Title
THREE-DIMENSIONAL RECONSTRUCTION IN ELECTRON MICROSCOPY

Permalink
https://escholarship.org/uc/item/3vz917n1

Author
Grano, David Allan

Publication Date
2011-08-16
THREE-DIMENSIONAL RECONSTRUCTION IN ELECTRON MICROSCOPY

David Allan Grano
(Ph. D. thesis)

May 1979

TWO-WEEK LOAN COPY
This is a Library Circulating Copy which may be borrowed for two weeks. For a personal retention copy, call Tech. Info. Division, Ext. 6782

Prepared for the U. S. Department of Energy under Contract W-7405-ENG-48
DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.
ABSTRACT

The development and implementation of a versatile system of image processing programs for electron microscopy is described. Both high-dose, negatively stained specimens and low-dose, unstained specimens can be analyzed by this system. The theory behind image analysis in electron microscopy is described together with the practical aspects of computer processing of electron micrographs.

The Fourier transform of cylindrically symmetric objects is studied in some detail. Structure factor formulas are derived by the use of a delta-function cylindrical net model. The range of structural deductions that may be made from the Fourier transforms of projections of such objects is discussed.

The program system is used to study several biological specimens. The methods of 2-D image filtering are applied to high-dose images of
negatively stained gap junction membranes and to frozen, hydrated, low-dose images of the hexagonally packed protein component of *Spirillum serpens* cell wall. The computer processed *Spirillum serpens* specimen reveals the presence of Y-linkers similar to those seen in negatively stained preparations. Computer processing of the gap junction images makes the presence of a central staining pit more obvious. The techniques of 3-D helical reconstruction are applied to high-dose images of negatively stained T4 bacteriophage tails, to demonstrate the successful transfer of the IBM-based MRC helical reconstruction programs to our Control Data Corporation computer system.

Finally, the tubular structures found in preparations of *Spirillum serpens* cell wall are analyzed by Fourier methods. The cylindrical symmetry of these tubes is confirmed along with their essential similarity to rolled-up sheets of hexagonally packed protein. The radial positions of the first diffraction maxima on the layerlines, used together with the tube outside diameters, leads to the conclusion that the position of the subunit-to-subunit contacts lies near the extracellular surface in the sheet form.
TABLE OF CONTENTS

List of Figures.................................iv

I. Introduction and Historical Perspective
   A. Image Interpretation..........................2
   B. Image Analysis and Reconstruction in EM........3
   C. EM Structure Analysis at
      Lawrence Berkeley Laboratory...................7

II. Image Formation in the Electron Microscope
   A. Introduction..................................10
   B. The Electron Optics of the Microscope........11
   C. Image Formation Theory.........................17
   D. The Transmittance Function....................21
   E. The Image in Terms of the Object..............26
      1. The weak phase object......................27
      2. The weak amplitude and phase object.........29
      3. Single-side-band image mode................30
   F. Summation and Conclusions....................31

III. Theory & Methods of Two-Dimensional Restoration
    and Three-Dimensional Reconstruction
   A. Introduction..................................32
   B. The Optical Diffractometer....................35
   C. The Scanner....................................41
   D. Scanning.......................................47
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>E.</td>
<td>The Discrete Fourier Transform</td>
<td>59</td>
</tr>
<tr>
<td>1.</td>
<td>The relationship between sample interval,</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>array size, and Fourier sampling interval.</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>The DFT of a periodic specimen</td>
<td>70</td>
</tr>
<tr>
<td>F.</td>
<td>Two-dimensional Processing</td>
<td>94</td>
</tr>
<tr>
<td>G.</td>
<td>The Act of Bilinear Interpolation</td>
<td>115</td>
</tr>
<tr>
<td>H.</td>
<td>Plane Group Rationalization</td>
<td>121</td>
</tr>
<tr>
<td>I.</td>
<td>Coordinate Origin Shifts</td>
<td>126</td>
</tr>
<tr>
<td>J.</td>
<td>Contrast Transfer Correction</td>
<td>130</td>
</tr>
<tr>
<td>K.</td>
<td>Translational, Rotational and Magnification</td>
<td>131</td>
</tr>
<tr>
<td></td>
<td>Alignment</td>
<td></td>
</tr>
<tr>
<td>L.</td>
<td>Three-Dimensional Helical Reconstruction</td>
<td>135</td>
</tr>
<tr>
<td>M.</td>
<td>The General Three-Dimensional</td>
<td>141</td>
</tr>
<tr>
<td></td>
<td>Reconstruction Problem</td>
<td></td>
</tr>
<tr>
<td>N.</td>
<td>Conclusion</td>
<td>148</td>
</tr>
<tr>
<td>IV.</td>
<td>Implementation of Image Processing and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Three-Dimensional Reconstruction Algorithms</td>
<td></td>
</tr>
<tr>
<td>A.</td>
<td>Introduction</td>
<td>150</td>
</tr>
<tr>
<td>B.</td>
<td>Large Interpolation</td>
<td>152</td>
</tr>
<tr>
<td>C.</td>
<td>Large Discrete Transforms</td>
<td>154</td>
</tr>
<tr>
<td>D.</td>
<td>Piecewise Transform</td>
<td>160</td>
</tr>
<tr>
<td>E.</td>
<td>Ordering of the Fourier Coefficients in the</td>
<td>168</td>
</tr>
<tr>
<td></td>
<td>DFT</td>
<td></td>
</tr>
<tr>
<td>F.</td>
<td>Picture Generation</td>
<td>175</td>
</tr>
<tr>
<td>G.</td>
<td>General Computer Considerations</td>
<td>183</td>
</tr>
<tr>
<td>H.</td>
<td>Scan Conversion Program</td>
<td>185</td>
</tr>
<tr>
<td>I.</td>
<td>The Histogram Program</td>
<td>187</td>
</tr>
<tr>
<td>J.</td>
<td>Lattice Definition and Viewing Program</td>
<td>190</td>
</tr>
<tr>
<td>K.</td>
<td>Phase Origin Refinement Program</td>
<td>194</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Figure 1</td>
<td>Schematic diagram of EM optics.</td>
<td>12</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Coordinates used to describe astigmatism orientation.</td>
<td>22</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Flow chart of the 3-D reconstruction process.</td>
<td>36</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Evidence of scanner defect found in Fourier space.</td>
<td>44</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Comparison of continuous and discrete Fourier transforms.</td>
<td>48</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Aliasing effects in non-band-limited Fourier transforms.</td>
<td>52</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Effect of sampling interval on aliasing error.</td>
<td>56</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Graphs of the aperture transform for various aperture sizes.</td>
<td>60</td>
</tr>
<tr>
<td>Figure 9</td>
<td>The discrete Fourier transform pair.</td>
<td>66</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Variables used to describe the discrete transform of a cosine wave.</td>
<td>71</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Variables used to describe a mis-sampled delta function.</td>
<td>78</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Correction graphs for a mis-sampled delta function.</td>
<td>80</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Two-dimensional convolution of a mis-sampled delta function.</td>
<td>83</td>
</tr>
</tbody>
</table>
Figure 14  Graphs for maximum correction factor and percent error for various data width to array width ratios.......................... 88

