1973

A computerized acoustic imaging technique incorporating automatic object recognition.

Mueller, Dieter Erwin.

Monterey, California. Naval Postgraduate School

A COMPUTERIZED ACOUSTIC IMAGING TECHNIQUE INCORPORATING AUTOMATIC OBJECT RECOGNITION

Dieter Erwin Mueller

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A Computerized Acoustic Imaging
Technique Incorporating Automatic Object Recognition

by

Dieter Erwin Mueller

Thesis Advisor: J. P. Powers

March 1973

Approved for public release; distribution unlimited.

A Computerized Acoustic Imaging Technique Incorporating Automatic Object Recognition

by

Dieter Erwin Mueller
Lieutenant Commander, Federal German Navy

Submitted in partial fulfillment of the
requirements for the degree of

ELECTRICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
March 1973

ABSTRACT

A computerized technique combining backward wave propagation and an automatic edge detection scheme is developed and tested. The class of objects considered is limited to those with edge boundaries since it is shown that a universal automatic reconstruction cannot be obtained for all possible objects. Using samples of the acoustic diffraction pattern as input this technique enables the computer to predict the most likely locations of objects and to produce graphical output of the objects. A discussion is given on implementing the diffraction equations on the computer. A simplified edge detection scheme conserving both memory space and computer time is described. Test results are presented for both computer generated diffraction patterns and one set of experimental data.

TABLE OF CONTENTS

LIST OF DRAWINGS

I. MOTIVATION

Since Dennis Gabor [Ref. 1] discovered holography - recording the amplitude and phase of the diffraction pattern of an object with the help of a reference wave and the subsequent reconstruction of the original object from this hologram - vast advances have been made in this field. Originally confined to optics holography has been extended to the fields of acoustics and microwaves; comparisons between the results obtained by the pioneers in holography and those of recent vintage are striking with the promise of even better results to come. (For a description of the development of holography see Gabor's lecture on the occasion of receiving the 1971 Nobel Prize for Physics in Stockholm, Dec. 13, 1971, reprinted in [Ref. 2]).

The interest in acoustical holography is motivated by the search for a means of probing media which cannot be investigated by any other method. For example, electro-magnetic waves have a very short propagation distance under water (those that can be propagated for a longer distance at very low frequency, have much too low a resolution to be of any significance for imaging). In murky water optical wavelengths cannot be used at all. In another application it has been found that X-rays which give a clear indication of the bony structure of a body are not very good at making distinction between two different kinds of soft tissue (e.g., in obstetrics). Acoustic waves however can show a difference in absorption for different soft tissues and do not result in damage to the insonified body at reasonable intensities. Investigation by insonification and interpretation of the reflected sound has long been used in probing the earth's crust and in nondestructive testing.

The application of holographic concepts brings to these fields the possibility of obtaining information that not only indicates the existence of an object of interest but also shows many of its features. The same statement holds for the advantages of acoustical holography versus the well known echo-ranging concepts as expressed in sonar. The result of the holographic approach is an image of the object, whereas without holography only detection of the object is realized. This is not meant to imply that the holographic approach is the only one feasible: Imaging systems incorporating acoustic lenses have also been developed with good results. (For the development and state of the art of acoustical holography the reader is referred to [Ref. 3]).

In general illumination of a recorded hologram with a duplicate of the reference wave results in two reconstructed object wavefronts, one being an exact duplicate of the original object wave (producing the "true" image), while the other wavefront is the complex conjugate of the object wave (the "conjugate" image). If the two obtained wavefronts are not spatially separated from each other by some suitable arrangement, they can interfere with each other which results in distortion of the images (the so-called twin-image problem).

Unfortunately it is not possible to realize in acoustical holography that most striking feature of optical holograms, namely the three-dimensional sensation experienced by observing the true image with both eyes. In order to be able to retain the depth information, the wavelengths of constructing and reconstructing beam have to be equal or nearly so. A hologram can be considered to be a sum of zone plates into which the object information is incorporated. (See [Ref. 4]). It then has an equivalent focal length f which is given by

$$f = \left(\frac{1}{v} - \frac{1}{u} \right)^{-1}$$

where v and u are object and reference source distance respectively, measured from the hologram plane along the line joining object and reference source. See Fig. 1-1.

Then if the linear dimension of the hologram is changed by a factor m from X to X^1, (i.e., $X^1 = mX$), and the wave length λ^1 used in reconstruction differs from the original wavelength λ by $\mu = \lambda^1/\lambda$, the effective equivalent focal length of the hologram is given by

$$f^1 = \pm \left(\frac{1}{v^1} - \frac{1}{u^1}\right)^{-1}$$

where $f^1 = \mu \cdot f/m^2$

v^1 and u^1 are the distance of reconstructed image and illuminating source respectively, measured as before, and the + sign gives the position of the true image, the - sign gives the position of the conjugate image. This gives rise to axial magnification

$$M_A = \frac{(\mu/m^2)\ (1/v^2)}{\{\pm\ (\mu/m^2)\ [(1/v) - (1/u)] + 1/u^1\}^2}$$

and transverse magnification

$$M_T = m \left(\frac{u}{u-v} \right) \left(1 - \frac{v^1}{u^1} \right)$$

These magnifications are different from one another unless $m = 1/\mu$ and thus give rise to distortion of the reconstructed image, if this condition is not met (as is usually the case). In acoustical holography the reconstruction wavelength is optical. The sound wavelengths used in the recording process are much longer and μ then is of the order of 10^{-3}. To avoid distortion this would mean reducing the size of an acoustical hologram by this same factor, i.e., a hologram of 1m diameter would be reduced to 1mm diameter. Naturally one cannot

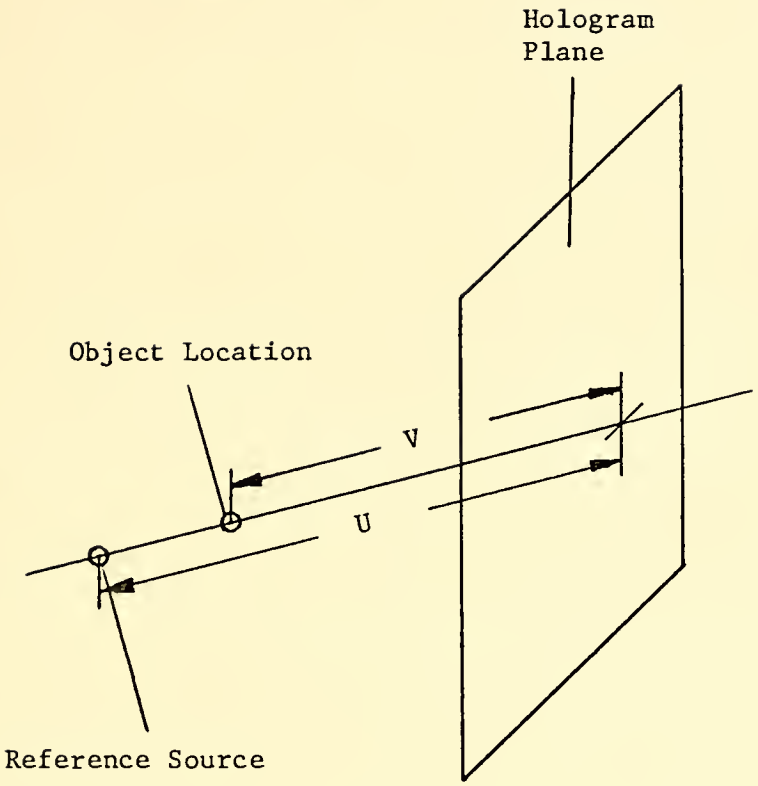


Figure 1-1

Geometry of General Hologram

expect to experience any three-dimensional illusion in looking at a hologram of this small size. Optical magnification of this small hologram introduces longitudinal distortion exactly analogous to that being eliminated by the scaling due to depth of field problems. This longitudinal distortion is one of the primary disadvantages of acoustical holography.

Another disadvantage encountered in acoustical holography is the absence to date, inspite of intensive research, of a recording medium which gives the analogous fine grain structure as that encountered in optical holography by using spectroscopic plates. This shortcoming is in part balanced by the possibility of recording both amplitude and phase of a sound wave impinging on a linear detector directly. Because of the presence of linear acoustic detectors (one of the fundamental differences of acoustical holography versus optical holography), it is not necessary to use an acoustic reference beam and to work with intensities as in optical holography. It might be argued that a direct recording of amplitude and phase of a diffraction pattern is not really a hologram (according to definition) because of the lack of the reference beam. But as a hologram is after all an ingenious method of recording a complex wavefront using a medium that is insensitive to phase and as the recorded diffraction pattern contains the same information as a hologram, only in simpler form, there is no difference in the information content of the two types of recording. Working with amplitude and phase can also eliminate the twin-image problem referred to earlier.

Assuming then the existence of a record of the amplitude and phase of the sound diffraction pattern due to some source (self-emitting or reflecting), in sampled form, whether obtained by means of an array of

detectors or by scanning the sound field or by any other means, it is possible to circumvent the difficulties of optical reconstruction (not only depth distortion, but also those due to degradation because of nonlinearities of the film or cathode ray tube, etc. as in [Ref. 5]) by using a digital computer technique to restore the original object. However there will be errors introduced by quantization, sampling of the complex amplitude and nonuniformities in the scan if such is used. These effects are analyzed in [Ref. 5]. Of course the digital computer is limited in the amount of data it can handle, whereas the optical hologram is comparatively unlimited in the amount of information stored. Thus in increasing the number of independent samples of the wavefront eventually a point is reached in computer reconstruction beyond which the amount of storage and computation time becomes unacceptable. Another disadvantage of digital reconstruction is that perfect reconstruction can only be achieved for planar objects. The results of trying to reconstruct a three-dimensional acoustical object are identical to observing the real image in a reconstruction of an optical hologram. The shape of a three-dimensional object can only be reconstructed in a series of sections through its real image.

One of the greatest drawbacks of digital reconstruction of objects from their diffraction pattern is that the distance from hologram to its generating object has to be known. This problem never occurs in the case of visual reconstruction of an object from its virtual image because the focus is found automatically by an eye - brain interaction. It is the same problem as that experienced in attempting to bring a real image to a focus on a screen, with the difference that the movement of the screen can be continuous and information on the validity of the placement is fed back by looking at the result and interpreting

it to be more or less correct than before. The final screen location is determined by the compatibility of the retinal image with some object hypothesis stored in the brain's memory. In the field of eye-brain interaction great amounts of research effort have been and still are expended; still the processes are not completely understood to date. It is therefore hardly promising to try and simulate this interaction on the digital computer. Nevertheless it would be of great help to find a method which brings about automatic focusing - or at least narrows down that region in space in which the object might be present. Problems can be envisioned in which acoustical holograms from quite a large number of different objects might be taken. Then one would not want to occupy a large number of specialists in the given field by having them look at the results of each backward propagation step but would rather have the computer make a preliminary decision on the object location and present this together with the computed sound field at that location. A single observer could then - using his pattern recognition ability - verify the computer decision or discard it. Naturally, computer time being expensive, the economics of either scheme have to be considered. In any case the computer program should take as little time and/or memory capability as possible, an obvious requirement for any efficient algorithm.

This then is the problem to be solved in this investigation: Given a sampled sound diffraction pattern it was required to find a method for a digital computer to make a tentative decision on the object location. In all that follows a diffraction pattern will be assumed

(except where stated otherwise), that results from propagation through a medium that is linear and not disturbed by turbulence or convection. In the following section the necessary theory for this investigation is introduced.

II. SCALAR THEORY OF DIFFRACTION

Diffraction is generally treated with a scalar wave theory. This is quite applicable for acoustical disturbances, more so than for electromagnetic waves which should rigorously be considered in a vectorial theory of diffraction as their electric and magnetic vector components are not independent but coupled through Maxwell's equations.

A. DIFFRACTION BY A PLANE SCREEN

The solution of the problem of diffraction by a plane screen is treated in numerous books, e.g., [Ref. 6]. As it in itself is not a suject of this investigation only those results which are essential for the further treatment are stated below. Figure 2-1 shows the geometry of the problem.

The resulting complex amplitude $\underline{U}\ (P_o)$ at an observation point P_o behind an impenetrable screen S, with an aperture A, which is insonified by a disturbance $\underline{U}$ can be written as

$$\underline{U}\ (P_o) = \frac{1}{4\pi} \int_{s_1} \left(\frac{\partial \underline{U}}{\partial \hat{n}} \cdot \underline{G} - \underline{U} \frac{\partial \underline{G}}{\partial \hat{n}} \right) ds \qquad (2\text{-}1)$$

where $\underline{G}$ is the so-called Green's Function, which is chosen here to be a spherical wave expanding about the point P_o and $\partial/\partial\hat{n}$ is the derivative in the direction of the outward normal to the plane of S.

The difficulty encountered now is that the boundary conditions which should be applied to the solution of Eq. (2-1) are never known exactly, for even though the screen is assumed to be impenetrable to sound (except at its opening A), diffraction is known to take place,

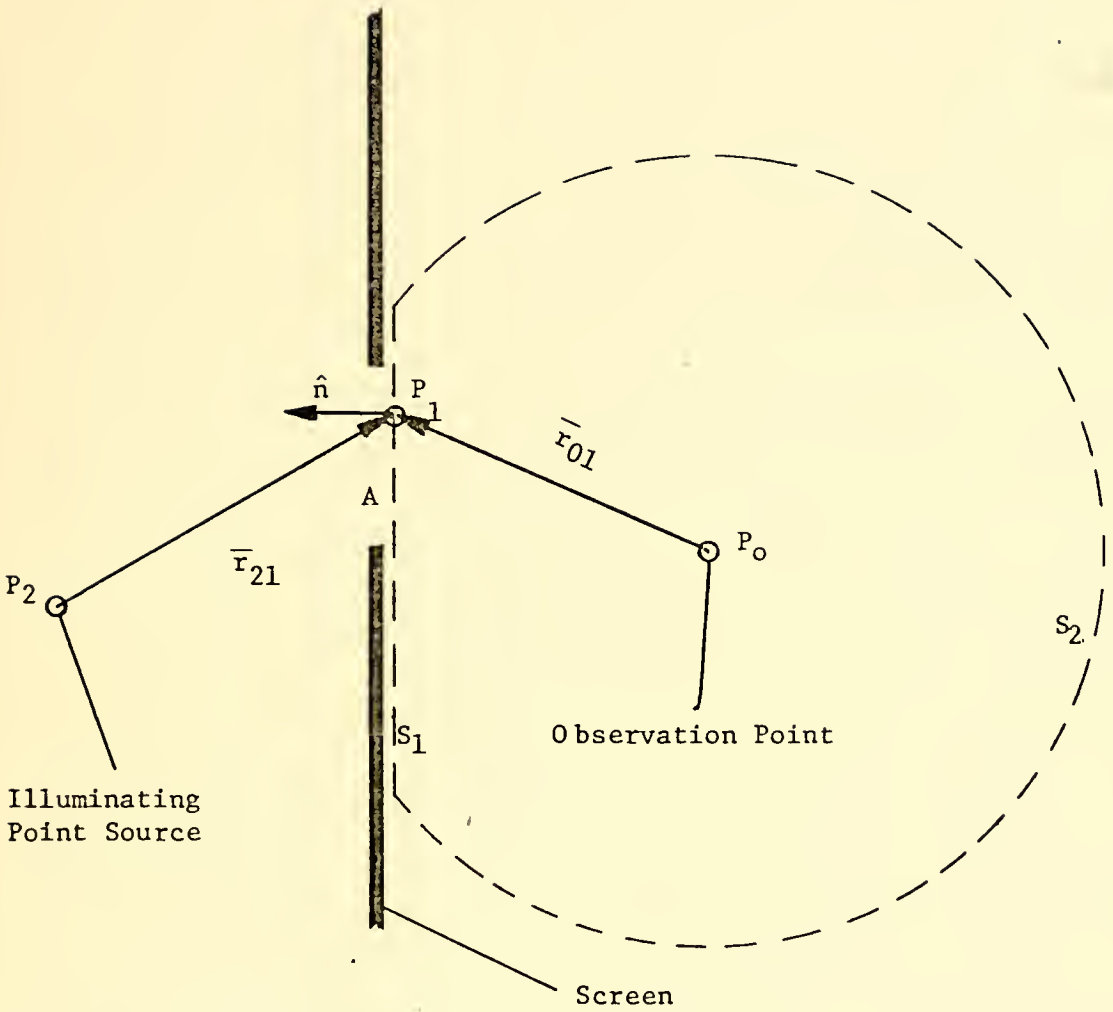


Figure 2-1
Geometry of Diffraction by a Plane Screen

the exact expression for which is to be found from Eq. (2-1). To simplify the problem, Kirchhoff set up the following conditions on S_1:

1. Across the aperture A the field distribution $\underline{U}$ and $\partial\underline{U}/\partial\hat{n}$ are the same as they would be without the screen.

2. Over that portion of S_1 that lies in the geometrical shadow. $\underline{U}$ and $\partial\underline{U}/\partial\hat{n}$ are zero.

It is noted that these boundary conditions are only approximations. Still Kirchhoff's diffraction theory leads to very good results which are widely used in practice.

Without going into the details (see [Ref. 6]) Kirchhoff's diffraction equation for the plane screen insonified by a unity amplitude spherical wave emanating from P_2 is,

$$\underline{U}(P) = \frac{1}{2j\lambda} \iint_A \frac{\exp[jk(r_{21} + r_{01})]}{r_{21} \cdot r_{01}} [\cos(\hat{n},\bar{r}_{01}) - \cos(\hat{n},\bar{r}_{21})]ds \tag{2-2}$$

where $(\hat{n},\bar{r}_{01})$ indicates the angle between the indicated vectors. Another severe inconsistency of the Kirchhoff boundary conditions is that by assuming a value of zero for both $\underline{U}$ and $\partial\underline{U}/\partial\hat{n}$ the problem is over-specified and requires the field to be zero over all space which clearly is not the case. Considerable criticism has therefore been raised against Kirchhoff's treatment. Sommerfeld removed this difficulty by assuming a Green's function which makes $\underline{G}$ or $\partial\underline{G}/\partial\hat{n}$ identically zero over all of S_1, without invalidating the development which leads to Eq. (2-1) (for details see [Ref. 7]). His result is incorporated into the Rayleigh-Sommerfeld diffraction formula:

$$\underline{U}(P_o) = \frac{1}{j\lambda} \iint_A \underline{U}(P_1) \frac{\exp(jkr_{01})}{r_{01}} \cos(\hat{n},\bar{r}_{01})ds \tag{2-3}$$

where $\underline{U}\ (P_1)$ is the field distribution in the aperture plane. This can be expanded into (2-4)

$$\underline{U}\ (P_o) = \frac{Z_0-Z_1}{j\lambda} \iint_A \underline{U}\ (P_1)\ \frac{\exp(jk((x_0-x_1)^2+(y_0-y_1)^2+(Z_0-Z_1)^2)^{1/2})}{(x_0-x_1)^2+(y_0-y_1)^2+(Z_0-Z_1)^2} dx_1 dy_1$$

which shows that diffraction is a space-invariant phenomenon, as the result depends on the difference between observation point and the point at which the disturbance is located.

Diffraction also is a linear phenomenon, because the response to a sum of inputs is equal to the sum of the individual responses to each input, (within the limitations of this study). The linearity and invariance of diffraction imply that the result $\underline{U}_o$ due to some disturbance $\underline{U}$ impinging on a diffracting aperture can be written in form of a convolution integral

$$\underline{U}_o = \underline{U} * \underline{h}$$

where $\underline{h}$ is the so-called impulse response of the diffracting system. It is also possible then to write:

$$F\{\underline{U}_o\} = F\{\underline{U} * \underline{h}\} = F\{\underline{U}\} \cdot F\{\underline{h}\} = F\{\underline{U}\} \cdot \underline{H}$$

where F is the symbol for the Fourier transform, $\underline{H}$ is the transfer function of the diffracting system and use has been made of the convolution theorem.

For a unity amplitude expanding spherical wave incident on the screen Eq. (2-3) becomes

$$\underline{U}\ (P\ ) = \frac{1}{j\lambda} \iint_A \frac{\exp[jk(r_{21}+r_{01})]}{r_{21} \cdot r_{01}} \cos(\hat{n}, \bar{r}_{01})\ ds \qquad (2\text{-}5)$$

which differs from Kirchhoff's result only in the so-called obliquity factor $\cos(\hat{n}, r_{01})$ compared to 1/2 $[\cos(\hat{n}, \bar{r}_{01}) - \cos(\hat{n}, \bar{r}_{21})]$.

The importance of Eq. (2-3) is that it allows the computation of the field at any observation point if the sound distribution in a plane across the direction of propagation is known.

B. SPATIAL FREQUENCY APPROACH TO DIFFRACTION BY PLANE SCREEN

The following discussion follows closely [Ref. 7 and Ref. 8]. Propagation and diffraction being linear, space invariant phenomena it is possible to define impulse responses and transfer functions for different systems. Every wave distribution can be expanded into a finite or infinite sum of weighted plane waves. This is accomplished by Fourier transforming the complex disturbance by use of the formula

$$\underline{A}\,(f_x,f_y) = \int_{-\infty}^{\infty}\int \underline{U}\,(x,y)\,\exp[-j2\pi(f_x\cdot x + f_y\cdot y)]\,dx\,dy \qquad (2\text{-}6)$$

where $\underline{A}\,(f_x,f_y)$ is then the complex spatial frequency spectrum of $\underline{U}$ and the spatial frequencies f_x and f_y are related to the direction cosines by

$$f_x = \alpha/\lambda \;;\; f_y = \beta/\lambda;\; f_z = \gamma/\lambda$$

The direction cosines α,β and γ are defined to be the cosines of the angles between the direction of propagation of the plane wave and the positive coordinate axes. The equation for a unit-amplitude plane wave propagating with direction cosines (α,β,γ) is simply

$$\underline{B}\,(x,y,z) = \exp\,[jk(\alpha x + \beta y + \gamma z)] \qquad (2\text{-}7)$$

where the direction cosines are geometrically constrained to obey the equation $\alpha^2 + \beta^2 + \gamma^2 = 1$

Writing $\underline{U}\,(x,y,z)$ as an inverse transform of its spectrum

$$\underline{U}\,(x,y,z) = \int_{\infty}\int \underline{A}\,(f_x,f_y)\,\exp[j2\pi(f_x\cdot x + f_y\cdot y)]\,df_x\,df_y \qquad (2\text{-}8)$$

it is obvious upon comparing Eq. (2-8) and Eq. (2-7) that $\underline{U}$ can be visualized as a sum of plane waves propagating with direction cosines

$$\alpha = \lambda \cdot f_x, \quad \beta = \lambda \cdot f_y \quad \gamma = (1-(\lambda^2 f_x^2)-(\lambda^2 f_y^2))^{1/2}$$

The complex amplitude of a plane wave component is simply $\underline{A}$ (fx,fy) $df_x\, df_y$, evaluated at $(f_x = \alpha/\lambda, f_y = \beta/\lambda)$.

In the following treatment the impulse response and transfer function of a diffracting aperture will be found. An equivalent form [Ref. 8] of the Rayleigh - Sommerfeld diffraction formula Eq. (2-3) is

$$\underline{U}(P_o) = \frac{j}{2\pi} \iint_A \underline{U}(P_1) \frac{\partial}{\partial z_1} \left[\frac{\exp(jkr_{01})}{r_{01}}\right] ds \qquad (2\text{-}9)$$

$$= \frac{1}{2\pi} \iint_A \underline{U}(P_1) \cdot K_{01} \, ds$$

where $K_{01} = \frac{\partial}{\partial z_1} \left[\frac{\exp(jkr_{01})}{r_{01}}\right]$

The plane wave expansion for a diverging spherical wave of the form $[\exp(jkr_{01})]/r_{01}$ is known to be (see [Ref. 9])

$$\frac{\exp(jkr_{01})}{r_{01}} = \frac{j}{\lambda} \int_{-\infty}^{\infty}\int \frac{1}{\gamma} \exp\{jk[(x_0-x_1)\alpha+(y_0-y_1)\beta + (z_0-z_1)\gamma]\} \, d\alpha d\beta \qquad (2\text{-}10)$$

To find K_{01} the differentiation according to z_1 is performed and yields:

$$K_{01} = \frac{2\pi}{\lambda^2} \int_{-\infty}^{\infty}\int \exp\{jk[(x_0-x_1)\alpha+(y_0-y_1)\beta+(z_0-z_1)\gamma]\} \, d\alpha d\beta \qquad (2\text{-}11)$$

Then

$$\underline{U}(P_o) = \frac{1}{\lambda^2} \iint_A \underline{U}(P_1) \cdot \int_{-\infty}^{\infty}\int \exp\{jk[(x_0-x_1)\alpha+(y_0-y_1)\beta + [(z_0-z_1)\gamma]\} d\alpha d\beta \, dx_1 dy_1 \qquad (2\text{-}12)$$

This can be brought into the form

$$\underline{U}(x_0,y_0,z_0) = \int\int_{-\infty}^{\infty} \underline{U}_1(x_1,y_1,z_1)\cdot t(x_1,y_1,z_1) \cdot \int\int_{\infty} \exp\{j\tfrac{2\pi}{\lambda}[(x_0-x_1)\alpha+(y_0-y_1)\beta+(z_0-z_1)\gamma]\} d\tfrac{\alpha}{\lambda} d\tfrac{\beta}{\lambda} \tag{2-13}$$

by introducing the aperture function $t(x_1,y_1,z_1)$ which is unity inside the aperture located in the opaque screen at z_1 and zero outside A. Kirchhoff boundary conditions are applied here to $\underline{U}$ but not to $\partial\underline{U}/\partial\hat{n}$. Separating the exponential and interchanging the order of integration leads to

$$\underline{U}_0(x_0,y_0,z_0) = \int\int_{-\infty}^{\infty} \underline{U}_t(x_1,y_1,z_1) \exp[-j2\pi(\tfrac{\alpha}{\lambda}x_1+\tfrac{\beta}{\lambda}y_1)]\, dx_1dy_1 \cdot \exp\{-j\tfrac{2\pi}{\lambda}\gamma z_1\} \int\int_{-\infty}^{\infty} \exp\{j\tfrac{2\pi}{\lambda}(\alpha x_0+\beta y_0+\gamma z_0)\}\, d\tfrac{\alpha}{\lambda}\, d\tfrac{\beta}{\lambda} \tag{2-14}$$

where $\underline{U}_t$ in Eq. (2-14) replaces $\underline{U}_1\cdot t$. But now the inner integral is just the Fourier transform or plane wave decomposition $\underline{A}_t$ of $\underline{U}_t$

$$\underline{A}_t = \int\int_{-\infty}^{\infty} \underline{U}_t(x_1,y_1,z_1) \exp[-j2\pi(\tfrac{\alpha}{\lambda}x_1+\tfrac{\beta}{\lambda}y_1)]\, dx_1dy_1 \tag{2-15}$$

and the outer integral can be written as

$$\underline{U}_0(x_0,y_0,z_0) = \int\int_{-\infty}^{\infty} \underline{A}_t(\tfrac{\alpha}{\lambda},\tfrac{\beta}{\lambda}) \exp\{j\tfrac{2\pi}{\lambda}\sqrt{1-\alpha^2-\beta^2}\,(Z_0-Z_1)\} \cdot \exp\{j\tfrac{2\pi}{\lambda}(\alpha x_0+\beta y_0)\} d\tfrac{\alpha}{\lambda}\, d\tfrac{\beta}{\lambda} \tag{2-16}$$

Eq. (2-16) can be seen as an inverse transform which relates $\underline{A}_0(\frac{\alpha}{\lambda},\frac{\beta}{\lambda})$ to $\underline{U}_0(x_0,y_0,z_0)$, so that

$\underline{A}_t(\frac{\alpha}{\lambda},\frac{\beta}{\lambda}) \exp[j\frac{2\pi}{\lambda}(z_0-z_1)\sqrt{1-\alpha^2-\beta^2}]$ must be the spectrum $\underline{A}_0(\frac{\alpha}{\lambda},\frac{\beta}{\lambda})$ of the diffracted and propagated disturbance. Therefore a transfer function $\underline{H}(f_x,f_y)$ can be defined

$$\underline{H}(f_x,f_y) = \frac{\underline{A}_0(f_x,f_y)}{\underline{A}_t(f_x,f_y)} = \exp[j\tfrac{2\pi}{\lambda}(z_0-z_1)\sqrt{1-(\lambda f_x)^2-(\lambda f_y)^2}] \tag{2-17}$$

The corresponding impulse response is found directly from the Rayleigh-Sommerfeld integral Eq. (2-3).

$$\underline{U}(P_o) = \frac{1}{j\lambda} \iint_A \underline{U}(P_1) \frac{\exp(jkr_{01})}{r_{01}} \cos(\hat{n},\bar{r}_{01})\, ds = \tag{2-18}$$

$$= \frac{1}{j\lambda} \iint_{\infty}^{\infty} \underline{U}(P_1) \cdot t(P_1) \frac{\exp(jkr_{01})}{r_{01}} \cos(\hat{n},\bar{r}_{01})\, ds$$

$$= \iint_{\infty}^{\infty} \underline{U}_t(P_1) \cdot \underline{h}(P_0,P_1)\, ds$$

The impulse response $\underline{h}(P_0,P_1)$ is given thus by

$$\underline{h}(P_0,P_1) = \frac{1}{j\lambda} \frac{\exp[jkr_{01}]}{r_{01}} \cos(\hat{n},\bar{r}_{01}) \tag{2-19}$$

By definition then Eq. (2-17) must be the Fourier transform of Eq. (2-18).

III. FRESNEL DIFFRACTION AND DIFFRACTION TREATMENT IN THE SPATIAL FREQUENCY DOMAIN

Having found the general form of the solution to diffraction by a plane screen it is soon evident that the Rayleigh-Sommerfeld diffraction formula is not very amenable to closed form solution. Even on the digital computer one would expect quite long computation times. Therefore approximations are introduced, which will allow reducing diffraction-pattern calculations to simpler mathematical operations.

A. FRESNEL DIFFRACTION

Consider the situation of Fig. 3-1. Single-frequency sound is diffracted by a finite aperture A in an infinite plane opaque screen which is located in the x_1-y_1- plane at the origin of the z- axis. The observation region is assumed to be parallel to the x_1-y_1- plane and has a coordinate system (x_0,y_0) attached to it. This observation plane is at a distance Z from the diffracting screen.

Using the Rayleigh-Sommerfeld formula the field at a point (x_o,y_o) can immediately be written as

$$\underline{U}_0(x_0,y_0,z_0) = \int\int_{-\infty}^{\infty} \underline{h}\,(x_0,y_0,x_1,y_1)\; \underline{U}_t(x_1,y_1,0)\; dx_1 dy_1 \qquad (3\text{-}1)$$

where

$$\underline{h}\,(x_0,y_0,x_1,y_1) = \frac{1}{j\lambda}\;\frac{\exp(jkr_{01})}{r_{01}}\;\cos(\hat{n},\bar{r}_{01}) \qquad (3\text{-}1a)$$

and the effect of the finite aperture A is incorporated into $\underline{U}_t$, it being assumed the $\underline{U}_t$ is zero outside the aperture region. This allows extending the integration limits to infinity.

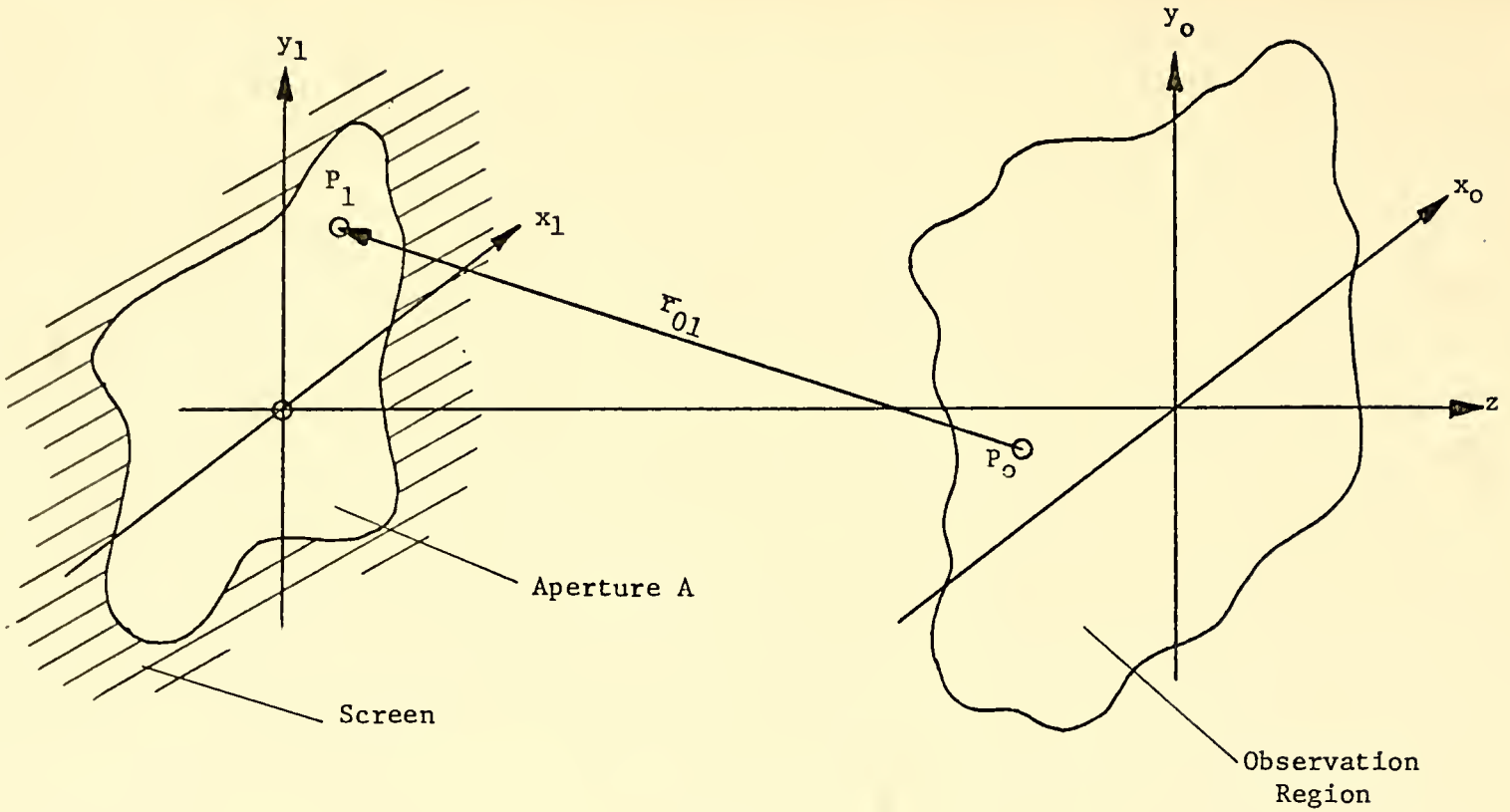


Figure 3-1

Geometry for Fresnel and Fraunhofer Diffraction

The first assumption to be made is that the distance z is very much greater than the maximum linear dimension of the aperture A and that in the observation plane only a small region around the z-axis is of interest. Again z is assumed to be very much greater than the extension of this region of interest. The initial approximations introduced are called the paraxial approximations because they hold only close to the axis of propagation. Using this assumption the obliquity factor is approximated by

$$\cos(\hat{n}, r_{01}) \cong 1$$

which is good to within 5% if the argument of the cosine is not greater than 18°.

The same assumption can also be used to replace the radius r_{01} by the distance z in the denominator of Eq. (3-1a) but not in the exponent, since the resulting error would be multiplied by the large number k, generating intolerable phase errors. The impulse response has now been approximated by:

$$\underline{h}(x_0, y_0, x_1, y_1) \cong \frac{1}{j\lambda z} \exp(jkr_{01}) \tag{3-2}$$

Further simplification is possible by using the Fresnel approximation. This consists of expanding the radius r_{01} in the exponent, which is given exactly by

$$r_{01} = \sqrt{(x_0-x_1)^2+(y_0-y_1)^2+z^2}$$

$$= z\sqrt{1+\left(\frac{x_0-x_1}{z}\right)^2+\left(\frac{y_0-y_1}{z}\right)^2}$$

in a binomial expansion and assuming that r_{01} is given adequately by the first two terms in the series. Then

$$r_{01} \cong z\left[1+\frac{1}{2}\left(\frac{x_0-x_1}{z}\right)^2+\frac{1}{2}\left(\frac{y_0-y_1}{z}\right)^2\right]$$

The impulse response can then be written as

$$\underline{h}\,(x_0,y_0,x_1,y_1) = \frac{\exp(jkz)}{j\lambda z}\exp\left\{j\frac{k}{2z}\left[(x_0-x_1)^2+(y_0-y_1)^2\right]\right\} \qquad (3\text{-}3)$$

The region for which this approximation is exact enough is called the region of Fresnel diffraction. The accuracy of this approximation hinges on the relative sizes of the aperture and of the observation region, the propagation distance z and the wavelength λ. One might require that the maximum phase change contributed by the third order term in the series expansion should be rather small, say much less than one radian. This is the case if

$$z^3 >> \frac{\pi}{4\lambda}\left[(x_0-x_1)^2+(y_0-y_1)^2\right]^2_{MAX} \qquad (3\text{-}4)$$

For an observation region and diffraction region of about 50λ by 50λ this would require the distance z to be much greater than about $250\text{-}300\lambda$. In fact the requirement of Eq. (3-4) is too restrictive because all that is required is that the higher order terms do not change the value of the superposition integral of Eq. (3-1) by an appreciable amount. For distances z smaller than those given by Eq. (3-4) the oscillations of the quadratic phase factor given by Eq. (3-3) will generally be so very rapid as the integration is performed that they virtually cancel out and only those points of the aperture where $x_0=x_1$ and $y_0=y_1$ will give a noticeable contribution to the integral. Near these points of so-called stationary phase the magnitude of higher order terms will normally be negligible. The principle of stationary phase is discussed in [Ref. 6]. None the less, Fresnel diffraction cannot be assumed to hold for distances which for the given example might be less than about 100λ (the paraxial approximation is implied also!).

If one is satisfied as to the validity of the Fresnel approximation for a given case, the superposition integral can be expressed in the form of the following equation

$$\underline{U}(x_0,y_0) = \frac{\exp(jkz)}{j\lambda z} \exp\left[\frac{jk}{2z}(x_0^2+y_0^2)\right] \cdot \int_{-\infty}^{\infty}\int \underline{U}_t(x_1,y_1) \exp\left[\frac{jk}{2z}(x_1^2+y_1^2)\right] \exp\left[-j\frac{2\pi}{\lambda z}(x_0x_1+y_0y_1)\right] dx_1 dy_1 \tag{3-5}$$

by expanding the quadratic terms in the exponent. This can be seen to be (aside from multiplicative phase factors and attenuation independent of (x_1,y_1)) the Fourier transform of $\underline{U}_t(x_1,y_1) \cdot \exp[\frac{jk}{2z}(x_1^2+y_1^2)]$, evaluated at the spatial frequencies $f_x = x_0/\lambda z$ and $f_y = y_0/\lambda z$, which assures correct scaling in the observation plane. As such the integral Eq. (3-5) is amenable to digital computer solution using a Fast Fourier Transform algorithm.

It is also possible to transform the impulse response of Eq. (3-3) because it is space-invariant (it depends only on the differences (x_0-x_1) and (y_0-y_1)) to get a transfer function:

$$\underline{H}(f_x,f_y) = \exp(jkz) \exp[-j\pi\lambda z(f_x^2+f_y^2)], \tag{3-6}$$

which relates the spatial frequencies in the diffracting and in the observation plane. This is immediately observed to be an approximation to the exact transfer function of Eq. (2-17).

B. FRAUNHOFER DIFFRACTION

For more severe restrictions on the propagation distance, it is possible to simplify the diffraction equation even more. If z is large enough to assume that the quadratic phase factor $\exp[jk(x_1^2+y_1^2)/2z]$ is approximately unity, then the diffraction equation reduces to a straightforward Fourier transform (again evaluated at the scaled values of spatial frequency) with an additional multiplication

$$\underline{U}(x_0,y_0) = \frac{\exp(jkz)}{j\lambda z} \exp[jk(x_0^2+y_0^2)/2z] \times \int_{-\infty}^{\infty}\int \underline{U}_t(x_1,y_1) \exp[-j2\pi(x_0x_1+y_0y_1)/\lambda z]\, dx_1 dy_1 \tag{3-7}$$

This equation is called the Fraunhofer diffraction formula. Unfortunately, the distance z would have to be very large indeed for Eq. (3-7) to be a good approximation to the actual physical situation. For the dimensions used previously to find the minimum distance for Fresnel diffraction z would have to be very much larger than 15000λ. Nevertheless, the Fraunhofer diffraction equation can be very useful in optics, because it fits the situation at the focal plane of a well-corrected lens; it is of less use in acoustical diffraction problems because of the difficulty of building good acoustical lenses. In the context of this investigation it can be used to check the results obtained by use of the Fresnel equation because Fraunhofer diffraction is the limit of very large propagation distance z in the Fresnel diffraction equation and therefore a Fourier transform of the diffracting aperture would be obtained.

