CONVERGENCE OF MISELL. CURRENTLY USES ONLY MAX ITERATION NUMBER CHECK

- SUBROUTINE create_guess.f
  CREATES THE INITIAL GUESS OF THE COMPLEX APERTURE PATTERN FOR THE MISELL ALGORITHM, USING THE SPECIFIED EDGE TAPER AND RANDOM PHASE FLUCTUATION

- SUBROUTINE cumulate.f
  SUMS THE VALUE OF COMPLEX ARRAYS TO ARRAY CCUM

- SUBROUTINE defocus.f
  CREATES DEFOCUSED COMPLEX APERTURE PATTERN

- FUNCTION defocus1.f
  CALCULATES PHASE VALUE AT RADIUS radius1 ON THE APERTURE

- SUBROUTINE divide.f
  PERFORMS SIMPLE DIVISION OF A COMPLEX ARRAY BY AN INTEGER

- SUBROUTINE fft.f
  A TWO-DIMENSIONAL,RADIX 2,DECIMATION IN FREQ. FFT ALGORITHM. WRITTEN BY BERNARD ARAMBEPOLA,ENGINEERING DEPT.,TRUMPINGTON ST. CAMBRIDGE, CB2 1PZ ,ENGLAND

- SUBROUTINE fftshift.f
  SHIFTS ARRAY VALUES AFTER FFT TO CONFORM TO DFT CONVENTION

- SUBROUTINE func2.f
  AN AUXILIARY ROUTINE FOR DEFOCUS

- SUBROUTINE gasdev.f
  A NUMERICAL RECIPES ROUTINE FOR CALCULATING NORMALLY DISTRIBUTED DEVIATE

- SUBROUTINE initialize.f
  INITIALIZES A COMPLEX ARRAY WITH ZEROS

- SUBROUTINE nearfield.f
  PERFORMS SECOND-ORDER NEAR FILED CORRECTION

- SUBROUTINE phasefit.f
  PERFORMS LEAST SQUARE FITTING ON APERTURE PHASE LARGE SCALE ERROR TERMS

- FUNCTION pythag1.f
  A NUMERICAL RECIPES FUNCTION FOR COMPUTING THE SQUARE ROOT OF THE SUM SQUARES OF TWO VALUES

- FUNCTION ran1.f
  A NUMERICAL RECIPES FUNCTION FOR GENERATING RANDOM NUMBERS

- SUBROUTINE read_amm.f
  READS IN FOUR DEFOCUSED AMPLITUDE FILES

- SUBROUTINE read_amph.f
  READS IN AMPLITUDE AND PHASE
- SUBROUTINE read_defocus.f
  READS IN TWO DEFOCUSED AMPLITUDE FILES
- SUBROUTINE read_diff.f
  READS IN DIFFRACTION PATTERN
- SUBROUTINE read_prm1.f
  READS PARAMETERS FOR USE IN WITHPHASE CALCULATION
- SUBROUTINE read_prm2.f
  READS PARAMETERS FOR USE IN MISELL CALCULATION
- SUBROUTINE rms.f
  FINDS RMS PHASE ACROSS APERTURE
- SUBROUTINE separate_ap.f
  SEPARTES THE AMPLITUDE AND PHASE FROM THE COMPLEX APERTURE ARRAY
- SUBROUTINE svbksb1.f
  A NUMERICAL RECIPES ROUTINE FOR SOLVING LINEAR SYSTEMS
- SUBROUTINE svdcmp1.f
  A NUMERICAL RECIPES ROUTINE FOR SOLVING LINEAR SYSTEMS USING SINGULAR VALUE DECOMPOSITION
- SUBROUTINE svdfit1.f
  MODIFIED NUMERICAL RECIPES ROUTINE FOR CHI-SQUARE FITTING USING SINGULAR VALUE DECOMPOSITION
- SUBROUTINE undefocus.f
  UN-DOES THE DEFOCUS ON APERTURE PATTERN
- SUBROUTINE write_amph.f
  WRITES AMPLITUDE AND PHASE ARRAYS INTO DATA FILES
- SUBROUTINE write_resiph.f
  WRITES RESIDUAL PHASE INTO DATA FILE

Include files

- holis.inc

Parameter files

- withphase.prm
- misell.prm

Plotting files

- holograf
  AN EXECUTABLE FILE BASED ON PGPLOT, FOR PLOTTING N BY N ARRAIES.
- grfont.dat
  FONT FILE USED IN CONJUNCTION WITH holograf

Routines Used in Directory "Panelfit"