Figure 15  Effect of the number of repeats on the DFT in the case of two waves of nearly identical frequency................. 91

Figure 16  Flow chart of two-dimensional processing..... 95

Figure 17  Effect of the filler value on the floated transform..........................101

Figure 18  Transform of the Fourier space mask function in normalized coordinates...........108

Figure 19  Plot of the normalized aperture transform for several mask sizes.....................110

Figure 20  The use of an optically filtered image for the determination of interpolation values........118

Figure 21  An illustration of plane group rationalization........................................123

Figure 22  A discrete helix illustrating the selection rule........................................138

Figure 23  Assignment of near side and far side to diffraction spots..............................142

Figure 24  Subdivision of a large array into smaller blocks for FFT calculation.............157

Figure 25  Subdivision of a large array for the piecewise transform.......................162

Figure 26  Frequency ordering in the FFT and array flipping..................................169

Figure 27  Patterning in a greyscale dot picture..............176
Figure 28  An example of scan histogram program output...188
Figure 29  An example of LTCLOOK output..................191
Figure 30  Organization of the PLTPROC program.........197
Figure 31  Comparison of array storage using row-major
and column-major orders.................................222
Figure 32  Geometry used for tilt and skew correction
by the MRC routine EM4HLX.............................225
Figure 33  Sections types available from
the MRC helical reconstruction program....233
Figure 34  Contour plots and greyscale pictures of
sections produced by the EM3DISP program....238
Figure 35  Geometry of the radial projection.............241
Figure 36  Cylindrical-tube-to-hexagonal-sheet
transition.................................................246
Figure 37  A single unit of the cylindrical tube............249
Figure 38  Negatively stained rat liver gap junction....269
Figure 39  Optical and computed diffractograms
of gap junction...........................................271
Figure 40  Gap junction image, before and after computer
"optical" filtration.......................................273
Figure 41  Power spectrum of an interpolated
gap junction image.......................................276
Figure 42  Delta function
filtration of the gap junction.........................279
Figure 43  Image and optical diffraction pattern
of a T4 tail..............................................282
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>44</td>
<td>PLTPROC picture of a typical scanned T4 tail</td>
</tr>
<tr>
<td>45</td>
<td>Different displays of the T4 tail reconstruction</td>
</tr>
<tr>
<td>46</td>
<td>Comparison of T4 tail reconstruction with published data</td>
</tr>
<tr>
<td>47</td>
<td>Negatively stained and frozen hydrated cell walls from <em>Spirillum serpens</em></td>
</tr>
<tr>
<td>48</td>
<td>Scanned image and diffraction pattern of frozen cell wall</td>
</tr>
<tr>
<td>49</td>
<td>Symmetrized and filtered frozen cell wall</td>
</tr>
<tr>
<td>50</td>
<td>Images and optical diffraction patterns of negatively stained <em>S. serpens</em> cell wall tubes</td>
</tr>
<tr>
<td>51</td>
<td>Computed diffraction pattern of <em>S. serpens</em> cell wall tube</td>
</tr>
<tr>
<td>52</td>
<td>Diagram illustrating the formation of tubes from sheets</td>
</tr>
<tr>
<td>53</td>
<td>n,l plot for <em>S. serpens</em> cell wall tubes</td>
</tr>
<tr>
<td>54</td>
<td>Computer generated picture of cell wall tube - case 1</td>
</tr>
<tr>
<td>55</td>
<td>Computer generated picture of cell wall tube - case 2</td>
</tr>
</tbody>
</table>
I. Introduction and Historical Perspective

The electron microscope has served the needs of biologists for more than thirty years using techniques that were developed or conceived during the infancy of the field. Although the extremely small wavelength of the electron seemed to hold out the hope of atomic resolution, such detail has never been achieved with the ease needed for practical use. Today, however, the atomic frontier again seems within reach. A new generation of electron microscopists is at hand wielding sophisticated techniques to apply both to the production and viewing of the specimen and to the analysis of the images thus produced. Although the hope of easily viewed atomic resolution is gone, the possibility of practical, achievable near-atomic resolution beckons.

The last decade has seen an elucidation of the problems confronting the electron microscopist in search of ultra-high resolution (<20Å). The traditional methods of preparing specimens by fixation, embedment, sectioning, staining, and viewing have been shown to be fatal to structures in the specimens smaller than 30-40Å. Work continues today on improvements in all these areas. The development of the negative staining technique circumvented the need to fix, embed, section, and stain. Of course, not all biological samples of interest are amenable to this treatment, but for isolated particles, viruses, and small crystallites, for example, the support and staining provided by the negative
stain preserves structural details to the 15-20Å level. The microcrystalline structure of the negative stain, together with its disruptive surface tension effects when drying, led to the search for more gentle staining and/or support techniques. The glucose embedment technique pioneered by Unwin and Henderson (1975), together with the frozen hydrated technique advanced by Taylor and Glaeser (1976), offer two approaches to preserving structural details, for some specimens, close to the limit of instrumental resolution.

The observation of more and more detail would be a useless exercise in many cases if the increased resolution could not be interpreted more clearly. Knowing the two-dimensional projection of an object to high resolution cannot answer many questions about structure, especially biological structures. What is needed is a full, three-dimensional knowledge of the object's structure. Obtaining this information is the goal of modern high-resolution structure determination from electron micrographs.

A. Image Interpretation

Image interpretation in electron microscopy has been both sophisticated and simple-minded. The strong scattering exhibited by electrons warned the early pioneers that image interpretation might be complex. The theoreticians knew that explanations based upon dynamical scattering theory were the only correct ones (Hirsch, 1965) yet the biologists who used the microscopes saw structures in their electron
micrographs that were recognizable from optical microscopy. Theory notwithstanding, images were interpreted (and still are) on a subjective basis. The variations in density in the photographic image were understood "intuitively". A more sophisticated approach was the mass-thickness interpretation (Zeitler and Bahr, 1965). According to this approximation, an increased density of scattering centers in the direction of the electron beam leads to increased scattering outside the effective aperture of the microscope and hence a darker area on a print of the micrograph. The elucidation of the idea of the thin EM (electron microscopy) specimen as a weak phase object has lead to a more rigorous understanding that the image represents a projection of the specimen's Coulomb potential in a direction parallel to the electron beam. For classes of objects that are known to violate the kinematic approximation, such as thick crystals, a dynamical treatment of image interpretation is needed. The Cowley and Moodie (1957) multislice dynamical approximation provides a tractable approach. In our laboratory, Jap (1975) has investigated dynamical approximations and their domains of validity in electron microscopy. He has advanced an invertible dynamical relationship between image and object.