The transfer function (Eq. (3-6)) approach to Fresnel diffraction can also be used in the Fraunhofer diffraction region, but one would hardly under normal computational circumstances try to implement the Fresnel (or Fraunhofer) diffraction equation by using the transfer function in the spatial frequency domain. This will take longer computation time by requiring a forward and an inverse Fourier transform, plus a matrix multiplication whereas in programming the Fresnel equation in the space domain one needs only a forward Fourier transform with additional matrix multiplications.

C. SPATIAL FREQUENCY APPROACH

If it has been decided that the use of longer computation time is feasible, then the implementation of the spatial frequency diffraction equation (Eq. (2-26)) is a much more powerful tool. (Indeed, for large array sizes the use of the spatial frequency approach may be more economical, as the number of computations for a Fast Fourier Transform increases like $N \cdot \log_2 N$, whereas N^2 multiplications are required for each of the quadratic phase factors). By using it not only the Fresnel requirement on propagation distance but also the paraxial approximations are circumvented. Any inaccuracies in the results obtained by the spatial frequency approach are thus due only to the use of Kirchhoff boundary conditions on $\underline{U}$ or on $\partial\underline{U}/\partial\hat{n}$. But as Kirchhoff's diffraction theory as used in the Fresnel-Kirchhoff diffraction formula or in the Rayleigh-Sommerfeld diffraction formulation (the relative accuracy of both equations being still a subject of active research) gives results which are in excellent agreement with practice, any reservation against the use of Kirchhoff's boundary conditions can only be very slight. Moreover they are applied in the Fresnel equation also.

To program diffraction by a plane screen using the spatial frequency approach one first multiplies the insonifying disturbance by the aperture function, performs a forward Fourier transform and multiplies each spatial frequency component by the appropriate phase delay as given by Eq. (2-26). Inverse transforming then results in the diffraction pattern in the observation plane.

In shorthand notation:

$$\underline{U}\ (x_0,y_0) = F^{-1}\ \{\underline{H}\ (f_x,f_y) \cdot F\ \{\underline{U_1}(x_1,\ y_1)\cdot\underline{t}\ (x_1,y_1)\}\} \qquad (3\text{-}8)$$

where F and F^{-1} symbolize forward and inverse Fourier transform and $\underline{H}\ (f_x,f_y)$ is given by Eq. (2-26)

$$\underline{H}\ (f_x,\ f_y) = \exp\ [j\frac{2\pi z}{\lambda}\ \sqrt{1-(\lambda f_x)^2-(\lambda f_y)^2}] \qquad (2\text{-}26)$$

$\underline{U}_1$ is the insonifying wave complex amplitude and $\underline{t}$ is the aperture function.

The aperture function is usually taken to be unity inside the aperture and zero outside but there is really no necessity for doing that. The aperture might also be characterized by the introduction e.g., of a constant phase shift or a quadratic phase shift (like that due to a lens) or a combination of amplitude attenuation and phase shifts. Of course the effect of the aperture could also be handled in the spatial frequency domain by using the convolution theorem of linear systems analysis and writing

$$F\ \{\underline{U_1}(x_1,y_1)\cdot\underline{t}(x_1,y_1)\} = \underline{A_1}(f_x,f_y)\ *\ \underline{T}\ (f_x,f_y)$$

where $\underline{A}\ (f_x,f_y)$ is the spatial frequency spectrum of $\underline{U_1}$, $\underline{T}\ (f_x,f_y)$ is the spatial frequency spectrum of $\underline{t}$ and * represents the convolution operation.

This explanation may be helpful in visualizing the effect of the aperture in broadening the angular spectrum of the disturbance but would not be used on the computer if it can be avoided.

The transfer function $\underline{H}\ (f_x,f_y)$ has to be investigated more closely. To recapitulate

$$\underline{H}\ (f_x,f_y) = \exp\ [j\frac{2\pi z}{\lambda}\ \sqrt{1-(\lambda f_x)^2-(\lambda f_y)^2}] \qquad (2\text{-}26)$$

When the direction cosines $\alpha=\lambda\cdot f_x$ and $\beta=\lambda\cdot f_y$ satisfy $\alpha^2+\beta^2<1$ (case of homogenous waves) the effect of propagation over a distance z is just a change in the relative phases of the spectral components: Since each component travels at a different direction it has to travel a different distance to reach the observation plane and thereby relative phase shifts are introduced. But when $\alpha^2+\beta^2>1$ (case of inhomogenous or evanescent waves) the square root in Eq. (2-26) is imaginary and

$$\underline{H}\ (f_x,f_y) = \exp\ [-\frac{2\pi z}{\lambda}\ \sqrt{\lambda f_x)^2+(\lambda f_y)^2-1}] = \exp\ [-pz]$$

where p is a positive real number. These wave components which travel perpendicular to the direction of propagation are severely attenuated by even small propagation distances. For propagation distances greater than a few wavelengths the influence of these evanescent waves is entirely negligible on the diffraction pattern and it is possible to band-limit the transfer function to

$$\underline{H}\ (f_x,f_y) = \begin{cases} \exp[j\frac{2\pi z}{\lambda}\ \sqrt{1-(\lambda f_x)^2-(\lambda f_y)^2}] & \text{for } f_x^2+f_y^2<\frac{1}{\lambda^2} \\ 0 & \text{elsewhere} \end{cases} \qquad (3\text{-}9)$$

D. COMPARISON OF FRESNEL EQUATION AND SPATIAL FREQUENCY APPROACH

A comparison between the two possible ways of implementing the diffraction problem is now possible.

The time needed for one propagation step using the Fresnel equation (one Fourier transform, two multiplication operations on a two-dimensional matrix - one before, one after the Fourier transform) will be less (for moderate array size) than that consumed by one propagation step using the spatial frequency approach (forward Fourier transform, matrix multiplication, inverse transform). The Fresnel equation cannot be used for small propagation distances because of its inherent approximations, which do not occur in the spatial frequency method. The spatial increment between array points increases with propagation distance in the use of the Fresnel equation due to the necessary evaluation of the Fourier transform at $f_x=x_o/\lambda z$, $f_y=y_o/\lambda z$, whereas it is constant when the spatial frequency approach is used. This leads to the fact that it is possible to use the Fresnel equation program for very large propagation distances whereas the spatial frequency method is limited in the propagation distances to be used because the diffraction patterns from "ghost" objects (which are postulated by the Discrete Fourier Transform) start overlapping with the useful diffraction pattern. These effects are described in more detail later in the section on the Discrete Fourier Transform. Reduction of this unwanted and damaging effect is possible but only by using a larger array size which can increase the computation time considerably. The last aspect under which the two methods will be compared is that of the validity of the results after backward propagation of the observed wavefront when the propagation distance differs by a small amount from the actual distance.

Suppose a diffraction pattern is observed at a distance z from a plane object. Then, on propagating the wave back a distance z-dz (an error which might occur in an automatic reconstruction scheme) using the transfer function approach one would expect to obtain the same diffraction pattern as would appear on forward propagation of the original diffracted wavefront at the object by a distance dz. It is then possible (by propagation backward using a propagation distance dz) to obtain the actual diffracting object. The case is quite different when the Fresnel equation is used. Here when the propagation distance in reconstructing differs by a small amount from the true one, the obtained "image" is only a very coarse approximation to the diffraction pattern at a distance dz in front of the diffracting aperture. The actual diffracting plane and that just reconstructed are related only by the fact that they give rise to the same diffraction pattern at a distance z and z-dz respectively when the Fresnel equation is used. In this second case it is not possible by another small propagation step of the correct difference dz to reach the correct result: The Fresnel equation does not hold true for small propagation distances. One would have to go back to the original diffraction pattern and to try again from there. This either requires storing the original diffraction pattern (because in the use of the Fourier transform the original data are replaced by the transformed components), which being a two-dimensional matrix of complex quantities with a rather large size requires a considerable amount of additional core memory or demands another propagation step for the restoration of the original pattern. Thus the use of the Fresnel equation in an automatic reconstruction scheme would take more computation time or core memory than the spatial frequency method and become uneconomical. (If the

propagation distance is known precisely beforehand this reversal of relative time consumption between the two methods would not occur). This last aspect settled the question which algorithm should be used in favor of the transfer function approach in that it is desirable in a restoration problem to reach a result which is as close to the actual disturbance at a particular distance as possible even though the distance might be slightly wrong. Moreover it is much more difficult to handle a diffraction problem by the Fresnel equation if two diffracting objects are located in planes which are displaced in depth, due to the self-scaling of the Fresnel equation; it is impossible to do if the distance between these objects is less than that required for the Fresnel equation to hold.

E. BACKWARD PROPAGATION

How 'backward propagation' is accomplished can now be described. The conceptually more satisfying spatial frequency approach is treated first because it is used in the actual program. Consider a given diffraction pattern $\underline{U}_0\ (x_0,y_0)$ at a distance z from an object $\underline{U}_1(x_1,y_1)$ as in Fig. 3-1. Then from the diffraction equations,

$$\underline{U}_0\ (x_0,y_0) = F^{-1}\ \{\underline{H}\ (f_x,f_y)\cdot F\ \{\underline{U}_1(x_1,y_1)\}\} \qquad (3\text{-}8)$$

To find $\underline{U}_1\ (x_1,y_1)$ one would first perform a Fourier transform on $\underline{U}_0$, multiply the result by $\underline{H}^{-1}$ and then inverse transform the result

$$\underline{U}_1\ (x_1,y_1) = F^{-1}\ \{\underline{H}^{-1}(f_x,f_y)\cdot F\ \{\underline{U}_0(x_0,y_0)\}\} \qquad (3\text{-}10)$$

where $\underline{H}^{-1}$ is defined by

$$\underline{H}\cdot\underline{H}^{-1} = 1 \qquad (3\text{-}11)$$

The symmetry between Eq. (3-8) and (3-10) is obvious. The reverse transfer function $\underline{H}^{-1}$ can easily be found from the definition Eq. (3-11) and Eq. (3-9) which defines the transfer function $\underline{H}$.

$$\underline{H}^{-1} = \exp\left[-j\frac{2\pi z}{\lambda}\sqrt{1-(\lambda f_x)^2-(\lambda f_y)^2}\right]. \qquad (3\text{-}12)$$

Observe that the same result would have been obtained by mechanically inserting (-z) as the propagation distance in the equation for the forward transfer function, where the minus-sign might be taken to signify 'backward propagation'. Another observation is in order here: Suppose $\underline{H}$ were not taken to be band-limited to those spatial frequencies which give a real square root. Then at least $\underline{H}^{-1}$ must be band-limited; else for those frequencies which give an imaginary square root

$$\underline{H}^{-1}(f_x,f_y) = \exp\left[\frac{2\pi z}{\lambda}\sqrt{(\lambda f_x)^2 + (\lambda f_y)^2-1}\right] = e^{+pz} \text{ for } f_x^2+f_y^2>\frac{1}{\lambda^2}$$

and an exponential growth would be postulated, which leads to conceptual problems (for complete treatment of this subject see [Ref. 8]). Apart from these conceptual problems: the computer will not handle exponentials with real positive arguments which are too large (as would be in the case for non-bandlimited $\underline{H}^{-1}$ used in a propagation over more than a minuscule distance). The frequency components of the angular spectrum corresponding to evanescent waves in a diffraction field that is several wavelengths away from the object would be very small in amplitude. There is a good chance that they would be prone to round-off errors on the computer and in an actual data acquisition system would be lost in the unavoidable noise anyway. Trying to work with a non-bandlimited reverse transfer function then is bound to fail. But it must always be kept in mind that the evanescent waves are physically existent and carry information about those object features which are smaller than

a wavelength. Setting them to zero destroys information irretrievably and therefore degrades system performance. Figures 3-2 through 3-5 give an example of this image degradation. Figure 3-2 shows the amplitude distribution of the object, which is 32 sampling points wide and 16 sampling points deep, centered in a 64 sampling points square array. If the emission at all sampling points is assumed to be in phase (as in a specular object) and the sampling points are one half wavelength apart the result after transforming and reconstruction neglecting evanescent waves is given in Fig. 3-3. The degradation is not very severe. For in-phase emission and sampling every 0.3 wavelengths the result of Fig. 3-4 is obtained which shows much lower slope and higher overshoots. It still gives a good indication of the actual shape of the original object. Figure 3-5 is the result of sampling 0.5 wavelengths apart and assuming a random phase distribution for the object to be reconstructed (as in a diffuse object). As a diffusely emitting object will have more energy in high frequency components one would expect the loss of information to be worse than in the specular case; still the degree of degradation as shown in Fig. 3-5 is rather unexpected. For experimental results of backward propagation using the spatial frequency approach see [Ref. 10].

From the Fresnel equation $\underline{U}_1\ (x_1,y_1)$ can be found by forming the complex conjugate of $\underline{U}_0\ (x_0,y_0)$ and propagating $\underline{U}_0^*$ a distance (+z)

$$\underline{U}_2(x_2,y_2) = \frac{\exp(jkz)}{j\lambda z} \exp\left[j\frac{k}{2z}(x_2^2+y_2^2)\right] \cdot \int_{-\infty}^{\infty}\int \underline{U}_0^*(x_0,y_0) \exp\left[j\frac{k}{2z}(x_0^2+y_0^2)\right] \exp\left[-j\frac{2\pi}{\lambda z}(x_0x_2+y_0y_2)\right] dx_o dy_o \qquad (3\text{-}13)$$

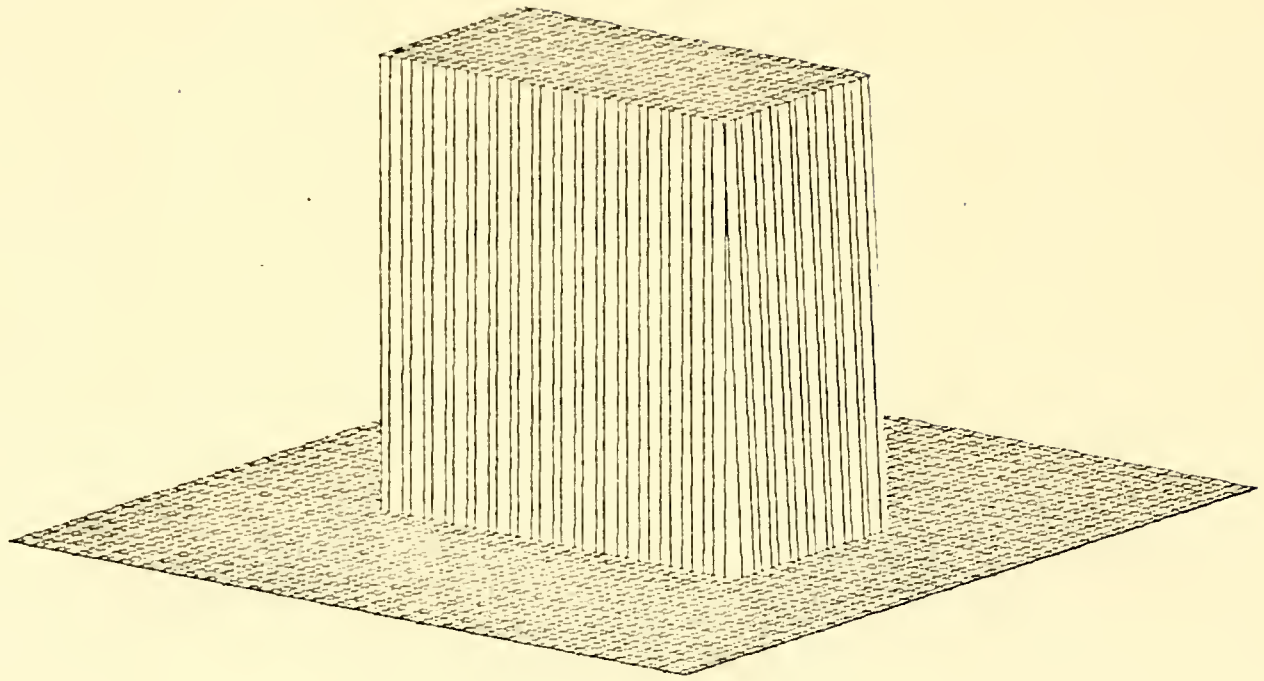


Figure 3-2
Amplitude Distribution of Input Object

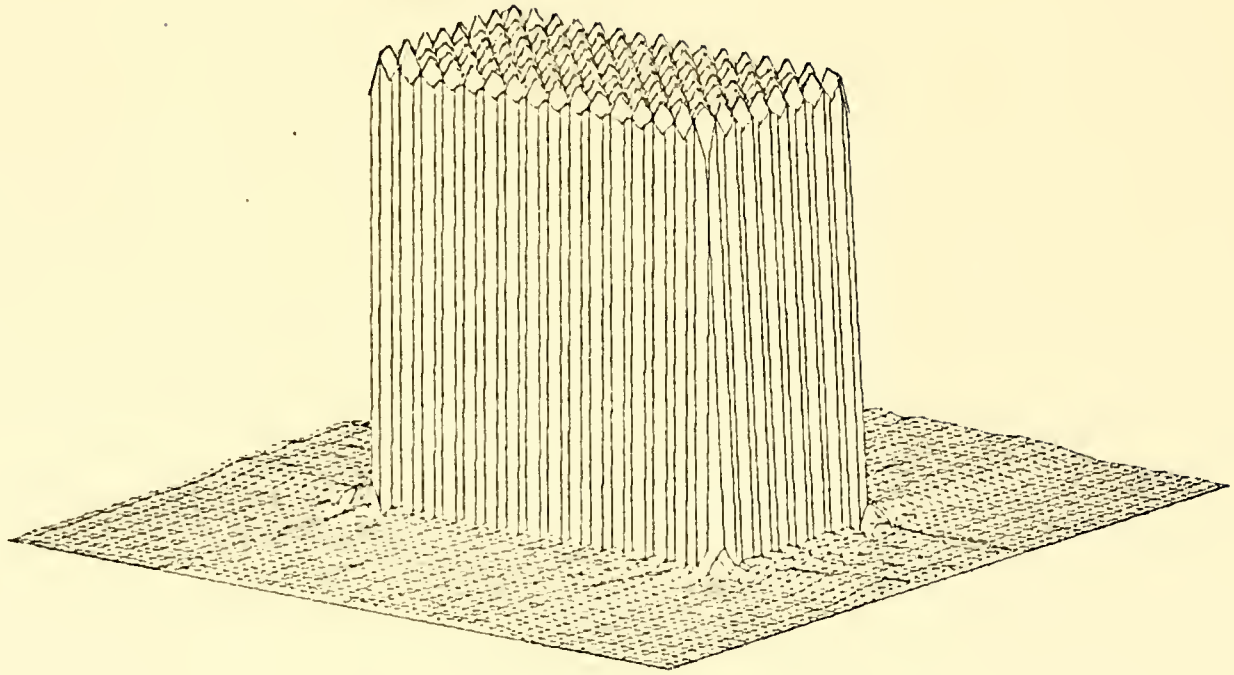


Figure 3-3
Reconstructed Specular Object with Nyquist Rate Sampling and Neglecting Evanescent Waves

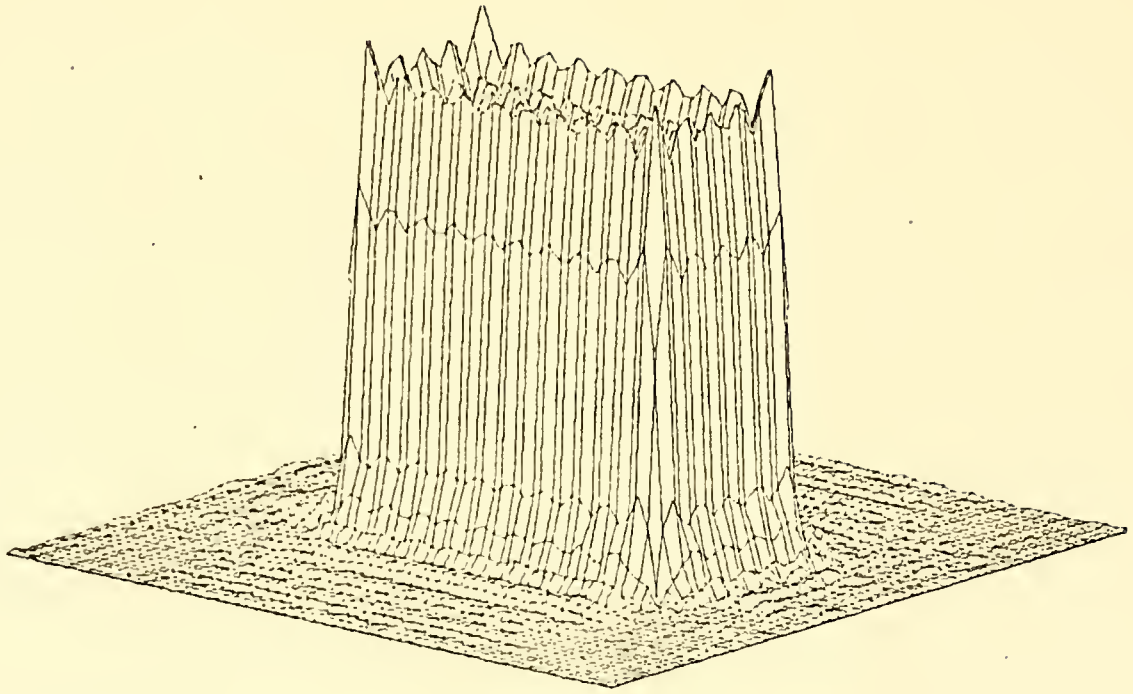


Figure 3-4
Reconstructed Specular Object with Sampling at Higher than Nyquist Rate and Neglecting Evanescent waves

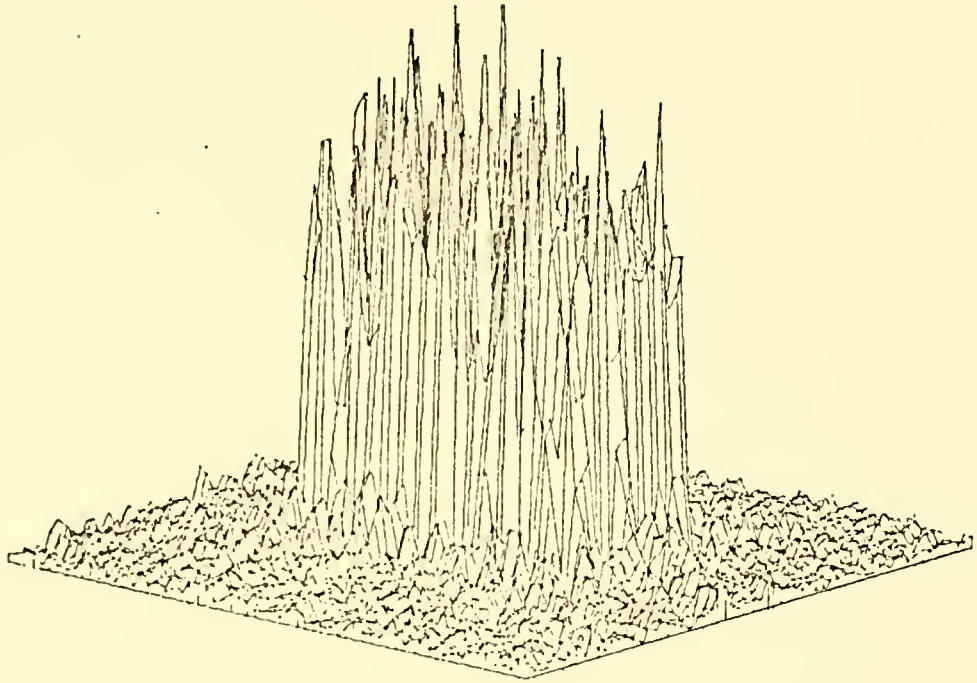


Figure 3-5
Reconstructed Diffuse Object with Nyquist Rate Sampling and Neglecting Evanescent Waves

Conceptually, the term 'backward propagation' is not applicable to this operation; it is rather a forward propagation of a different though closely related diffraction pattern. The result of this operation is the original object with the positive sense of the x_1 and y_1- axes reversed, i.e., $\underline{U}_2(x_2,y_2) = \underline{U}_1(-x_1,-y_1)$; this usually is no reason for not using this approach, since the inversion is easily corrected. The main reason for discarding the Fresnel equation approach to backward propagation is that it is not possible to use repeated small propagation steps in the Fresnel equation due to its inherent approximations.

IV. DISCRETE AND FAST FOURIER TRANSFORM

Diffraction pattern computations can be accomplished in various ways, but if one of the approaches as outlined in the last section is chosen (for the sake of simplicity and short computation time) it will be necessary to take the Fourier transform of some input array. Since it is possible to work only with discrete samples on the digital computer a discrete Fourier transform has to be used.

A. DISCRETE FOURIER TRANSFORM

Suppose a function $f(x)$ is represented on the interval $[0,X]$ by N samples $f(n\cdot a)$ where

$$a = \frac{X}{N} \qquad \text{and}$$

$$0 \leq n \leq N - 1$$

The Discrete Fourier Transform (DFT) of the sequence $f(o)$, $f(a)$,$\cdots$ $f((N-1)\cdot a)$ is defined to be the sequence $F(o)$, $F(\Omega)$,$\cdots$ $F((N-1)\cdot\Omega)$ where

$$F(k\Omega) = \sum_{n=o}^{N-1} f(n\cdot a)\, \exp(-jn\cdot a\cdot k\cdot\Omega) \qquad 0 \leq k \leq N-1 \tag{4-1}$$

and

$$\Omega = \frac{2\pi}{X} = \frac{2\pi}{N\cdot a} \tag{4-1a}$$

The inverse DFT is given by

$$f(n\cdot a) = \frac{1}{N} \sum_{k=o}^{N-1} F(k\cdot\Omega)\, \exp(jk\Omega na) \tag{4-2}$$

Remarks:

1. The similarity between the DFT as defined above and the exponential form of the Fourier series expansion is obvious. There

$$f(t) = \frac{1}{T} \sum_{n=-\infty}^{\infty} F(n\omega) \exp(jn\omega t) \qquad (\omega = \frac{2\pi}{T})$$

and

$$F(n\omega) = \int_{-T/2}^{T/2} f(t) \exp(-jn\omega t)\, dt$$

In the case of the DFT the integral is replaced by a sum as there are only discrete samples of f(x).

2. The DFT gives as many frequency components as there are samples of the function to be transformed.

3. In order to make the definition of the DFT in form of a Fourier series consistent it has to be assumed that the sampled function is periodic with period X. Otherwise it could not give rise to discrete frequency components. Thus each object must be taken to have 'ghost objects' of equal shape located next to it at a periodic distance X in both horizontal and vertical directions. Analogously, the sequence of spectral components must be assumed to be periodic outside the actually computed spectral period.

4. Even if the sampled function values are real, the sequence of frequency components is in general going to be complex.

5. The spacing between frequency components is the reciprocal of the period X, multiplied by 2π. As the spatial frequency approach to diffraction was shown to be bandlimiting in the last section, it is possible to compute the minimum sampling rate, which is necessary to avoid overlapping of spectral components. In the spatial frequency domain the spectrum of the transformed object is repeated with a frequency period of 1/a. If a is chosen to be large the individual spectra will begin overlapping; it is then not possible to extract the

original object from this distorted spectrum. This effect is called 'aliasing'. To recapitulate:

$$\underline{H}(f_x,f_y) = \begin{cases} \exp\left(j\frac{2\pi z}{\lambda}\sqrt{1-(\lambda f_x)^2-(\lambda f_y)^2}\right) & (f_x^2+f_y^2<\frac{1}{\lambda^2} \\ 0 & \text{elsewhere} \end{cases}$$

This means that in the spatial frequency plane $\underline{H}(f_x,f_y)$ is bandlimited to those frequencies which lie inside the circle of radius $1/\lambda$ centered at the origin. (See Fig. 4-1) If efficient sampling is desired (meaning sampling that is not higher than the Nyquist rate) the sampling scheme becomes very complicated even for such a simple band-limited frequency region as a circle. This can be avoided at the cost of slightly higher sampling rate (about 12%, see [Ref. 11]) by assuming the band-limited region to be the smallest square that circumscribes the actual circle. Using the Nyquist sampling criterion on this new band-limited region leads to the maximum allowable spacing between sampling points, namely $a = \lambda/2$. This is found the following way:

For the f_x -direction: $f_{xMAX} = \frac{1}{\lambda} = \Delta f_x \, N/2$

$$\Delta f_x = \frac{2}{N\cdot\lambda} = \frac{\Omega}{2\pi}$$

from Eq. (4-1a) $\Omega = \frac{2\pi}{X} = \frac{2\cdot 2\pi}{N\cdot\lambda}$

then $X = \frac{N\cdot\lambda}{2}$; $a = \frac{X}{N} = \frac{\lambda}{2}$

A similar result holds analogously for the f_y -direction.

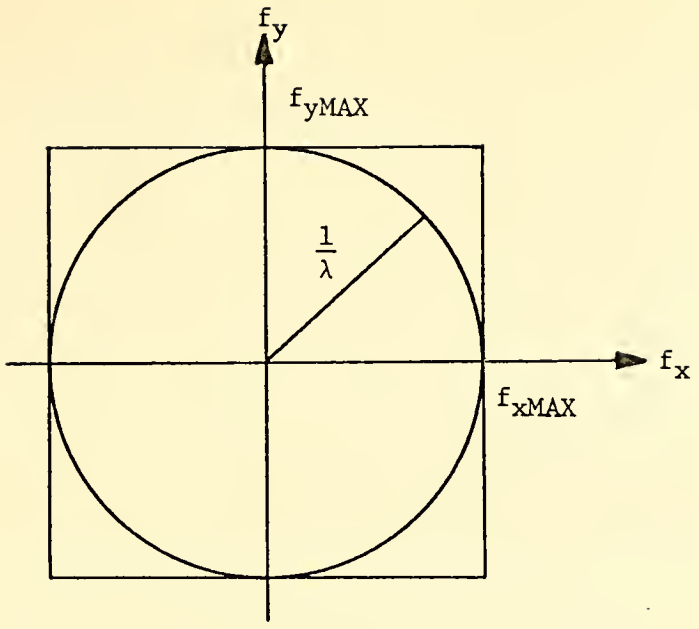


Figure 4-1
Propagation Transfer Function in the Spatial Frequency Domain, Showing Band-limitation at Nyquist Rate Sampling.

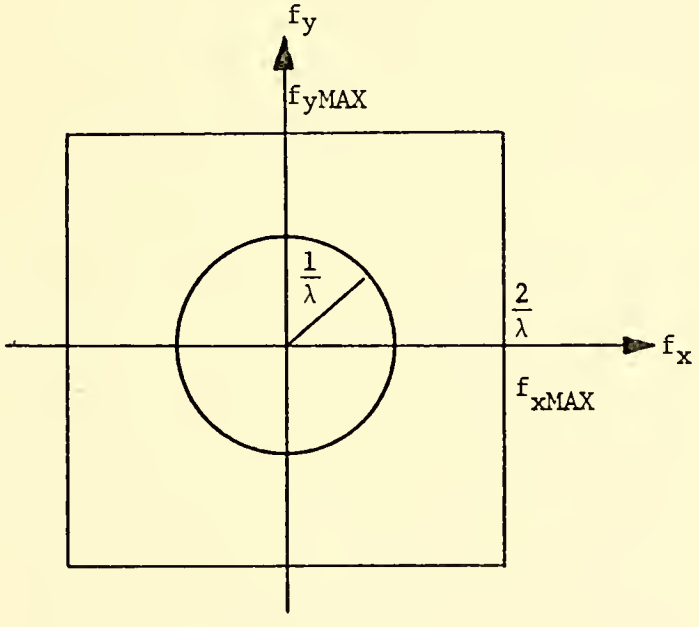


Figure 4-2
Propagation Transfer Function in the Spatial Frequency Domain, Showing Band-limitation at Twice Nyquist Rate Sampling.

Now, suppose for some reason it would be desirable to introduce a guard band in the spatial frequency domain or to sample at a higher than the Nyquist rate, e.g.,

$$a^1 = \frac{\lambda}{4} < a = \frac{\lambda}{2}$$

without increasing the total number of samples.

Then $f_{xMAX} = \frac{1}{2a^1} = \frac{2}{\lambda}$. (See Fig. 4-2)

But the Discrete Fourier Transform will space the calculated frequency components evenly from $-f_x max$ to $+ f_x max$, and from $-f_y max$ to $+f_y max$ and all the spectral components outside the bandlimiting circle will be set to zero on propagation. Therefore the (necessarily discrete) propagation algorithm will have to work on fewer frequency components than when the Nyquist rate is used. The results will be in error because on inverse transforming fewer spatial frequency components are available. There is then a trade-off between fine sampling in space and a coarsening of results of the propagation algorithm which has to be taken into account.

Of course if the object spectrum itself is band-limited to a band which is narrower than that introduced by the propagation transfer function, the sampling rate could be adjusted accordingly to a coarser sampling (see [Ref. 12]). Since the object is assumed to be unknown, this possibility cannot be used here.

The definition of the Discrete Fourier Transform is readily extended to two (or more) dimensions. In particular for two dimensions:

$$F(k\Omega, l\Delta) = \sum_{n=o}^{N-1} \sum_{m=o}^{M=1} f(na,mb) \exp[-j(nak\Omega + mbl\Delta)]$$

$$\text{for } 0 \leq k \leq N-1 \text{ and } 0 \leq l \leq M-1 \qquad (4-3)$$

where $\Omega = \frac{2\pi}{X}$ and $\Delta = \frac{2\pi}{Y}$; $a = \frac{X}{N}$; $b = \frac{Y}{M}$

Then

$$f(na,mb) = \frac{1}{N \cdot M} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} F(k\Omega,l\Delta) \exp[j(nak\Omega+mbl\Delta)] \qquad (4\text{-}4)$$

The postulated periodicity of the object to be transformed is very unfortunate for a system that is to simulate imaging and diffraction. Consider the one-dimensional case as shown in Fig. 4-3. An object and its two nearest 'ghost objects' are shown. It is well known that diffraction means a spreading out of the diffracted wavefront. The waveform of Fig. 4-3 diffracted and propagated out some distance might look like Fig. 4-4. Now at an even larger propagation distance the spreading out might be so severe that the wavefronts due to the object of interest and those due to the postulated ghost objects to either side of it will begin overlapping and interfering constructively or destructively as in Fig. 4-5. Naturally the waveform of Fig. 4-5 looks quite different towards the boundaries of the sampling period from the one that would be obtained without any ghost objects and in fact is obtained and observed in a physically existing system as in Fig. 4-6. The observed waveform of Fig. 4-6 sampled and fed to the computer to be propagated back the correct distance using a DFT routine cannot reproduce exactly the central of Fig. 4-3, because of the limitations of the DFT routine. It is this fact that gives rise to the limitation in propagation distance of the transfer function approach to diffraction that was mentioned in the last section. There is an easy wasy to overcome this limit by increasing the guard space between the desired object and its ghost images by increasing the array size, if the implied increase in core memory needed and associated computing time can be afforded. In this investigation it was decided that a 64 by 64 element matrix was as large as could be afforded without running into time and memory problems.

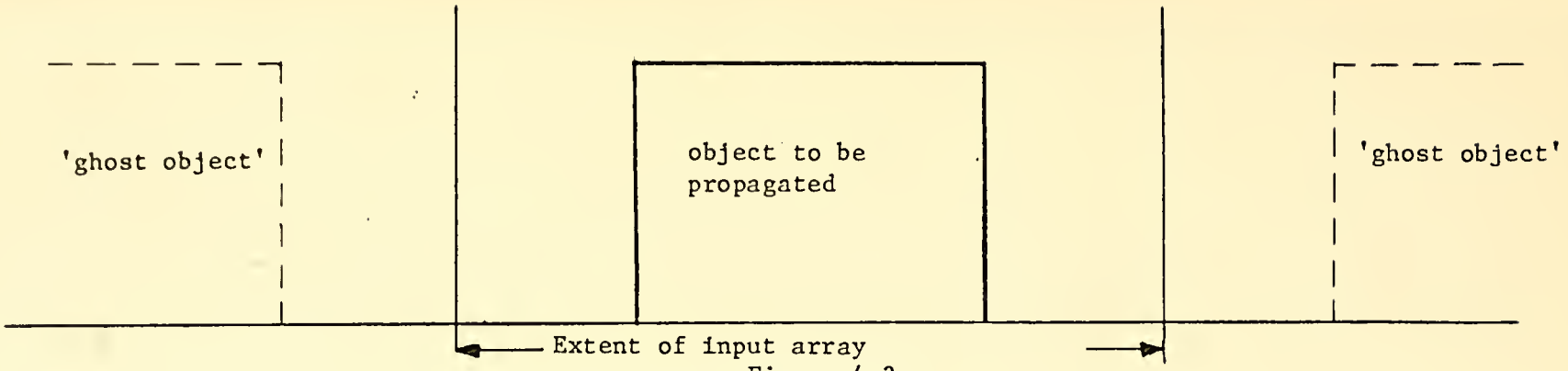


Figure 4-3
Amplitude of Object Waveform

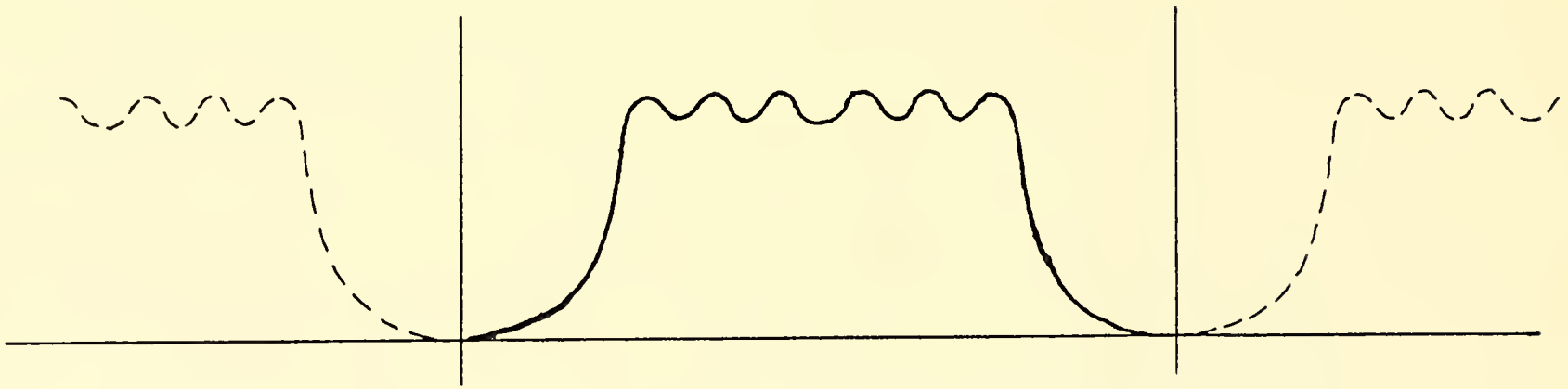


Figure 4-4
Waveform of Fig. 4-3 Propagated a Small Distance

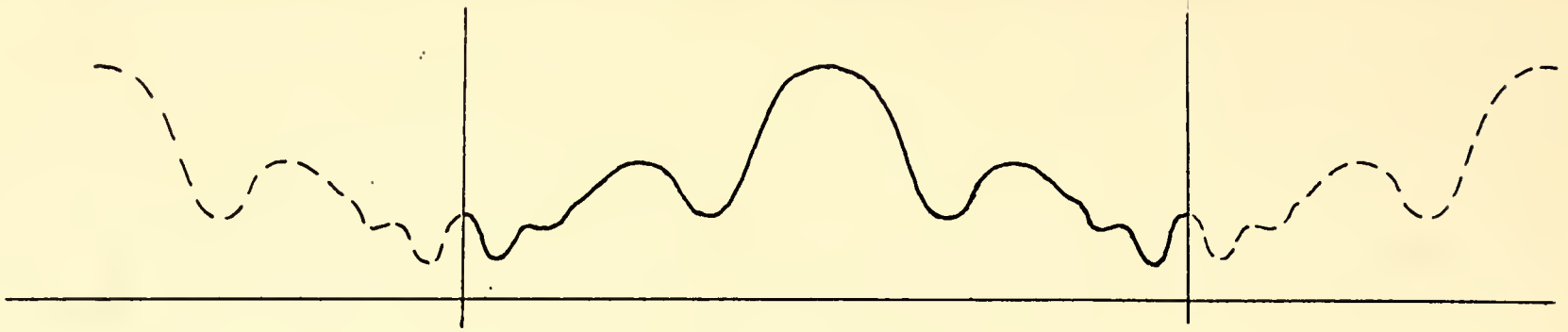


Figure 4-5
Waveform of Fig. 4-3 propagated a large distance, showing distortion due to overlap of central object and 'ghost objects'.