The directory "**panelfit**" contains the following files:

- makefile

Fortran routines

- PROGRAM nodes.f
  GENERATES THE POLAR COORDINATES OF THE VERTEXES AND SCREWS OF THE PANELS FOR SMA ANTENNAS
- PROGRAM panelfit.f
  GENERATES THE TABLE OF SCREW MOVEMENTS FOR EACH PANEL. ALL 72 PANELS OF SMA ANTENNAS ARE FITTED AT THE SAME TIME. THE INPUT IS THE RESIDUAL PHASE MAP AT THE REFERENCE PLANE OF THE ANTENNA, WHICH IS THE OUTPUT FROM "HOLIS.F". THE COORDINATE TABLES OF VERTEX COORDINATES AND SCREW COORDINATES MUST ALREADY BE STORED IN "NODES.TABLE" AND "SCREWS.TABLE" BEFORE

THIS PROGRAM CAN BE RUN. THE OUTPUT IS IN THE FORM OF SCREW MOTIONS PERPENDICULAR TO THE ANTENNA SURFACE FOR EACH PANEL, AND ARE STORED IN "PANELFIT.TABLE" (IN THE TABLE, POSITIVE NUMBERS SUGGEST OUTWARD MOVEMENT OF PANELS TOWARDS SUBREFLECTOR, NEGATIVE NUMBERS SUGGEST INWARD MOVEMENT TOWARDS THE BACKUP STRUCTURE. CONSISTENT WITH THE CONVENTION OF D.MORRIS)

- SUBROUTINE errorcal.f
  CALCULATES THE SURFACE ERROR ARRAY FROM THE PHASE ERROR ARRAY

- SUBROUTINE newerror.f
  CALCULATES THE NEW ERROR ARRAY AFTER PANELFIT

- FUNCTION pythag.f
  A NUMERICAL RECIPES FUNCTION FOR COMPUTING THE SQUARE ROOT OF THE SUM SQUARES OF TWO VALUES

- SUBROUTINE read_resiphase.f
  READS IN RESIDUAL PHASE FILE

- SUBROUTINE readprm_pan.f
  READS PARAMETERS FOR PANEL FITTING PROGRAMS

- SUBROUTINE search.f
  SEARCHES FOR POINTS ON A PARTICULAR PANEL DEFINED BY RMAX, RMIN, THETAMAX, THETAMIN. WRITE THE RESULT OUT TO 1D ARRAYS X,Y,ERROR

- SUBROUTINE surfacerms.f
  FINDS RMS SURFACE ERROR ACROSS APERTURE

- SUBROUTINE svbksb.f
  A NUMERICAL RECIPES ROUTINE FOR SOLVING LINEAR SYSTEMS

- SUBROUTINE svdcmp.f
  A NUMERICAL RECIPES ROUTINE FOR SOLVING LINEAR SYSTEMS USING SINGULAR VALUE DECOMPOSITION

- SUBROUTINE svddrv.f
  MODIFIED NUMERICAL RECIPES ROUTINE FOR SOLVING THE COEFFICIENTS OF ERROR FUNCTION FOR A PARTICULAR PANEL

FROM THE KNOWN LOCATION ARRAYS X,Y AND ERROR ARRAY ERROR, USING THE SINGULAR VALUE DECOMPOSITION ROUTINES SVDCMP AND SVBKSB

- SUBROUTINE write_error.f
  WRITES OUT THE ERROR MAP ON THE APERTURE

- SUBROUTINE write_resierror.f
  WRITES OUT THE RESIDUAL ERROR MAP ON THE APERTURE

Include files

- holis.inc

Parameter files

- panelfit.prm

Routines Used in Directory "Diffraction"

The directory "**diffraction**" contains the following files:

- makefile

Fortran routines

- PROGRAM pattern.f
  CREATES FAR-FIELD PATTERNS BY (1) MASKING THE APERTURE DATA FILE WITH THE DATA POINTS OUTSIDE THE MAIN REFLECTOR AND INSIDE THE SUBREFLECTOR PUT TO ZERO (2) ALSO INCLUDES DEFOCUS (3) HAS OPTION OPTION FOR BLOCKING STRUTS, SHADOW, AND SECONDARY SUPPORT STRUCTURE FOR SMA ANTENNA (4) HAS OPTION FOR PANEL MOVEMENT

- PROGRAM flatdiff1d.f
  CALCULATES THE 1-D DIFFRACTION PATTERN OF ONE FLAT SURFACE

- PROGRAM backtrans1d.f
  USES THE OUTPUT OF "FLATDIFF1D.F", PERFORMING A ONE-DIMENSIONAL FOURIER TRANSFORM AND NEAR-FIELD CORRECTION TO GET APERTURE AMPLITUDE AND PHASE.