B. Image Analysis and Reconstruction in EM

Klug and Berger (1964) at the MRC Laboratory for Molecular Biology in Cambridge, England made use of the optical diffraction technique to study periodicities present in bio-
logical images. They also recognized that the Fraunhofer diffraction pattern thus produced carries information about the two "sides" of a helical object. Klug and DeRosier (1966) actually separated "near" and "far" side images of the surface of helical viruses. The next step was taken by DeRosier and Klug (1968) in 1968 when they succeeded in demonstrating that the 3-D structure of a helix could be recovered from a single 2-D micrograph. The key was the fact that the image represented a projection of the structure. The Fourier projection theorem, known to all crystallographers, relates the 2-D transform of a projection of a 3-D structure to the 3-D Fourier transform of the same structure. DeRosier and Klug exploited the connection between the image and the 3-D Fourier transform, analyzed and made use of the symmetry of the helical specimen, and developed computer methods to carry out the reconstruction.

The next several years saw the development of the reconstruction theory together with investigations of the conditions that must be met for the reconstruction to work. The techniques were expanded to cover the case of "spherical" or icosahedral viruses (Crowther et al., 1970a, 1970b; Crowther, 1971). At the same time, other methods of three-dimensional reconstruction besides the Fourier (or reciprocal space) method were being tried. Gordon, Bender, and Herman (1970) advanced the Algebraic Reconstruction Technique (ART), Gilbert (1972a, 1972b) developed the Simultaneous Iterative Reconstruction Technique (SIRT), Ramachandran
and Lakshminarayanan (1971) used a method based on convolutions, and Vainshtein (1971) proposed another technique based on projections. Arguments over the relative merits, deficiencies, accuracies and meanings appeared in the literature while still other ideas were proposed. Zwick and Zeitler (1973) introduced a new set of basis functions for the projection, real and reciprocal spaces which had the advantage of being calculable from one another. The field of nuclear medicine and X-ray tomography have made use of similar reconstruction techniques. The MRC Laboratory and the Biocenter at Basel emerged as two centers of EM reconstruction research. Amos and Klug at MRC (1975) developed the programs needed to correlate data from more than one image of a helical object (T4 bacteriophage tail) with the view toward reducing statistical noise and staining effects, while Smith and Aebi at Basel (1974) made us aware of the D(z,k) plot to distinguish more clearly helical signal from non-helical "noise" while combining data from many images. Data handling became more sophisticated with the correction for the contrast transfer function and the use of several differently defocused micrographs to gather information about all parts of the Fourier transform. Alignment techniques useful for asymmetric as well as symmetric objects were developed for electron microscopy by Frank (1975) and Saxton (1974).

The transition between high resolution and ultra-high resolution was accomplished by Henderson and Unwin (1975).
They made use of all of the advances in image processing, radiation damage theory, image formation theory and hydrated specimen techniques to do a three-dimensional reconstruction of the purple membrane from *Halobacterium halobium* to 7Å resolution. The resulting structure shows clearly the alpha helices which are present in the protein component. The first step in an exciting new field has been taken. With electron diffraction patterns from many unstained, stained, frozen, and unfrozen biological specimens extending to 3Å, three-dimensional reconstructions to these resolutions is imminent.

Three-dimensional image reconstruction is essential in any attempt to extract maximum information from the electron microscope. Even though the specimens used in electron microscopy are thin, the fact that the image is a projection can obscure spatial relationships found in the specimens. Stereo microscopy is best suited for statistically defined images which are dominated by surface detail. The new unstained specimen techniques, applied in a manner to minimize radiation damage to the specimen, will yield "invisible images" (Caspar, 1975) in most cases. These images would be useless in the viewing of traditional stereo pairs. Computer reconstructed three-dimensional views of the specimens will bridge the gap between an educated guess at structure-function relationships deduced from high resolution projections and a hard-evidence backed conclusion drawn from the knowledge of the 3-D structure.
C. EM Structure Analysis at Lawrence Berkeley Laboratory

The acquisition of the software tools needed to do biological structure analysis by electron microscopy necessarily preceded their use in solving biological problems. The development and/or implementation of the programs in our processing system was often done without reference to any biological problem at hand. Some biological specimens with known structure were used as a source of test data to debug programs. As a core of abilities emerged, they were used to answer some limited questions about biological specimens actively under study in our laboratory. Finally, enough processing capability was in place so that the biological questions determined the programs' usefulness, not vice versa.

The development of our system has been a pragmatic one. In order to have programs that fit our needs and our computer environment, most were developed from scratch, albeit guided by previous work done elsewhere. When clearly defined tasks could be accomplished by programs from other laboratories (notably the three-dimensional helical reconstruction algorithms from the MRC), these have been hybridized to work with our system. The programs developed here have been designed to be very economic in their use of computer resources and very flexible in their range of applications. Since no processing system is the ultimate, ours is capable of being upgraded and expanded. The usefulness of
the programs is enhanced by their orientation toward the electron microscopist doing structural analysis who needs answers but does not want to become a computer expert to get them. In this same vein, the efficient production of visually displayed results has been a major part of this work. It is my belief that the axiom of "one picture is worth a thousand words" is especially applicable in the area of EM structure analysis, where the mountains of numbers can quickly obscure their own significance.

Although many of the programs developed here have similar counterparts elsewhere, the almost total lack of practical software exchange from laboratory to laboratory has necessitated this duplication of effort. In fact, this has been very helpful since the literature is weak in the practical aspects of this type of work. For the most part, the limitations of the techniques are best learned by the step-by-step development of the programs and their testing with artificial data. The quirks of the programs and of the methods can be found and understood. Unfortunately, the naive application of these powerful computer techniques to real data can lead to erroneous results and interpretations.