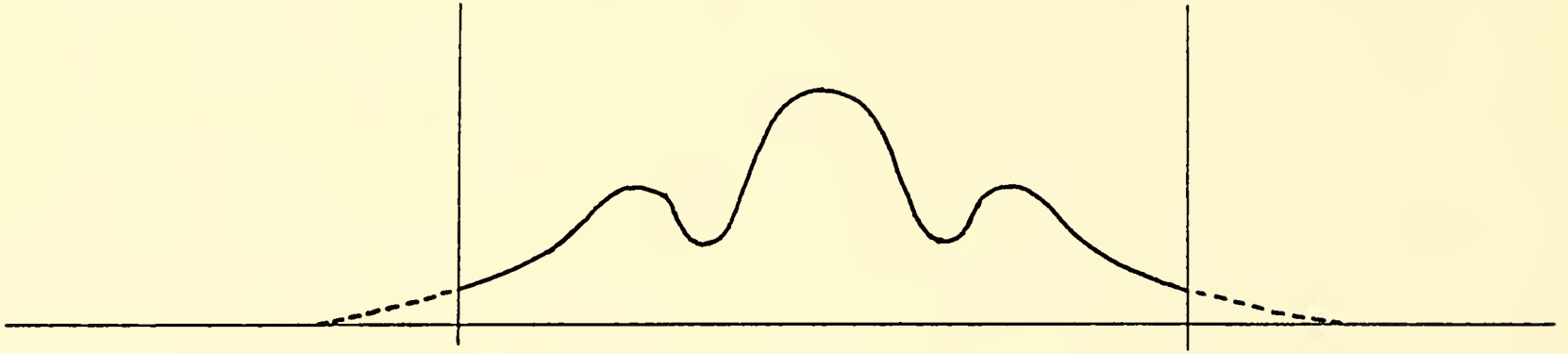


Figure 4-6
Central Object of Fig. 4-3 propagated the same distance as in Fig. 4-5 (no distortion)

On this basis the largest useful propagation distance was found to be about 200 wavelengths for an object aperture with a linear dimension of the order of 15λ.

In image processing it is often desirable to have both the center of the input object and the zero frequency component (or undiffracted or d.c. component) of the spatial frequency spectrum centered in the array. When the DFT relationship of Eqs. (4-3) and (4-4) is used the zero frequency component appears in one corner of the transformed array. Andrews [Ref. 13] recommends multiplying the array to be transformed by $(-1)^{n+m}$ before transformation, which will indeed shift the spectrum in such a way that its origin will be centered in the transform domain array. In this investigation a different approach was used because Andrews' method still leaves an unwanted phase shift for an object that is centered in the middle of the array: the DFT routine asks for an object that is centered at the origin for an output without phase shift. The center of the array was defined to be at the point (1+N/2, 1+N/2). A subroutine (called SHUFL, see Appendix) was written which interchanges the four quadrants of the array such that the array point (1+N/2, 1+N/2) after rearranging occupies the point (1,1). (See Fig. 4-7)

In interchanging the four quadrants of the array by a simple translation of the object, advantage was taken of the otherwise undesirable postulated existence of the ghost images. After transforming the array, the transform was once again operated on by SHUFL, which produced the desired effect of moving the origin of the angular spectrum to the center of the array (as previously defined) without any unwanted associated phase shift.

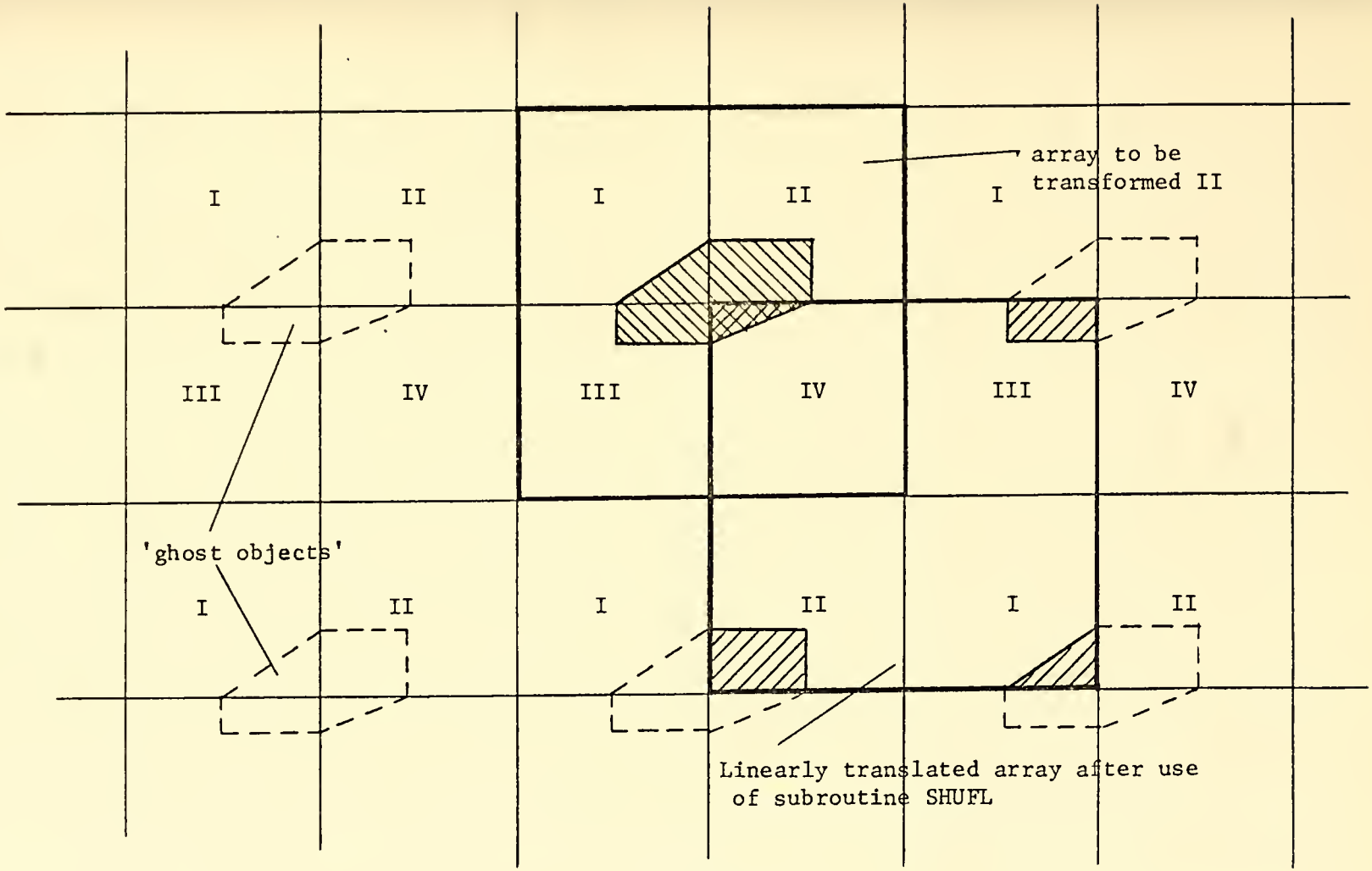


Figure 4-7

Array Before and After Use of Subroutine SHUFL

B. FAST FOURIER TRANSFORM

The discrete Fourier transform requires N^2 complex multiplications and additions for a one-dimensional record of N samples; this can be prohibitive for even moderately large N. The Fast Fourier Transform (FFT) is just a different way of computing a DFT (see [Ref. 14], but as it reduces the number of complex multiplications and additions on the order of $N \cdot \log_2 N$ its importance can hardly be overestimated. The reduction in number of computations is even more dramatic for arrays of higher dimension. Reference [15] lists a few examples. Without the economy of the Fast Fourier Transform idea the DFT would be a useful tool, but to be used very sparingly because of the large time penalty to be paid. Since Cooley and Tukey published their report [Ref. 16] digital signal processing has received new impetus because transformation operations are now possible at a tolerable even though still not a very economical time expenditure. The FFT algorithm used in this investigation (subroutine FOUR2) [Ref. 17], is included in the Appendix. The time needed for a transformation of a 64 by 64 element matrix using FOUR2 was less than 2 seconds.

V. THE DETECTION PROBLEM

To recapitulate, the problem to be solved was to reconstruct the original object from its observed sound diffraction pattern without previous knowledge of the distance of the object from the observation plane.

A. GENERAL OBJECT AND UNIVERAL RECONSTRUCTION CRITERION

For an unknown object distance a given diffraction pattern can be due to an infinite number of'objects'. It is therefore not possible without a prior knowledge to reconstruct a general object (not constrained in any way in amplitude and phase) by using a simple criterion function, i.e. one which does not contain the object distance in some coded form. The proof of this statement follows.

Consider an object U_1 belonging to a rather general class U of objects, all of which are related to each other by some common amplitude and phase characteristics. Suppose also there is a reconstruction criterion W which when applied to all diffraction patterns A from all objects belonging to U successfully reconstructs the respective object U_i. It is hoped that W is the 'universal reconstruction criterion'.

Suppose a diffraction pattern A_1 appears at a distance z from U_1. Now take the diffraction pattern A* which appears at a distance z* in front of U_1 to be a new object to be reconstructed. A* gives rise to A_1 at a distance z_1-z* in front of A*. Application of W on A_1 will always reconstruct U_1 (according to the second hypothesis). Therefore it will never be possible to extract A* from A_1 by using the criterion W. So W cannot be the desired universal reconstruction

criterion. Since U and W were chosen arbitrarily in this example, one can generalize that no single simple criterion can be used to recover all objects.

The expression 'simple criterion' has been used in the above discussion for the reason that it was proposed to make two diffraction patterns of the same object using different wavelengths. Then, on backward propagation both images should coincide at the object location but not at different propagation distances, at least over a considerable portion of the array. This idea was not investigated any further because it was thought to circumvent the issue of the problem. The difficulties in this approach are obvious at once: The need of recording two diffraction patterns with the associated long time expenditure, the need of almost twice as much computer storage and time etc. Moreover it was hoped that a single diffraction pattern would contain enough information to find the original object if the class of objects to be reconstructed were somewhat limited.

Recently a reconstruction time has been proposed by Kalra and Rodgers [Ref. 18] that works reliably on a class of objects, among them the point scatterer or source, which have a magnitude of the acoustical disturbance which reaches its maximum at the object location. Unfortunately this very simple criterion will not reconstruct successfully objects whose phase distribution departs much from a linear one. In the above reference [Ref. 18] the possibility of extending the analysis from a point source to a general object is suggested. Of course any object can be thought of as a (possibly infinite) sum of point sources and the diffraction pattern due to that object as resulting from the appropriate accumulation of point sources if the phase relationship between all

point sources is accounted for. But the assumption that maximum amplitude is a general reconstruction criterion can be disproven (without going to such exotic 'objects' as other diffraction patterns) by considering an uniformly insonified object, whose phase varies in a quadratic fashion such that it is advanced towards the edges and retarded in the center, i.e., a lens-type phase distribution. Then the sonic energy in propagation will concentrate near the focal point of this 'lens'. As the energy density there is much higher than at the uniformly insonified object (see [Ref. 19]), the amplitude of the sound field must be higher than at the object. This is due to the constructive interference introduced by the chosen phase relationship. The situation as above was simulated on the computer: A 16λ by 8λ slit with a phase distribution that had an equivalent focal length of 10λ was insonified by a unity amplitude plane wave. The computed amplitude at the focal point was found to be more than a factor of 8 higher than the amplitude at the slit plane. So the proposed 'unbiased estimator' for the true object distance z* of above reference [Ref. 18] will always choose the focal plane as object location. It may be argued that a lens-type phase relationship is very uncommon in nature, but random phase distributions (diffuse scattering) are widely observed, and random phase relationships between the point sources that make up an object can also give rise to amplitudes of the disturbance which are higher at other planes, than at the object location. As example a 16λ by 8λ unity amplitude object was taken to have random phase distribution between array points, uniformly distributed from 0 to 2π. The highest amplitude was found to be 2.1 times the value at the slit at a location 5λ in front of the object. This must not necessarily be so; for other distributions the

amplitude may be highest at the object location (the linear distribution might be considered a very special case of a random distribution, as might be the parabolic distribution), but it introduces an element of uncertainty into the reconstruction scheme which may well be unbearable.

In the initial stages of this investigation it was hoped that cross-correlation of the diffraction pattern with a point source at the correct distance would give the necessary high output to discriminate between the object giving rise to the diffraction pattern and some other diffraction pattern due to this object.

Suppose there is a unity amplitude point source at the orgin. Then

$$\underline{U}_p\ (x,y,o) = \delta(x,y,o) \quad \text{and}$$

$$\underline{A}_p\ (f_x,f_y,o) = 1$$

If the disturbance due to this point source is propagated a distance z then

$$\underline{A}_{pz}\ (f_x,f_y,z) = \underline{A}_p\ (f_x,\ f_y,\ o)\cdot \exp\ [j\tfrac{2\pi}{\lambda}z\ \sqrt{1-(\lambda f_x)^2-(\lambda f_y)^2}]$$

$$= 1 \cdot \exp\ [j\tfrac{2\pi}{\lambda}z\ \sqrt{1-(\lambda f_x)^2-(\lambda f_y)^2}]$$

Now cross-correlation can be accomplished by multiplying the spectrum of one function by the complex conjugate of the spectrum of the other function and inverse Fourier transforming the result.

Let the Fourier transform of the diffraction pattern $\underline{U}_z(x,y,z)$ be given by $\underline{A}_z(f_x,\ f_y,\ z)$. Then the result of the proposed crosscorrelation can be written as:

$$\underline{R}\ (x,y,o) = F^{-1}\{\underline{A}_z\ (f_x,f_y,z)\cdot \exp[-j\tfrac{2\pi}{\lambda}z\ \sqrt{1-(\lambda f_x)^2-(\lambda f_y)^2}]\}$$

which is recognized as the result obtained by backward propagation of the diffracted wavefront by a distance z, namely the original object. This method of maximizing the crosscorrelation cannot be used as it has been shown that the amplitude of the wavefront at the object need not be higher than everywhere else in its propagation field.

B. OBJECTS WITH BOUNDARIES AND THEIR RECONSTRUCTION

In the search for a class of objects which might be more useful than the one reconstructed by Kalra and Rodgers [Ref. 18] a brief study of psychopictorics, that subfield of psychophysics which deals with natural images as pictorial stimuli (see Lipkin and Rosenfeld [Ref. 20]) showed that the three most important psychopictorial variables are: contrast and border; shape and geometry; texture. Of these contrast and border is easiest to relate to what is actually available in a digitized picture. To treat shape and geometry would mean going to a pattern recognition system, probably by using a crosscorrelation scheme, with the attendant necessity of storing a large number of objects to be recognized. Trying to base a detection system on the texture of an object compared to that of its surroundings would probably be bound to fail with today's still rather crude acoustical holography systems. Thus, trying to detect objects which are characterized by contrast and boundary seems most promising. In fact, many objects that are of interest to acoustical holography fall under this heading in that they are characterized by emitting or reflecting large amounts of sonic energy and being separated from the non-emitting (or less emitting) background by a rather sharp boundary or vice versa. The 'picture' of a mine lieing on the ocean floor will show a sharp step in intensity as will that of a bone imbedded in soft tissue and there are more examples that might be cited.

1. Objects of Interest

It is hoped then that this class of objects will be general enough to warrant investigation. Therefore in all the following discussion it will be assumed that the object plane to be reconstructed from its diffraction pattern has associated with it (or ideally is characterized by) a sharp change in the emitted amplitude (ideally a discontinuity) separating regions of large amplitude from those of low amplitude. This change in amplitude will not be assumed to be confined to a single point or to a number of disconnected points because the word 'boundary' or 'edge' implies connection between quite a large number of points. In this sense then a point source is not a valid object for this discussion. A line object cannot be hoped to be detected, because it really has not any area associated with it; the same objection will hold for an object that is checkerboard-shaped with the amplitude changing from high to low and back again from one sampling point to the next one. The smallest object that one could hope to detect is composed of four sampling points of high intensity next to each other, arranged in a square. This probably would not be a great obstacle to the use of the proposed routine as it implies detection of at least all one wavelength by one wavelength objects if the Nyquist sampling rate is used.

2. Reconstruction Schemes

The first reconstruction scheme for the given class of objects was very simple: As the object was assumed to have a sharp step in amplitude it was thought that the difference in amplitude between one sampling point and its neighboring sampling point would have a maximum at the object location and not in any other plane. Therefore that

plane which contains this maximum would be taken to be the object plane and the reconstructed diffraction pattern in this plane would be the image of the original object. Unfortunately the above assumption was proven to be not valid for the Nyquist sampling rate if the phase distribution over the object departs markedly from a linear one, by assuming such an object and simulating its diffraction on the computer. Random phase objects as well as such with a lens-type phase relationship showed maximum differences between some sampling points which were higher (even very much higher in the case of the parabolic phase object) in other planes than the discontinuity in the object plane. As examples: A 16λ by 8λ unity-amplitude object with uniformly distributed random phase had a maximum difference of 1.73 at a distance 3 wavelengths in front of the object. The same object with lens-type phase distribution and a focal length of 10λ showed a maximum difference of 3.85 in the focal plane. At first this was not taken to invalidate the whole idea, because theoretically by finer and finer sampling it would be possible to reduce the maximum difference between sampling points in any other plane so far that it would be smaller than the given discontinuity in the object plane, which would not be influenced by finer sampling. But apart from the obvious difficulty in an actual system to sample fractions of a wavelength apart (the detector would have to have very small area indeed) the hoped for gain could only be realized at an immense increase in amount of data and additional computation time and core memory necessary to handle that many data. If one samples finer without increasing the number of samples (which might be possible for very small objects) nothing is gained. This hinges on the fact that by sampling above the Nyquiest rate too few frequency components will be

available to faithfully reconstruct the original object. (see Sect. IV) The image will lose its clear amplitude step and instead will develop a lower amplitude slope and overshoots; in this way the characteristic that discriminates between actual object and its diffraction pattern is lost. Some other way has then to be found to detect the edge of the intensity distribution.

In the above scheme no use was made of the fact that an edge consists of more than one spot where the amplitude changes rapidly, that a number of such changes have to occur at adjacent sampling points and that these places of rapid change have to separate a region of high amplitude from one of low amplitude.

The problem of enhancing or detecting an edge in a digital picture has been treated in the literature, e.g., by Rosenfeld, Lee, and Thomas in [Ref. 21]. The usual treatment is to take differences of averages, e.g. for a one-dimensional case

$$d_{mk} = \frac{1}{m} \cdot \left| a_{k+m} + \cdots + a_{k+1} - a_k - \cdots - a_{k-m+1} \right|$$

where a_k is the amplitude of the point being investigated. The difference $d_{1k}=a_{k+1}-a_k$ is sensitive to every local fluctuation, whether it occurs at an edge or not. The difference d_{mk} for large m has the disadvantage of detecting even sharp transitions in a blurred manner since it registers a difference for all k such that some of its terms are in one segment, some in the other segment. A simple method of overcoming these difficulties is using a product of d_{mk}'s, which involves both small and large m's. This product can only be large if all its factors are large. For the two-dimensional case this edge detection scheme naturally becomes more involved. Operations of the form

$$d_{rs,nk} = \frac{1}{r(2s+1)} \Big| (a_{h+r,k+s} + \cdots + a_{h+r,k-s} + \cdots + a_{h+1,k+s} + \cdots + a_{h+1,k-s}) - (a_{h,k+s} + \cdots + a_{h,k-s} + \cdots + a_{h-r+1,k+s} + \cdots + a_{h-r+1,k-s}) \Big|$$

are performed and products of the results for various large and small r's and s's are formed. Again the product can only be relatively large if all its factors are large. The example $d_{rs,hk}$ shown above is used for the detection of a horizontal edge, the parameter s increasing results in the sensitivity towards orientations departing from the horizontal being accentuated. Each possible edge orientation has to be treated by a different combination of differences of averages, obviously a time-consuming task for an array of say 64 by 64 sample points. The edge detection operation would normally then be followed by a curve detection routine to filter out any spurious output, as the output of the edge detection step will be noisy for an edge which is not perfectly smooth and straight. The curve detection idea is based on the fact that the values resulting from the edge detection should be high on moving along the edge and low in a direction perpendicular to it. In other words a high point on an edge should have neighboring high points in approximately opposite directions. Points which do not satisfy this criterion are regarded as spurious and discarded. For a more profound discussion of this topic reference is made to Rosenfeld, Thomas and Lee [Ref. 22]. Again, the time needed for this step in the recognition process cannot be regarded as negligible, because it involves checking the neighborhood of N^2 points and a quite involved decision-making process.

To receive some information on the computation time needed using the above method of edge detection a first approximation to it was developed and tested. In order to make the operation $d_{rs,nk}$ relatively

insensitive to edge orientation s was taken to be 1. Differences $d_{21,hk}$, $d_{41,hk}$ and $d_{81,hk}$ were formed and multiplied together for the horizontal and vertical edge orientation and the products for the horizontal and vertical operation added together. It was hoped in this way to overcome the necessity of investigating all different orientations. The result obtained with this technique was promising. (The test was performed using a circle, whose area was filled with high values, as a boundary comprising all edge orientations as input to the routine. Input and output are shown in Fig. 5-1 and Fig.5-2). Nevertheless this method had to be shelved because of the long computation time needed, over 20 seconds for each 64 by 64 element array. To this time would have to be added the time needed for the previously discussed curve detection algorithm also.

The difficulty of having to treat each orientation separately is analogous to that which is encountered in trying to accomplish edge detection by a cross-correlation scheme, which also is very sensitive to a relative rotation between the reference object and the object to be tested, and additionally has to account for different phase relationships also.

In order to surmount these difficulties due to various edge orientations a new, rather simple (and thus more error-prone) procedure was worked out. In this algorithm first the largest difference between any two adjacent array points in the plane under investigation is found. This is used to set two thresholds, an upper (THRUP) and a lower one (THRLO). These thresholds are kept variable, so that they can be adjusted according to the kind of object one is looking for (if such previous information is available). A setting of 60% of the maximum difference

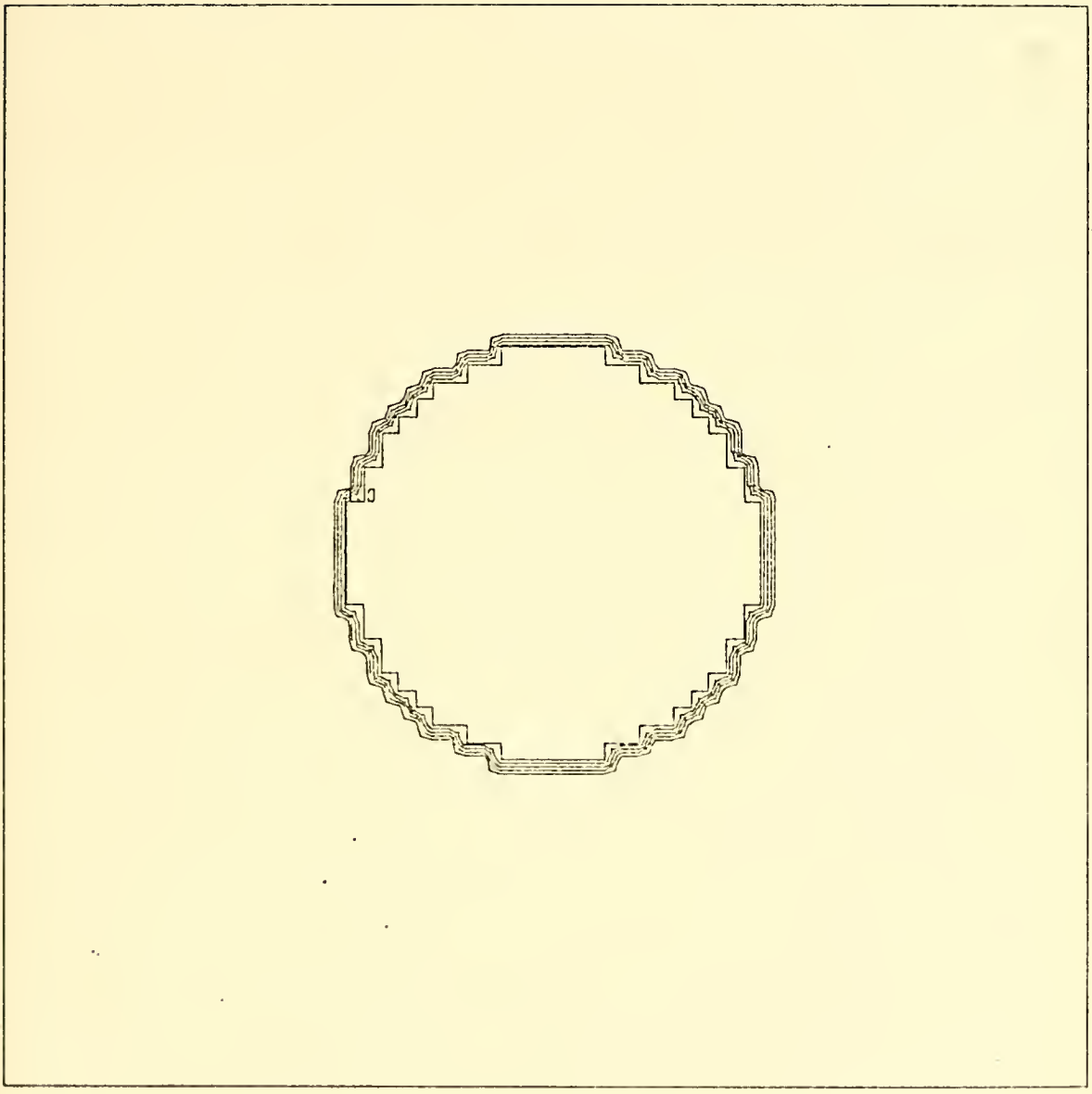


Figure 5-1
Amplitude Contour Plot of Input to
Edge Detection Scheme to be Tested

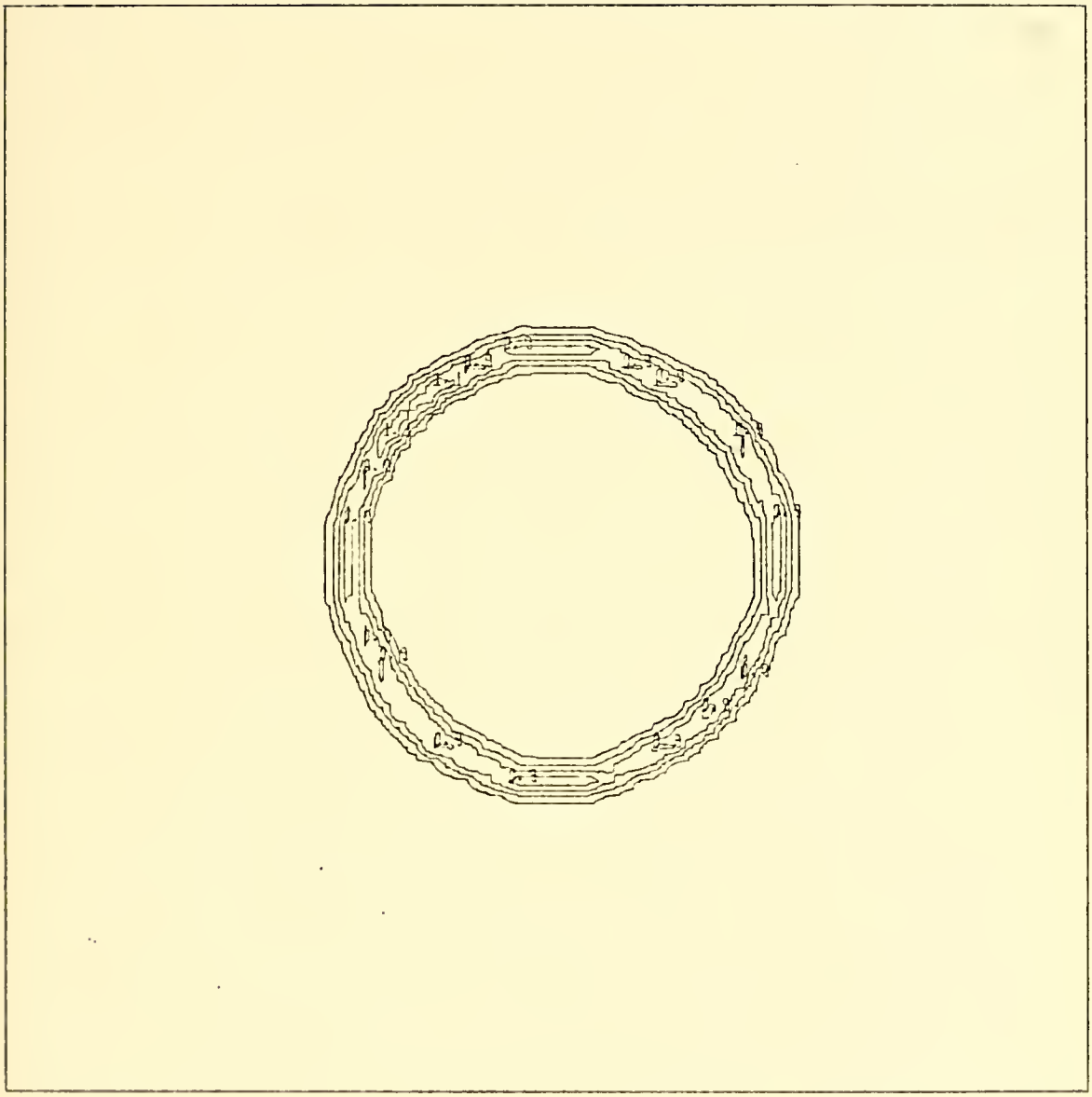


Figure 5-2
Amplitude Contour Plot of Output of
Edge Detection Scheme to be Tested

between array apoints for the upper threshold and 20% for the lower one seems in general to give good results. An edge in one direction is assumed to exist at an array point (i,k) if for a downgoing step $(a_{i,k}-a_{i,k+1})$ is greater than the upper threshold and both $|a_{i,k+1}-a_{i,k+2}|$ and $|a_{i,k-1}-a_{i,k}|$ are less than the lower threshold. This criterion is analogous to the edge detection routine described earlier with S = 0 and the product operation replaced by its logical equivalent 'AND': The output in both cases will be positive ('YES' in this case, large in the former) if all the inputs (differences) to the operation are positive ('YES' or large).

This edge criterion is used twice at each array point, for horizontal and for vertical differences. In this manner corners as points belonging to two edges at the same time are weighted more than simple edge points. (The deemphasis of corners is one unfortunate feature of the earlier edge detection scheme). Also, a distinction is made between upward steps $((a_{i,k+1}-a_{i,k})$ must be greater than the upper threshold, while the other two conditions stay unchanged in this case) and downward steps. This feature is used in that part of this scheme which assigns different values to each edge point according to whether it is connected to other edge points or not: The step heights for an array point (i,k) are computed in both the horizontal and vertical directions; values of positive going steps are kept separate from negative steps; then the four array points next to (i,k) which have been treated previously in the sequential edge search are tested for the highest absolute value of assigned 'edgemeasure'. If this is due to an upward going edge the height of the positive step at the point in question is added to it and stored at (i,k). The treatment is analogous if the highest value stored next to (i,k) has been found

from a negative edge; the negative step height at (i,k) is added to it and kept at (i,k). If the highest adjoining edge measure is positive, but only negative steps occur at (i,k) this negative step height is stored at (i,k) without any addition of previously computed edge measure. The same thing happens if the adjoining array points contain zero. Zero is stored at (i,k) regardless of adjoining edge measures if the situation at (i,k) and its vicinity does not meet the edge criterion using the upper and lower thresholds. For a more exhaustive treatment of all different cases that may arise see the computer flow diagram for subroutine ADERIV in the Appendix. After the whole array has been treated in this manner, all edge measures computed are added in absolute value; the result is taken to be the overall edge measure of the given array.

It must be admitted that the only justification for this scheme that can be given, is that it seems natural in looking for edges to discriminate between large and small differences between adjoining points and that longer connected boundaries should be weighted more than short ones: three points of rapid change next to each other might be considered at least as a beginning of an edge, whereas two such points could be just a random occurrence. The normal height of the thresholds as given is found entirely from practice with this routine and has no profound mathematical background. It may be argued that in the decribed algorithm too much importance is attributed to connection between edge points and too little to the height of the single steps. But it has been observed that if an array containing fewer, large steps is compared with another array with many smaller steps next to each other, the latter is more likely to be the looked for object than the former. Besides that, a shift in the relative weight balance between

'connection' and 'step height' is easily accomplished by multiplying the largest absolute value of edge measure adjoining the point (i,k) by some factor less than one before the addition and storage at (i,k).

The algorithm can easily be made more discriminating against steps which do not belong to an actual edge by taking into account not merely two differences next to the point in question but investigating more points in its neighborhood. Then it must be borne in mind that as more small differences in one direction are required to satisfy the step criterion the smallest detectable object will become larger. Another means of increasing the rejection capability is by increasing the upper threshold and reducing the lower threshold if that is compatible with the kind of object to be investigated: setting a lower threshold of 0.1 of the maximum difference, if the amplitude of the background of the object is uniformly random distributed between zero and 20% of the object discontinuity, could reject quite a number of bona fide edge points thus reducing the probability of detection. It is inadvisable to work with an edge detection routine which is too discriminating from another point of view also: this routine might not pass the diffraction patterns a small distance behind and in front of the object plane with a high enough edge measure to make them obvious. Thus in stepping through the volume of interest the computer would have to investigate very near to the actual object to receive an indication of its presence or it would be missed altogether and the computer would settle for some other than the true object distance and output the diffraction pattern there as the reconstructed object. This would mean that many small steps would have to be taken, which implies very long computation times. This case could not appear if the overall edge measure as computed above were

steadily increasing towards the actual object location. Then it would be just a matter of finding the maximum edge measure and the object would be detected. Unfortunately, the situation is not that simple. Figure 5-3 shows the computed edge measures for a 16 by 8 wavelength, unity amplitude object with random phase, versus propagation distance in steps of 2.5 wavelengths. It shows quite a number of planes in which the edge measure has a relative maximum. This is not only due to the way in which the edge measure is computed but also due to the rapid variation of the diffracted wavefront with distance.

A reconstruction method then has to start from a diffraction pattern and taking advantage of all previous information (which might have been obtained by rangegating) propagate this diffracted wavefront back to some starting distance. From there to some given final distance the wavefront will be propagated back in as small a number of steps (as large a step size) as possible without missing the object altogether, meaning that the step size has to be so small that an indication of a possible object is received by a relative edgemeasure maximum less than half a step size behind or in front of the actual object location. In the proposed algorithm this is accomplished by setting the parameters ZFIRST for the starting distance, ZLAST for the final distance and NRSTEP for the number of steps to be used. The choice of step size for this first pass through the volume of interest is thus left to the operator. For an object of 16 by 8 wavelengths and Nyquist rate sampling the step size should not be chosen larger than about 2 wavelengths. Larger objects can bear a larger step size.

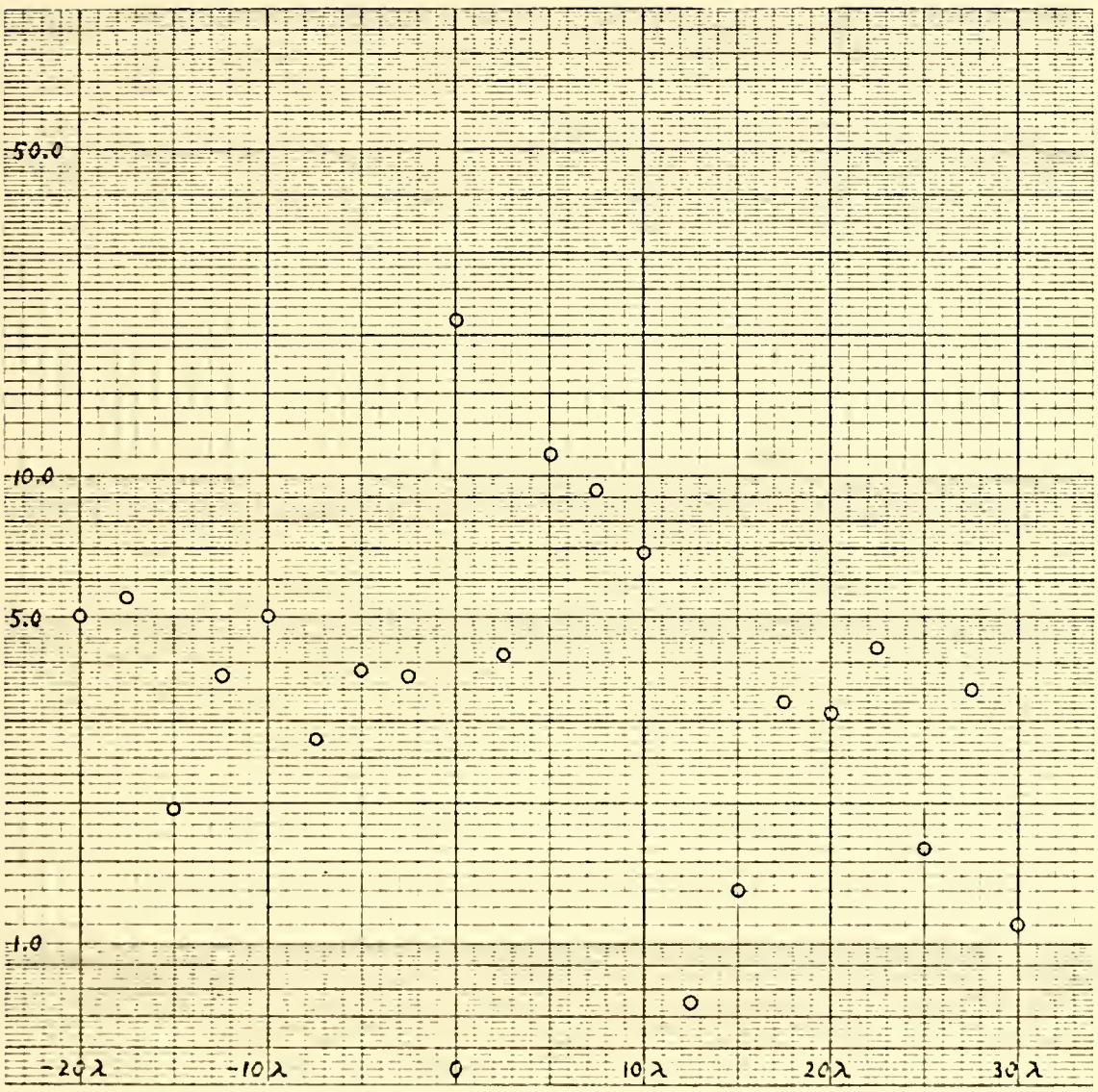


Figure 5-3
Edge Measure of Diffraction Pattern of Random
Phase Slit Versus Propagation Distance

All relative maxima in overall edge measure observed on this first pass have to be stored for future reference, together with the propagation distances at which they occur.

Different treatments of the result of this first pass are possible now. For instance the reconstructed wavefront at all locations of a relative maximum might be displayed on a plotter, or graphics terminal and the decision as to the actual location(s) of object(s) left to the experience of the operator, who would then investigate those locations closer. The method chosen for the proposed algorithm differs from the above by making further use of the computer's decision-making capabilities.

After the first pass through the whole distance of interest the vector of edge measure maxima is reordered according to their size and those possible object locations discarded, whose edge measure is only 10% or less of the maximum edge measure encountered on the first pass, it being assumed that closer investigation of these locations will not yield the looked for object. The remaining locations are investigated in a fixed step size which is set equal to the sampling increment, i.e, to one half a wavelength if the Nyquist rate is used, from a distance one half of the original step size in front of the relative maximum to one half that step size behind this location. The absolute maximum in edge measure in this interval is found and stored in a different array. After all relative maxima resulting from the first pass have been investigated the array of second pass maxima is scanned for the overall maximum and those locations discarded where the edge value is below a certain fraction of the overall maximum. This is accomplished by

setting a threshold THRCON to a fraction of 1.0: one would not expect maximum of less than 0.2 to 0.3 of the overall maximum to be due to actual objects. Also incorporated into the algorithm was the possibility of investigating fewer relative maxima resulting from the first pass by reading in a value for NRMAX, e.g., if it is known that the object exists only in a single plane parallel to the observation plane. This option should be used with caution because there is no guarantee that the highest relative maximum after the first pass will be associated with the actual object for all objects (although this is true for many simple objects).

The results of propagating the wavefront to the locations of the remaining second pass maxima are then plotted on the CALCOMP - plotter, either in form of a contour plot of the computed amplitude or as a surface composed of the values of amplitude at all array points, from which the hidden lines are removed (other graphical output devices can also be used). The contour plot will give a much better pictorial representation of the outline of the object, but a good part of the amplitude information is lost by plotting only a number of contour lines. The three-dimensional representation is to be preferred on the basis of information contained in the plot (it also suffers from loss of information because it is not possible to see through the surface) but it takes much longer computation times and requires much more core memory than the contour plot. Also it must be remembered that the apparent dimensionality is not due to any three-dimensional object of that shape but rather represents the relative amplitude of the wavefront in a single plane. Subroutines used for this output are for

the contour plot CONTUR [Ref. 23] and PRO1 for the surface plot [Ref. 24]. For a 64 by 64 element array to be treated CONTUR takes on the order of 50 sec if six contour lines are plotted; PRO1 needs for the same array from 50 sec up to 10 min depending on how complicated the surface to be plotted is.