- PROGRAM flatdiff2d.f
  CALCULATES THE 2-D DIFFRACTION PATTERN OF A FLAT ANTENNA SURFACE AT ANOTHER SPHERICAL SURFACE A DISTANCE R AWAY (DISTANCE MEASURED FROM THE CENTERS OF BOTH SURFACES). BY DEFAULT, THE AMPLITUDE IS 1 INSIDE THE ANTENNA SURFACE (EXCLUDING THE SECONDARY) AND ZERO EVERYWHERE ELSE. THE PHASE PATTERN IS ZERO EVERYWHERE TO BEGIN WITH HAS OPTION TO HAVE A INITIAL DEFOCUSED PHASE PATTERN ON THE ANTENNA SURFACE HAS OPTION TO RAISE THE PHASE OF A SPECIFIED ANTENNA PANEL BY ARBITRARY AMOUNT OF PHASE, BOTH SPECIFIED IN THE .PRM FILE

- PROGRAM subdiff.f
  A SCALAR CALCULATION OF SECONDARY MIRROR DIFFRACTION PATTERN IN A CASSEGRAIN SYSTEM. BY RICHARD HILLS, MRAO, CAVENDISH LAB, CAMBRIDGE

- PROGRAM diff2d.f
  CONVERTS THE OUTPUT OF THE 1-DIMENSIONAL DIFFRACTION CALCULATION OF PROGRAM SUBDIFF.F (DIFF.DAT) INTO A 2D DIFFRACTION MAP (DIFF2D.DAT)

- PROGRAM add_noise.f
  ADDS VARIOUS AMOUNT OF NOISE TO THE BEAM PATTERN

- PROGRAM spar
  A PROGRAM WRITEN BY COLIN MASSON WHICH CALCULATES THE BLOCKAGE OF SPARS

- PROGRAM twod2oned.f CONVERTS A TWO DIMENSIONAL ARRAY INTO AZIMUTHALLY-AVERAGED ONE-DIMENSIONAL ARRAY.

- SUBROUTINE add_diff.f
  ADDS SECONDARY MIRROR DIFFRACTION PATTERN ONTO THE APERTURE PHASE

- SUBROUTINE create_phase.f
  CREATES INITIAL PHASE PATTERN ON THE APERTURE

(NORMALLY CIRCULAR) AT ANOTHER SPHERICAL SURFACE A DISTANCE R (FROM THE CENTER OF THE FLAT SURFACE) AWAY.

- SUBROUTINE gasdev.f
  A NUMERICAL RECIPES ROUTINE FOR CALCULATING NORMALLY DISTRIBUATED DEVIATE
- SUBROUTINE panelphase.f
  CREATES COMPLEX PANEL PHASE ARRAY FROM A REAL PHASE ARRAY
- FUNCTION ran1.f
  A NUMERICAL RECIPES FUNCTION FOR GENERATING RANDOM NUMBERS
- SUBROUTINE readprm.f
  READS PARAMETERS FOR FLATDIFF2D.F
- SUBROUTINE search_raise.f
  SEARCHES FOR POINTS ON A PARTICULAR PANEL DEFINED BY RMAX, RMIN, THETAMAX, THETAMIN. CHANGE THE PHASE OF POINTS IN SPECIFIED BOUNDARY BY THE SPECIFIED AMOUNT
- SUBROUTINE write_phase.f
  WRITES APERTURE PHASE INTO OUTPUT FILE

Include files

- holis.inc

Parameter files

- flatdiff2d.prm
- pattern.prm

References

Morris, D. 1985, IEEE Trans. Ap-33, 749

Ruze, J. 1969, "Small Displacements in Parabolic Reflectors," Research Report, MIT Lincoln Laboratory

Zhang, X., Levine, M., Bratko, P., Test, J., Papa, C. and Masson, C. 1995, "Planned Panel Alignment Procedure for the SMA Antennas Using the Microwave Holography Technique," Submillimeter Array Technical Memorandum, No. 85

Zhang, X. 1995, "Certain Optics Considerations for the Holography Experiment," Submillimeter Array Techical Memorandum, No. 86

Zhang, X., Bratko P., Oberlander D., Patel N., Sridharan T. K., and Stark, A.A. 1996, "First Result of the SMA Holography Experiment", Submillimeter Array Techical Memerandum, No. 102