To date, our system is capable of extensive two-dimensional processing and limited three-dimensional processing. It has been used to do 2-D restorations on a variety of biological specimens. The 3-D reconstruction of T4 bacteriophage tail was undertaken to test our understand-
ing and implementation of the MRC helical reconstruction programs. The full range of two-dimensional image processing and three-dimensional image reconstruction abilities have been brought to bear on the problem of the three-dimensional structure of the outer cell membrane of the bacterium *Spirillum serpens*. An analysis has been made of the crystallographic symmetry of the "tubular forms" which are found in negatively stained preparations of the cell wall. Theoretical calculations for generalized tubular structures were done to decide whether the techniques of helical reconstruction could be successfully applied to specimens with cylindrical symmetry. The experimental findings from the *Spirillum serpens* tubes were compared with those derived from theory in order to confirm the symmetry of the structure. An analytic model was used to investigate how much can be deduced about the 3-D structure of a cylinder from a single view. The results of the analysis of this model were used to propose extensions to the helical reconstruction programs to allow the reconstruction of objects having cylindrical symmetry.
II. Image Formation in the Electron Microscope

A. Introduction

The foundation for any attempts to process an image, i.e., correct it for systematic errors and/or extract information from it, must be an understanding of the formation of the image. Since images are the building blocks in the reconstruction of three-dimensional information, the approximations employed to describe images, together with the attendant limitations and exclusions, must be appreciated if one is to have confidence in a derived structure. This section attempts to outline the different ways of analyzing the image formation process and to underscore the essential limitations of each.

The photographic plate pulled from the microscope displays a varying pattern of exposed silver grains. The most common method of interpretation has been visual, subjective, and easy to conceptualize. Until recently, the electron microscopic image was viewed as analogous to the light optical image by the majority of EM users. What is seen and comprehended represents what really exists. This gut-level approach to image interpretation is still prevalent among the users of the electron microscope; and for qualitative results at resolutions of object detail preserved by the common sectioning methods, it is a very good method of interpretation. With the need to get quantitative results and with the push to smaller image details,
image formation theories have been spawned. Since the specimens introduced into the microscope vary so greatly, from unstained thin biological crystals to thick material-science alloys, it is not surprising to find a variety of theories that have their own domains of validity (Jap, 1975). For practical image processing and three-dimensional reconstruction work, we will focus attention on a viewpoint which will cover the majority of specimens of interest, albeit with a dash of unproven extension and pragmatic optimism here and there.

B. The Electron Optics of the Microscope

A brief description of the imaging systems found in a transmission electron microscope will serve to introduce some terminology that will be used often, as well as to give a physical understanding of the image-forming process.

Figure 1 shows the major components of the imaging system. At the top of this schematic drawing of the microscope is found the electron "gun" or source of electrons. The most common type of gun used is the thermionic type, although a more sophisticated gun, the field emission gun (FEG), offers several advantages. The properties of the thermionic emitter will be explained first.

A thin piece of tungsten wire bent into a hairpin curve acts as the cathode or filament in the thermionic gun. A biasing shield surrounds the filament and contains a small
Figure 1

Schematic of the imaging system of an electron microscope
High voltage bias shield
Anode
1st Condenser lens
2nd Condenser lens
Condenser aperture
Object plane
Objective lens
Objective aperture
Intermediate lens
Projector lens
Viewing screen
Image recording
aperture used both to limit the emission of electrons to a small angle and to form a focusing electrostatic lens when held at a potential different from that of the filament. The anode accelerates the emitted electrons and its small bore again serves to limit the beam. The heated filament emits electrons with a spread of wavelengths (energies) related to the emission temperature. The distribution of electron energies is Maxwellian (Hall, 1966) with a peak of approximately 0.25 electron volts and the majority of electrons falling below 1.0eV for a typical tungsten temperature of 2900° Kelvin. The focusing effect of the biasing shield is significant because it increases the intensity of the beam while decreasing its size. Since the aberrations of magnetic lenses dictate that only paraxial electrons be used in the imaging process, a thin pencil of electrons is the only suitable source of illumination, and having enough electrons in this small angle is of great concern. The useful lower limit to the angle of illumination is determined by the brightness of the source.

Seeking to improve source brightness over that achieved by the thermionic gun, a gun operating under different physical principles has been developed. The FEG uses the immense electric field found near a point of extremely small radius to pull electrons from a cold or "lukewarm" filament. Using an oriented crystal of tungsten with an approximate radius of a few hundred Ångstroms, the FEG manufactured by JEOL, Ltd. achieves a 100-fold increase in brightness.
The next major component in the electron optical system is the condenser lens system - usually consisting of two magnetic condenser lenses and an adjustable physical aperture. This double lens system is used to control both the intensity of illumination at the specimen and the angular spread of the beam. The physical aperture, located in the second condenser lens, defines the largest angle of illumination possible. The first condenser lens is used to demagnify the electron source size. The second lens is used to control the beam intensity on the object by either focusing the demagnified source on the specimen (maximum intensity) or focusing it above or beyond the specimen (decreasing intensity). Astigmatism in the condenser lens system merely limits the maximum intensity obtainable.

The next lens, the objective lens, is the lens which actually images the electrons scattered by the specimen. The aberrations exhibited by the objective lens are not compensated by the rest of the imaging system, so they are of primary importance. Of the imaging defects, only spherical and chromatic aberrations are important in the objective lens because the specimen is small and on-axis and the angle of scattering physically allowed by the instrument is very small (less than $10^{-2}$ radians). These two criteria minimize the effects of pincushion and barrel distortion, coma, and anisotropic distortions, which are only significant for
large scattering angles and/or far-from-axis object distances. Another aberration, astigmatism, is caused by a magnetic lens which is not cylindrically symmetric. All electron microscopes contain stigmatizing devices which can compensate for this defect to a limit that is governed, in practice, by the skill of the operator.

The objective aperture, located in the focal plane of the objective lens on the downstream side of the lens (the back focal plane), plays a crucial part in the imaging process. This aperture limits the electrons which pass it to those scattered through an angle which is less than that subtended by the aperture. The aperture, operating as it does in the back focal plane, can also be used for dark-field imaging and single-side-band imaging (Hoppe, Langer, and Thon, 1970). Through the buildup of contamination, which in turn becomes charged by the electron beam, the aperture can exert a severe degrading effect on the image by acting as an electrostatic lens possessing unknown and variable characteristics.

The final lenses in the microscope, the intermediate and one or more projector lenses, serve to magnify the image to any number of convenient values. They also provide the means for using the microscope as a selected area (or limited area) electron diffraction device. These lenses do not have the ability to blur the final image because the errors introduced by them in the final magnified image are on a
much smaller scale than those introduced by the objective lens. The final image, however, does display pincushion, barrel and/or anisotropic distortions. When images are analyzed in a point-to-point, local-resolution manner, as they usually are when viewed by eye, these distortions are not important, but when the images are subjected to the type of scrutiny that will be outlined in later chapters, scrutiny that could extend across the entire 10cm plate, the distortions caused by the projector lens(es) will be, perhaps, fatally important if their effects are not corrected.