VI. TESTS AND RESULTS

Part of the tests performed were designed to check the propagation algorithm, the greater part were used to try the automatic reconstruction scheme. For this latter purpose cumputer generated diffraction patterns and one set of actually observed data were used.

A. TEST OF THE PROPAGATION ALGORITHM

As the spatial frequency approach to diffraction is limited in the maximum useful propagation distance due to the presence of the postulated ghost images it is not possible to apply the usual test for diffraction algorithms in which the amplitude distribution at a very large progagation distance has the functional form of a Fourier transform of the diffracting aperture. Instead the fact was used that illuminating an aperture with a spherical wave converging to a point P_o will give rise to an amplitude distribution in a plane perpendicular to the direction of propagation and containing P_o, which again is simply related to the Fourier transform of the diffracting aperture. [Ref. 7].

A 10λ by 10λ square aperture was simulated to be insonified by a spherical wave converging to a point 40λ behind the aperture. Using the proposed propagation algorithm and propagating the diffracted wavefront 40λ resulted in the output of Fig. 6-1. It is readily apparent that this output has the functional form of $(\sin(x)/x)\cdot(\sin(y)/y)$ (which is the Fourier transform of a uniformly insonified rectangular aperture). The fact that the surface of Fig. 6-1 is not very smooth is due to the relatively small number of surface points. Due to the presence of the ghost images the output is deteriorated towards the

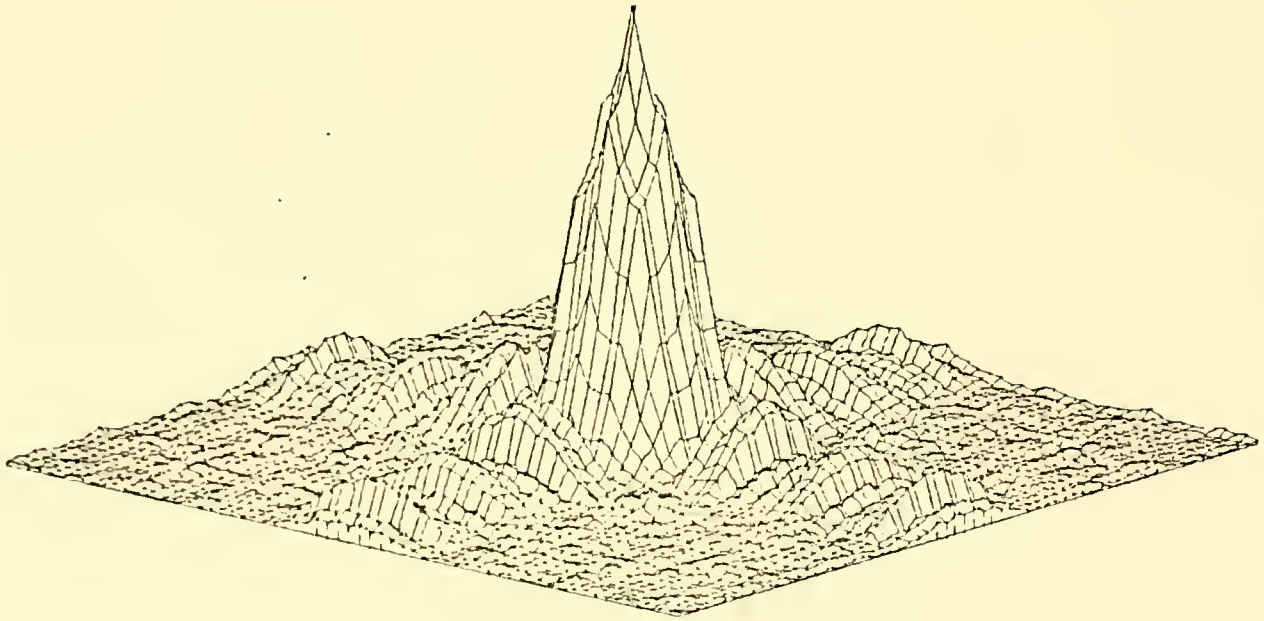


Figure 6-1
Amplitude Distribution in the Focal Plane of a Square Aperture Insonified by a Converging Spherical Wave

boundary of the presented surface (the amplitude does not fall off as fast as it should in following a $(\sin(x)/x)\cdot(\sin(y)/y)$ functional relationship).

It is of interest to note that even after about 30 sequential propagation steps the result differs from that obtained in one step only in the fourth decimal place. This good agreement could never be expected if the Fresnel equation were used.

B. TESTS OF THE RECONSTRUCTION SCHEME

1. Tests Using Computer Generated Diffraction Patterns

Most of the tests of the automatic reconstruction scheme were performed using computer generated diffraction patterns. They have the advantage of being perfect in the sense of having no distortion introduced by noise in the detection system, by reverberation and by scan non-uniformities. In all tests involving computer generated diffraction patterns the distance from object to observation plane was chosen arbitrarily (subject to the constraint of being small enough for the results of the spatial frequency approach to diffraction to be valid). The numbers of steps given in the following discussions are those for the first pass through the volume of interest (i.e, those given by setting the input NRSTEP). The number of steps for the closer investigation of relative maxima in edge measure is computed by the detection algorithm (as discussed in the last section). All computation times are without those necessary for the output subroutines (CONTUR or PR01).

Case a: The simplest object whose reconstruction was attempted was a 16λ by 8λ uniformly insonified rectangular slit with all its sample points in phase thus simulating a specular object. The diffraction

pattern was observed 175.3λ in front of the object; in the reconstruction the closest distance to be investigated was set to 160λ, the furthest distance was 192λ; this volume of interest was investigated in 12 steps. A reconstruction (see Fig. 6-2) was obtained at a distance of 175.25 wavelengths from the observation plane, i.e. only 0.05λ away form the original object. The time needed was about 2.6 min.

Case b: The same object as in case a was assumed with the additional difficulty of a background insonification with an amplitude which was randomly distributed between zero and 0.2 of the object amplitude. Also the phase of the background insonfication was taken to be uniformly random distributed between zero and 2π. The distance between object and observed diffraction pattern was 177.2λ. The volume of interest and number of steps were as in case a. The obtained reconstruction of Fig. 6-3 was located 0.05λ behind the actual object location. In order to show the random amplitude variation of the background here the output is given in form of a 3-dimensional surface. Computation time was about 2.6 min.

Case c: An object of the same size and shape as in case b, was treated in this case. The random background, too, was the same as in case b. This case differed from case b, in the assumption of a parabolic phase relationships for the object which gave it an equivalent focal length of 10 wavelengths. The observation plane was located 177.1λ in front of the object. Initial and final distances remained unchanged from cases a and b, as did the number of steps. An object reconstruction was obtained at a distance of 177.25, requiring a computation time of 3.3 min.

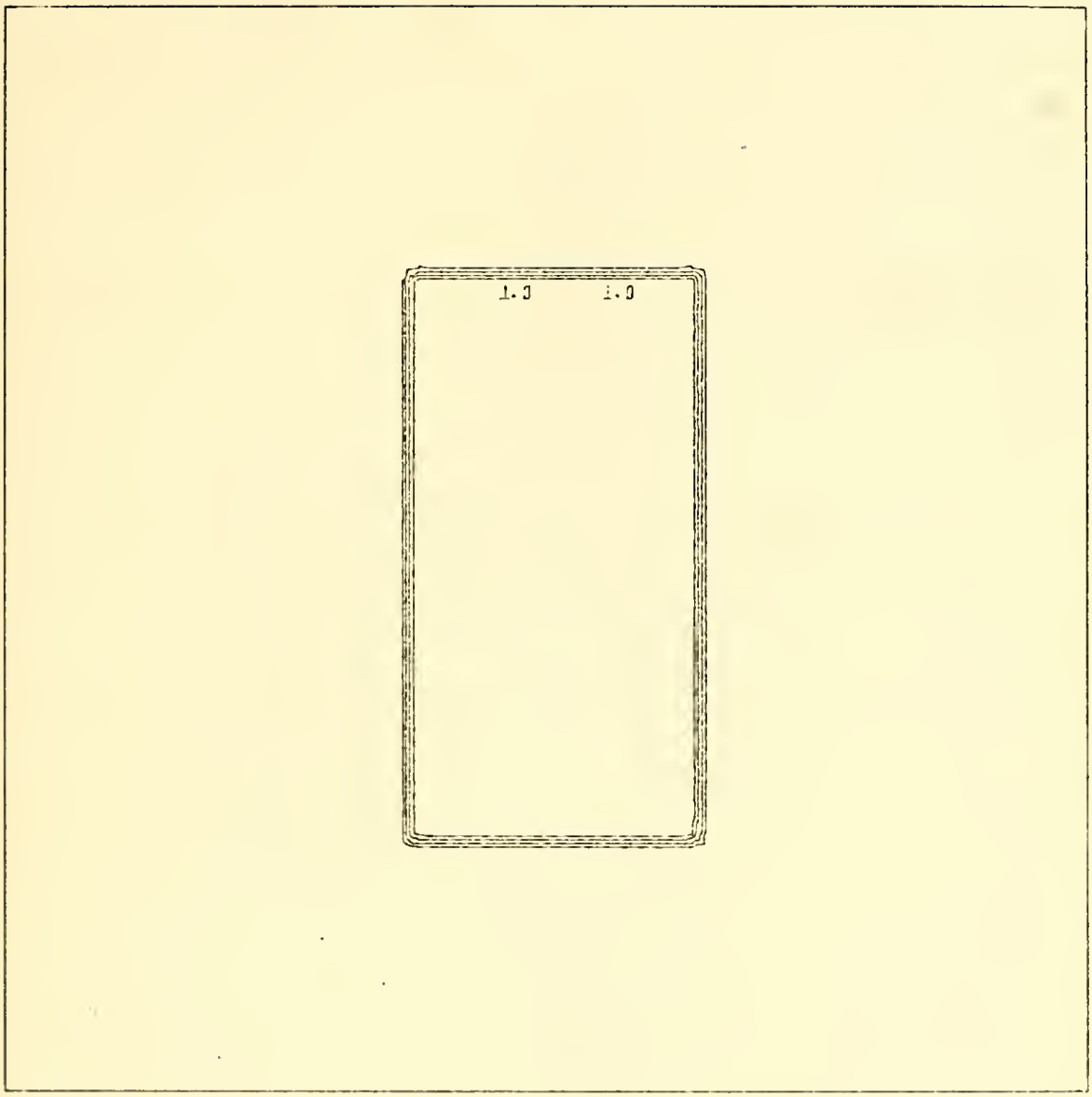


Figure 6-2
Amplitude Contour Plot of
Reconstructed Specular Object
(Case a)

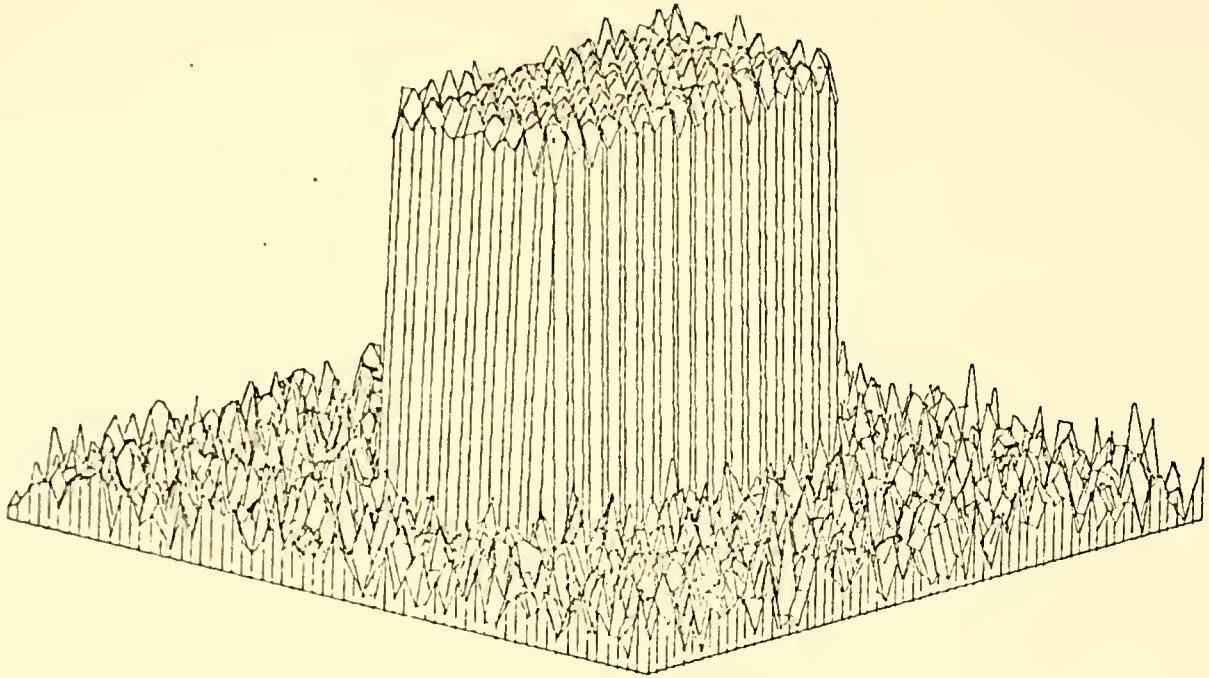


Figure 6-3
Surface Plot of Amplitude of Reconstruction of Specular Object with Random Phase, Random Amplitude Background (Case b)

Case d: To show that the proposed reconstruction scheme works also for other than rectangular objects a shape was chosen for the input object which can best be visualized by viewing the reconstructed output of Fig. 6-4. Background and phase distribution of the object are the same as in case c. Propagation distance to the observation plane was 177.1λ. Initial and final distances for the investigation were set to 160.0 and 190.0 wavelengths respectively, the investigation was performed in 12 steps. The reconstructed object (see Fig. 6-4) was located 177.25λ from the observation plane. Time needed for the reconstruction was about 4 min. Figure 6-5 shows the same output as Fig. 6-4, only in the 3-dimensional surface plot. This figure was included here to illustrate some of the shortcomings of this output scheme: Even for a relatively simple shape as that from Fig. 6-4 the 3-dimensional plot does not show the object shape clearly. Moreover it exhibits certain errors due to the rapid variation of the amplitude function which cannot sufficiently be accounted for in the interpolation scheme. The contour plot (Fig. 6-4) on the other hand, does not show the random variation of amplitude in the background insonification while exhibiting quite clearly the shape of the object.

Case e: For this case a 16λ by 8λ rectangular uniformly emitting object was assumed to have a uniformly distributed random phase distribution between zero and 2π whcih simulates a diffuse object. No background sound was taken to exist. The diffraction pattern was calculated for a observation distance of 177.5λ. The volume of interest remained unchanged between 160λ and 190λ, as did the number of steps in the investigation. One would expect this case to be harder to solve due to the strong deterioration of the best possible reconstructed object

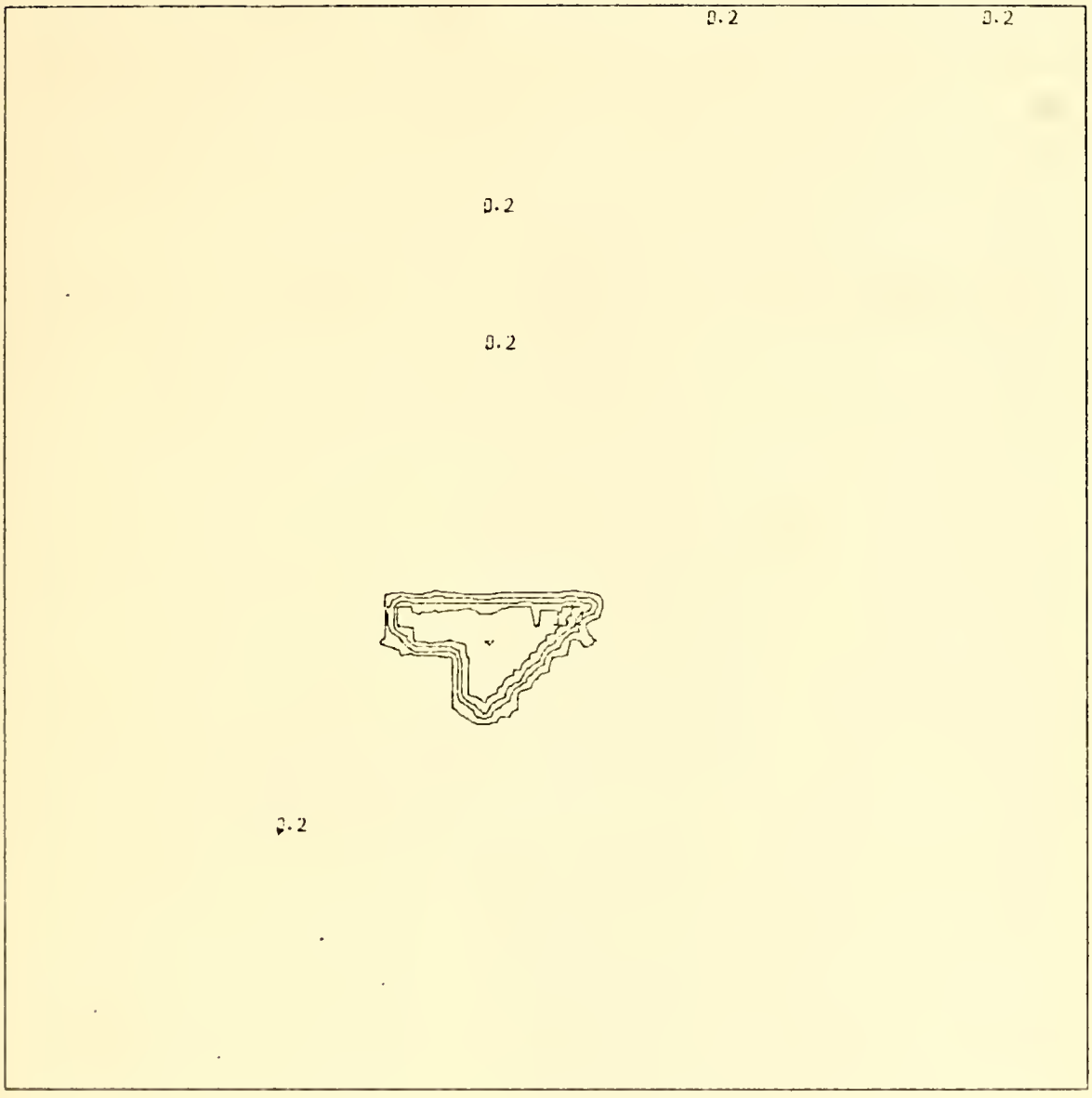


Figure 6-4
Amplitude Contour Plot of Reconstructed Parabolic Phase
Object with Random Phase, Random Amplitude Background (Case d)

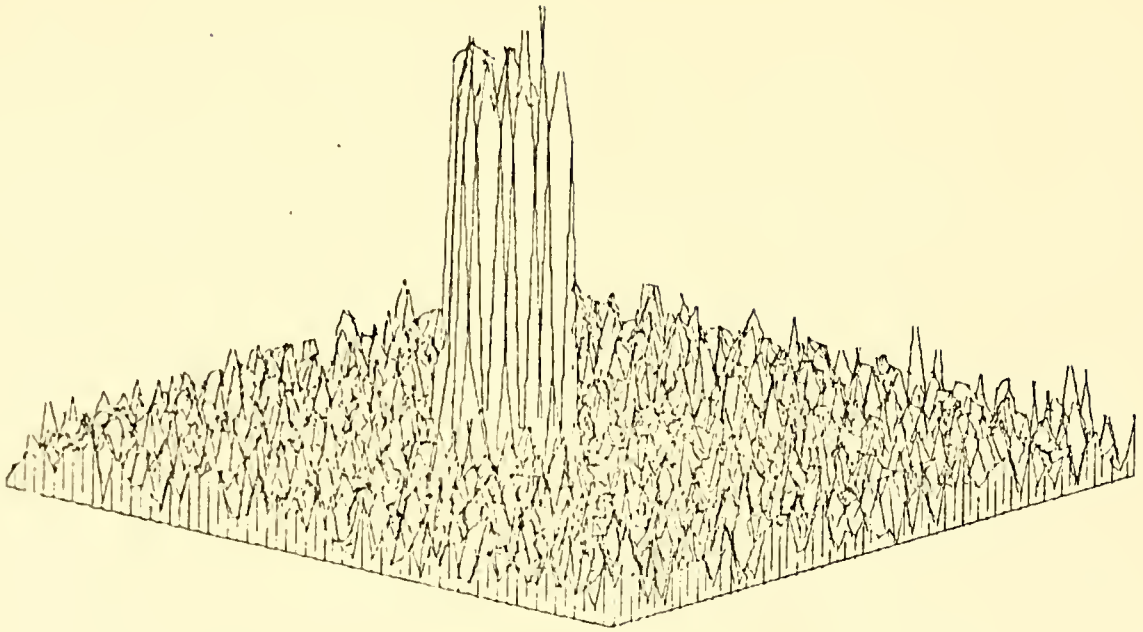


Figure 6-5
Surface Plot of Amplitude of Reconstructed Parabolic Phase
Object with Random Phase, Random Amplitude Background (Case d)

(see Fig. 3-6). The computer found two possible object locations. Plots for the diffraction patterns at these locations (177.5λ and 188.25λ) are shown in Fig. 6-6 and 6-7 respectively. The first of these locations is the actual object location and the plot Fig. 6-6 the best output that can be expected. Computation time was 5.5 min. It is surprising that the edge-detection routine does not discard the array of Fig. 6-7, but instead assigns it a value of overall edge-measure which is 70% of that of the actual object. Here it would be necessary for a human operator to use his pattern-recognition capability to discard the second object hypothesis.

Case f: As the last test for the reconstruction of objects which exist only in a single plane a 16λ by 8λ object was simulated with a background insonification whose amplitude was uniformly random distributed between zero and 0.2 and whose phase was assumed to vary uniformly between zero and 2π. The object itself had an amplitude which consisted of a constant term of 0.9 to which was added a randomly varying value which was uniformly distributed between zero and 0.2. Also, a random phase variation between zero and 2π was assumed for the object. The actual distance between object and observation plane was 177.5λ. The initial distance, final distance and number of steps were left the same as in case c. The computer required 5 min. to calculate two tentative object locations at 177.5λ and 163.75λ. The diffraction patterns at these locations are shown in Figures 6-8 and 6-9 respectively. The edge-measure assigned by the computer to the spurious output of Fig. 6-9 was 0.75 of that of the correct output.

From all these tests it would seem that one could expect the computer to be able to reconstruct any object which has a sharp edge

Figure 6-6
Amplitude Contour Plot of Reconstructed Random Phase Object
(Case e)

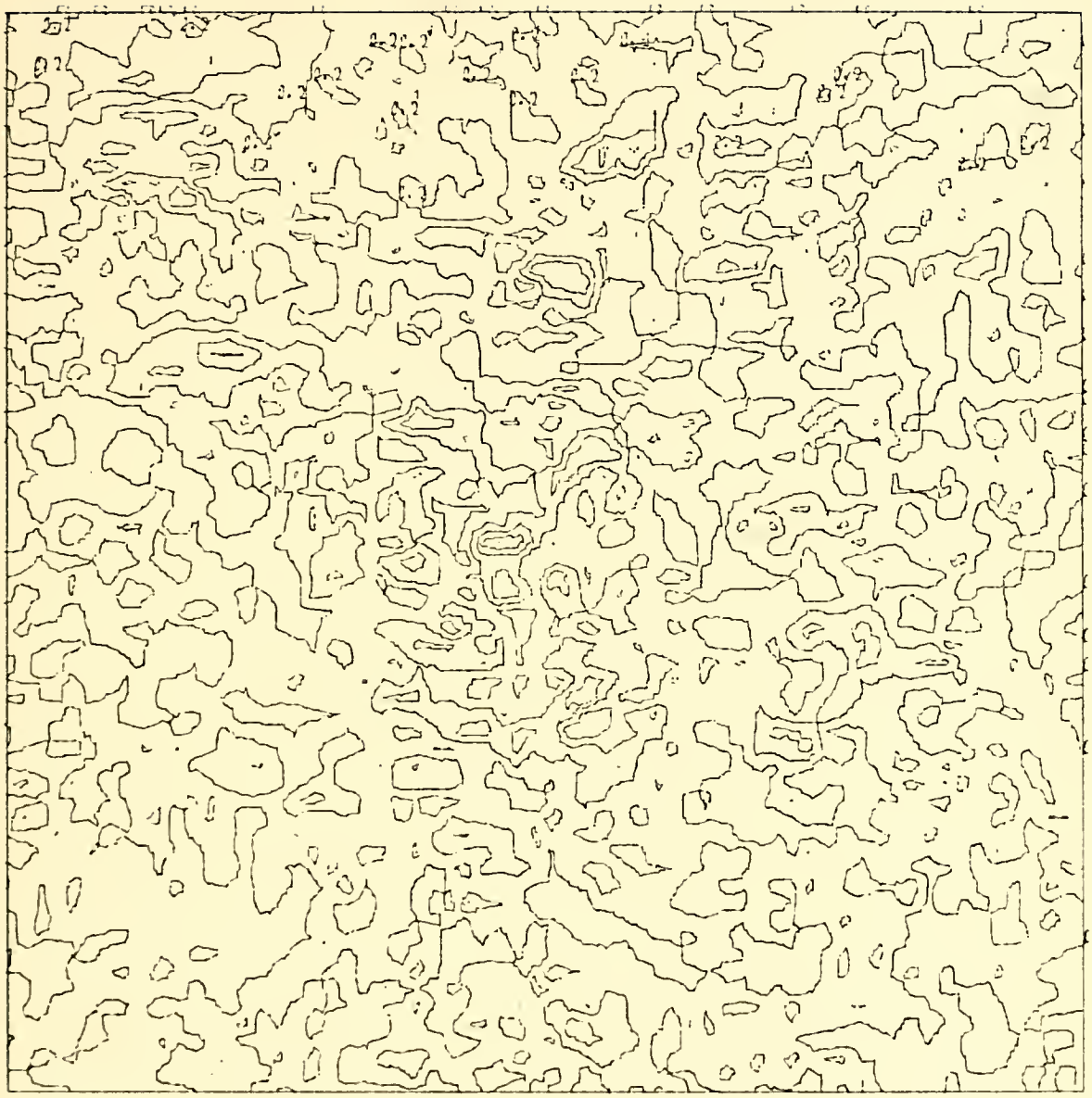


Figure 6-7
Amplitude Contour Plot of Spurious Output
(Case e)

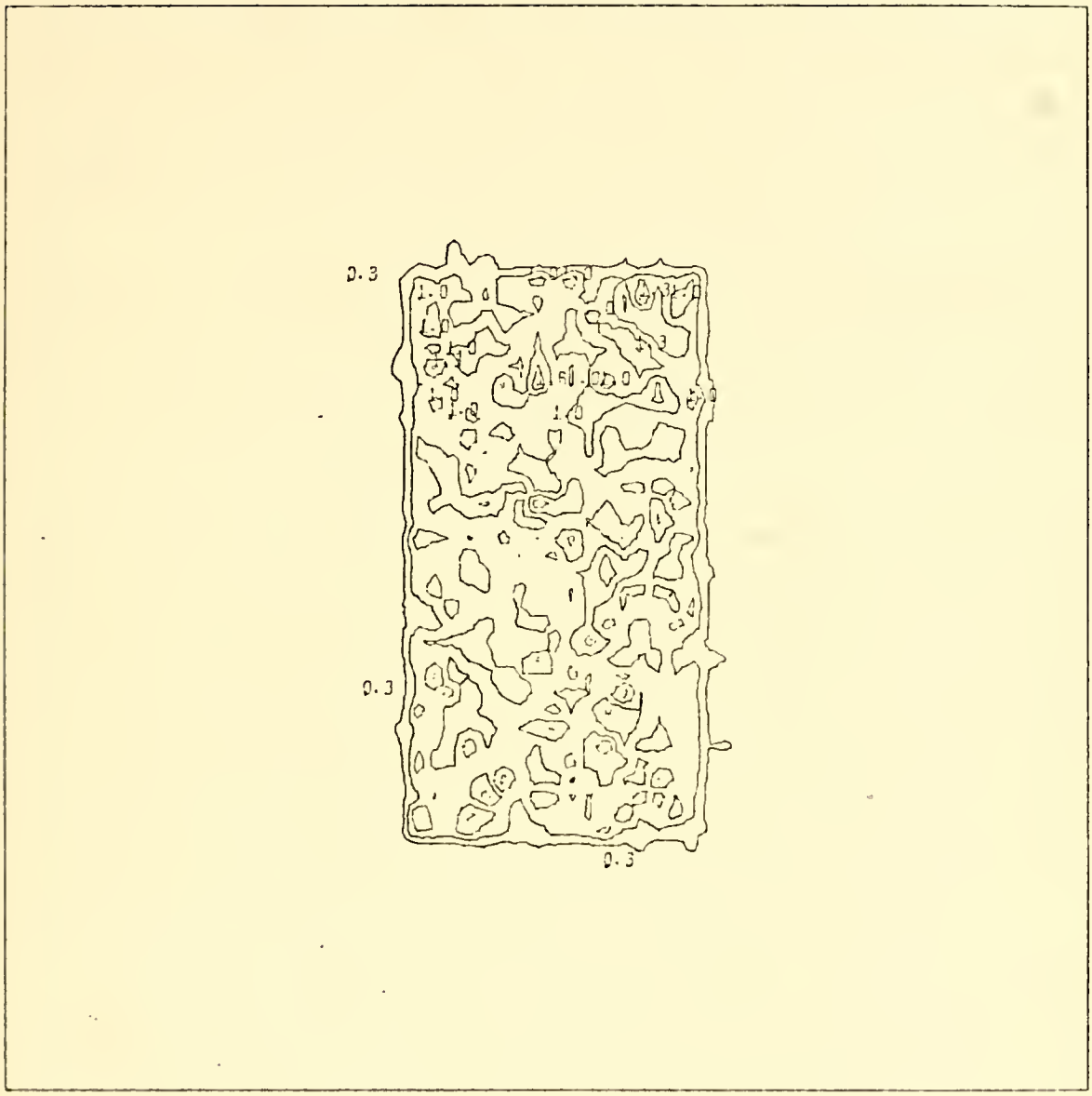


Figure 6-8
Amplitude Contour Plot of Reconstructed Random Phase Object with Random Amplitude Variation. Background Assumed to have Random Phase, Random Amplitude (Case f)

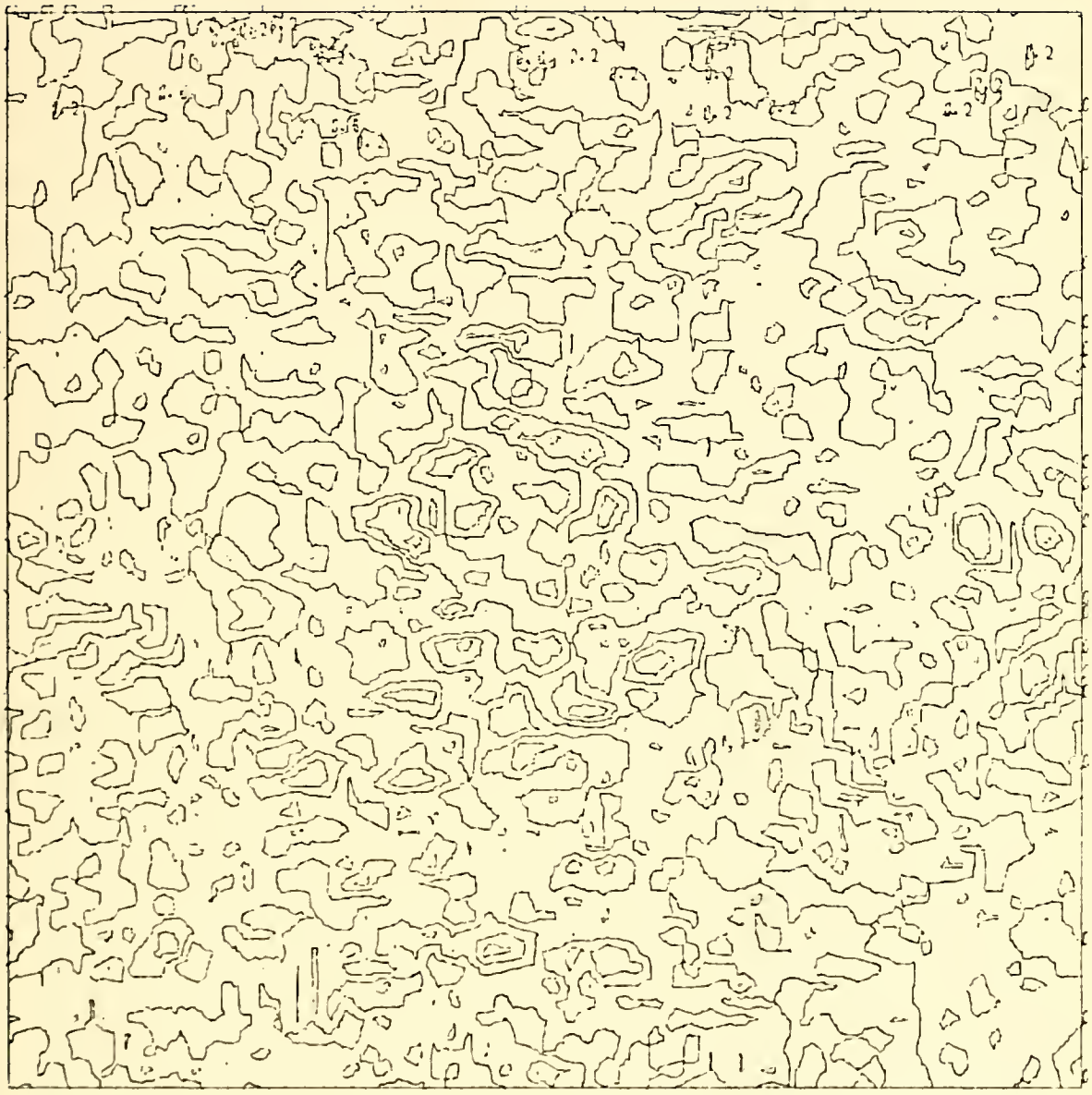


Figure 6-9
Amplitude Contour Plot of Spurious Output
(Case f)

and exists only in a single plane with rather high probability. Naturally each output would have to be scrutinized by an operator such that spurious output which could appear in addition to the actual object reconstruction would be discarded. The proposed edge detection routine is not infallible, but it performs its operation in rather short time which was the main reason for setting it up in the existing manner.

The next set of tests performed consisted of trying to reconstruct input objects existing in two different parallel planes. As in backward propagation the real image is reconstructed, the first difficulty is immediately obvious: the diffraction pattern converges to that existent in the plane of the object on backward propagation, but the sound waves emanating from the object do not stop on the reconstructed object but are propagated farther back on the next propagation step. This poses no great problem for an object that exists only in a single plane parallel to the observation plane. But consider an object consisting of two extended sound sources in two different parallel planes. The sound from the source farther from the observation plane is propagated into the plane of the nearer source there to be absorbed, phase shifted and/or diffracted, as the case may be. The sound field that then exists in the plane of the front source is propagated to the observation plane. On backward propagation the field converges to that previously existent in the front plane. So far, no fault can be found with the described situation. But on further backward propagation (the second source should also be reconstructed) the sound field introduced by the front source is also propagated back into the plane of the farther object, there to give rise to a disturbance which did not exist in actuality. In fact

this is not too different from the case of the optical hologram and the eye because the eye converts the virtual image to a real image on the retina. By focusing on a part of the image that is in a plane farther away, that part which is nearer is still seen, in a blurred fashion, in that same plane, but the brain rules out the hypothesis of a composite sharp and blurred object: The unwanted information is largely ignored. How recognition is accomplished, i.e., why certain object hypotheses are accepted, other not, remains unknown to date, even though the necessity of experience is generally assumed (apart from the obvious advantage of stereoscopic vision if the distance is not too large). Huang [Ref. 25] and Ichioka, Izumi and Suzuki [Ref. 26] show a method of generating diffraction patterns due to such objects. The disturbance due to the back object is propagated into the plane of the front object. There that part of the resulting disturbance that corresponds to the extension of the front object is replaced by the disturbance due to the front object and the resulting composite disturbance is propagated into the observation region. This scheme was used for generating the necessary diffraction patterns in this investigation.

Case g: The simplest object of this category consisted of two 16λ by 8λ rectangles, assumed to be specularly emitting (i.e., all sample points were in phase). These objects were located in planes which were 15λ behind one another. One rectangle was rotated 90° with respect to the other around the propagation axis and displaced in such a manner that the projections of both rectangles along the axis of propagation overlapped by only a small amount (about 0.1 of the area of one rectangle). The actual object configuration can very well be

visualized by studying Figs. 6-10 and 6-11, which are the results of the computer reconstruction. The diffraction pattern was observed 100.4 wavelengths in front of the front rectangle. The volume of interest was taken to extend from 85λ to 130λ from the diffraction pattern. Therefore the number of steps in the investigation had to be increased to 30. The reconstructions obtained were located at 100.25λ and 115.25λ from the observation plane, i.e., 0.15λ in front of the first and second rectangle respectively. The total computation time was about 8.5 min.

Case h: The same geometry as for case g was assumed but here both rectangles had a random phase distribution uniformly distributed from zero to 2π. The volume of interest and number of steps for the investigation were not changed from case g, but the observation plane was located 99.63λ in front of the front rectangle. The computer obtained 8 object locations (at 93.53λ, 99.69λ, 103.85λ, 106.95λ, 116.26λ, 120.91λ, 123.79λ and 126.12λ), a clear indication that this reconstruction was not quite successful. The array corresponding to the actual location of the front rectangle, Fig. 6-12, was only third in overall edge-measure. An output which is typical for the spurious objects computed is shown in Fig. 6-13. The amplitude distribution of Fig. 6-13 was the fourth highest in edgemeasure; it is computed at a distance of 7.32λ behind the front object. Figure 6-14 shows the output which is the lowest of all eight computed in edgemeasure. Its generating diffraction pattern is located 1.6λ behind the second rectangle. It is not evident that this output is strongly related to the second rectangle (a comparison with Fig. 6-11 is helpful). The existence of the front rectangle can be seen quite clearly in Fig. 6-12 (compare with Fig. 6-10).

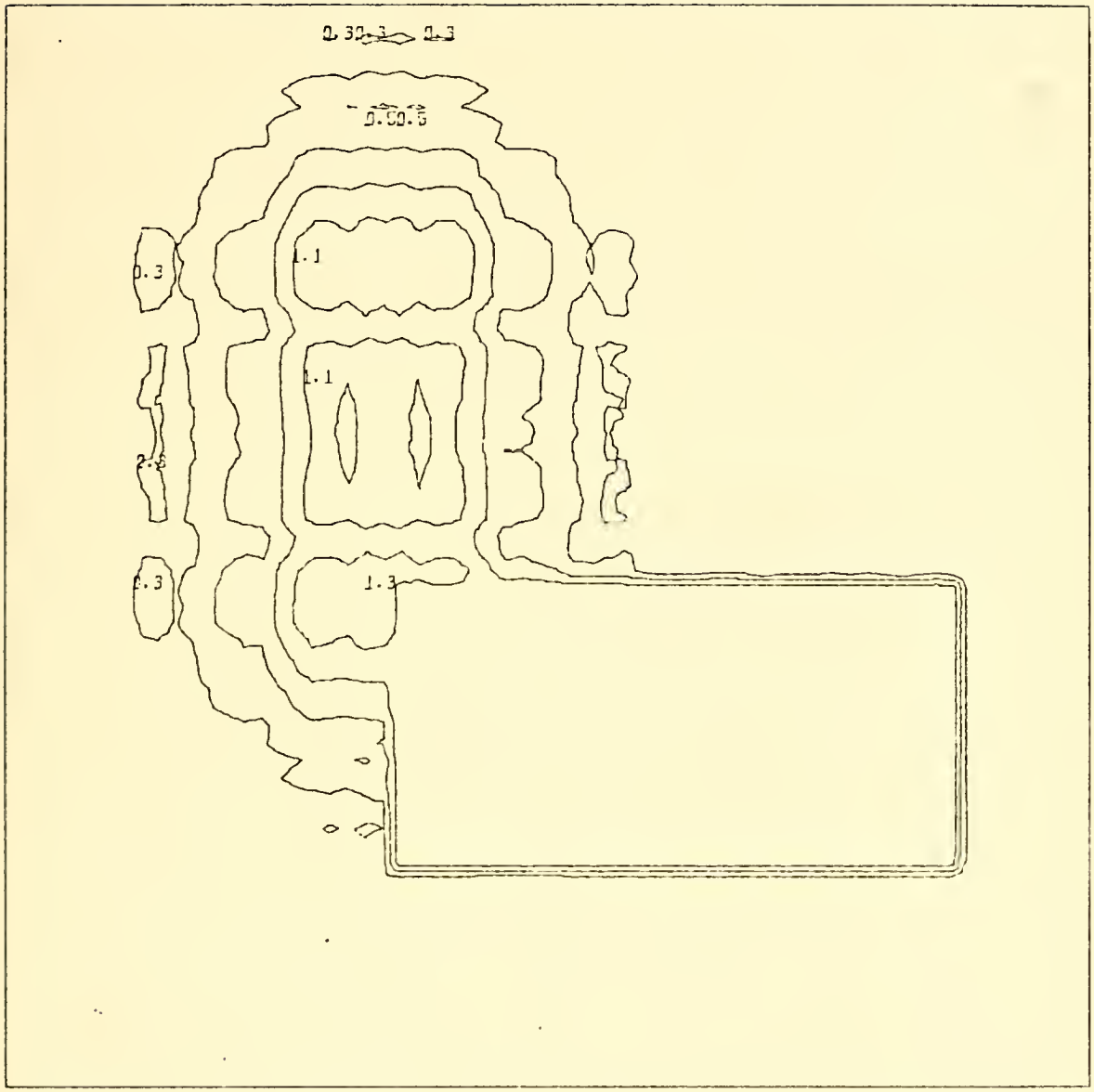


Figure 6-10
Amplitude Contour Plot of Reconstructed Diffraction Pattern in Front Plane of Double Specular Object (Case g)

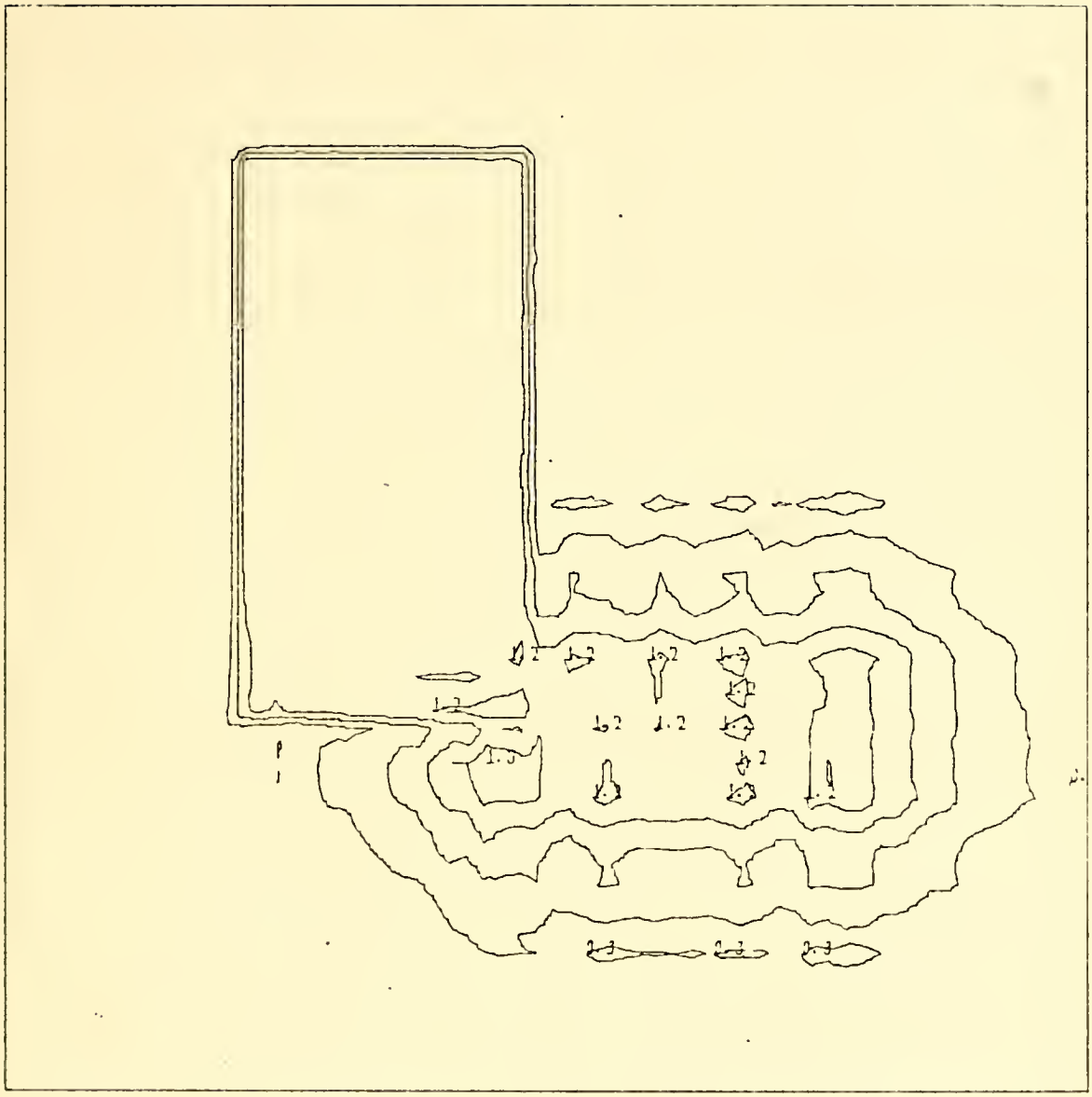


Figure 6-11
Amplitude Contour Plot of Reconstructed Diffraction Pattern in Back Plane of Double Specular Object
(Case g)

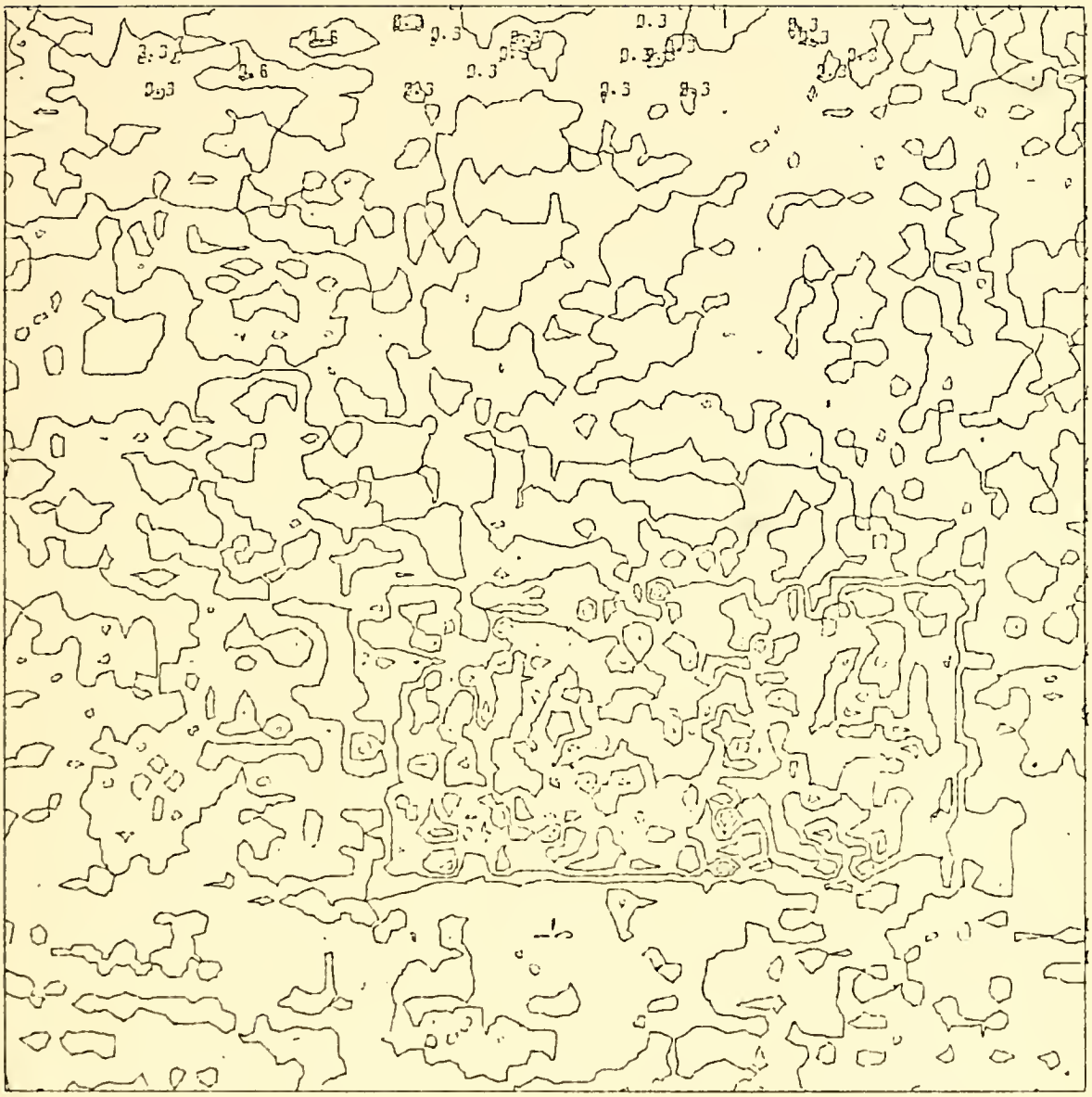


Figure 6-12
Amplitude Contour Plot of Reconstructed Diffraction Pattern in Front Plane of Double Random Phase Object (Case h)

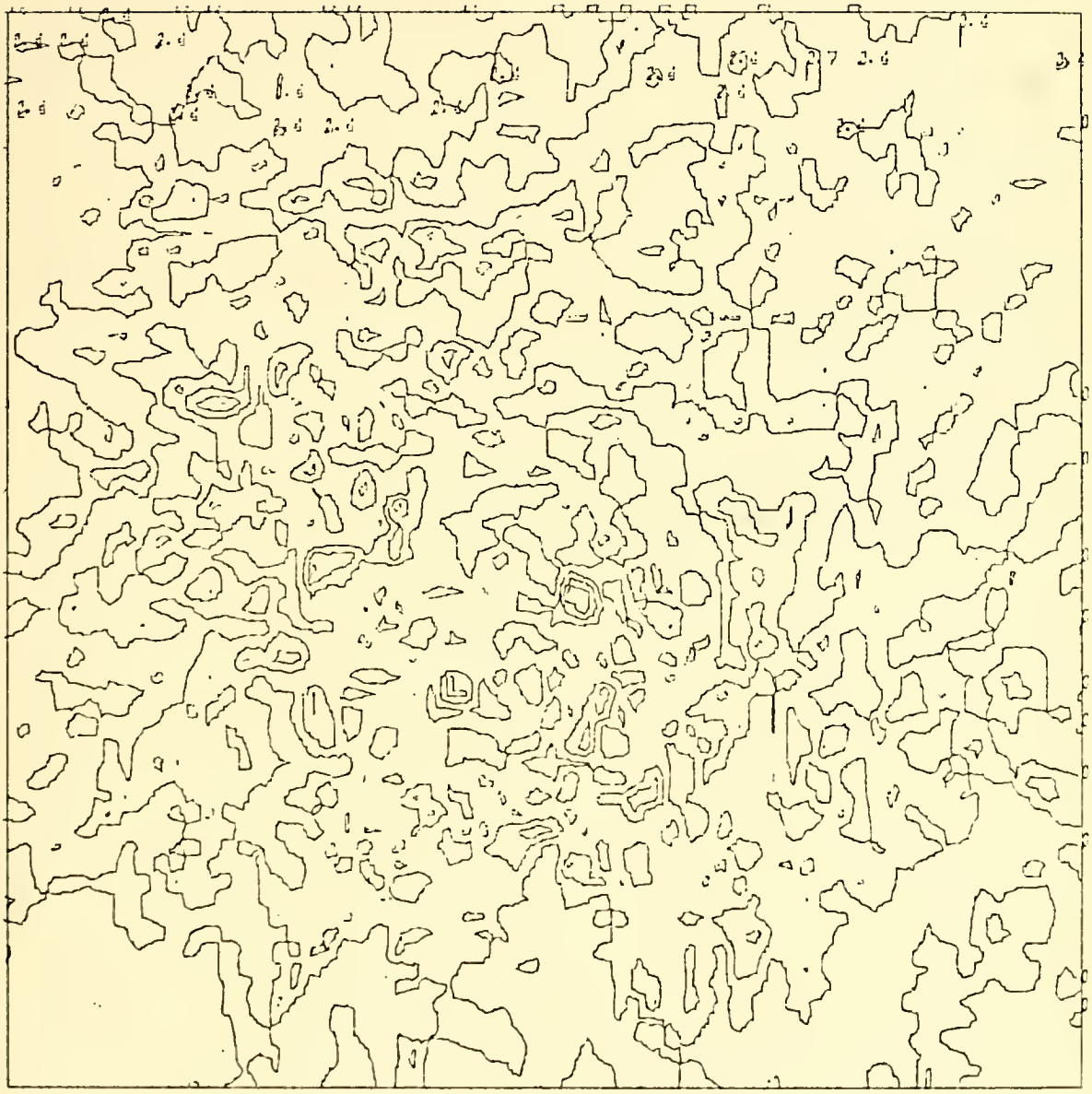


Figure 6-13
Amplitude Contour Plot of Spurious Output
(Case h)

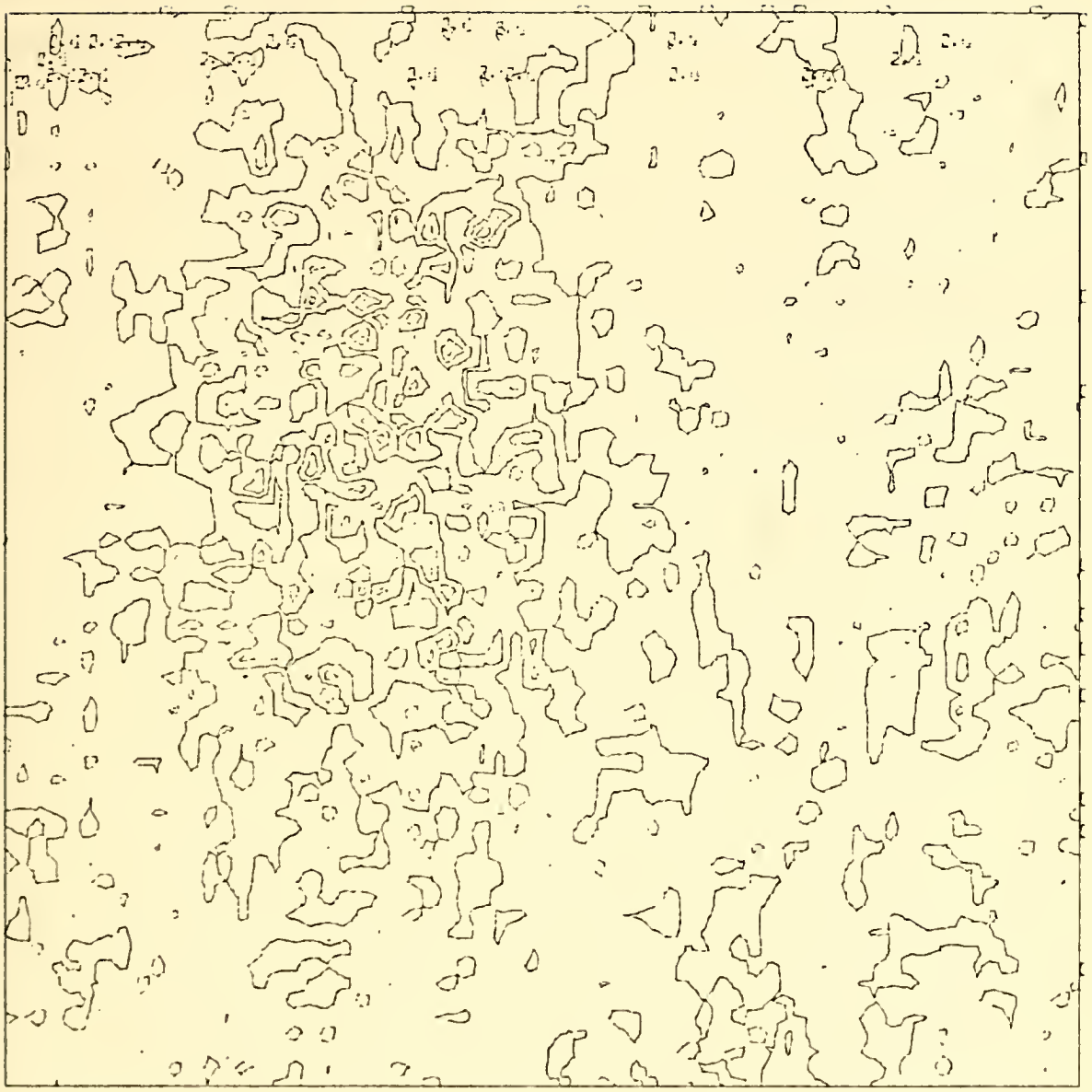


Figure 6-14
Amplitude Contour Plot of Reconstructed Diffraction Pattern
Near Back Plane of Double Random Phase Object
(Case h)

On studying Fig. 6-12 and 6-13 it is not surprising that the computer cannot clearly recognize the output of Fig. 6-12 as the actual object. The deterioration of the output due to the loss of evanescent waves is very apparent in a situation as complicated as the one treated in this case.

Case i: In order to investigate the effects of varying phases between both rectangular objects and those of larger amounts of overlap, two rectangles of unity emission amplitude were simulated. These were assumed to be emitting in phase, but their separation in the direction of propagation was taken to be an odd number of half wavelengths, namely 14.5λ. Both rectangles were of size 16λ by 8λ, one was rotated 90° versus the other around the propagation direction and both were centered in the input array. The overlap thus became one half the area of one rectangle. The diffraction pattern was observed at 100.4λ from the front object. Investigation took place in 30 steps between 85λ and 130λ from the observation plane. The computer generated two possible object locations at 100.75λ (Fig. 6-15) and at 113.5λ (Fig. 6-16) from the diffraction pattern. This took 6.2 min. of computation time. The front rectangle being reconstructed 0.35λ behind its actual location shows up very well. Due to the destructive interference between front and back object the location of the second rectangle could not be found exactly but the shape of the second object is quite apparent in Fig. 6-16. Removing the front object from the diffraction pattern, once it is recognized, can be helpful for finding the location of the second object in a case like this (see case j, below).

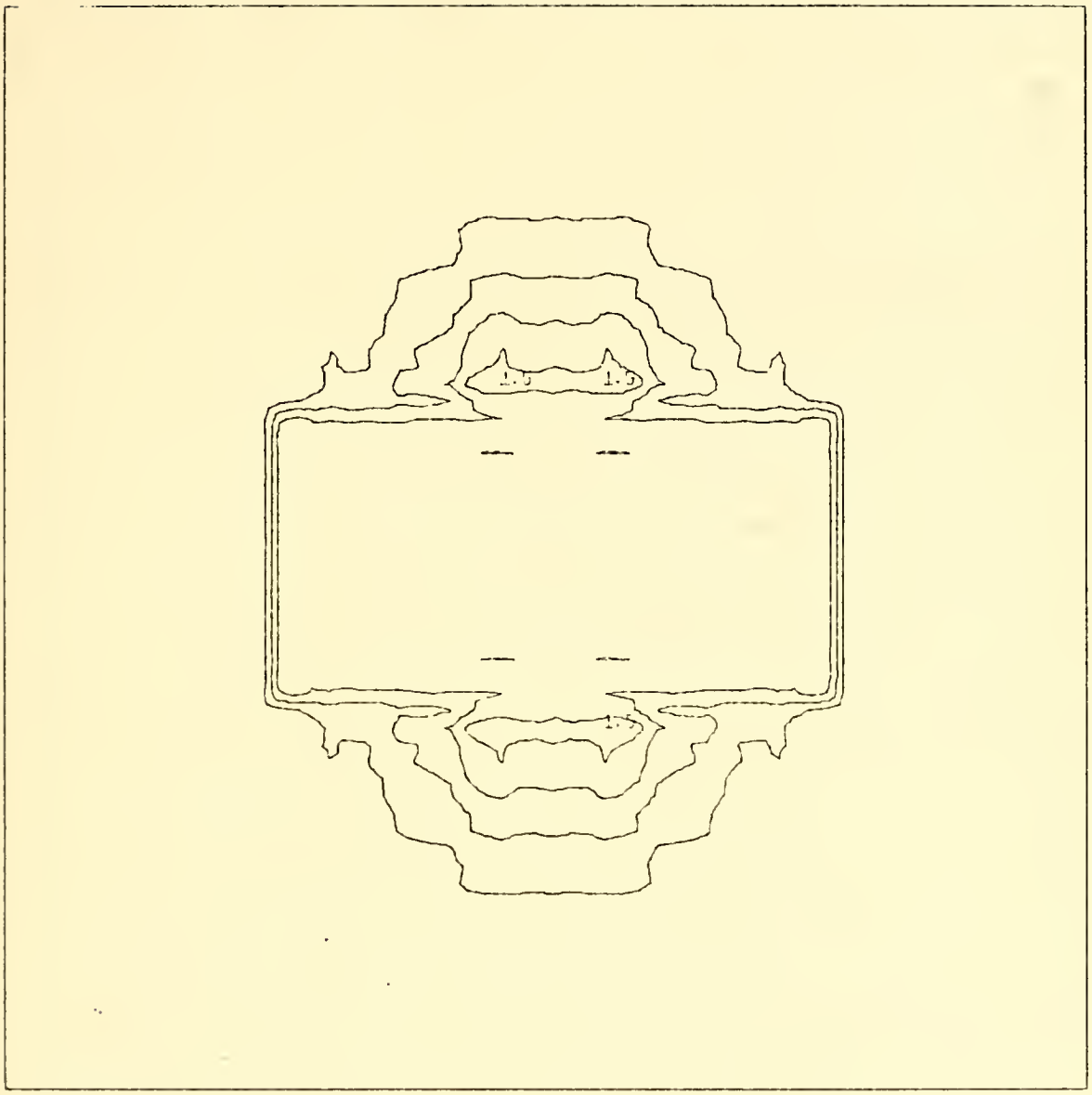


Figure 6-15
Amplitude Contour Plot of Reconstructed Diffraction
Pattern in Front Plane of Double Specular Object
(Case i)

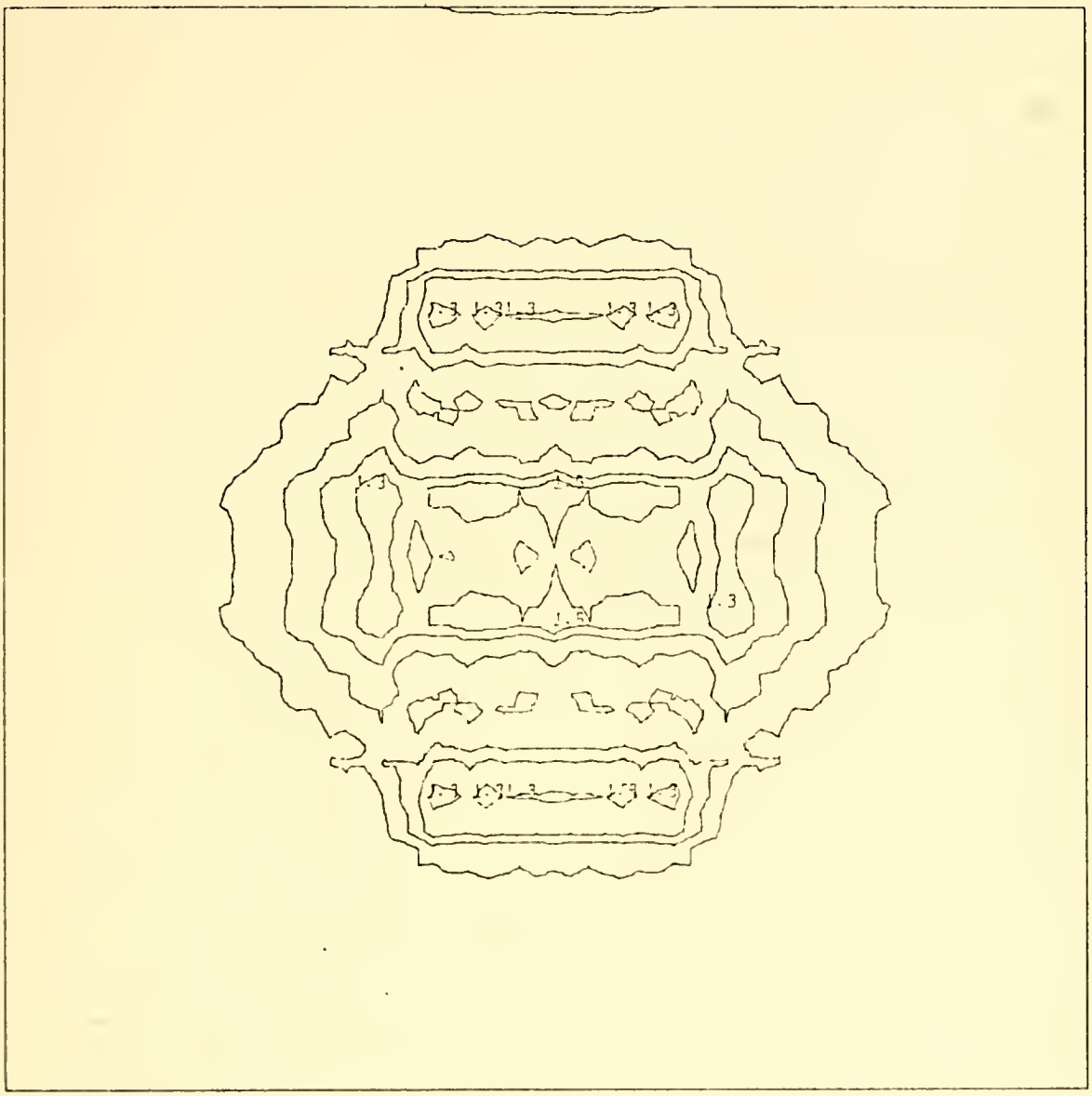


Figure 6-16
Amplitude Contour Plot of Reconstructed Diffraction
Pattern Near Back Plane of Double Specular Object
(Case i)

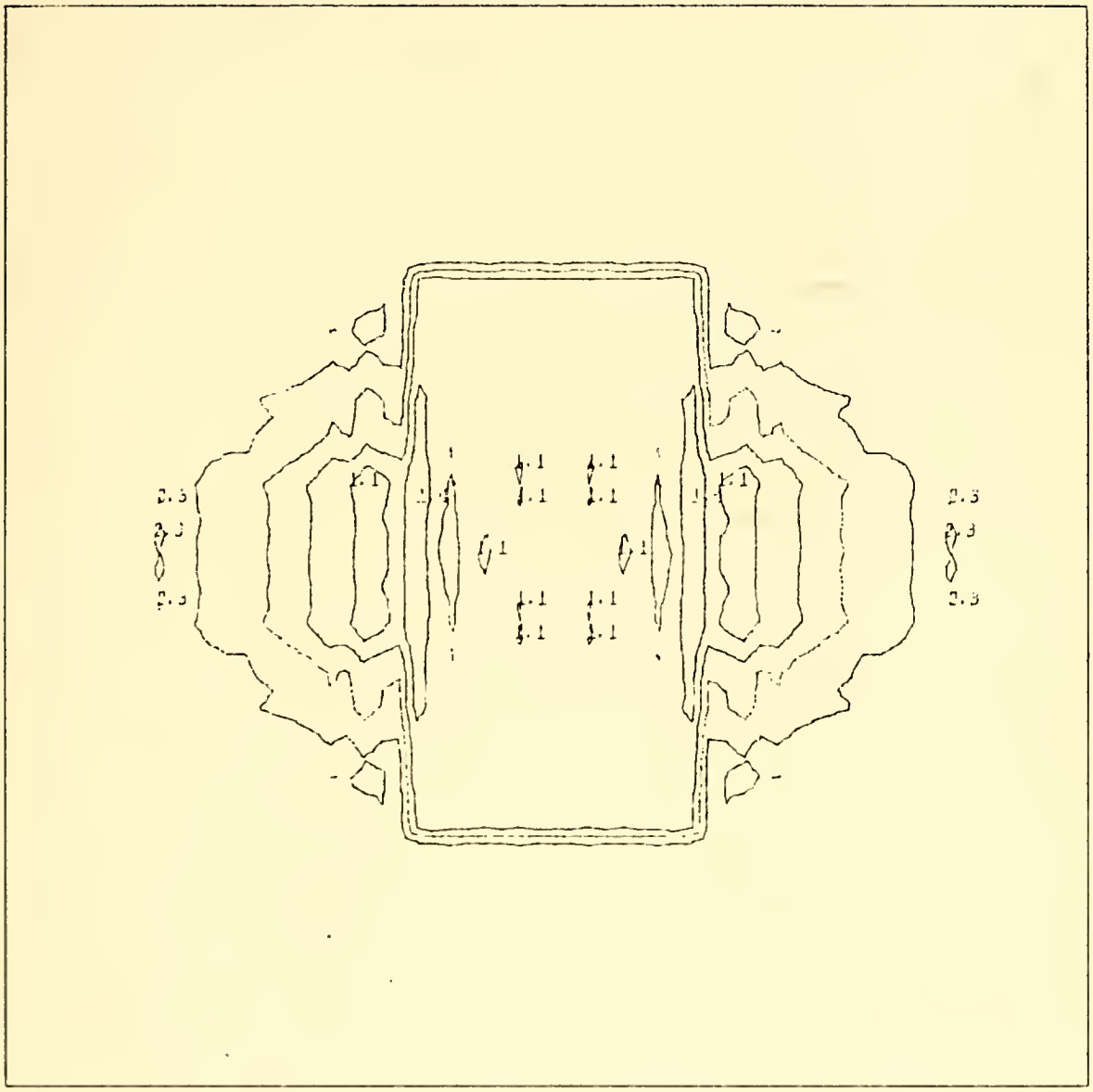


Figure 6-17
Amplitude Contour Plot of Reconstructed Diffraction
Pattern in Back Plane of Double Specular Object
(Case j)

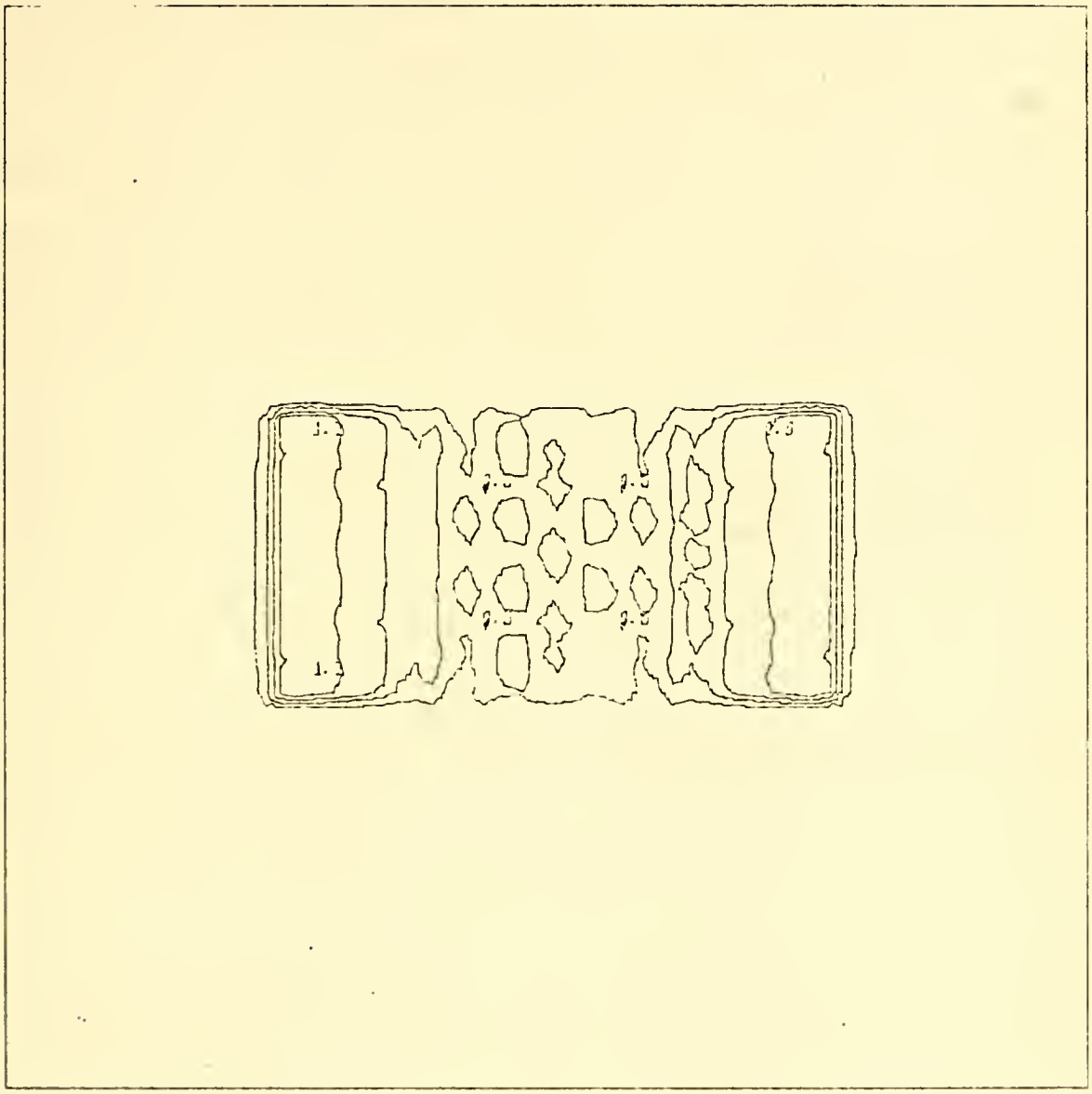


Figure 6-18
Amplitude Contour Plot of Diffraction Pattern in Front Plant of Double Specular Object, Obtained After Removing Back Object. (Case j)

Case j: A final test using computer generated diffraction patterns was performed with an input geometry as in case i, with the difference that the distance between rectangles was set to 15.0λ and both were taken to be specularly emitting in phase. Distances to the observation plane and from these to the first and last plane of interest remained unchanged as did the number of investigation steps. The computer recovered the location only of the back rectangle at 115.25λ from the observation plane (see Fig. 6-17). This was accomplished using 7.5 min of computer time. It is surprising that the plane of the nearer rectangle was not found. In a case like this it may be helpful to remove the clearly recognizable object from the output by setting a portion of the reconstructed array to zero and to use the remaining diffraction pattern to try to recover the rest of the composite object. This was done here: The clearly indicated back rectangle was zeroed out and the remaining disturbance propagated forward 25λ. A new volume of interest was defined to extend from 5λ in front of the new observation plane (as the diffraction pattern there was not displayed, it was not on actual observation plane) to 40λ, which was investigated in 25 steps. This took 6.3 min. of computation. The computer recovered the front object 0.15λ in front of its actual location. The resulting diffraction pattern there is shown in Fig. 6-18. The fact that the center portion of the rectangle is distorted is attributable to the removal of the center portion in the back object plane.

The obtained results from this series of tests imply that automatic reconstruction of objects from their diffraction pattern is feasible. Of course for this to be accomplished not only the objects but also their reconstructions must meet the criterion which is applied to make

the object - no object decision. This is the reason why in the above tests certain objects could not be recognized by the computer. Neglecting the evanescent wave destroys a larger or smaller amount of edge information, depending on the object. This has to be kept in mind when judging the merits of the proposed scheme.

2. Testing the Reconstruction Scheme with Experimental Data

As the input for the final test a set of data from an actual diffraction pattern was used. This diffraction pattern had been observed by C. Griggs (see [Ref. 27]) for a different purpose. The object giving rise to the diffraction pattern was a square transducer of 2 inch side length. The active area was 1.75" by 1.75". The pattern was sampled with a linear transducer approximately 73λ away from the transducer by use of a scanning system. These data suffer from a few minor details (see [Ref. 27]), the most important of these being that only one half of the diffraction pattern was observed to conserve time. It was therefore necessary to fill the rest of the input array by a mirror image of the actually observed data. This explains the unnatural symmetry that can be observed in inputand output (Fig 6-19 through 6-24). Another drawback of this set of data, which has to be taken into account, is the fact that the difference between adjacent sampling points is 0.9λ, i.e., the sampling was performed at a rate very much below the Nyquist rate; aliasing effects are therefore to be expected, mostly at high spatial frequencies which means that distortion of edges by enhancement or deemphasis is probable. But the most important reservation against these data is not due to the manner in which they were recorded but rather to the object which generated this diffraction pattern: As the transducer had to be clamped at its boundary to get a watertight

seal it is quite possible that it does not constitute a good object for the proposed reconstruction scheme. One cannot assume that its amplitude dropped abruptly from a rather high value to a very low one, but would think that his transition would be more gradual.

Nevertheless the test was performed with these data as they were readily available.

Amplitude and phase of the diffraction pattern after reading in and reflecting about the center are shown in Figs. 6-19 and 6-20.

The volume of interest was defined to be between 60λ and 80λ from the observed diffraction pattern. This valume was investigated in 8 steps. This investigation took about 4.4 min. Two possible object locations were found at 74.66 λ (edge-measure 29.927) and at 67.68λ (edge-measure 26.438) from the observation plane. At first sight there is nothing in the amplitude display (Fig. 6-21) from the location with the highest edge-measure, to prefer it as recovered object to that of Fig. 6-22, which shows the amplitude distribution at 67.68λ from the observation plane (or for that matter from the input amplitude distribution), except for the fact that it appears close to the given nominal distance of 73λ. In fact the amplitude pattern for such an object does not change very much in the volume of interest. The plausibility that the correct location was assigned the highest edge-measure becomes quite high when the phase distribution in Fig. 6-23 (at74.66λ) and in Fig. 6-24 (at 67.68λ) are studied. It can be seen that in Fig. 6-23 all points on the face of the transducer have the same phase (or very nearly so), whereas in Fig. 6-24 a distinct variation can be observed.