C. Image Formation Theory

The electron microscope can be viewed as an imaging system that obeys Abbe's theory of imaging (Goodman, 1968). If the specimen is conceptualized as an object which converts the incident plane wave into a collection of diffracted waves, the lens is a device which collects these diffracted rays and forms the Fourier transform of them in the back focal plane of the lens. In other words, the wave function at the back focal plane is the Fraunhofer diffraction pattern of the wavefunction at the object plane, modified by terms that take into account the finite aperture of the lens and the distortions that the lens imparts to the waves. The wavefunction at the image plane is the inverse Fourier transform of the wave function at the back focal plane.
Accepting that Abbe theory describes the transfer of information through the electron microscope, we can use the power of linear systems analysis and Fourier optics to describe the properties of the image wavefunction in terms of the object wavefunction (Goodman, 1968). Especially well elucidated are the relationships between the response of the electron optical system to theoretical, simple wavefunctions and the response of the system to a real, complicated wavefunction.

The electron microscope is a linear system since input and output are related by a linear operator, the Fourier transform. The properties of any linear system can be expressed in terms of an impulse response function, which is the response of the system to a delta-function-like input. Since any input can be decomposed into a weighted set of impulse functions, the properties of the system are known with regards to an arbitrary input.

Let \( f_o(x_o,y_o) \) be the input to a linear system (i.e. the object function) and \( f_i(x_i,y_i) \) be the output (i.e. the image function). Input is related to output by a linear operator \( O \) in a linear system

\[
 f_i(x_i,y_i) = O [f_o(x_o,y_o)] \quad (II-1)
\]

Any function \( f_o \) can be decomposed into a set of weighted and displaced delta functions

\[
 f_o(x_o,y_o) = \int \int f_o(\xi,\eta) \delta(x_o-\xi,y_o-\eta) d\xi d\eta \quad (II-2)
\]
We can substitute \((2)\) into \((1)\), and using the linearity of the operator \(0\), we can bring it through the integral to get:

\[
f_i(x_i, y_i) = \iint f_0(\xi, \eta) O[\delta(x_o - \xi, y_o - \eta)] \, d\xi d\eta \quad (II-3)
\]

Let \(h(x_i, y_i; \xi, \eta) = O[\delta(x_o - \xi, y_o - \eta)]\). This function represents the output of the system at \(x_i, y_i\) to an impulse at \(\xi, \eta\) and is called the impulse response of the system.

An important and reasonable approximation can be invoked to simplify the subsequent mathematics. A two-dimensional linear system is termed isoplanatic if the image of a point source (delta function) is the same, no matter where the delta function is placed in the object plane. To the extent that we can ignore the off-axis aberrations discussed above, this is a good approximation. The impulse response of an isoplanatic system only depends on the distance \(x_i - \xi, y_i - \eta\). Therefore

\[
h(x_i, y_i; \xi, \eta) = h(x_i - \xi, y_i - \eta) \quad (II-4)
\]

Substituting \((4)\) into \((3)\), the image function becomes

\[
f_i(x_i, y_i) = \iint f_0(\xi, \eta) \, h(x_i - \xi, y_i - \eta) \, d\xi d\eta \quad (II-5)
\]

which is recognizable as the two-dimensional convolution of the input function with the impulse response. There exists a relationship between convolutions and Fourier transforms which can be applied here. Let \(F_i\) be the Fourier transform
of \( f_i \); likewise \( F_o \) and \( f_o \), \( H \) and \( h \). The convolution theorem states that \( F_i = F_o \ast H \). The Fourier transform of the impulse response, \( H \), is called the coherent transfer function of the system (Goodman, 1968). Its attractiveness lies in the ease of implementation in the Fourier domain, multiplication versus convolution.

Thus the coherent transfer function of the system, the electron microscope, can be invoked to relate the image and the object Fourier transforms. The effect of spherical aberration and focusing errors, as well as the imposition of the finite objective aperture, can be taken into account through the transfer function. Spherical aberration and objective lens defocus and astigmatism contribute a phase distortion term, \( e^{ijy(\vec{s})} \) where \( \vec{s} \) is a vector in reciprocal space.

\[
y(\vec{s}) = y(s, \phi) = +2\pi \left[ \frac{\lambda}{2} \Delta Z s^2 - c_s \frac{\lambda^3}{4} s^4 + \frac{\lambda}{4} \Delta Z_a s^2 \sin(2(\phi - \phi)) \right]
\]

where

\[
s = \sqrt{s_x^2 + s_y^2}
\]

\( c_s = \) Spherical aberration coefficient of the objective lens

\( \Delta Z = \) Defocus value

\( (+ \) sign for weak excitation of the objective lens)

\( \phi = \arctan \left( \frac{|s_x|}{|s_y|} \right) \)

\( \Delta Z_a = \) defocus difference caused by axial astigmatism
\[ \phi_0 = \text{reference angle for axial astigmatism} \]

(see Figure 2)

The effect of the aperture can be described by a step-function which has a value of 1 within the physical aperture and 0 outside of it.

\[ A(s) = \begin{cases} 
1 & s \text{ inside the aperture} \\
0 & s \text{ outside the aperture} 
\end{cases} \]

With these definitions, the coherent transfer function becomes:

\[ H(s) = A(s) \cdot e^{i\gamma(s)} \]

We will delay the description of the image until the functional form of the object wave function is described.

D. The Transmittance Function

Up to now, the wave function at a plane immediately behind the object has been the starting point of the analysis. The nature of this function and its relationship to the three-dimensional structure of the specimen must be explored.

The relationship between the functions at the entering and exiting surfaces of the specimen is complicated and reflects the many types of interaction that the electron can have with the object. One of the first divisions that can be made in these interactions is between elastic and
Figure 2

Definition of the variables found in the contrast transfer term, \( \sin\gamma(s) \)
\[ \Delta Z = \frac{\Delta Z_S + \Delta Z_L}{2} \]

\[ \Delta Z_a = \Delta Z_L - \Delta Z_S \]

\[ \Delta Z_{(L,S)} = \frac{1}{\lambda} \frac{1}{s^2_{(L,S)}} \]

\[ \lambda = \text{electron wavelength} \]
inelastic scattering events. Although the various types of inelastic processes are important to the materials scientist, they are, at present, a nuisance to the biological transmission electron microscopist. Since the focal length of a magnetic lens depends on the energy of the electrons traversing it, electrons which have experienced an energy loss will not follow the same trajectories as do the elastically scattered ones. The inelastically scattered electrons will therefore be out of focus with respect to electrons with no energy loss. Inelastic scattering is also "incoherent", that is to say that the scattered wave no longer has a predictable phase relationship to other waves scattered by the object. Thus imaging done with inelastically scattered electrons possessing a wide range of energies will not display any coherent, phase-contrast effects. The net effect of the inelastic electrons can be to reduce the contrast in the image by contributing an out-of-focus background of intensity. Various types of energy filters have been used in transmission electron microscopy to remove energy loss electrons and thereby improve contrast in the image, but such filters are not yet available on standard commercial instruments.

The elastically scattered electrons are the primary carriers of information in the TEM. While passing through the specimen, the electrons are scattered by the net Coulomb potential associated with the nuclei and electrons of the atoms making up the specimen. The probability for an
interaction or scattering event is very high for electrons, five to six orders of magnitude greater than that for X-rays for example. While traversing a specimen, it is possible that an electron, having been scattered by one atom in the object, could undergo further scattering. This possibility of either single scattering or multiple scattering is the basis for much of the controversy and uncertainty in image analysis in transmission electron microscopy. Image theory that assumes a single scattering event for the electron is termed kinematic, while theory that allows for multiple scattering is called dynamic.

Kinematic theory is by far the easier to employ mathematically. If an electron exits an object at a certain angle relative to its initial direction, a single event must have caused this to happen, in the kinematic view. The dynamic view holds that an unknown number of events lead to the final scattering direction. The path of the electron can not be determined (in a classical sense) by the final scattering angle.

Since we are neglecting inelastic and dynamic processes, the object can be described as a phase object, that is, an object which merely retards the electron wave as it passes through. This view of the object is consistent both with semi-classical electron scattering theory and with the full quantum mechanical scattering theory in the limit of small angle scattering (Schiff, 1956). Assuming a nor-
mally incident plane wave of amplitude $A$, the transmission function for a pure phase object would be 

$$T(x,y) = A e^{-i\sigma \phi(x,y)}$$

where $\phi(x,y)$ is the projection of the Coulomb potential distribution in the direction of the beam and $\sigma$ is the interaction constant ($\pi E/\lambda$, $E$ is the electron accelerating voltage). In most of biological electron microscopy, the objects are thin and the projected potential small. A further approximation can be made, the weak phase object (WPO) approximation. Under the assumption $\sigma \phi(x,y) \ll 1$, the transmission function becomes 

$$T(x,y) = A [1 - i\sigma \phi(x,y)].$$

We shall now see how this wave function changes as it passes through the electron microscope.

**E. The Image in Terms of the Object**

The tools needed for an investigation of the form of the image wavefunction in terms of the transmitted object wave and the transfer characteristics of the EM are now in our hands. Each assumption as to the nature of the object leads to a different mathematical form for the image. We will first assume that symmetrical bright-field imaging conditions are being used i.e. a symmetrical aperture is centered on the optical axis in the back focal plane and the illumination is not tilted. This is the most common imaging mode in use. We will analyze two classes of objects - the weak phase object and the weak mixed amplitude and phase object. Lastly, we will see how the use of a half-plane, or
single-side-band, aperture affects the image.

To simplify the mathematics and clarify the results, let us use a simple form for each of these two objects.

The weak phase object is represented by

$$\Psi(x) = 1 - iB \cos(2\pi g x + \phi), \quad B << 1 \quad (\text{II-7})$$

The mixed weak amplitude and phase object is represented by

$$\Psi(x) = 1 - A \cos(2\pi g_1 x + \phi_1) - iB \cos(2\pi g_2 x + \phi_2); \quad A, B << 1 \quad (\text{II-8})$$

More complicated (more realistic) wavefunctions can be constructed from a set of cosine functions with different periods and relative phase shifts, so we lose no generality by using these simple functions.

Abbe theory states that the wavefunction in the back focal plane of the objective lens is the Fourier transform of the object wavefunction. Multiplying this function by the coherent transfer function of the electron microscope accounts for the imaging defects. The function resulting from this multiplication is, by Abbe theory, the Fourier transform of the image wavefunction. Thus the image wavefunction can be calculated by using an inverse Fourier transform. Taking each object in turn, what is the image?

1. The weak phase object (WPO)

The Fourier transform of the WPO (Equation II-7) is composed of three terms - the undiffracted wave ($F(\theta) = 1$)
and two waves of equal strengths diffracted in opposite
directions \( F(\pm g) = \frac{\Omega B}{2} e^{\pm i\xi} \). The finite size of the ob-
jective aperture will either pass these diffracted waves or
absorb them. Assuming that they fall within the aperture,
and introducing the coherent contrast transfer function,
e\( i\gamma(s) \), we get the resultant wavefunction:

\[
\Phi(s) = \delta(s) - \frac{\Omega B}{2} e^{i\xi} \delta(s-g)e^{i\gamma(s)} + \frac{\Omega B}{2} e^{-i\xi} \delta(s+g)e^{i\gamma(s)}
\]

The image wavefunction is found by Fourier transforma-
tion and is:

\[
\Psi_i = 1 + \frac{\Omega B}{2} e^{i\xi} e^{i\gamma(g)} e^{i2\omega g x} + \frac{\Omega B}{2} e^{-i\xi} e^{i\gamma(-g)} e^{-i2\omega g x}
\]

Since photographic film is sensitive to the intensity
of the electrons and not their phase relationships, the
square of this wavefunction is what we seek.

\[
I = \Psi_i^* \Psi_i = 1 - 2\Omega B \cos(2\omega g x + \xi) \gamma(g) + \mathcal{O}(B^2)
\]

, since \( \gamma(g) = \gamma(-g) \)

The image is identical to the object except that the
factor of \( 2\sin\gamma(g) \) is modifying the cosine wave and a term
of the order \( B^2 \) is present. The latter term is negligible
for our weak phase object and the former may be corrected if
we know the form of \( \gamma(s) \). It can be seen that the coherent
transfer function is responsible for the observed image
variations (phase contrast), for if the information transfer
were "perfect", \( Y(s) \) would be 0 for all \( s \), and \( \sin Y(s) \) would also be 0, leading to an image with no variations in intensity. The effect of the \( \sin Y(s) \) term must be countered by processing of the image if the true object wavefunction is to be recovered.

2. The weak amplitude and phase object

The Fourier transform of the weak amplitude and phase object is similar to that of the weak phase object. Applying the contrast transfer function \( \exp(iY(s)) \) and assuming both \( g \) and \(-g\) fall within the objective aperture, we get the wave function at the exit plane of the objective lens:

\[
\phi(s) = 6(s-0) + \frac{A}{2} e^{iE_1} 6(s-g) e^{iY(s)} + \frac{A}{2} e^{-iE_1} 6(s+g) e^{iY(s)} + \frac{B}{2} e^{iE_2} 6(s-g) e^{iY(s)} + \frac{B}{2} e^{-iE_2} 6(s+g) e^{iY(s)}
\]

Again, the wavefunction at the image plane is the Fourier transform of this function and the recorded image is proportional to the intensity of this wave:

\[
I = 1 + 2A \cos Y(g) \cos(2wg^*x + \frac{E_1}{4}) - 2B \sin Y(g) \cos(2wg^*x + \frac{E_2}{4}) + O(A^2, B^2, AB)
\]

In this case, the image differs from the object by two factors, in addition to the negligible higher order terms. The amplitude modulation is transferred with weight \( 2\cos Y(s) \) while the phase modulation again is transferred with weight
2sinY(s). It can be seen that the image will show contrast at all values of Y(s), i.e. all defocus values. The exact nature of the contrast, however, will depend upon the value of defocus that was in effect when the image was formed.