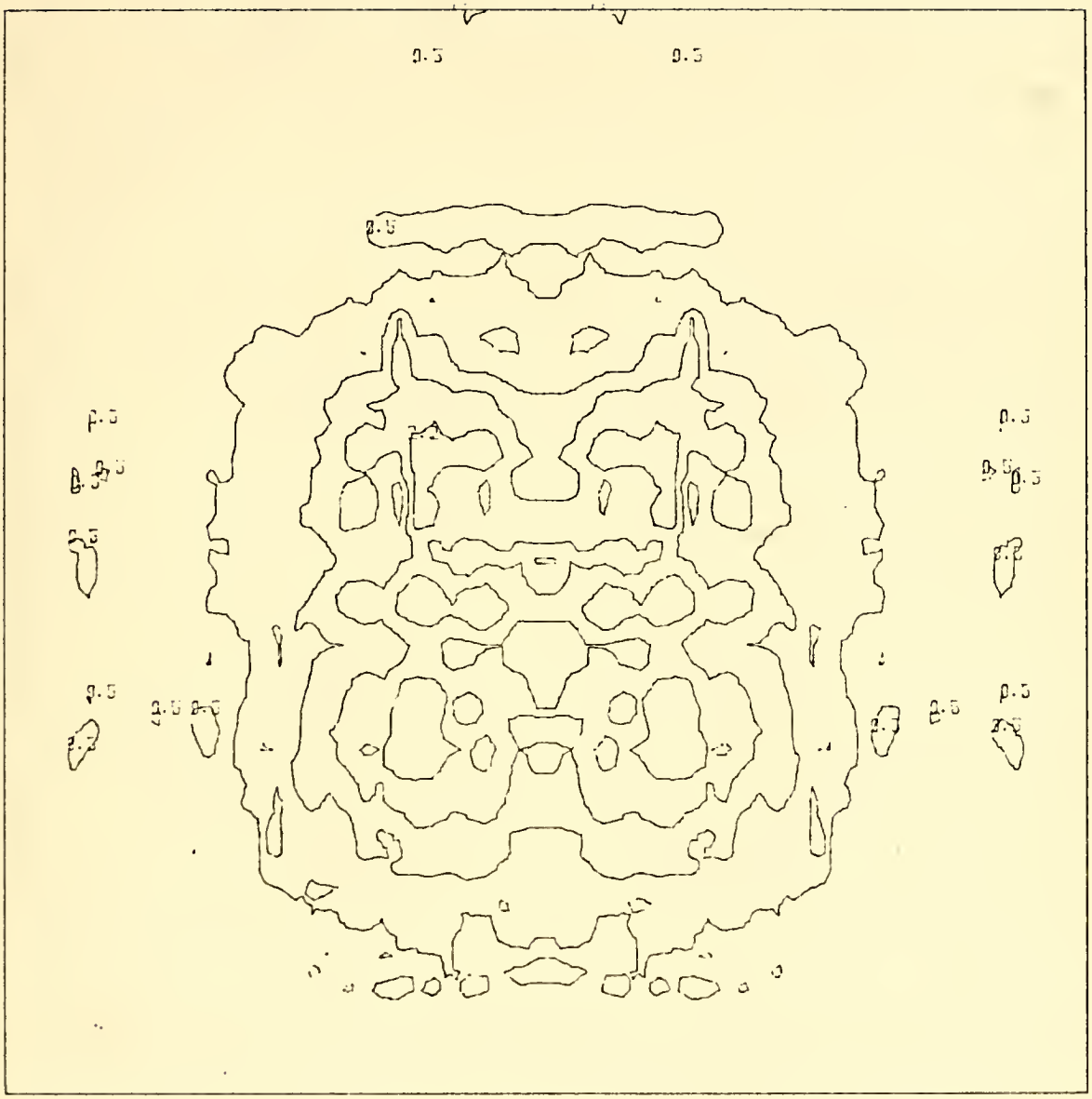


Figure 6-19
Amplitude Contour Plot of Experimental
Data; Input to Reconstruction Routine

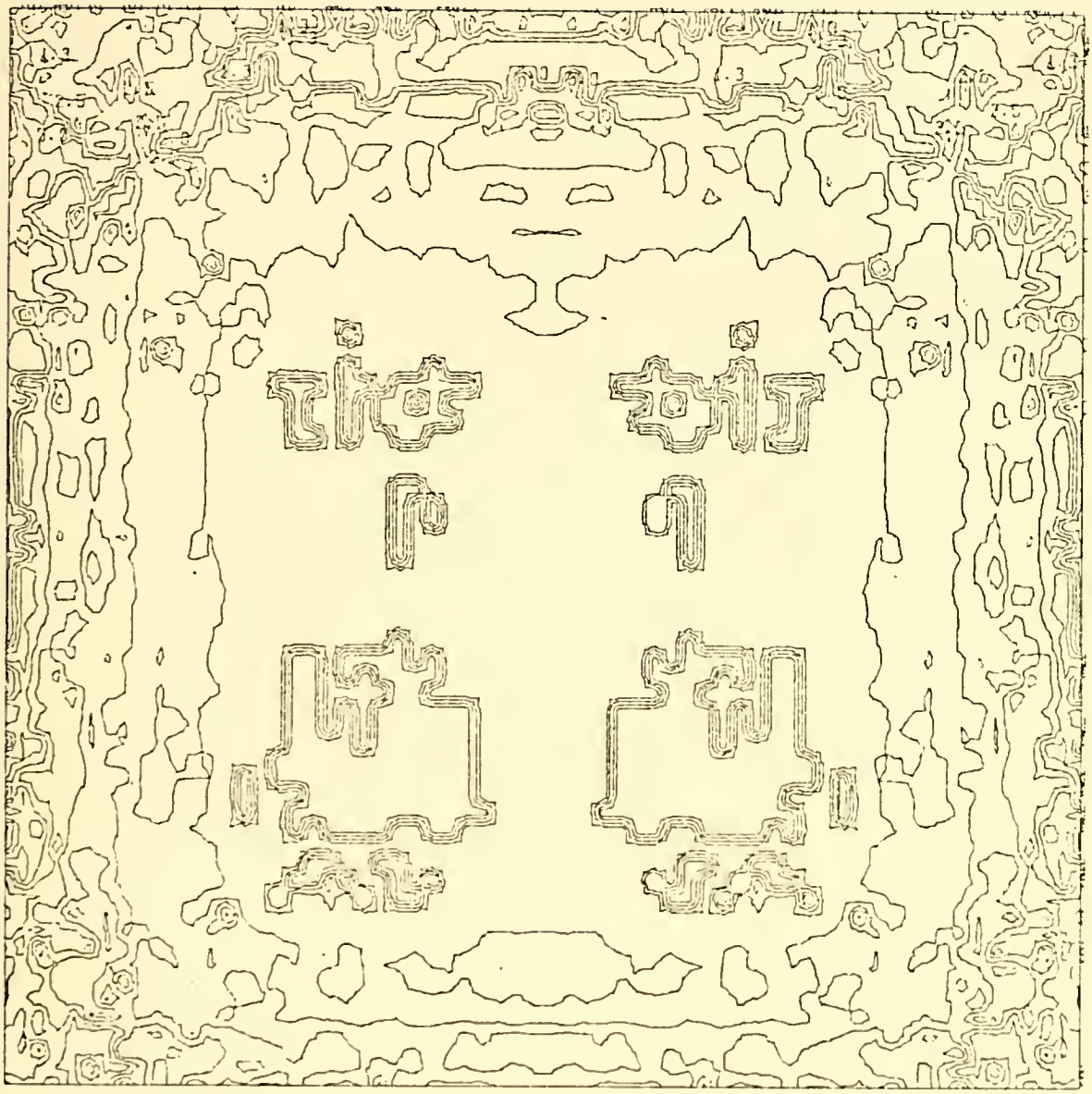


Figure 6-20
Phase Contour Plot of Experimental Data;
Input to Reconstruction Routine

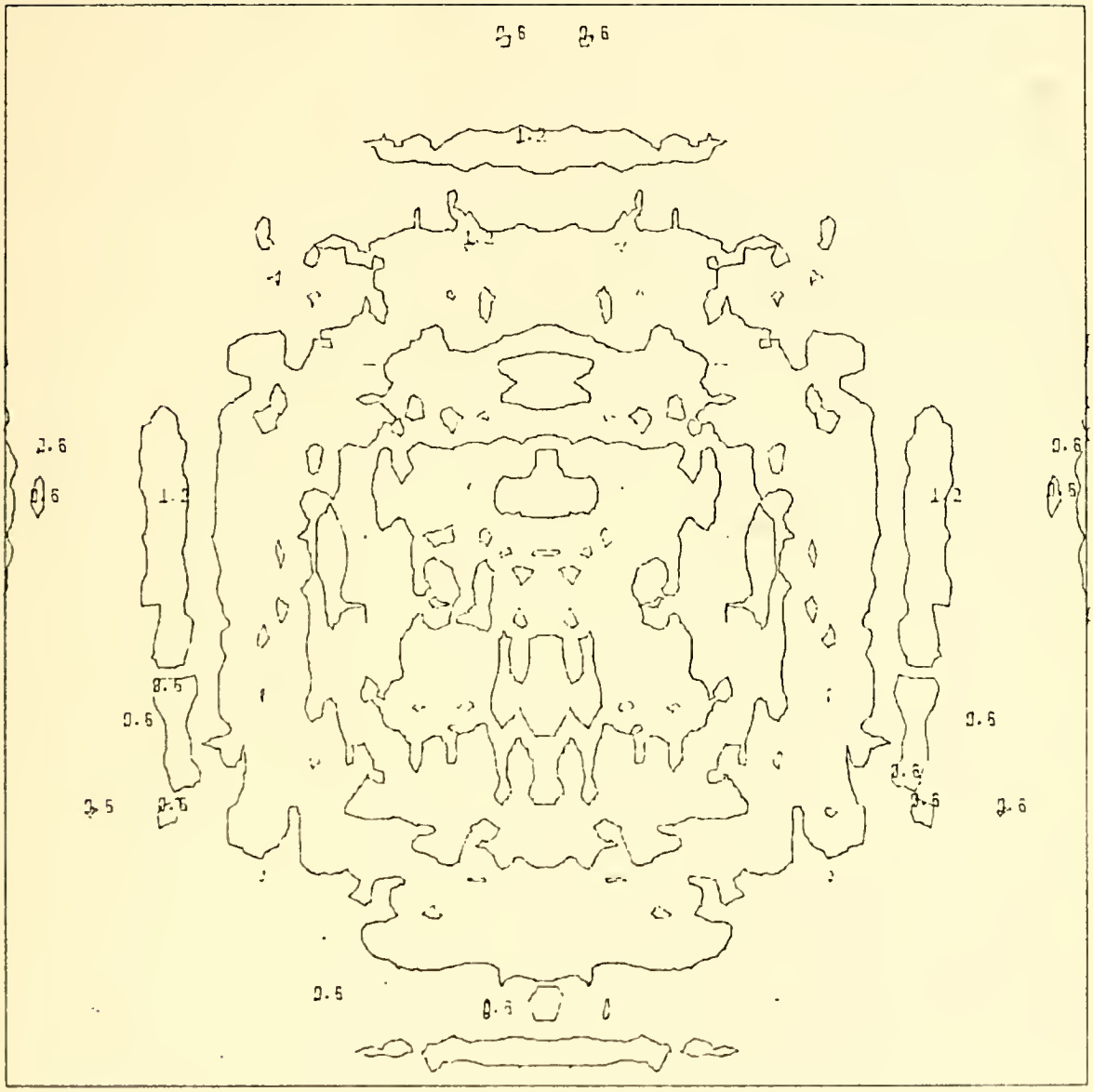


Figure 6-21
Amplitude Contour Plot of Reconstructed
Diffraction Pattern with Highest Edgemeasure

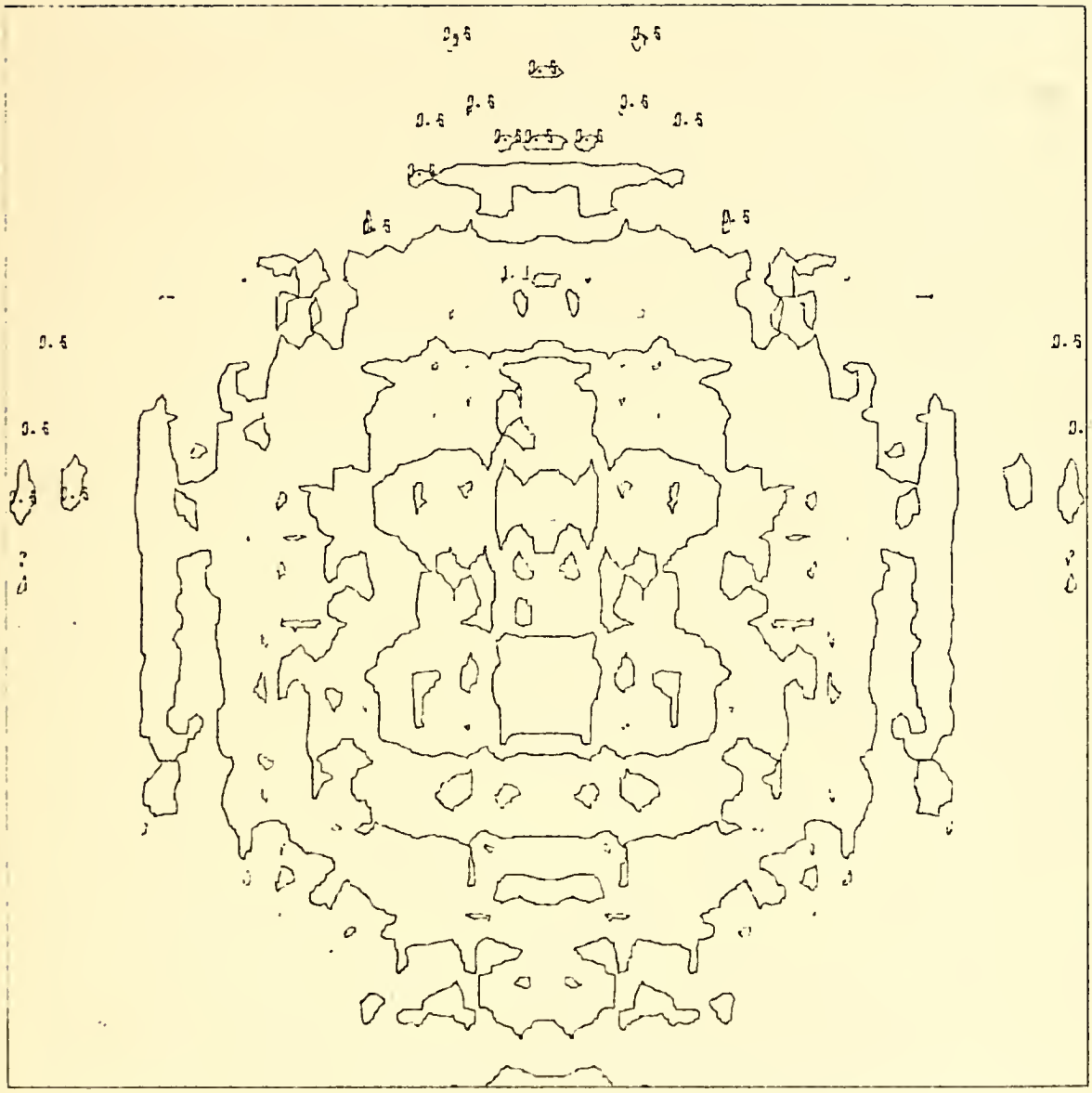


Figure 6-22
Amplitude Contour Plot of Second Obtained Reconstruction

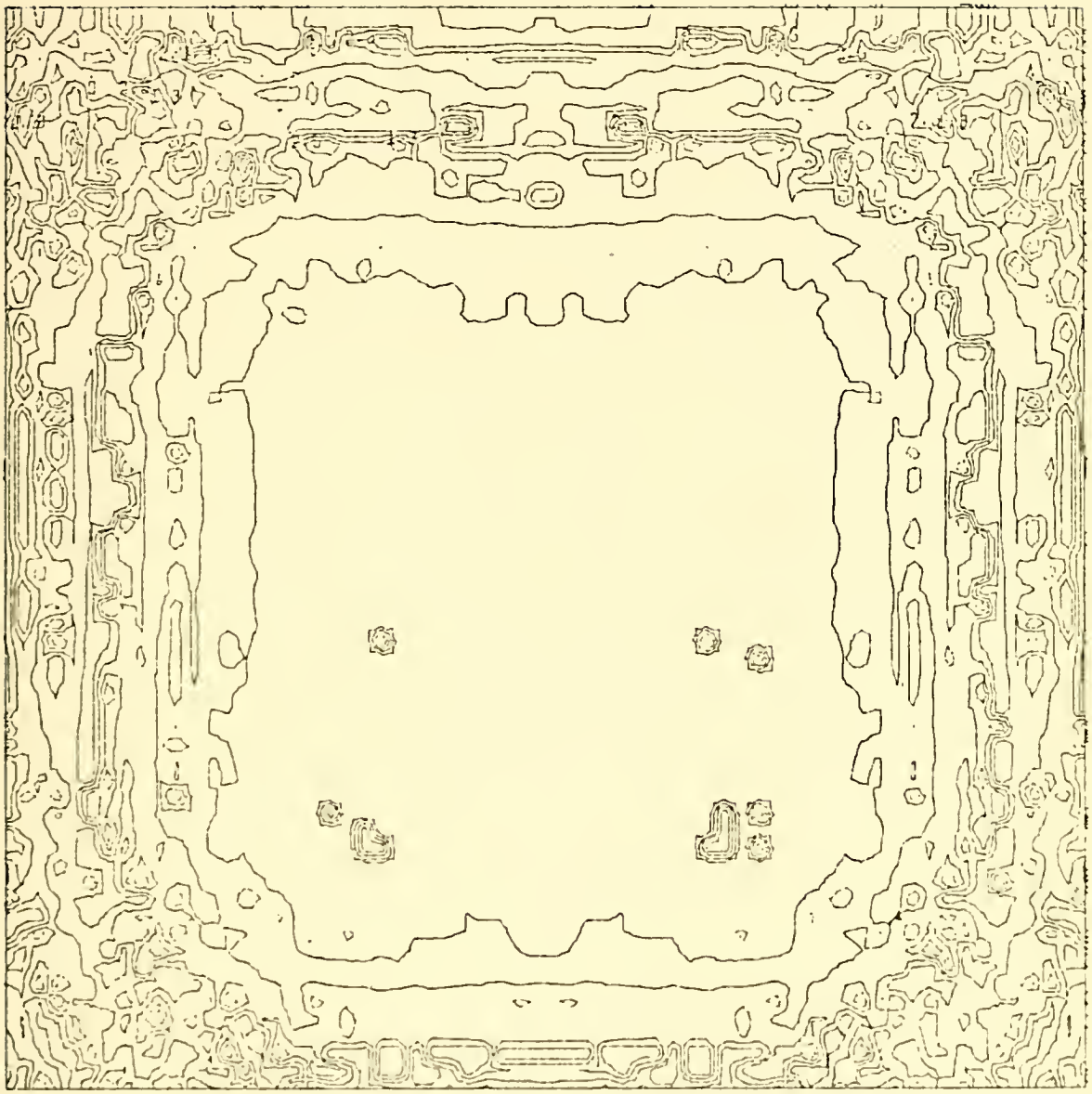


Figure 6-23
Phase Contour Plot of Reconstructed Diffraction Pattern with Highest Edgemeasure

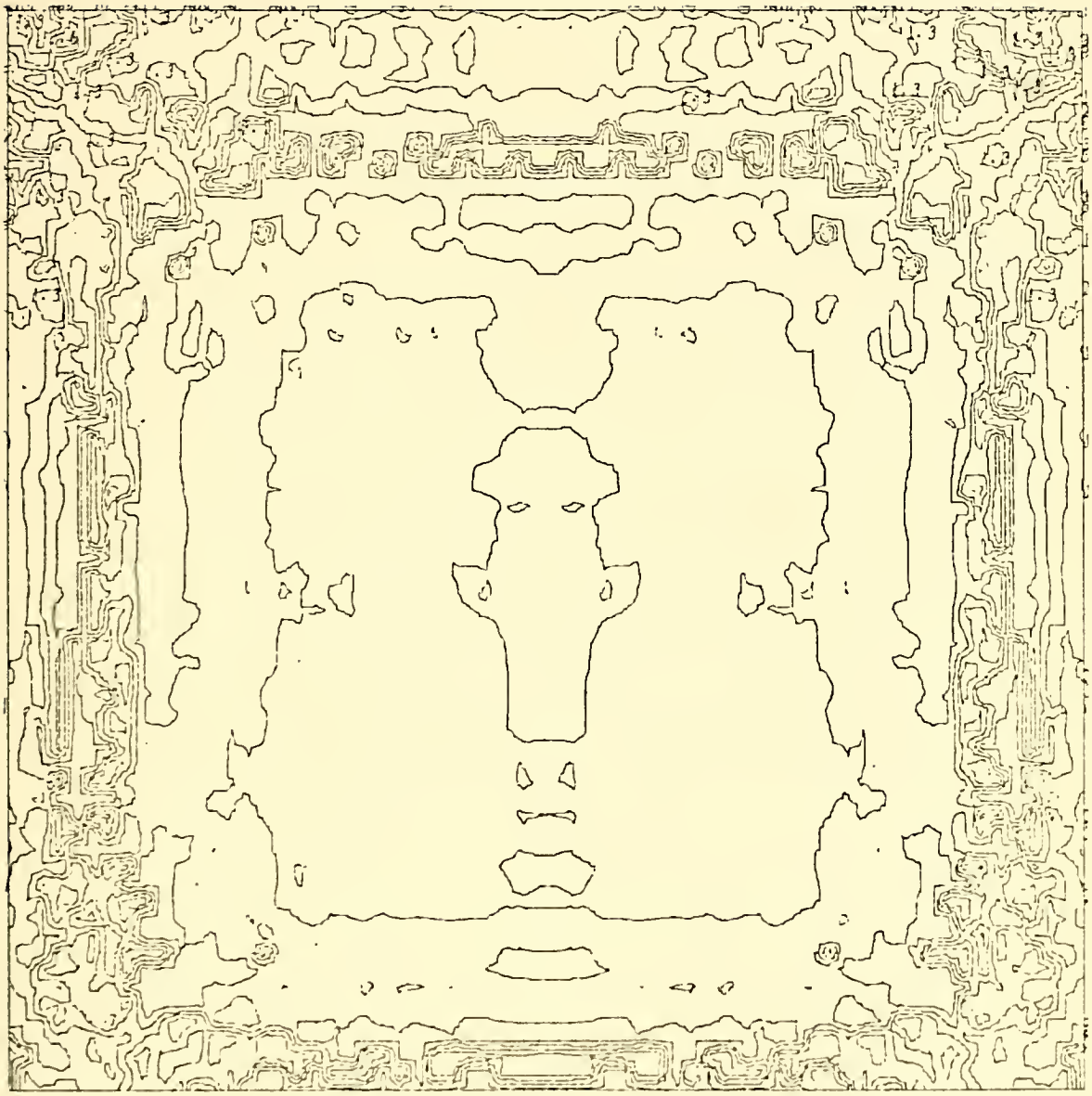


Figure 6-24
Phase Contour Plot of Second Obtained Reconstruction

The result of this test seems rather inconclusive, but this is due more to the nature of data available rather than the proposed reconstruction scheme.

VII. CONCLUSION

Computer reconstruction of objects from their diffraction patterns has certain advantages versus optical reconstruction, notably the avoidance of longitudinal distortion due to the difference in acoustical generating and optical reconstruction wavelength. The main drawback of computer reconstruction is the need of knowing exactly the distance of the object from its diffraction pattern. For an unknown object distance one would have to use many small steps of backward propagation, display the diffraction pattern at each location and have an operator make the decision as to the object location according to his preconceived ideas of what an object looks like. As the manipulation of data into form useable for output takes up considerable amounts of computer time it will be helpful if only a small number of possible objects is displayed. Automatic object recognition by computer can be an aid here.

Automatic object detection by a computer routine can only succeed for objects belonging to that class for which the routine was written, since it is impossible to find a criterion which will reconstruct all objects. A particularly important class of objects is that which includes objects with sharp boundaries and edges, i.e., rapid changes in amplitude separating regions of high emission from those with much lower amplitude. Therefore the reconstruction scheme presented in this report was tailored to detect edges and boundaries, without using great amounts of computer time and core memory. A technique as simple as the one given here can be improved in various ways; in doing this it should always be remembered that any improvement obtained at the cost

of much increased computation time will reduce the usefulness of the whole automatic reconstruction idea. The present edge detection scheme could readily be adopted for other classes of objects, e.g., those characterized by a sharp phase change or by a difference in texture. Once different values have been assigned to different types of texture it is possible to operate with the same edge detection routine on the thus interpreted array.

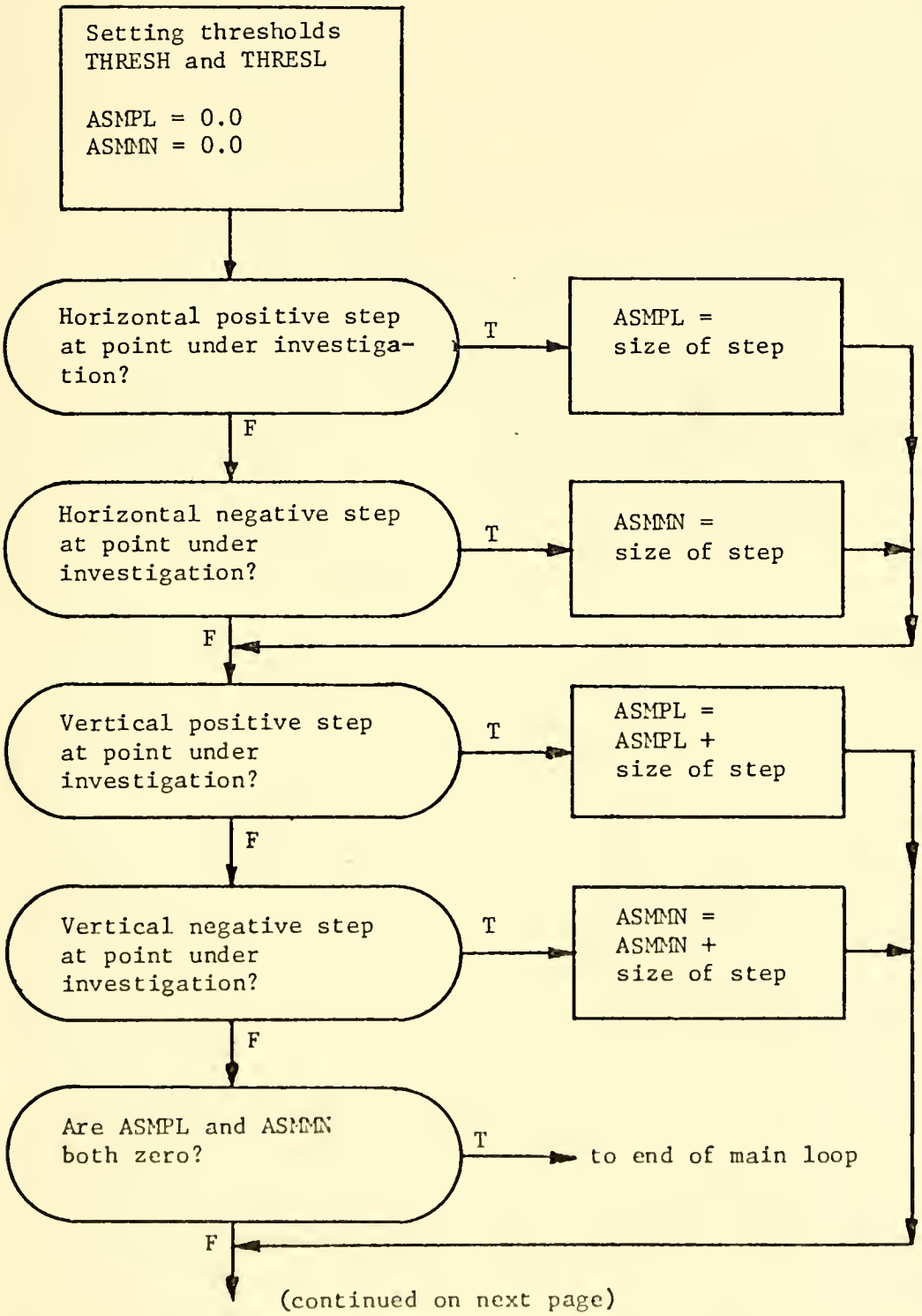
In this investigation the spatial frequency approach to diffraction has been preferred to programming the Fresnel equation, mainly because repeated small propagation steps are possible when the transfer function is used, but also because it circumvents the Fresnel approximations and therefore gives better results in its region of validity. It was found that the necessary neglect of evanescent waves can give rise to very strong deterioration of the output, most of all for diffusely emitting objects. It is tempting to use a lower sampling rate than the Nyquist rate because this avoids the necessity of neglecting evanescent waves: perfect reconstructions can be obtained this way, but only for computer generated diffraction patterns, or those obtained in an actual data acquisition system from periodic objects. For all other diffraction patterns the aliasing introduced by undersampling could be expected to be quite severe.

For cases where the neglect of evanescent waves destroys edge information to a high degree, the reconstruction procedure developed in this investigation will not succeed since the resulting diffraction pattern at the object location does not belong to that class of

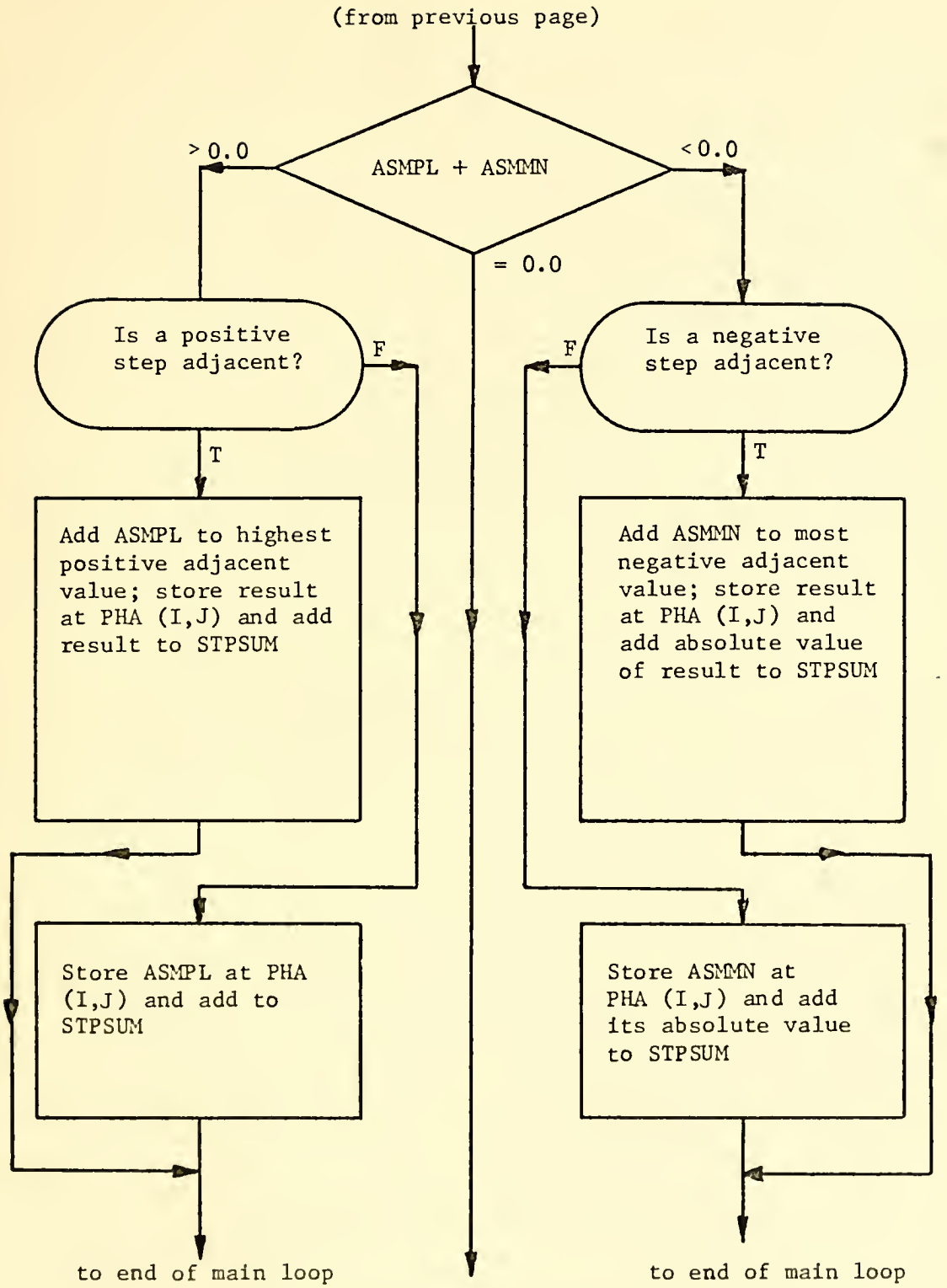
diffraction patterns which the computer accepts as objects because of the deterioration of the edge features. However, the test results presented are quite encouraging as they show that for many situations the computer can indeed make correct decisions as to the object location.

APPENDIX

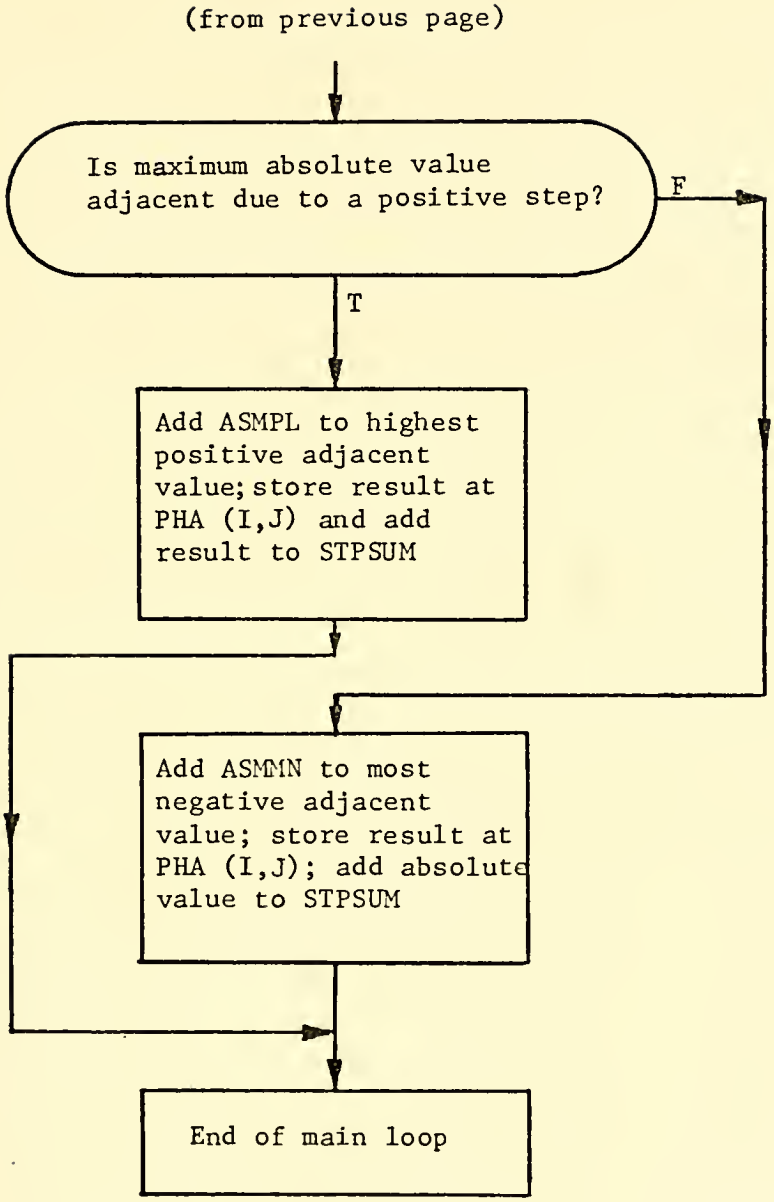
Main Loop of Subroutine ADERIV
This computes the edgemeasure of the array AMAG



(continued on next page)



(continued on next page)