3. Single-side-band image mode

In the above derivations, a symmetric aperture was assumed. If an aperture that passes only one of the pair of diffracted waves together with the undiffracted wave is used, the image will appear quite different. If two images are recorded, one using one half-plane and the other using the complementary half-plane, the images will be as follows for the weak phase object:

\[
I_1 = 1 - B\sin(2\pi g'x + \xi + Y) \quad \text{(II-14)}
\]
\[
I_2 = 1 + B\sin(2\pi g'x + \xi - Y)
\]

The advantage of these images (made with a single-side-band aperture) over those made with a full aperture is that the Y(s) value enters as a phase shift rather than an attenuating multiplicative factor. All spatial frequencies present in the object will lead to modulations of unit weight in the image.

The single-side-band (SSB) images for the mixed amplitude and phase objects are similar:

\[
I_1 = 1 + A\cos(2\pi g'x + \xi_1 + Y) - B\sin(2\pi g'x + \xi_2 + Y) \quad \text{(II-15)}
\]
\[
I_2 = 1 + A\cos(2\pi g'x + \xi_1 + Y) + B\sin(2\pi g'x + \xi_2 - Y)
\]
Unfortunately, the asymmetric shape of the aperture and the consequent asymmetric buildup of contamination on the aperture tends to produce new electrostatic fields which distort the image in an unknown manner. Until a practical aperture can be built which is free of distorting fields, the advantages of SSB microscopy will have to remain unrealized.

F. Summation and Conclusions

The theoretical path from object to image is almost as complex as the actual path of electrons from object to image. In order to relate image to object, we have made crucial approximations concerning the extent and number of interactions between the electrons and the object, the properties of the imaging system (linear and isoplanatic), and the form of the contrast transfer function. We have explored only one imaging mode in any detail because it accounts for the vast majority of experimental work undertaken. The following sections of the thesis will develop the image restoration and reconstruction methods which will be used to recover the object structure from the image. It should be kept in mind, though, that the answers are only as good as the approximations used in relating the projection of the object to the form of the image. Happily, practical implementation has shown that the objects which have been reconstructed are consistent with our knowledge of them based on methods other than electron microscopy.
III. Theory & Methods of Two-Dimensional Restoration and Three-Dimensional Reconstruction

A. Introduction

The practice of three-dimensional reconstruction in electron microscopy involves both a theoretical framework and a practical approach. The methods are still being mastered and there is a lot of room left for improvements and fresh ideas. In this section the theory and practice will be detailed.

From the results developed in the previous chapter, it can be seen that there is a relationship between the image of an object and the projection of the scattering potential of that object, within the bounds of the WPO approximation. The Fourier projection theorem identifies the Fourier transform of a 2-D projection with a central section of the 3-D Fourier transform. Let \( V'(x, y) = \int V(x, y, z) dz \) be the projection of the three-dimensional Coulomb potential function in the direction of the beam. The two-dimensional Fourier transform of this function is:

\[
F_1(S_x', S_y') = \iint V'(x, y) e^{-2\pi i (S_x' x + S_y' y)} \, dx \, dy \quad (III-1)
\]

A central section of a three-dimensional Fourier transform is a plane in Fourier space which passes through the origin. The \( F(S_x', S_y', 0) \) central section of the 3-D Fourier transform of \( V(x, y, z) \) would be:
\[
F(S_x, S_y, \theta) = \iiint V(x, y, z) e^{2\pi i (S_x \cdot x + S_y \cdot y + \theta \cdot z)} \, dx \, dy \, dz \\
= \iiint e^{2\pi i (S_x \cdot x + S_y \cdot y)} \, dx \, dy \, \int V(x, y, z) \, dz \\
= \iint e^{2\pi i (S_x \cdot x + S_y \cdot y)} \, V'(x, y) \, dx \, dy \\
= F_1(S_x, S_y)
\]

This identification is the basis for the various 3-D reconstruction methods based on Fourier transforms. By tilting the specimen, different projections of the structure are imaged. Each image gives rise to another plane in Fourier space. If enough of these planes are known, a three-dimensional Fourier synthesis, akin to that performed by X-ray crystallographers, can be done, and the true three-dimensional structure of the object will be recovered. Of course, the precise meaning of the phrases "enough ... planes" and "true ... structure" are at the heart of an analysis of any method, and these topics will be covered below.

As mentioned in chapter I, there do exist many algorithms for performing reconstructions from projections in real space, that is, without transforming the projection data to another space, be it Fourier or otherwise. Since our course has been to use Fourier methods, real space methods will not be explained. There is a healthy literature on these methods and comparisons between Fourier and non-Fourier approaches (Gordon and Herman, 1974; Brooks and DiChiro, 1976).
There are several reasons for the lack of use of direct-space methods for three-dimensional reconstruction in electron microscopy. One reason is human inertia. The initial proponents of reconstruction for EM were from an X-ray crystallography background and favored an approach that was familiar. The initial direct space work was not very convincing either, thus attitudes of pro-Fourier and anti-direct space were established. Another reason, and one with more objective weight, is that Fourier methods lend themselves to the practical aspects of imaging and reconstructing biological specimens using electrons. As discussed above, the images need to be corrected for the transfer characteristics of the microscope before they can be treated as projections of the structure. For many specimens, the radiation damage problem requires the use of a crystalline specimen and the symmetry of such crystals can be incorporated into the Fourier method at much less cost than into the current real space methods. For crystalline specimens, the Fourier transform performs an averaging function (Kuo, 1975) which can be extremely helpful in combating noise and structural disorder. This is another practical advantage that Fourier-based methods enjoy. Direct space methods have advantages also, but these advantages are more suited for reconstruction studies in other fields, such as nuclear medicine and X-ray tomography. The fact that all reconstructed points must be physically meaningful (positive mass or density at each point) is usable in the direct space
approach. Corrections for attenuation of the imaging radiation such as found in radionuclide studies are also possible in real space (Budinger and Gullberg, 1977). The lack of radiation damage and the ability to tilt the specimen arbitrarily through known tilt angles that characterize the non-EM work make the direct space methods more usable. The lack of Fourier series truncation error is very appealing, together with the related ability to reconstruct sharp edges.