(from previous page)
Is maximum absolute value adjacent due to a positive step?
F
T
Add ASMPL to highest positive adjacent value; store result at PHA (I,J) and add result to STPSUM
Add ASMMN to most negative adjacent value; store result at PHA (I,J); add absolute value to STPSUM
End of main loop

```
//MUELLERS JOB (1560,0855,WE04),'MUELLER,SMC2954',TIME=10
// EXEC FORTCLGP,REGION.GO=180K
//FORT.SYSIN DD *
      IMPLICIT COMPLEX*8(D)
      DIMENSION AMAGVC(4096),PHAVC(4096),DATAVC(4096),XVEC(64),YVEC(64)
     1,YVECP(64),ZVEC(30),CONVEC(30),RMAXVC(20)
      COMMON/DAREA1/DATA(64,64)/AREA2/AMAG(64,64),PHA(64,64)/DAREA3/DJ/
     1AREA4/Z,ALAM,PI/IAREA5/ISIGN,NDIM,NN(2)/AREA6/AINCR/AREA7/AMDER,
     2STPSUM,THRUP,THRLO/AREA8/ZFIRST,ZLAST,THRCON/IAREA9/NRSTEP,NRMAX /
     3AREA10/CONVAL(20),ZZMAX(20)
      EQUIVALENCE (DATA(1,1),DATAVC(1)),(AMAG(1,1),AMAGVC(1)),(PHA(1,1),
     1PHAVC(1))
      PI=3.14159265
      DJ=(0.0,1.0)
C
CC    SETTING ARRAY SIZE
C
      NN(1)=64
      NN(2)=NN(1)
      NNN=NN(1)
      NDIM=2
C
CC    SETTING VALUE FOR PROPAGATION WAVELENGTH
C
      ALAM=1.0
C
CC    SETTING INCREMENT BETWEEN ARRAY POINTS
C
      AINCR=0.5
C
CC    SETTING 7 INPUT PARAMETERS FOR RECONSTRUCTION ROUTINE
C
      ZFIRST=80.0
      ZLAST=120.0
      NRSTEP=40
      THRCON=0.6
      THRLO=0.2
      THRUP=0.6
      NRMAX=10
      WRITE(6,8002) NN(1),NN(1),ALAM,AINCR
 8002 FORMAT('1','ARRAY SIZE IS ',I4,' BY ',I4,'.',//,' WAVELENGTH IS',
     1F8.3,' UNITS.',//,' INTERVAL BETWEEN ARRAY POINTS IS ',F8.3,' UNIT
     2S.',//)
      WRITE (6,8003) ZFIRST,ZLAST,NRSTEP,NRMAX
 8003 FORMAT(' ','DIFFRACTION FIELD WILL BE INVESTIGATED BETWEEN',F9.3,
     1' AND',F9.3,' UNITS IN ',I3,' STEPS.',I3,' MAXIMA IN CONTRAST WILL
     2 BE LOOKED AT.',//)
      WRITE (6,8004) THRUP,THRLO,THRCON
```

```
 8004 FORMAT(' '.'UPPER THRESHOLD IS SET TO ',F5.3,',LOWER THRESHOLD TO'
     1,F6.3,' OF MAXIMUM DIFFERENCE BETWEEN ARRAY VALUES.',/,' CONTRAST
     2MAXIMA OF LESS THAN ',F5.3,' OF THE MAXIMUM CONTRAST VALUE WILL BE
     3 DISREGARDED.'./)
C
C       ZEROING OUT INPUT ARRAY
C
        NSQ=NN(1)*NN(2)
        DO 1001 K=1,4096
        DATAVC(K)=(0.0,0.0)
        AMAGVC(K)=0.0
        PHAVC(K)=0.0
 1001   CONTINUE
C
C       READING IN OBSERVED VALUES FOR AMPLITUDE AND PHASE INTO ARRAY
C       AMAGVC AND PHAVC RESPECTIVELY. AMAGVC IS EQUIVALENT TO AMAG, PHAVC
C       IS EQUIVALENT TO PHA
C
        READ (5,6660) (AMAGVC(K),K=1,NSQ)
        READ (5,6660) ( PHAVC(K),K=1,NSQ)
 6660   FORMAT (F9.4,4F14.4)
        DO 998 K=1,NSQ
        AIMAGP=AMAGVC(K)*SIN(PHAVC(K))
        REALP=AMAGVC(K)*COS(PHAVC(K))
        DATAVC(K)=CMPLX(REALP,AIMAGP)
  998   CONTINUE
C
C       CALLING RECONSTRUCTION SUBROUTINE
C
        CALL RECON
        STOP
        END


        SUBROUTINE RECON
C
C       SUBROUTINE RECON ACCOMPLISHES AUTOMATIC RECONSTRUCTION BY FINDING
C       MAXIMA IN EDGEMEASURE
C       POSITIVE DISTANCES FOR THE OBSERVER ARE NEGATIVE FOR THE PROPA-
C       GATION ALGORITHM
C
        IMPLICIT COMPLEX*8(D)
        REAL*8 TITLE(12)
        LOGICAL*1 LTG(3)
        DIMENSION AMAGVC(4096),PHAVC(4096),DATAVC(4096),XVEC(64),YVEC(64)
       1,YVECP(64),ZVEC(30),CONVEC(30),RMAXVC(20)
        DIMENSION CL(10)
        COMMON/DAREA1/DATA(64,64)/AREA2/AMAG(64,64),PHA(64,64)/DAREA3/DJ/
```

```
     1AREA4/Z,ALAM,PI/IAREA5/ISIGN,NDIM,NN(2)/AREA6/AINCR/AREA7/AMDER,
     2STPSUM,THRUP,THRLO/AREA8/ZFIRST,ZLAST,THRCON/IAREA9/NRSTEP,NRMAX /
     3AREA10/CONVAL(20),ZZMAX(20)
      EQUIVALENCE (DATA(1,1),DATAVC(1)),(AMAG(1,1),AMAGVC(1)),(PHA(1,1),
     1PHAVC(1))
      DATA TITLE/'MUELLER ',
     1                  'A',
     2                 'MPLITUDE',
     3                 ' CONTOUR',
     4                 ' OF OBJE',
     5                 'CT      ',
     6                 '        ',
     7                 '        ',
     8                 '        ',
     8                 '        ',
     C                 '        ',
     C                 '        '/
      LTG(1)=.TRUE.
      LTG(2)=.TRUE.
      LTG(3)=.FALSE.
C
CC    COMPUTING SIZE OF PROPAGATION STEP FOR THE FIRST PASS THROUGH THE
CC    VOLUME OF INTEREST
C
      ZSTEP=(ZLAST-ZFIRST)/FLOAT(NRSTEP)
      Z=-ZFIRST
      ZVEC(1)=ZFIRST
C
CC    FIRST PASS THROUGH VOLUME OF INTEREST
C
      NREP=NRSTEP+1
 5454 DO 9876 L=1,NREP
      ZVEC(L+1)=ZVEC(L)+ZSTEP
      CALL PROPAG
      CALL CONVRT
      CALL ADERIV
C
CC    STORING COMPUTED EDGEMEASURE AT GIVEN DISTANCE IN CONVEC(L)
C
      CONVEC(L)=STPSUM
      Z=-ZSTEP
      ZLAST=ZVEC(L)
 9876 CONTINUE
C
CC    END OF FIRST PASS
C
CC    FINDING RELATIVE MAXIMA OF ARRAY VALUES IN CONVEC
C
```

```
      MAXNR=0
      DO 9765 L=2,NRSTEP
      IF((CONVEC(L+1).LE.CONVEC(L)).AND.(CONVEC(L-1).LE.CONVEC(L)))GO TO
     1 2345
      GO TO 9765
 2345 MAXNR=MAXNR+1
      CONVEC(MAXNR)=CONVEC(L)
      RMAXVC(MAXNR)=ZVEC(L)
 9765 CONTINUE
      IF (MAXNR.LE.0) GO TO 5400
      IF (MAXNR.EQ.1) GO TO 4710
C
C     ORDER RELATIVE MAXIMA ACCORDING TO THEIR SIZE
C
      MANR=MAXNR-1
      DO 9654 L=1,MANR
      NSTOP=MAXNR-L
      DO 9654 LL=1,NSTOP
      IF (CONVEC(LL).GE.CONVEC(LL+1)) GO TO 9654
      TEMPC=CONVEC(LL)
      TEMPZ=RMAXVC(LL)
      CONVEC(LL)=CONVEC(LL+1)
      RMAXVC(LL)=RMAXVC(LL+1)
      CONVEC(LL+1)=TEMPC
      RMAXVC(LL+1)=TEMPZ
 9654 CONTINUE
C
C     DISREGARDING MAXIMA OF LESS THAN 10 PERCENT OF THE OVERALL MAX-
C     IMUM
C
      MSTOP=MAXNR
      CONTHR=0.1*CONVEC(1)
      DO 9543 LM=2,MSTOP
      IF (CONVEC(LM).GT.CONTHR) GO TO 9543
      CONVEC(LM)=0.0
      MAXNR=MAXNR-1
 9543 CONTINUE
C
C     COMPUTING STEPSIZE AND NUMBER OF PROPAGATION STEPS FOR SECOND PASS
C
      ZZSTEP=AINCR/ALAM
      MSTP=1+IFIX(ZSTEP/ZZSTEP)
      WRITE(6,8880)MSTP
 8880 FORMAT(' ','INVESTIGATING IN ',I3,'STEPS')
 4710 IF(MAXNR.LT.NRMAX) GO TO 5411
 4711 NINV=NRMAX
      GO TO 4713
 4712 NINV=MAXNR
```

```
C
CC      SECOND PASS INVESTIGATES VICINITY OF FIRST PASS MAXIMA
C
 4713 DO 7676 NM=1,NINV
      Z=ZLAST-RMAXVC(NM)+ZSTEP/2.0
      ZVEC(1)=RMAXVC(NM)-ZSTEP/2.0
      CONTST=CONVEC(NM)
      ZVCMAX=RMAXVC(NM)
      DO 7677 NP=1,MSTP
      ZVEC(NP+1)=ZVEC(NP)+ZZSTEP
      CALL PROPAG
      CALL CONVRT
      CALL ADERIV
      IF(STPSUM.LT.CONTST) GO TO 7666
      CONTST=STPSUM
      ZVCMAX=ZVEC(NP)
 7666 Z=-ZZSTEP
      ZLAST=ZVEC(NP)
 7677 CONTINUE
      CONVAL(NM)=CONTST
      ZZMAX(NM)=ZVCMAX
 7676 CONTINUE
C
CC      END OF SECOND PASS
C
      GO TO 4554
 5400 WRITE(6,6606)
 6606 FORMAT(' ','NO CONTRAST MAXIMUM FOUND')
      GO TO 9323
 5411 WRITE(6,6707) MAXNR
 6707 FORMAT(' ','THERE ARE ONLY',I3,'CONTRAST MAXIMA')
      GO TO 4712
 4554 CONTINUE
      IF(NINV.EQ.1) GO TO 9322
C
CC      ORDERING SECOND PASS MAXIMA ACCORDING TO SIZE
C
      NIN=NINV-1
      DO 9321 LP=1,NIN
      KSTOP=NINV-LP
      DO 9321 LO=1,KSTOP
      IF(CONVAL(LO).GT.CONVAL(LO+1)) GO TO 9321
      TTEMPC=CONVAL(LO)
      TTEMPZ=ZZMAX(LO)
      CONVAL(LO)=CONVAL(LO+1)
      ZZMAX(LO)=ZZMAX(LO+1)
      CONVAL(LO+1)=TTEMPC
      ZZMAX(LO+1)=TTEMPZ
```

```
 9321 CONTINUE
C
C      DISREGARDING MAXIMA OF SMALLER VALUE THAN GIVEN BY INPUT PARA-
C      METER THRCON
C
       NNSTOP=NINV
       CVLTHR=THRCON*CONVAL(1)
       DO 9322 LQ=2,NNSTOP
       IF(CONVAL(LQ).GE.CVLTHR) GO TO 9322
       CONVAL(LQ)=0.0
       NINV=NINV-1
 9322 CONTINUE
       WRITE(6,3437) NINV
 3437 FORMAT(' ',I4,' OF THE INVESTIGATED MAXIMA ARE HIGHER THAN POSTULA
     1TED THRESHOLD')
C
C      OUTPUTTING LOCATIONS OF COMPUTED MAXIMA
C
       DO 9323 LR=1,NINV
       Z=ZLAST-ZZMAX(LR)
       ZLAST=ZZMAX(LR)
       CALL PROPAG
       CALL CONVER
       WRITE(6,3030) ZLAST,CONVAL(LR)
 3030 FORMAT (' ',T40,'OBJECT AT ',F9.3,' , EDGEMEASURE',F9.3)
       CALL CONTUR(AMAG,NNN,NNN,NNN,CL,-6,TITLE,6,6,LTG)
 9323 CONTINUE
       RETURN
       END


       SUBROUTINE ADERIV
C      SUBROUTINE ADERIV COMPUTES THE EDGEMEASURE OF THE ARRAY AMAG AND
C      STORES IT IN STPSUM
       COMMON/AREA2/AMAG(64,64),PHA(64,64)/IAREA5/ISIGN,NDIM,NN(2)/
      1AREA7/AMDER,STPSUM,THRUP,THRLO
       NNN=NN(1)
       NNMIN=NNN-1
C
C      COMPUTING MAXIMUM DIFFERENCE OF AMPLITUDES AT TWO ADJACENT ARRAY
C      POINTS AND STORING IT IN AMDER
C
       AMDER=0.0
       DO 1222 I=1,NNMIN
       DO 1222 J=1,NNMIN
       TEST=AMAG(I,J+1)-AMAG(I,J)
       TESTA=ABS(TEST)
       IF (TESTA.GE.AMDER) AMDER=TESTA
```

```
      TEST=AMAG(I+1,J)-AMAG(I,J)
      TESTA=ABS(TEST)
      IF (TESTA.GE.AMDER) AMDER=TESTA
 1222 CONTINUE
C
C      SETTING UPPER THRESHOLD THRESH AND LOWER THRESHOLD THRESL
C
      THRESH=THRUP*AMDER
      THRESL=THRLO*AMDER
      STPSUM=0.0
C
C     MAIN LOOP STARTS HERE
C
 1440 DO 1444 I=3,NNMIN
      DO 1444 J=3,NNMIN
C
C     ZEROING OUT VALUES FOR POSITIVE AND NEGATIVE STEPS
C
      ASMPL=0.0
      ASMMN=0.0
C
C     COMPUTING VALUES FOR STEPS AT LOCATION UNDER INVESTIGATION AND
C     CHECKING IF POSITIVE STEP AS DEFINED
C
      STEP=AMAG(I,J)-AMAG(I,J-1)
      ASTEP=ABS(AMAG(I,J)-AMAG(I,J+1))
      BSTEP=ABS(AMAG(I,J-2)-AMAG(I,J-1))
      IF((STEP.GE.THRESH).AND.(ASTEP.LE.THRESL).AND.(BSTEP.LE.THRESL))
     1GO TO 1500
C
C     CHECKING IF NEGATIVE STEP AS DEFINED
C
      STEPM=-1.0*STEP
      IF((STEPM.GE.THRESH).AND.(ASTEP.LE.THRESL).AND.(BSTEP.LE.THRESL))
     1ASMMN=STEP
      GO TO 1600
C
C     ENTER VALUE FOR FOUND POSITIVE STEP IN ASMPL
C
 1500 ASMPL=STEP
C
C     COMPUTING VALUES FOR STEPS AT LOCATION UNDER INVESTIGATION AND
C     CHECKING IF POSITIVE STEP AS DEFINED
C
 1600 STEP=AMAG(I,J)-AMAG(I-1,J)
      ASTEP=ABS(AMAG(I,J)-AMAG(I+1,J))
      BSTEP=ABS(AMAG(I-2,J)-AMAG(I-1,J))
      IF((STEP.GE.THRESH).AND.(ASTEP.LE.THRESL).AND.(BSTEP.LE.THRESL))
```

```
     1GO TO 1550
C
C      CHECKING IF NEGATIVE STEP AS DEFINED
C
       STEPM=-1.0*STEP
       IF((STEPM.GE.THRESH).AND.(ASTEP.LE.THRESL).AND.(BSTEP.LE.THRESL))
     1GO TO 1650
       IF (ASMPL.GT.0.0) GO TO 1700
       IF (ASMMN.LT.0.0) GO TO 1700
C
C      NO BONA FIDE STEP AT LOCATION IN QUESTION. GO TO NEXT ARRAY POINT
C
       GO TO 1444
C
C      POSITIVE STEPS IN BOTH HORIZONTAL AND VERTICAL DIRECTION. ADDING
C      THEM UP
C
 1550  ASMPL=ASMPL+STEP
       GO TO 1700
C
C      NEGATIVE STEPS IN BOTH HORIZONTAL AND VERTICAL DIRECTION. ADDING
C      THEM UP
C
 1650  ASMMN=ASMMN+STEP
C
C      IS POSITIVE OR NEGATIVE STEP LARGER IN ABSOLUTE VALUE ?
C
 1700  ASM=ASMPL+ASMMN
       IF(ASM) 1800,1850,1900
C
C      FINDING MINIMUM OF EDGEMEASURE STORED IN 4 PREVIOUSLY TREATED
C      LOCATIONS
C
 1800  SMM=AMIN1(PHA(I-1,J-1),PHA(I,J-1),PHA(I-1,J+1),PHA(I-1,J))
       IF(SMM.LE.0.0) GO TO 1801
C
C      NO NEGATIVE STEP ADJACENT. STORE COMPUTED NEGATIVE STEP AT
C      LOCATION AND ADD ITS ABSOLUTE VALUE TO STPSUM
C
       PHA(I,J)=ASMMN
       STPSUM=STPSUM-PHA(I,J)
       GO TO 1444
C
C      FINDING MAXIMUM OF EDGEMEASURE STORED IN 4 PREVIOUSLY TREATED
C      LOCATIONS
C
 1900  SMP=AMAX1(PHA(I-1,J-1),PHA(I,J-1),PHA(I-1,J+1),PHA(I-1,J))
       IF(SMP.GE.0.0) GO TO 1901
```

```
C
C       NO POSITIVE STEP STORED IN ADJACENT LOCATION. STORE COMPUTED
C       POSITIVE STEP AT LOCATION UNDER INVESTIGATION AND ADD ITS VALUE
C       TO STPSUM
C
        PHA(I,J)=ASMPL
        STPSUM=STPSUM+PHA(I,J)
        GO TO 1444
C
C       IS EDGEMEASURE AT ADJACENT LOCATIONS HIGHER FOR POSITIVE OR FOR
C       NEGATIVE STEPS ?
C
 1850   SMP=AMAX1(PHA(I-1,J-1),PHA(I,J-1),PHA(I-1,J+1),PHA(I-1,J))
        SMM=AMIN1(PHA(I-1,J-1),PHA(I,J-1),PHA(I-1,J+1),PHA(I-1,J))
        SM=SMM+SMP
        IF(SM) 1801,1801,1901
C
C       ADD ABSOLUTE VALUE FOR NEGATIVE STEP TO STPSUM
C
 1801   PHA(I,J)=SMM+ASMMN
        STPSUM=STPSUM-PHA(I,J)
        GO TO 1444
C
C       ADD VALUE FOR POSITIVE STEP TO STPSUM
C
 1901   PHA(I,J)=SMP+ASMPL
        STPSUM=STPSUM+PHA(I,J)
 1444   CONTINUE
C
C       END OF MAIN LOOP
C
        RETURN
        END


        SUBROUTINE PROPAG
C
C       SUBROUTINE PROPAG ACCOMPLISHES THE PROPAGATION OF THE ARRAY DATA
C       IN THE SPATIAL FREQUENCY APPROACH
C
        IMPLICIT COMPLEX*8 (D)
        DIMENSION DATAVC(4096)
        COMMON/DAREA1/DATA(64,64)/DAREA3/DJ/AREA4/Z,ALAM,PI/IAREA5/ISIGN,
       1NDIM,NN(2)/AREA6/AINCR
        EQUIVALENCE (DATA(1,1),DATAVC(1))
        NNN=NN(1)
        CALL SHUFL
C
```

```
C       CALLING FORWARD TRANSFORM
C
        CALL FOUR2(DATA,NN,2,-1)
C
CC      DATA CONTAINS NOW THE SPATIAL FREQUENCY COMPONENTS
C
        CALL SHUFL
        DARG=DJ*2.0*PI*Z/(ALAM*AINCR*NN(1))
        ALSQ=(AINCR*NN(1))**2
        NNPL=1+NN(1)/2
C
CC      MULTIPLYING EACH FREQUENCY COMPONENT BY THE APROPRIATE PHASE
CC      DELAY
C
        DO 900 I=1,NNN
        AI=FLOAT(I-NNPL)
        AISQ=AI**2
        DO 900 J=1,NNN
        AJ=FLOAT(J-NNPL)
        AJSQ=AJ**2
        FREQ=ALSQ-(AISQ+AJSQ)*ALAM**2
        IF (FREQ.LE.0.0) GO TO 800
        ARG=SQRT(FREQ)
        DAXP=CEXP(DARG*ARG)
        GO TO 888
C
CC      SETTING EVANESCENT COMPONENTS TO ZERO
C
  800   DATA(I,J)=(0.0,0.0)
        GO TO 900
  888   DATA(I,J)=DATA(I,J)*DAXP
  900   CONTINUE
        CALL SHUFL
C
CC      INVERSE TRANSFORM
C
        CALL FOUR2(DATA,NN,2,1)
C
CC      DIVIDING EACH ARRAY VALUE BY THE TOTAL NUMBER OF ARRAY POINTS FOR
CC      CORRECT SCALING
C
        NNSQ=NN(1)**2
        ANSQ=FLOAT(NNSQ)
        DO 5555 K=1,NNSQ
        DATAVC(K)=DATAVC(K)/ANSQ
 5555   CONTINUE
        CALL SHUFL
        RETURN
```

```
      END

      SUBROUTINE SHUFL
C
CC    SUBROUTINE SHUFL REARRANGES THE 4 QUADRANTS OF THE ARRAY DATA
C
      IMPLICIT COMPLEX*8 (D)
      COMMON/DAREA1/DATA(64,64)/IAREA5/ISIGN,NDIM,NN(2)
      NNN=NN(1)
      NNHF=NN(1)/2
C
CC    INTERCHANGING QUADRANTS I AND IV
C
      DO 4001 I=1,NNHF
      IPR=I+NNHF
      DO 4001 J=1,NNHF
      JPR=J+NNHF
      DEMP=DATA(I,J)
      DATA(I,J)=DATA(IPR,JPR)
      DATA(IPR,JPR)=DEMP
 4001 CONTINUE
C
CC    INTERCHANGING QUADRANTS II AND III
C
      NNPL=NNHF+1
      DO 4002 I=NNPL,NNN
      IPR=I-NNHF
      DO 4002 J=1,NNHF
      JPR=J+NNHF
      DEMP=DATA(I,J)
      DATA(I,J)=DATA(IPR,JPR)
      DATA(IPR,JPR)=DEMP
 4002 CONTINUE
      RETURN
      END

      SUBROUTINE CONVRT
C
CC    SUBROUTINE CONVRT FINDS THE MAGNITUDE OF THE ARRAY POINTS IN DATA
CC    AND STORES IT IN AMAG . PHA IS ZEROED OUT AS NEEDED BY SUB-
CC    ROUTINE ADERIV
C
      IMPLICIT COMPLEX*8 (D)
      DIMENSION DATAVC(4096),AMAGVC(4096),PHAVC(4096)
      COMMON/DAREA1/DATA(64,64)/AREA2/AMAG(64,64),PHA(64,64)
     1/IAREA5/ISIGN,NDIM,NN(2)
```

```
      EQUIVALENCE (DATA(1,1),DATAVC(1)),(AMAG(1,1),AMAGVC(1)),(PHA(1,1),
     1PHAVC(1))
      NNSQ=NN(1)**2
      DO 1080 K=1,NNSQ
      REALP=REAL(DATAVC(K))
      AIMAGP=AIMAG(DATAVC(K))
      AMAGVC(K)=SQRT(REALP**2+AIMAGP**2)
      PHAVC(K)=0.0
 1080 CONTINUE
      RETURN
      END


      SUBROUTINE CONVER
C
C     SUBROUTINE CONVER CONVERTS THE COMPLEX NUMBERS IN DATA TO THEIR
C     MAGNITUDE AND PHASE AND STORES THEM IN ARRAYS AMAG AND PHA
C
      IMPLICIT COMPLEX*8 (D)
      DIMENSION DATAVC(4096),AMAGVC(4096),PHAVC(4096)
      COMMON/DAREA1/DATA(64,64)/AREA2/AMAG(64,64),PHA(64,64)
     1/IAREA5/ISIGN,NDIM,NN(2)
      EQUIVALENCE (DATA(1,1),DATAVC(1)),(AMAG(1,1),AMAGVC(1)),(PHA(1,1),
     1PHAVC(1))
      NNSQ=NN(1)**2
      DO 1080 K=1,NNSQ
      REALP=REAL(DATAVC(K))
      AIMAGP=AIMAG(DATAVC(K))
      AMAGVC(K)=SQRT(REALP**2+AIMAGP**2)
      IF(AMAGVC(K).LT.0.00001) GO TO 1070
      PHAVC(K)=ATAN2(AIMAGP,REALP)
      GO TO 1080
 1070 PHAVC(K)=0.0
 1080 CONTINUE
      RETURN
      END


      SUBROUTINE FOUR2(DATA,NN,NDIM,ISIGN)
C
C     THE COOLEY-TUKEY FAST FOURIER TRANSFORM IN USASI BASIC FORTRAN
C     TRANSFORM(J1,J2,...) = SUM(DATA(I1,I2,...)+W1**((I1-1)*(J1-1))
C                                  *W2**((I2-1)*(J2-1)*...).
C     WHERE I1 AND J1 RUN FROM 1 TO NN(1) AND W1=EXP(ISIGN*2*PI*
C     SQRT(-1)/NN(1)),ETC.
C     DATA IS A MULTIDIMENSIONAL FLOATING POINT ARRAY ALL OF WHOSE
C     DIMENSIONS ARE POWERS OF TWO. THE LENGTH OF EACH DIMENSION IS
C     STORED IN THE INTEGER ARRAY NN, OF LENGTH NDIM. ISIGN IS
```

```
C      +1 OR -1. GIVING THE SIGN OF THE TRANSFORM. THE REAL
C      AND IMAGINARY PARTS OF A DATUM ARE IMMEDIATELY ADJACENT IN STORAGE
C      (SUCH AS FORTRAN IV PLACES THEM). TRANSFORM RESULTS ARE RETURNED
C      IN ARRAY DATA. REPLACING THE ORIGINAL DATA. TIME IS PROPORTIONAL
C      TO N*LOG2(N), RATHER THAN THE USUAL N**2. NOTE THAT IF A FORWARD
C      TRANSFORM IS FOLLOWED BY AN INVERSE TRANSFORM, THE ORIGINAL DATA
C      WILL REAPPEAR MULTIPLIED BY NN(1)*NN(2)*.... EXAMPLE--
C      FORWARD FOURIER TRANSFORM OF A TWO-DIMENSIONAL ARRAY IN FORTRAN IV
C
C      DIMENSION DATA(64,32),NN(2)
C      COMPLEX DATA
C      DATA NN/64,32/
C      DO 1 I=1,64
C      DO 1 J=1,32
C    1 DATA(I,J)=COMPLEX VALUE
C      CALL FOUR2(DATA,NN,2,-1)
C
C      PROGRAMM BY NORMAN BRENNER FROM THE BASIC PROGRAM BY CHARLES
C      RADER.  MAY 1967. THE IDEA FOR THE DIGIT REVERSAL WAS SUGGESTED
C      BY RALPH ALTER.
C
C      THIS VERSION OF THE FAST FOURIER TRANSFORM IS THE FASTEST KNOWN
C      TO THE AUTOR. LOOKING UP SINES AND COSINES IN A TABLE INSTEAD OF
C      COMPUTING THEM WOULD DECREASE RUNNING TIME SEVEN PERCENT
C      SEE-- IEEE AUDIO TRANSACTIONS (JUNE 1967), SPECIAL ISSUE ON FFT.
C
       DIMENSION DATA(1),NN(1)
       IF(NDIM-1)700,1,1
     1 NTOT=2
       DO 2 IDIM=1,NDIM
       IF(NN(IDIM))700,700,2
     2 NTOT=NTOT*NN(IDIM)
       TWOPI=6.2831853070
       RTHLF=.7071067812
C
C      MAIN LOOP FOR EACH DIMENSION
C
       NP1=2
       DO 600 IDIM=1,NDIM
       N=NN(IDIM)
       NP2=NP1*N
       IF(N-1)700,600,100
C
C      SHUFFLE DATA BY BIT REVERSAL, SINCE N=2**K. AS THE SHUFFLING
C      CAN BE DONE BY SIMPLE INTERCHANGE, NO WORKING ARRAY IS NEEDED
C
   100 NP2HF=NP2/2
       J=1
```

```
      DO 160 I2=1,NP2,NP1
      IF(J-I2)110,130,130
  110 I1MAX=I2+NP1-2
      DO 120 I1=I2,I1MAX,2
      DO 120 I3=I1,NTOT,NP2
      J3=J+I3-I2
      TEMPR=DATA(I3)
      TEMPI=DATA(I3+1)
      DATA(I3)=DATA(J3)
      DATA(I3+1)=DATA(J3+1)
      DATA(J3)=TEMPR
  120 DATA(J3+1)=TEMPI
  130 M=NP2HF
  140 IF(J-M)160,160,150
  150 J=J-M
      M=M/2
      IF(M-NP1)160,140,140
  160 J=J+M
C
C     MAIN LOOP. PERFORM FOURIER TRANSFORMS OF LENGTH FOUR, WITH ONE OF
C     LENGTH TWO IF NEEDED. THE TWIDDLE FACTOR W=EXP(ISIGN*2*PI*
C     SQRT(-1)*M/(4+MMAX)). CHECK FOR THE SPECIAL CASE W=ISIGN*SQRT(-1)
C     AND REPEAT FOR W=W*(1+ISIGN*SQRT(-1))/SQRT(2).
C
      NP1TW=NP1+NP1
      IPAR=N
  310 IF(IPAR-2)350,330,320
  320 IPAR=IPAR/4
      GO TO 310
  330 DO 340 I1=1,NP1,2
      DO 340 K1=I1,NTOT,NP1TW
      K2=K1+NP1
      TEMPR=DATA(K2)
      TEMPI=DATA(K2+1)
      DATA(K2)=DATA(K1)-TEMPR
      DATA(K2+1)=DATA(K1+1)-TEMPI
      DATA(K1)=DATA(K1)+TEMPR
  340 DATA(K1+1)=DATA(K1+1)+TEMPI
  350 MMAX=NP1
  360 IF(MMAX-NP2HF)370,600,600
  370 LMAX=MAXO(NP1TW,MMAX/2)
      DO 570 L=NP1,LMAX,NP1TW
      M=L
      IF(MMAX-NP1)420,420,380
  380 THETA=-TWOPI*FLOAT(M)/FLOAT(4*MMAX)
      IF (ISIGN)400,390,390
  390 THETA=-THETA
  400 WR=COS(THETA)
```

```
      WI=SIN(THETA)
  410 W2R=WR*WR-WI*WI
      W2I=2.0*WR*WI
      W3R=W2R*WR-W2I*WI
      W3I=W2R*WI+W2I*WR
  420 DO 530 I1=1,NP1,2
      KMIN=IPAR*M+I1
      IF(MMAX-NP1)430,430,440
  430 KMIN=I1
  440 KDIF=IPAR*MMAX
  450 KSTEP=4*KDIF
      DO 520 K1=KMIN,NTOT,KSTEP
      K2=K1+KDIF
      K3=K2+KDIF
      K4=K3+KDIF
      IF(MMAX-NP1)460,460,480
  460 U1R=DATA(K1)+DATA(K2)
      U1I=DATA(K1+1)+DATA(K2+1)
      U2R=DATA(K3)+DATA(K4)
      U2I=DATA(K3+1)+DATA(K4+1)
      U3R=DATA(K1)-DATA(K2)
      U3I=DATA(K1+1)-DATA(K2+1)
      IF(ISIGN)470,475,475
  470 U4R=DATA(K3+1)-DATA(K4+1)
      U4I=DATA(K4)-DATA(K3)
      GO TO 510
  475 U4R=DATA(K4+1)-DATA(K3+1)
      U4I=DATA(K3)-DATA(K4)
      GO TO 510
  480 T2R=W2R*DATA(K2)-W2I*DATA(K2+1)
      T2I=W2R*DATA(K2+1)+W2I*DATA(K2)
      T3R=WR*DATA(K3)-WI*DATA(K3+1)
      T3I=WR*DATA(K3+1)+WI*DATA(K3)
      T4R=W3R*DATA(K4)-W3I*DATA(K4+1)
      T4I=W3R*DATA(K4+1)+W3I*DATA(K4)
      U1R=DATA(K1)+T2R
      U1I=DATA(K1+1)+T2I
      U2R=T3R+T4R
      U2I=T3I+T4I
      U3R=DATA(K1)-T2R
      U3I=DATA(K1+1)-T2I
      IF(ISIGN)490,500,500
  490 U4R=T3I-T4I
      U4I=T4R-T3R
      GO TO 510
  500 U4R=T4I-T3I
      U4I=T3R-T4R
  510 DATA(K1)=U1R+U2R
```

```
      DATA(K1+1)=U1I+U2I
      DATA(K2)=U3R+U4R
      DATA(K2+1)=U3I+U4I
      DATA(K3)=U1R-U2R
      DATA(K3+1)=U1I-U2I
      DATA(K4)=U3R-U4R
  520 DATA(K4+1)=U3I-U4I
      KMIN=4*(KMIN-I1)+I1
      KDIF=KSTEP
      IF(KDIF-NP2HF)450,450,530
  530 CONTINUE
      M=M+LMAX
      IF(M-MMAX)540,540,570
  540 IF(ISIGN)550,560,560
  550 TEMPR=WR
      WR=(WR+WI)*RTHLF
      WI=(WI-TEMPR)*RTHLF
      GO TO 410
  560 TEMPR=WR
      WR=(WR-WI)*RTHLF
      WI=(TEMPR+WI)*RTHLF
      GO TO 410
  570 CONTINUE
      IPAR=3-IPAR
      MMAX=MMAX+MMAX
      GO TO 360
  600 NP1=NP2
  700 RETURN
      END

      SUBROUTINE CONTUR(AM,M,N,MX,CL,NL,TITLE,IW,IH,LTG)
C.....CNTR0010
CC                                                                    CNTR0020
CC                    SUBROUTINE CONTUR
CC                                                                    CNTR0040
CC    1. IDENTIFICATION                                               CNTR0050
CC                                                                    CNTR0060
CC       A. NAME: GENERATE CONTOUR GRAPH (CONTUR) - J5                CNTR0070
CC                                                                    CNTR0080
CC       B. PROGRAMMER: BERNADETTE R.PEAVEY, 7 JULY 1969              CNTR0090
CC                                                                    CNTR0100
CC    2. PURPOSE                                                      CNTR0110
CC                                                                    CNTR0120
CC           CONTUR GENERATES A CONTOUR GRAPH ON WHICH ONE OR MORE    CNTR0130
CC       (USUALLY MORE) CONTOUR LEVELS ARE DRAWN; THAT IS, GIVEN A MATRIX CNTR0140
CC       OF NUMERICAL VALUES OF Z=F(X,Y), CONTUR FINDS THE LOCI OF    CNTR0150
C        SPECIFIED DISCRETE VALUES OF Z AND PLOTS THESE LOCI (CONTOUR CNTR0160
```

```
C      LINES) USING THE OFFLINE PLOTTER.                                  CNTR0170
C                                                                         CNTR0180
C   3. USAGE                                                              CNTR0190
C                                                                         CNTR0200
C      A. DEFINITIONS                                                     CNTR0210
C                                                                         CNTR0220
C          IN WHAT FOLLOWS THE WORD GRAPH OR CONTOUR GRAPH WILL BE        CNTR0230
C      TAKEN TO MEAN A COMPLETE CONTOUR PICTURE; THAT IS, ONE PIECE       CNTR0240
C      OR FRAME OF GRAPH PAPER ON WHICH ONE OR MORE (USUALLY MORE)        CNTR0250
C      CONTOUR LEVELS ARE DRAWN.                                          CNTR0260
C                                                                         CNTR0270
C          A CONTOUR OR CONTOUR LEVEL WILL MEAN THE PLOT OF THE LOCUS     CNTR0280
C      ON THE XY PLANE OF ALL Z EQUAL TO A CONSTANT VALUE. NORMALLY,      CNTR0290
C      SUCH A CONTOUR WILL CONTAIN MORE THAN ONE CONTOUR SEGMENT.         CNTR0300
C                                                                         CNTR0310
C          A CONTOUR SEGMENT IS EITHER A CLOSED FIGURE (INTERIOR          CNTR0320
C      CONTOUR SEGMENT) OR A CURVE, EACH END OF WHICH TERMINATES AT       CNTR0330
C      A MARGIN OF THE CONTOUR GRAPH (EXTERIOR CONTOUR SEGMENT). IT       CNTR0340
C      REPRESENTS, IN GENERAL, ONLY A PART OF THE LOCUS ON THE XY         CNTR0350
C      PLANE OF ALL Z EQUAL TO SOME VALUE. FOR EXAMPLE, IF Z=F(X,Y)       CNTR0360
C      REPRESENTS A RANGE OF THREE MOUNTAINS OF EQUAL ALTITUDE, EACH      CNTR0370
C      A CIRCULAR CONE, THEN EVERY CONTOUR SEGMENT WOULD APPEAR AS A      CNTR0380
C      CIRCLE. THREE CIRCLES, ONE FOR EACH MOUNTAIN, WOULD MAKE UP        CNTR0390
C      ONE CONTOUR OR CONTOUR LEVEL.  THE COMPLETE GRAPH WOULD APPEAR     CNTR0400
C      AS THREE NESTS OF CONCENTRIC CIRCLES.                              CNTR0410
C                                                                         CNTR0420
C      B. CALLING STATEMENT                                               CNTR0430
C                                                                         CNTR0440
C          CALL CONTUR(AM,M,N,MX,CL,NL,TITLE,IW,IH,LTG)                   CNTR0450
C                                                                         CNTR0460
C          WHERE:                                                         CNTR0470
C                                                                         CNTR0480
C          1) AM IS THE REAL*4 DATA MATRIX TO BE CONTOURED AND PLOTTED.   CNTR0490
C                IT IS A TWO-DIMENSIONAL ARRAY WHOSE ROW DIMENSION IS     CNTR0500
C                PASSED TO CONTUR AS THE PARAMETER MX.                    CNTR0510
C                                                                         CNTR0520
C          2) M IS AN INTEGER VARIABLE WHICH IS THE NUMBER OF ROWS IN THE CNTR0530
C             DATA MATRIX TO BE CONTOURED.                                CNTR0540
C                                                                         CNTR0550
C          3) N IS AN INTEGER VARIABLE WHICH IS THE NUMBER OF COLUMNS IN  CNTR0560
C             THE DATA MATRIX TO BE CONTOURED.                            CNTR0570
C                                                                         CNTR0580
C          4) MX IS AN INTEGER WHICH SPECIFIES THE ROW DIMENSION OF AM.   CNTR0590
C                THIS IS THE EXACT NUMBER APPEARING IN THE DIMENSION      CNTR0600
C                STATEMENT DEFINING AM.                                   CNTR0610
C                                                                         CNTR0620
C          5) CL IS A REAL*4 ARRAY CONTAINING THE CONTOUR LEVELS TO BE    CNTR0630
C                PLOTTED DIMENSIONED AT LEAST CL(|NL|).                   CNTR0640
```

```
C                                                                       CNTR0650
C     6) NL IS AN INTEGER VARIABLE WHOSE ABSOLUTE VALUE IS THE          CNTR0660
C        NUMBER OF CONTOUR LEVELS TO BE FOUND FOR THIS GRAPH. IF        CNTR0670
C        NL IS NEGATIVE THEN CONTUR SCANS THE DATA MATRIX FOR THE       CNTR0680
C        LOWEST AND HIGHEST LEVELS TO DETERMINE THE DELTA VALUE         CNTR0690
C        FOR PRODUCING |NL| LEVELS OF CONTOURING. IF NL IS              CNTR0700
C        POSITIVE, THEN CONTUR ASSUMES THE CL ARRAY CONTAINS            CNTR0710
C        NL ENTRIES OF CONTOUR LEVELS TO BE PLOTTED.                    CNTR0720
C                                                                       CNTR0730
C     7) TITLE IS A REAL*8 ARRAY DIMENSIONED TITLE(12). CONTUR PICKS    CNTR0740
C           UP THE MAIN TITLE FROM TITLE(1)-TITLE(6). THE               CNTR0750
C           SUBTITLE IS PICKED UP FROM TITLE(7)-TITLE(12). IF THE       CNTR0760
C           USER WISHES HIS TITLES TO BE CENTERED THEN HE MUST          CNTR0770
C           STORE HIS CHARACTERS IN TITLE ACCORDINGLY.                  CNTR0780
C                                                                       CNTR0790
C     8) IW IS AN INTEGER VARIABLE SPECIFYING THE WIDTH OF THE          CNTR0800
C        CONTOUR GRAPH IN INCHES (1 IW 9).                              CNTR0810
C                                                                       CNTR0820
C     9) IH IS AN INTEGER VARIABLE SPECIFYING THE HEIGHT OF THE         CNTR0830
C        CONTOUR GRAPH IN INCHES (IH>0).                                CNTR0840
C                                                                       CNTR0850
C    10) LTG IS A LOGICAL*1 ARRAY DIMENSIONED LTG(3) WHERE:             CNTR0860
C                                                                       CNTR0870
C        A) LTG(1)=.TRUE. REQUESTS LABELLING OF ALL EXTERIOR CONTOUR    CNTR0880
C                 SEGMENTS AND THOSE INTERIOR CONTOUR SEGMENTS          CNTR0890
C                 WHICH REPRESENT A LOCAL MAXIMUM.                      CNTR0900
C                                                                       CNTR0910
C           LTG(1)=.FALSE. OMITS LABELLING OPTION.                      CNTR0920
C                                                                       CNTR0930
C        B) LTG(2)=.TRUE. REQUESTS TIC MARKS AND CORRESPONDING SCALE    CNTR0940
C                 VALUES ONE INCH APART ON THE EXTERIOR FRAME OF        CNTR0950
C                 THE CONTOUR GRAPH.                                    CNTR0960
C                                                                       CNTR0970
C           LTG(2)=.FALSE. OMITS "TIC" OPTION.                          CNTR0980
C                                                                       CNTR0990
C        C) LTG(3)=.TRUE. REQUESTS CONTUR TO SUPERIMPOSE A ONE INCH     CNTR1000
C                 BY ONE INCH STRAIGHT LINE GRID ON THE CONTOUR         CNTR1010
C                 GRAPH.                                                CNTR1020
C                                                                       CNTR1030
C           LTG(3)=.FALSE. OMITS THE GRID OPTION.                       CNTR1040
C                                                                       CNTR1050
C  C. ERROR MESSAGES                                                    CNTR1060
C                                                                       CNTR1070
C     1) "CONTOUR LINE AT LEVEL (E12.5) WAS TERMINATED AT X=(E12.5),    CNTR1080
C        Y=(E12.5) BECAUSE IT CONTAINED MORE THAN 1799 POINTS." (THE    CNTR1090
C        APPROPRIATE VALUES ARE PRINTED WHERE FORMATS ARE INDICATED)    CNTR1100
C                                                                       CNTR1110
C     2) "WIDTH OF CONTOUR GRAPH ILLLEGAL."  (IW.LE.0)                  CNTR1120
```

```
C                                                                       CNTR1130
C        3) "HEIGHT OF CONTOUR GRAPH ILLEGAL." (IH.LE.0)                CNTR1140
C                                                                       CNTR1150
C        4) "DATA MATRIX HAS ONLY ONE LEVEL."  AFTER SCANNING THE DATA  CNTR1160
C           MATRIX FOR THE HIGHEST AND LOWEST LEVEL ENTRIES, CONTUR     CNTR1170
C           FINDS THAT THE MATRIX TO BE CONTOURED CONTAINS THE SAME     CNTR1180
C           LEVEL VALUE IN ALL ENTRIES.                                 CNTR1190
C                                                                       CNTR1200
C        5) "NUMBER OF LEVELS REQUESTED=0."  (NL=0)                     CNTR1210
C                                                                       CNTR1220
C        6) "NO GRAPH WILL BE PRODUCED." THIS MESSAGE FOLLOWS 2)-5)     CNTR1230
C           ABOVE.                                                      CNTR1240
C                                                                       CNTR1250
C        7) " IW PARAMETER GREATER THAN 9. CONTUR WILL SET IW=9."       CNTR1260
C           PROCESSING CONTINUES WITH IW=9.                             CNTR1270
C                                                                       CNTR1280
C     4. SUBROUTINES USED                                               CNTR1290
C                                                                       CNTR1300
C         THE SUBROUTINES DESCRIBED IN "THE PLOTTING PACKAGE FOR NPGS   CNTR1310
C     IBM 360/67" ARE USED TO DRAW THE CONTOUR GRAPH.  SUBROUTINES      CNTR1320
C     WHICH FIND THE POINTS TO BE PLOTTED WERE PROVIDED BY              CNTR1330
C     DR. M.O. DAYHOFF OF THE NATIONAL BIOMEDICAL RESEARCH FOUNDATION,  CNTR1340
C     SILVER SPRING, MARYLAND.                                          CNTR1350
C                                                                       CNTR1360
C         SUBROUTINE PLOTT DRAWS THE INTERIOR AND EXTERIOR CONTOUR      CNTR1370
C     SEGMENTS AND LABELS (IF REQUESTED) THE EXTERIOR CONTOUR           CNTR1380
C     SEGMENTS.  CONTUR DOES THE LABELLING OF INTERIOR SEGMENTS         CNTR1390
C     WHICH REPRESENT LOCAL MAXIMA.                                     CNTR1400
C                                                                       CNTR1410
C     5. RESTRICTIONS                                                   CNTR1420
C                                                                       CNTR1430
C        A) INPUT                                                       CNTR1440
C                                                                       CNTR1450
C           1) 1.LE.IW.LE.9                                             CNTR1460
C                                                                       CNTR1470
C           2) IH.GE.1                                                  CNTR1480
C                                                                       CNTR1490
C           3) WHENEVER NL IS POSITIVE THE DATA MATRIX IS NOT SCANNED FOR CNTR1500
C              THE MINIMUM AND MAXIMUM LEVELS; CONSEQUENTLY THE USER IS  CNTR1510
C              CAUTIONED AGAINST THE POSSIBILITY THAT NO GRAPH WILL BE   CNTR1520
C              PRODUCED IF                                               CNTR1530
C                                                                       CNTR1540
C              A) ALL ENTRIES IN THE DATA MATRIX ARE EQUAL               CNTR1550
C                                                                       CNTR1560
C              B) ENTRIES IN THE DATA MATRIX ARE ALL BELOW OR ALL ABOVE  CNTR1570
C                 THE REQUESTED LEVELS.                                 CNTR1580
C                                                                       CNTR1590
C        B) OUTPUT                                                      CNTR1600
```

```
C                                                                       CNTR1610
C        1) ACCURACY IS LIMITED BY THE LINEAR INTERPOLATION PROCESS     CNTR1620
C           USED IN FINDING THE VALUES OF THE POINTS ALONG CONTOUR      CNTR1630
C           SEGMENTS. IT IS ALSO LIMITED BY THE RESOLUTION OF THE       CNTR1640
C           OFFLINE PLOTTER; I.E., .01 INCH IN BOTH THE X AND Y         CNTR1650
C           DIRECTIONS.                                                 CNTR1660
C                                                                       CNTR1670
C        2) THE X-SCALE VALUES ALONG THE TIC MARKS ARE 1.0.LE.X.LE.N.   CNTR1680
C           THE Y-SCALE VALUES ARE 1.0.LE.Y.LE.M.                       CNTR1690
C                                                                       CNTR1700
C        3) ONLY UP TO 20 INTERIOR SEGMENTS WHICH REPRESENT             CNTR1710
C           MAXIMA ARE LABELLED.                                        CNTR1720
C                                                                       CNTR1730
C   6. STORAGE REQUIREMENTS                                             CNTR1740
C                                                                       CNTR1750
C        CONTUR AND ALL ITS SUBORDINATE SUBROUTINES REQUIRE 54,570      CNTR1760
C        BYTES OF CORE.                                                 CNTR1770
C                                                                       CNTR1780
C   7. HARDWARE CONFIGURATION                                           CNTR1790
C                                                                       CNTR1800
C      A) IBM 360/67EQUIPMENT CONFIGURATION (INCLUDING SPOOLING DISKS,  CNTR1810
C         ETC.) WITH APPROPRIATE SYSOUT AND SYSIN UNITS.                CNTR1820
C                                                                       CNTR1830
C      B) CALCOMP MODEL 765 PLOTTER                                     CNTR1840
C                                                                       CNTR1850
C   8. REFERENCES                                                       CNTR1860
C                                                                       CNTR1870
C      A) DAYHOFF,M.O.,"A CONTOUR-MAP PROGRAM FOR X-RAY CRYSTALLOGRAPHY"CNTR1880
C         COMMUNICATIONS OF ASSOCIATION FOR COMPUTING MACHINERY,        CNTR1890
C         OCTOBER 1963                                                  CNTR1900
C                                                                       CNTR1910
C      B) JOHNSON, PATRICIA C.,"PLOTTING PACKAGE FOR NPGS IBM 360/67",  CNTR1920
C         TECHNICAL NOTE NO. 0211-03, FEBRUARY 1969                     CNTR1930
C                                                                       CNTR1940
C      C) HILLEARY, R.R., "J7-NPG-CONTOUR (F-60)", SEPTEMBER 1964.      CNTR1950
C                                                                       CNTR1960
C   9. MATHEMATICAL METHOD                                              CNTR1970
C                                                                       CNTR1980
C        THE ALGORITHM USED IN FINDING THE CONTOURS IS DEFINED IN       CNTR1990
C        REFERENCE A).                                                  CNTR2000
C.......................................................................CNTR2010
      REAL*8 TITLE(1)                                                   CNTR2030
      REAL*8 WIDTH/'WIDTH'/,HEIGHT/'HEIGHT'/,WHICH                      CNTR2040
      DIMENSION AM(MX,1),CL(1)                                          CNTR2050
      DIMENSION           REC(900),  X(1800), Y(1800)                   CNTR2060
      DIMENSION IPT(3,3),INX(8),INY(8)                                  CNTR2070
      COMMON /DAYHOF/ MT,NT,NI,IX,IY,IDX,IDY,ISS,IT,IV,NP,NQ,JT,        CNTR2080
     1                PY,REC,CV,    IPT,INX,INY,DL,RA,THE               CNTR2090
```

```
      COMMON /INTFAC/ X,Y                                                    CNTR2100
      DIMENSION DITSX(5),DITSY(5)                                            CNTR2110
      LOGICAL*1 LTG(1),MINUS,LABL                                            CNTR2120
      COMMON/TABL/ TABC(20,6),JC                                             CNTR2130
      COMMON/DITS/XMIN,YMIN,SLOPEX,SLOPEY,DITSDX,DITSDY,IDIR,LABL,MINUS      CNTR2140
      JC=0                                                                   CNTR2150
      LABL=LTG(1)                                                            CNTR2160
C     CHECK IW PARAMETER                                                     CNTR2170
      WHICH=WIDTH                                                            CNTR2180
      IF(IW) 1,1,2                                                           CNTR2190
    1 WRITE(6,60) WHICH                                                      CNTR2200
   60 FORMAT('0',T7,A8,'OF CONTOUR GRAPH ILLEGAL.')                          CNTR2210
   71 WRITE(6,64)                                                            CNTR2220
   64 FORMAT('0',T7,'NO GRAPH WILL BE PRODUCED.')                            CNTR2230
      RETURN                                                                 CNTR2240
C     CHECK IF IW IS TOO WIDE                                                CNTR2250
    2 IF(IW-9) 3,3,40                                                        CNTR2260
   40 WRITE(6,61)                                                            CNTR2270
   61 FORMAT('0',T7,'IW PARAMETER GREATER THAN 9. CONTUR WILL SET IW=9.'     CNTR2280
     1)                                                                      CNTR2290
      IW=9                                                                   CNTR2300
C     NOW CHECK IH PARAMETER                                                 CNTR2310
    3 IF(IH) 4,4,5                                                           CNTR2320
    4 WHICH=HEIGHT                                                           CNTR2330
      GO TO 1                                                                CNTR2340
    5 DITSDX=(N-1.0)/IW                                                      CNTR2350
      DITSDY=(-1.0+M)/IH                                                     CNTR2360
      XMIN=1.0                                                               CNTR2370
      YMIN=-M                                                                CNTR2380
      SLOPEX=1.0/DITSDX                                                      CNTR2390
      SLOPEY=1.0/DITSDY                                                      CNTR2400
      DITSX(1)=1.0                                                           CNTR2410
      DITSX(4)=1.0                                                           CNTR2420
      DITSX(5)=1.0                                                           CNTR2430
      DITSX(2)=N                                                             CNTR2440
      DITSX(3)=N                                                             CNTR2450
      DITSY(1)=-1.0                                                          CNTR2460
      DITSY(2)=-1.0                                                          CNTR2470
           DITSY(5)=-1.0                                                     CNTR2480
      DITSY(3)=-M                                                            CNTR2490
      DITSY(4)=-M                                                            CNTR2500
      DO 2011 I=1,5                                                          CNTR2510
      DITSX(I)=SLOPEX*(DITSX(I)-XMIN)                                        CNTR2520
 2011 DITSY(I)=SLOPEY*(DITSY(I)-YMIN)                                        CNTR2530
      STARTP=(9.0-IW)/2.0                                                    CNTR2540
      CALL PLOTS                                                             CNTR2550
      CALL PLOT(STARTP,0.0,-3)                                               CNTR2560
      CALL LINE(DITSX,DITSY,5,1,1)                                           CNTR2570
```

```
      DITSX(1)=DITSX(1)-.5                                              CNTR2580
      DITSX(5)=DITSX(1)                                                 CNTR2590
      DITSX(4)=DITSX(4)-.5                                              CNTR2600
      DITSX(2)=DITSX(2)+.5                                              CNTR2610
      DITSX(3)=DITSX(3)+.5                                              CNTR2620
      DITSY(1)=DITSY(1)+.5                                              CNTR2630
      DITSY(5)=DITSY(1)                                                 CNTR2640
      DITSY(2)=DITSY(2)+.5                                              CNTR2650
      DITSY(3)=DITSY(3)-.5                                              CNTR2660
      DITSY(4)=DITSY(4)-.5                                              CNTR2670
      CALL LINE(DITSX,DITSY,5,1,1)                                      CNTR2680
      SLOPEX=1.0/DITSDX                                                 CNTR2690
      SLOPEY=1.0/DITSDY                                                 CNTR2700
      IENDX=SLOPEX*N+1                                                  CNTR2710
      IENDY=SLOPEY*M+1                                                  CNTR2720
      IF(.NOT.LTG(2)) GO TO 34                                          CNTR2730
C     DRAW TIC MARKS ON OUTER FRAME                                     CNTR2740
C     START ON LEFT EDGE GOING DOWNWARD                                 CNTR2750
      IFLAG=0                                                           CNTR2760
      ZINGX=-.1                                                         CNTR2770
      ZINGY=0.0                                                         CNTR2780
      ZX=0.0                                                            CNTR2790
      ZY=-1.0                                                           CNTR2800
      CX=DITSX(1)                                                       CNTR2810
      CY=DITSY(1)-.5                                                    CNTR2820
      IEND=IENDY                                                        CNTR2830
 2222 IFLAG=IFLAG+1                                                     CNTR2840
      DO 2022 I=1,IEND                                                  CNTR2850
      CALL PLOT(CX,CY,3)                                                CNTR2860
      COORDX=CX+ZINGX                                                   CNTR2870
      COORDY=CY+ZINGY                                                   CNTR2880
      CALL PLOT(COORDX,COORDY,2)                                        CNTR2890
      CX=CX+ZX                                                          CNTR2900
 2022 CY=CY+ZY                                                          CNTR2910
      GO TO (21,22,23,24),IFLAG                                         CNTR2920
C     NOW DO THE RIGHT EDGE GOING DOWNWARD                              CNTR2930
   21 ZINGX=.1                                                          CNTR2940
      CX=DITSX(2)                                                       CNTR2950
      CY=DITSY(2)-.5                                                    CNTR2960
      GO TO 2222                                                        CNTR2970
C     NOW DO TOP EDGE                                                   CNTR2980
   22 ZINGX=0.0                                                         CNTR2990
      ZINGY=.1                                                          CNTR3000
      ZX=1.0                                                            CNTR3010
      ZY=0.0                                                            CNTR3020
      CX=DITSX(1)+.5                                                    CNTR3030
      CY=DITSY(1)                                                       CNTR3040
      IEND=IENDX                                                        CNTR3050
```

```
      GO TO 2222                                                        CNTR3060
C     NOW DO THE BOTTOM EDGE                                            CNTR3070
   23 ZINGY=-.1                                                         CNTR3080
      CX=DITSX(4)+.5                                                    CNTR3090
      CY=DITSY(4)                                                       CNTR3100
      ZINGY=-.1                                                         CNTR3110
      GO TO 2222                                                        CNTR3120
C     NOW LABEL TIC MARKS                                               CNTR3130
C     DO X-DIRECTION FIRST, TOP EDGE                                    CNTR3140
C     POSITION PEN                                                      CNTR3150
   24 DELTAX=DITSDX                                                     CNTR3160
      IFLAG=0                                                           CNTR3170
      ZX=1.0                                                            CNTR3180
      ZY=0.0                                                            CNTR3190
      CX=DITSX(1)+.35                                                   CNTR3200
      CY=DITSY(1)+.12                                                   CNTR3210
 3033 IFLAG=IFLAG+1                                                     CNTR3220
      XZERO=1.0                                                         CNTR3230
      DO 3333 I=1,IEND                                                  CNTR3240
      CALL NUMBER(CX,CY,.07,XZERO,0.0,1)                                CNTR3250
      CX=CX+ZX                                                          CNTR3260
      CY=CY+ZY                                                          CNTR3270
 3333 XZERO=XZERO+DELTAX                                                CNTR3280
      GO TO (31,32,33,34),IFLAG                                         CNTR3290
C     LABEL BOTTOM EDGE TIC MARKS                                       CNTR3300
   31 CX=DITSX(4)+.35                                                   CNTR3310
      CY=DITSY(4)-.19                                                   CNTR3320
      GO TO 3033                                                        CNTR3330
C     LABEL LEFT EDGE OF TIC MARKS                                      CNTR3340
   32 CX=DITSX(4)-.4                                                    CNTR3350
      CY=DITSY(4)+.46                                                   CNTR3360
      DELTAX=DITSDY                                                     CNTR3370
      IEND=IENDY                                                        CNTR3380
      ZX=0.0                                                            CNTR3390
      ZY=1.0                                                            CNTR3400
      GO TO 3033                                                        CNTR3410
C     NOW LABEL RIGHT EDGE TIC MARKS                                    CNTR3420
   33 CX=DITSX(3)+.12                                                   CNTR3430
      CY=DITSY(3)+.46                                                   CNTR3440
      GO TO 3033                                                        CNTR3450
C     CHECK IF GRID DESIRED                                             CNTR3460
   34 IF(.NOT.LTG(3)) GO TO 35                                          CNTR3470
C     DRAW INCH BY INCH GRID                                            CNTR3480
      IEND=IENDX-2                                                      CNTR3490
C     POSITION PEN                                                      CNTR3500
      IFLAG=0                                                           CNTR3510
      CX=DITSX(1)+.5                                                    CNTR3520
      CY=DITSY(1)-.5                                                    CNTR3530
```

```
      COORDX=0.0                                                        CNTR3540
      COORDY=-IH                                                        CNTR3550
      DX=1.0                                                            CNTR3560
      DY=0.0                                                            CNTR3570
 4044 DO 4444 I=1,IEND                                                  CNTR3580
      CX=CX+DX                                                          CNTR3590
      CY=CY+DY                                                          CNTR3600
      CALL PLOT(CX,CY,3)                                                CNTR3610
      ZX=CX+COORDX                                                      CNTR3620
      ZY=CY+COORDY                                                      CNTR3630
 4444 CALL PLOT(ZX,ZY,2)                                                CNTR3640
      IF(IFLAG) 35,42,35                                                CNTR3650
   42 IFLAG=1                                                           CNTR3660
      IEND=IENDY-2                                                      CNTR3670
      CY=DITSY(4)+.5                                                    CNTR3680
      CX=DITSX(4)+.5                                                    CNTR3690
      COORDX=IW                                                         CNTR3700
      COORDY=0.0                                                        CNTR3710
      DX=0.0                                                            CNTR3720
      DY=1.0                                                            CNTR3730
      GO TO 4044                                                        CNTR3740
   35 CONTINUE                                                          CNTR3750
      NLEV=NL                                                           CNTR3760
C     CHECK IF CONTUR IS TC  COMPUTE LEVELS                             CNTR3770
      IF(NLEV) 6,7,11                                                   CNTR3780
C     NL NEGATIVE  MEANS CONTOUR MUST COMPUTE LEVELS                    CNTR3790
C     SCAN MATRIX FOR LOWEST AND HIGHEST LEVELS                         CNTR3800
    6 NLEV=-NLEV                                                        CNTR3810
      FMIN=AM(1,1)                                                      CNTR3820
      FMAX=FMIN                                                         CNTR3830
      DO 9 I=1,M                                                        CNTR3840
      DO 9 J=1,N                                                        CNTR3850
      IF(AM(I,J).LT.FMIN) FMIN=AM(I,J)                                  CNTR3860
      IF(AM(I,J).GT.FMAX) FMAX=AM(I,J)                                  CNTR3870
    9 CONTINUE                                                          CNTR3880
C     CHECK IF MIN AND MAX ARE EQUAL                                    CNTR3890
      IF(FMIN.GE.FMAX) GO TO 300                                        CNTR3900
      DELTAL=(FMAX-FMIN)/(NLEV-1)                                       CNTR3910
C     SET UP CL ARRAY                                                   CNTR3920
      DO 10 I=1,NLEV                                                    CNTR3930
   10 CL(I)=FMIN+(I-1)*DELTAL                                           CNTR3940
   11 DO 20 I=1,NLEV                                                    CNTR3950
   20 CALL SCAN(AM,M,N,MX,CL(I))                                        CNTR3960
      IF(.NOT.LABL) GO TO 778                                           CNTR3970
      IF(JC.EQ.0) GO TO 778                                             CNTR3980
      DO 777 I=1,JC                                                     CNTR3990
      COORDX=TABC(I,4)                                                  CNTR4000
      COORDY=TABC(I,5)                                                  CNTR4010
```

```
      CLEV=TABC(I,6)                                                    CNTR4020
      CALL NUMBER(COORDX,COORDY,.07,CLEV,0.0,1)                         CNTR4030
  777 CONTINUE                                                          CNTR4040
  778 CALL SYMBOL(-STARTP,IH+1.0,.21,TITLE(7),0.0,48)                   CNTR4050
      CALL SYMBOL(-STARTP,IH+1.5,.21,TITLE(1),0.0,48)                   CNTR4060
      CALL PLOT(-STARTP,IH+5.0,-3)                                      CNTR4070
      CALL PLOTE                                                        CNTR4080
      RETURN                                                            CNTR4090
  300 WRITE(6,62) FMIN                                                  CNTR4100
   62 FORMAT('0',T7,'DATA MATRIX HAS ONLY ONE LEVEL=',E15.3)            CNTR4110
      GO TO 71                                                          CNTR4120
    7 WRITE(6,63)                                                       CNTR4130
   63 FORMAT('0',T7,'NUMBER OF LEVELS REQUESTED=0.')                    CNTR4140
      GO TO 71                                                          CNTR4150
      END                                                               CNTR4160

      SUBROUTINE SCAN(AM,M,N,MX,CL)                                     CNTR4170
C THIS PROGRAM IS WRITTEN BY M.O.DAYHOFF                                CNTR4180
      DIMENSION AM(MX,1 ),REC(900),  X(1800), Y(1800)                   CNTR4190
      DIMENSION IPT(3,3),INX(8),INY(8)                                  CNTR4200
      COMMON /DAYHOF/ MT,NT,NI,IX,IY,IDX,IDY,ISS,IT,IV,NP,NQ,JT,        CNTR4210
     1                PY,REC,CV,    IPT,INX,INY,DL,RA,THE               CNTR4220
      COMMON /INTFAC/ X,Y                                               CNTR4230
      LOGICAL *1 LABL,MINUS                                             CNTR4240
      COMMON/DITS/XMIN,YMIN,SLOPEX,SLOPEY,DITSDX,DITSDY,IDIR,LABL,MINUS CNTR4250
      D=0.                                                              CNTR4260
      R=1.                                                              CNTR4270
      TH = 1.570796                                                     CNTR4280
      NP=0                                                              CNTR4290
      DL=D                                                              CNTR4300
      RA=R                                                              CNTR4310
      THE=TH                                                            CNTR4320
      MT=N                                                              CNTR4330
      NT=M                                                              CNTR4340
      CV=CL                                                             CNTR4350
      IF(IZW-120631) 1,3,1                                              CNTR4360
    1 IPT(1,1)=8                                                        CNTR4370
      IPT(1,2)=1                                                        CNTR4380
      IPT(1,3)=2                                                        CNTR4390
      IPT(2,1)=7                                                        CNTR4400
      IPT(2,3)=3                                                        CNTR4410
      IPT(3,1)=6                                                        CNTR4420
      IPT(3,2)=5                                                        CNTR4430
      IPT(3,3)=4                                                        CNTR4440
      INX(1)=-1                                                         CNTR4450
      INX(2)=-1                                                         CNTR4460
      INX(3)=0                                                          CNTR4470
```

```
      INX(4)=1                                                          CNTR4480
      INX(5)=1                                                          CNTR4490
      INX(6)=1                                                          CNTR4500
      INX(7)=0                                                          CNTR4510
      INX(8)=-1                                                         CNTR4520
      INY(1)=0                                                          CNTR4530
      INY(2)=1                                                          CNTR4540
      INY(3)=+1                                                         CNTR4550
      INY(4)=+1                                                         CNTR4560
      INY(5)=0                                                          CNTR4570
      INY(6)=-1                                                         CNTR4580
      INY(7)=-1                                                         CNTR4590
      INY(8)=-1                                                         CNTR4600
      IZW=120631                                                        CNTR4610
    3 XT=MT                                                             CNTR4620
      DO 58 J=1,900                                                     CNTR4630
   58 REC(J)=0                                                          CNTR4640
      ISS=0                                                             CNTR4650
    2 MT1=MT-1                                                          CNTR4660
      IDIR=1                                                            CNTR4670
      DO 110 I=1,MT1                                                    CNTR4680
      IF(AM(1,I)-CV) 55,110,110                                         CNTR4690
   55 IF(AM(1,I+1)-CV) 110,57,57                                        CNTR4700
   57 IX=I+1                                                            CNTR4710
      IY=1                                                              CNTR4720
      IDX=-1                                                            CNTR4730
      IDY=0                                                             CNTR4740
      CALL TRACE (AM,MX)                                                CNTR4750
  110 CONTINUE                                                          CNTR4760
      NT1=NT-1                                                          CNTR4770
      IDIR=2                                                            CNTR4780
      DO 20 I=1,NT1                                                     CNTR4790
      IF(AM(I,MT)-CV) 15,20,20                                          CNTR4800
   15 IF(AM(I+1,MT)-CV) 20,17,17                                        CNTR4810
   17 IX=MT                                                             CNTR4820
      IY=I+1                                                            CNTR4830
      IDX=0                                                             CNTR4840
      IDY=-1                                                            CNTR4850
      CALL TRACE (AM,MX)                                                CNTR4860
   20 CONTINUE                                                          CNTR4870
      IDIR=3                                                            CNTR4880
   22 DO 30 I=1,MT1                                                     CNTR4890
      MT2=MT+1-I                                                        CNTR4900
      IF(AM(NT,MT2)-CV) 25,30,30                                        CNTR4910
   25 IF(AM(NT,MT2-1)-CV) 30,27,27                                      CNTR4920
   27 IX=MT2-1                                                          CNTR4930
      IY=NT                                                             CNTR4940
      IDX=1                                                             CNTR4950
```

```
      IDY=0                                                              CNTR4960
      CALL TRACE (AM,MX)                                                 CNTR4970
   30 CONTINUE                                                           CNTR4980
      IDIR=4                                                             CNTR4990
      DO 40 I=1,NT1                                                      CNTR5000
      NT2=NT+1-I                                                         CNTR5010
      IF(AM(NT2,1)-CV) 35,40,40                                          CNTR5020
   35 IF(AM(NT2-1,1)-CV) 40,37,37                                        CNTR5030
   37 IX=1                                                               CNTR5040
      IY=NT2-1                                                           CNTR5050
      IDX=0                                                              CNTR5060
      IDY=1                                                              CNTR5070
      CALL TRACE (AM,MX)                                                 CNTR5080
   40 CONTINUE                                                           CNTR5090
      IDIR=5                                                             CNTR5100
      ISS=1                                                              CNTR5110
      NT1=NT-1                                                           CNTR5120
      MT1=MT-1                                                           CNTR5130
      DO 10 J=2,NT1                                                      CNTR5140
      DO 10 I=1,MT1                                                      CNTR5150
      IF(AM(J,I)-CV)5,10,10                                              CNTR5160
    5 IF(AM(J,I+1)-CV) 10,7,7                                            CNTR5170
    7 COM=100*(I+1)+J                                                    CNTR5180
      IF (NP) 12,11,12                                                   CNTR5190
   12 DO 9 ID= 1,NP                                                      CNTR5200
      IF (REC(ID)-COM) 9,10,9                                            CNTR5210
    9 CONTINUE                                                           CNTR5220
   11 IX= I+1                                                            CNTR5230
      IY=J                                                               CNTR5240
      IDX=-1                                                             CNTR5250
      IDY=0                                                              CNTR5260
      CALL TRACE (AM,MX)                                                 CNTR5270
   10 CONTINUE                                                           CNTR5280
      RETURN                                                             CNTR5290
      END                                                                CNTR5300

      SUBROUTINE TRACE (AM,MY)                                           CNTR5310
      DIMENSION AM(MY,1 ),REC(900),  X(1800), Y(1800)                    CNTR5320
      DIMENSION IPT(3,3), INX(8),INY(8)                                  CNTR5330
      COMMON /DAYHOF/ MT,NT,NI,IX,IY,IDX,IDY,ISS,IT,IV,NP,N ,JT,         CNTR5340
     1                PY,REC,CV,    IPT,INX,INY,DL,RA,THE                CNTR5350
      COMMON /INTFAC/ X,Y                                                CNTR5360
      PY=0.0                                                             CNTR5370
      RC=  COS (THE)*RA                                                  CNTR5380
      RS=  SIN (THE)*RA                                                  CNTR5390
  501 JT=0                                                               CNTR5400
      N=0                                                                CNTR5410
```

```
      IX0=IX                                                                CNTR5420
      IY0=IY                                                                CNTR5430
      ISX=IDX+2                                                             CNTR5440
      ISY=IDY+2                                                             CNTR5450
      IS=IPT(ISX,ISY)                                                       CNTR5460
      JTB=0                                                                 CNTR5470
      IS0=IS                                                                CNTR5480
      IF(IS0-8)18,18,17                                                     CNTR5490
   17 IS0=IS0-8                                                             CNTR5500
   18 IT=0                                                                  CNTR5510
    5 CONTINUE                                                              CNTR5520
      CALL CALC  (AM,MY)                                                    CNTR5530
      NZ=N                                                                  CNTR5540
      N=NZ                                                                  CNTR5550
      IF (IT+JT-1) 49,49,47                                                 CNTR5560
   47 XS=X(N-1)                                                             CNTR5570
      YS=Y(N-1)                                                             CNTR5580
      X(N-1)=X(N)                                                           CNTR5590
      Y(N-1)=Y(N)                                                           CNTR5600
      X(N)=XS                                                               CNTR5610
      Y(N)=YS                                                               CNTR5620
   49 IS=IS+1                                                               CNTR5630
      JT=IT                                                                 CNTR5640
    9 IF (IS-9) 8,7,7                                                       CNTR5650
    7 IS=IS-8                                                               CNTR5660
    8 IDX=INX(IS)                                                           CNTR5670
      IDY=INY(IS)                                                           CNTR5680
      IX2=IX+IDX                                                            CNTR5690
      IY2=IY+IDY                                                            CNTR5700
      JTB=JTB+1                                                             CNTR5710
      IF (JTB-1799) 51,51,308                                               CNTR5720
  308 PRINT              103,CV,X(N),Y(N)                                   CNTR5730
  103 FORMAT(1H0,23HA CONTOUR LINE AT LEVEL,E12.5,21H WAS TERMINATED AT     CNTR5740
     1X=,E12.5,3H Y=,E12.5/                                                 CNTR5750
     2 48H BECAUSE IT CONTAINED MORE THAN 1799 PLOT POINTS  )               CNTR5760
      RETURN                                                                CNTR5770
C                                                                           CNTR5780
C    SHOULD TEST HERE FOR MAXIMUM NUMBER OF PLOTTABLE POINTS IN SEGMENT.    CNTR5790
   51 CONTINUE                                                              CNTR5800
      IF (ISS) 10,10,20                                                     CNTR5810
   20 IF(IX-IX0) 12,21,12                                                   CNTR5820
   21 IF(IY-IY0) 12,22,12                                                   CNTR5830
   22 IF(IS-IS0) 12,23,12                                                   CNTR5840
   23 CONTINUE                                                              CNTR5850
      CALL CALC  (AM,MY)                                                    CNTR5860
      GO TO 73                                                              CNTR5870
   10 IF(IX2) 13,50,13                                                      CNTR5880
   13 IF (IX2-MT) 19,19,50                                                  CNTR5890
```

```
 19 IF (IY2) 11,50,11                                                    CNTR5900
 11 IF (IY2-NT) 12,12,50                                                 CNTR5910
 12 IF(CV-AM(IY2,IX2)) 206,206,5                                         CNTR5920
206 IF (IDX**2+IDY**2-1) 213,6,213                                       CNTR5930
213 DCP=(AM(IY,IX)+AM(IY,IX2)+AM(IY2,IX)+AM(IY2,IX2))/4.0                CNTR5940
    IF (DCP-CV) 5,217,217                                                CNTR5950
217 IF (INX(IS-1)) 214,215,214                                           CNTR5960
214 IX=IX+IDX                                                            CNTR5970
    IDX=-IDX                                                             CNTR5980
    PY=2.0                                                               CNTR5990
    CALL CALC  (AM,MY)                                                   CNTR6000
    IX=IX+IDX                                                            CNTR6010
    GO TO 6                                                              CNTR6020
215 IY=IY+IDY                                                            CNTR6030
    IDY=-IDY                                                             CNTR6040
    PY=2.0                                                               CNTR6050
    CALL CALC  (AM,MY)                                                   CNTR6060
    IY=IY+IDY                                                            CNTR6070
  6 IF(AM(IY,IX-1)-CV) 306,16,16                                         CNTR6080
306 NP=NP+1                                                              CNTR6090
    REC(NP)=100*IX+IY                                                    CNTR6100
 16 IS=IS+5                                                              CNTR6110
    IX=IX2                                                               CNTR6120
    IY=IY2                                                               CNTR6130
    GO TO 9                                                              CNTR6140
 50 XT=MT                                                                CNTR6150
    IF(AM(IY,IX-1)-CV) 307,73,73                                         CNTR6160
307 NP=NP+1                                                              CNTR6170
    REC(NP)=100*IX+IY                                                    CNTR6180
 73 DO 74 I=1,N                                                          CNTR6190
    X(I)=X(I)+RC*Y(I)                                                    CNTR6200
 74 Y(I)=RS*Y(I)                                                         CNTR6210
    CALL PLOTT(N,CV)                                                     CNTR6220
    RETURN                                                               CNTR6230
    END                                                                  CNTR6240


    SUBROUTINE CALC(AM,MY)                                               CNTR6250
    DIMENSION AM(MY,1 ),REC(900),  X(1800), Y(1800)                      CNTR6260
    DIMENSION IPT(3,3),INX(8),INY(8)                                     CNTR6270
    COMMON /DAYHOF/ MT,NT,NI,IX,IY,IDX,IDY,ISS,IT,IV,NP,N ,JT,           CNTR6280
   1                PY,REC,CV,    IPT,INX,INY,DL,RA,THE                 CNTR6290
    COMMON /INTFAC/ X,Y                                                  CNTR6300
    IT=0                                                                 CNTR6310
    N=N+1                                                                CNTR6320
    IF (IDX**2 + IDY**2 -1) 20,1,20                                      CNTR6330
  1 IF (IDX) 10,2,10                                                     CNTR6340
  2 X(N)=IX                                                              CNTR6350
```

```
      Z=IY                                                                    CNTR6360
      IY2=IY+IDY                                                              CNTR6370
      DY=IDY                                                                  CNTR6380
   41 Y(N)=((AM(IY,IX)-CV)/(AM(IY,IX)-AM(IY2,IX)))*DY+Z                       CNTR6390
      RETURN                                                                  CNTR6400
   10 Y(N)=IY                                                                 CNTR6410
      W=IX                                                                    CNTR6420
      DX=IDX                                                                  CNTR6430
      IX2=IX+IDX                                                              CNTR6440
   44 X(N)=((AM(IY,IX)-CV)/(AM(IY,IX)-AM(IY,IX2)))*DX+W                       CNTR6450
      RETURN                                                                  CNTR6460
   20 IX2=IX+IDX                                                              CNTR6470
      IY2=IY+IDY                                                              CNTR6480
      W=IX                                                                    CNTR6490
      Z=IY                                                                    CNTR6500
      DX=IDX                                                                  CNTR6510
      DY=IDY                                                                  CNTR6520
      DCP=(AM(IY,IX)+AM(IY,IX2)+AM(IY2,IX)+AM(IY2,IX2))/4.0                   CNTR6530
      IF (PY-2.0) 24,21,24                                                    CNTR6540
   24 IF (DCP-CV) 21,21,25                                                    CNTR6550
   21 AL=AM(IY,IX)-DCP                                                        CNTR6560
   23 V=.5*(AL+DCP -CV)/AL                                                    CNTR6570
   27 X(N)=V*DX+W                                                             CNTR6580
      Y(N)=V*DY+Z                                                             CNTR6590
      PY=0.0                                                                  CNTR6600
      RETURN                                                                  CNTR6610
   25 IT=1                                                                    CNTR6620
      AL=AM(IY2,IX2)-DCP                                                      CNTR6630
   33 V=.5*(AL+DCP-CV)/AL                                                     CNTR6640
   28 X(N)=-V*DX+W + DX                                                       CNTR6650
      Y(N)=-V*DY+Z + DY                                                       CNTR6660
      Y(N)=-V*DY+Z + DY                                                       CNTR6670
      RETURN                                                                  CNTR6680
      END                                                                     CNTR6690

      SUBROUTINE PLOTT(NP,CV)                                                 CNTR6700
      COMMON/INTFAC/X(1800),Y(1800)                                           CNTR6710
      LOGICAL*1 MINUS,LABL                                                    CNTR6720
      COMMON/TABL/ TABC(20,6),JC                                              CNTR6730
      COMMON/DITS/XMIN,YMIN,SLOPEX,SLOPEY,DITSDX,DITSDY,IDIR,LABL,MINUS       CNTR6740
C     SCALE POINTS FOR PLOT ROUTINE                                           CNTR6750
      DO 100 I=1,NP                                                           CNTR6760
      X(I)=SLOPEX*(X(I)-XMIN)                                                 CNTR6770
  100 Y(I)=SLOPEY*(-Y(I)-YMIN)                                                CNTR6780
      CALL LINE(X,Y,NP,1,1)                                                   CNTR6790
      IF(.NOT.LABL) RETURN                                                    CNTR6800
C     SHOULD ADJUSTMENT OF PEN LOCATION BE MADE FOR LABELLING CURVE?          CNTR6810
```

```
      DIR=0.0                                                          CNTR6820
      GO TO (1,2,3,4,6), IDIR                                          CNTR6830
    1 DIR=90.                                                          CNTR6840
    2 COORDX=X(1 )                                                     CNTR6850
      COORDY=Y(1 )                                                     CNTR6860
    5 CALL NUMBER(COORDX,COORDY,.07,CV,DIR,1)                          CNTR6870
      RETURN                                                           CNTR6880
C     MOVE PEN DOWN ONE HALF INCH                                      CNTR6890
    3 DIR=90.                                                          CNTR6900
      COORDX=X(1 )                                                     CNTR6910
      COORDY=Y(1 )-.3                                                  CNTR6920
      GO TO 5                                                          CNTR6930
C     MOVE PEN TO THE LEFT                                             CNTR6940
    4 COORDX=X(1 )-.3                                                  CNTR6950
      COORDY=Y(1 )                                                     CNTR6960
      GO TO 5                                                          CNTR6970
C     SEARCH FOR XMAX,XMIN,YMAX,YMIN,AND SAVE YMINX                    CNTR6980
    6 XMAX=X(1)                                                        CNTR6990
      SMIN=XMAX                                                        CNTR7000
      YMINX=Y(1)                                                       CNTR7010
      YMAX=YMINX                                                       CNTR7020
      VMIN=YMINX                                                       CNTR7030
      DO 200 I=2,NP                                                    CNTR7040
      IF(X(I).GT.XMAX) XMAX=X(I)                                       CNTR7050
      IF(Y(I).LT.VMIN) VMIN=Y(I)                                       CNTR7060
      IF(Y(I).GT.YMAX) YMAX=Y(I)                                       CNTR7070
      IF(X(I).GE.SMIN) GO TO 200                                       CNTR7080
      SMIN=X(I)                                                        CNTR7090
      YMINX=Y(I)                                                       CNTR7100
  200 CONTINUE                                                         CNTR7110
C     JC=NUMBER OF ENTRIES IN TABC                                     CNTR7120
      IF(JC) 400,500,400                                               CNTR7130
C     SEARCH TABLE TO SEE IF THIS IS INTERIOR TO ANOTHER INTERIOR SEGMENCNTR7140
  400 DO 900 I=1,JC                                                    CNTR7150
      IF(XMAX.LT.TABC(I,1).AND.YMAX.LT.TABC(I,2).AND.VMIN.GT.TABC(I,3). CNTR7160
     1AND.SMIN.GT.TABC(I,4)) GO TO 700                                 CNTR7170
  900 CONTINUE                                                         CNTR7180
C     DID NOT FIND THIS CONTOUR TO BE INTERIOR TO ANOTHER              CNTR7190
C     CHECK IF EXTERIOR                                                CNTR7200
      DO 1000 I=1,JC                                                   CNTR7210
      IF(XMAX.GT.TABC(I,1).AND.YMAX.GT.TABC(I,2).AND.VMIN.LT.TABC(I,3). CNTR7220
     1AND.SMIN.LT.TABC(I,4)) GO TO 800                                 CNTR7230
 1000 CONTINUE                                                         CNTR7240
C     THIS CONTOUR SEGMENT WAS NEITHER INTERIOR NOR EXTERIOR TO ANOTHER CNTR7250
  500 IF (JC.EQ.20) RETURN                                             CNTR7260
      JC=JC+1                                                          CNTR7270
      MC=JC                                                            CNTR7280
  600 TABC(MC,1)=XMAX                                                  CNTR7290
```

```
      TABC(MC,2)=YMAX                                                   CNTR7300
      TABC(MC,3)=VMIN                                                   CNTR7310
      TABC(MC,4)=SMIN                                                   CNTR7320
      TABC(MC,5)=YMINX                                                  CNTR7330
      TABC(MC,6)=CV                                                     CNTR7340
      RETURN                                                            CNTR7350
C     CHECK IF THIS INTERIOR ONE IS OF HIGHER LEVEL                     CNTR7360
  700 IF(CV.LE.TABC(I,6) ) RETURN                                       CNTR7370
 2000 MC=I                                                              CNTR7380
      GO TO 600                                                         CNTR7390
C     CHECK IF LEVEL OF THIS EXTERIOR ONE IS HIGHER                     CNTR7400
  800 IF(CV.LT.TABC(I,6)) RETURN                                        CNTR7410
      GO TO 2000                                                        CNTR7420
      END                                                               CNTR7430
```

LIST OF REFERENCES

1. Gabor, D., "A New Microscopic Principle," Nature, v. 161, p. 778, 1948.

2. Gabor, D., "Holography, 1948-1971," Proceedings of the IEEE, v. 60, no. 6, p. 655, June 1972.

3. Acoustical Holography v. 1-4, Plenum Press, 1967-1972.

4. El-Sum, H. M. A., Chap. 1, p.5, Acoustical Holography, v. 1, Plenum Press, 1967.

5. Goodman, J. W., Chap. 12, p. 174, Acoustical Holography, v. 1, Plenum Press, 1967.

6. Born, M. and Wolf, E., Principles of Optics, 3d ed., Pergamon Press, 1970.

7. Goodman, J. W., Introduction to Fourier Optics, McGraw-Hill, 1968.

8. Shewell, T. R. and Wolf, E., "Inverse Diffraction and a New Reciprocity Theorem," Journal of the Optical Society, v. 58, no. 12, p. 1596, Dec. 1968.

9. Banos, A., Dipole Radiation in the Presence of a Conducting Half-Space, Pergamon Press, 1966.

10. Boyer, A. L., and others, Chap. 18, p. 333, Acoustical Holography v. 3, Plenum Press, 1971.

11. Peterson, D. P. and Middleton D., "Sampling the Reconstruction of Wavenumber - limited Functions in N-Dimensional Euclidean Spaces," Information and Control, 5; p. 279, 1962.

12. Sondhi, M. M., "Reconstruction of Objects from Their Sound Diffraction Patterns," Journal of the Acoustical Society of America, v. 46, no. 5, p. 1158, 1969.

13. Andrews, H. C., Computer Techniques in Image Processing, p. 116, Academic Press, 1970.

14. Gold and Rader, Digital Processing of Signals, McGraw-Hill, 1969.

15. Huang, T. S. and Kasnitz, H. L, Proceedings of the Computerized Imaging Techniques Seminar, Society for Photo-Optical Instrumentation Engineers, XVII - 1, 1967.

16. Cooley, T. W. and Tukey, T. W., "An Algorithm for the Machine Calculation of Complex Fourier Series," Mathem. of Computation, v. 19, p. 297.

17. Brenner, N., Subroutine FOUR2, May 1967, not published (this program is listed in the Appendix to this report).

18. Kalra, A. V. and Rodgers, P. W., "Detection of Buried Geological Ore Bodies by Reconstructed Wavefronts," p. 687, in Acoustical Holography, v. 4, Plenum Press, 1972.

19. Papoulis, A., Systems and Transforms with Applications in Optics, McGraw-Hill, 1968.

20. Lipkin, B. S. and Rosenfeld, A., Editors, Picture Processing and Psychopictorics, Academic Press, 1970.

21. Rosenfeld, A., Lee, Y. H. and Thomas, R. B., Edge and Curve Detection for Texture Discrimination, p. 381, in Lipkin and Rosenfeld, Picture Processing and Psychopictorics, Academic Press, 1970.

22. Rosenfeld, A., Thomas, R. B. and Lee, Y. H., Edge and Curve Enhancement in Digital Pictures, University of Maryland Computer Science Cneter Technical Report 69-93, May 1969.

23. Peavey, B. R., Subroutine CONTUR, July 1969, not published (this program is listed in the Appendix to this report).

24. Lintner, M. A., Algorithm and Computer Programs for the Hidden Line Problem for Single Valued Surfaces, Idaho Nuclear Corporation Report IN-1342, Dec. 1969.

25. Huang, T. S., "Digital Holography," Proceedings of the IEEE, v. 59, no. 9, Sept. 1971.

26. Ichioka, Y., Izumi, M. and Suzuki, T., "Scanning Halfton Plotter and Computer Generated Continuous-Tone Hologram," Applied Optics, v. 10, no.2, p. 403, Feb. 1971.

27. Griggs, C. A., A Computer Technique for Near-Field Analysis of an Ultrasonic Transducer, M.S. Thesis, Naval Postgraduate School, Monterey, March 1973.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Asst Professor J. P. Powers Code 52Po Naval Postgraduate School Monterey, California 93940	1
4.	Assoc. Professor G. L. Sackman Code 52Sa Naval Postgraduate School Monterey, California 93940	1
5.	LCDR D. E. Mueller 8000 Muenchen 40 Mommsenstr. 3 Germany	1
6.	Mr. B. Saltzer Code 6513 Naval Underseas Research and Development Center San Diego, California 92132	1
7.	Marineamt, Inspektion der Offz./UOffz. AusbM. 294 Wilhelmshaven Germany	1
8.	DOKZENT Bw-See 53 Bonn Friedrich-Ebert-Allee 34 Germany	1
9.	Bundesministerium der Verteidigung - T V 6 - 53 Bonn Postfach 161 Germany	1

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

ORIGINATING ACTIVITY (Corporate author)
Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

REPORT TITLE
A Computerized Acoustic Imaging Technique Incorporating Automatic Object Recognition

DESCRIPTIVE NOTES (Type of report and inclusive dates)
Electrical Engineer; March 1973

AUTHOR(S) (First name, middle initial, last name)
Dieter Erwin Mueller

REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
March 1973	153	27

CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
PROJECT NO.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

DISTRIBUTION STATEMENT
Approved for public release; distribution unlimited.

SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

ABSTRACT

A computerized technique combining backward wave propagation and an automatic edge detection scheme is developed and tested. The class of objects considered is limited to those with edge boundaries since it is shown that a universal automatic reconstruction cannot be obtained for all possible objects. Using samples of the acoustic diffraction pattern as input this technique enables the computer to predict the most likely locations of objects and to produce graphical output of the objects. A discussion is given on implementing the diffraction equations on the computer. A simplified edge detection scheme conserving both memory space and computer time is described. Test results are presented for both computer generated diffraction patterns and one set of experimental data.

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Sound Imaging						
Acoustical Holography						
Backward Wave Propagation						
Edge Detection						
Computer Holography						
Automatic Object Recognition						