In a sense, then, the relative separation of the reconstructors into two camps is a natural outgrowth of the different types of specimens, the different objectives of the reconstructions and the different backgrounds of the leaders in each field.

B. The Optical Diffractometer

What are the practical steps that must be taken to perform three-dimensional reconstructions? Figure 3 is a flow chart of the process. Before computer analysis is begun, an examination of the micrographs must be done with an optical diffraction apparatus. The optical system, to be described later, can provide a quick and inexpensive assessment of the quality of a micrograph. For negatively stained specimens which are imaged with higher than "minimal" exposure (greater than 10 to 50 electrons/Å² for example), the Fraunhofer diffraction pattern produced by the optical dif-
Figure 3

Organization of the three-dimensional reconstruction process
fractometer can indicate the degree of astigmatism and specimen drift in the image. Astigmatism leads to an elliptically shaped Fourier transform for the amorphous (carbon) supporting film as opposed to the circular pattern present for well-corrected micrographs (Thon, 1965). Specimen drift will lead to loss of Fourier components in the direction of drift.

Apart from these instrumental factors, the preservation of periodicity of crystalline or other symmetric specimens can be judged by the optical transform. The degree of preservation will be echoed by the highest diffraction orders (largest radii in Fourier space) present with observable intensity. Only the "best" plates need be retained for subsequent computer processing. Since the computer side of processing is comparatively slow and expensive, there is little justification for going ahead with less than the best specimens possible. A diligent effort put into optical diffractometry is rewarded many times over.

Specimens without peaked Fourier transforms (i.e. non-periodic objects) can be evaluated as to their preservation of Fourier components if two micrographs of the same specimen area taken in succession are available. Both plates can be put into the diffractometer at the same time and shifted relative to one another by a small amount. The Young's fringes thus produced will only extend to a Fourier radius which represents the preservation of common detail between
The illuminating beam used in the optical diffractometer is generally limited by an aperture to confine the diffraction analysis to an area of interest on the plate. When looking at isolated particles, rather than large crystals, aperturing is especially important. The diffraction arising from the specimen (a T4 virus tail, for example) is very weak when compared to that arising from a large surrounding area of support film. If an aperture is used which does not exclude as much of the background as possible, the ordered diffraction may be lost to view. Apertures that are continuously adjustable (circular and rectangular) would be ideal but a satisfactory substitute are custom-cut cardboard (or index card) apertures. These may be aligned with the specimen using a light box and secured on the plate with easy-release (drafting) tape. The use of small apertures leads to a noticeable broadening of the diffraction spots for ordered specimens, as is expected since the resulting transform is the convolution of the Fourier transform of the specimen with that of the aperture. The smaller the linear dimension of the aperture, the less peaked the transform of the aperture. This broadening is preferable to the total masking of the diffraction spot by support film noise.

The optical diffractometer can perform another useful function, this one related to the digitization of the image that is necessary before computer processing can be done.
As will be developed, crystalline specimens should be digitized, if possible, along a line parallel to one of the edges of the unit cells (or in the case of a helical specimen, along the helical axis). With images taken with electron exposures high enough to define statistically each unit cell, it is possible, although sometimes difficult in actual practice, to determine visually a direction parallel to one of the unit cell edges. With low dose micrographs, which are needed to do high-resolution studies, visual alignment is impossible since the image of a single unit cell is not statistically defined and the brain can not integrate nearly enough information to carry out the task. The optical diffractometer does such an integration, however, and from the alignment of the statistically defined reciprocal lattice spots, the needed orientation of the plate can be deduced.

Of course, some method of reproducibly relating transform rotation to the alignment of the physical plate in the diffractometer is needed. A simple holder that would force all plates to adopt the same orientation vis-à-vis the diffraction pattern recording camera would suffice. The need to rotate the plate in the diffractometer and accurately measure that rotation is not really that great. Such a scheme would be very useful for specimens which give eye-visible diffraction patterns because the alignment could be determined in real-time by rotating the plate until the diffraction pattern was horizontally or vertically (or both) aligned, for example. For weakly diffracting specimens or
high optical density plates with small apertures, the eye can not see the diffraction pattern and alignment would require the photographic recording of the pattern. At that point, the need to rotate the plate disappears.

A last point with regard to plate orientation and scanning. The decision as to what areas on a plate to scan are made on the basis of the resulting optical diffraction from those areas. For delimited objects (small crystal-like patches or helical specimens, for example), the subsequent identification of the "good" area is not too difficult, if one is careful and the image is statistically defined. One merely has to make a note of the location of the object on the plate, using its surroundings to describe that location. The large crystalline case is different. Location cues are either absent or easily lost. For these plates, as well as for any that are not statistically defined, a means of establishing a plate-based coordinate system is necessary. Preferably this coordinate system could be used to position the plate on the scanner but, barring that, the ability to construct scanning masks to overlay the plates and thus delimit areas to be scanned would work fine. Of course, trial and error approaches are popular and have been the foundation of a good portion of the work undertaken to date.

C. The Scanner

Once a set of micrographs has been chosen for further
study, it is necessary to translate the information they contain into a form suitable for use by the computer. The plates are scanned and the optical density of small areas are recorded along a regular lattice of sampling points by a scanning microdensitometer. The rotating drum scanner and the flat bed scanner are the commercially available types, each having advantages and disadvantages compared to the other. The rotating drum scanner (Sandor, 1974) requires that the images be on a flexible substrate (film) which means preparing a film negative if the original micrograph was on a glass plate. This intermediate step is subject to errors especially if an enlargement of the micrograph is undertaken. Enlargement entails the use of a magnifying optical system while a 1:1 contact print of the plate onto film could be done without using lenses. The imaging properties (Modulation Transfer Function - MTF) of the photo enlarging step could seriously affect the information transferred. Without knowledge of the MTF, this change could not be corrected. The drum scanner has a larger limit to the smallest area which can be sampled at each point. For high-resolution images taken at low magnification, this can lead to failure to be able to sample the negative at the Nyquist limit (see below). Overcoming this obstacle would require the production of an enlarged negative, thereby involving the MTF of the enlarging step. The advantages of this type of scanner are that it is less expensive to purchase and it scans faster.
The type of scanner used in all of our work at Berkeley is the flat-bed type. As the name implies, the plate (or film) to be scanned is placed on a horizontal bed or negative holder. This bed is driven by stepping motors past a stationary set of measurement optics. Two different scanners were used during the research done for this thesis. Initially, a "home-made" scanner located in the Remote Sensing unit of the UC Space Sciences Laboratory was used. This unit suffered from several known (and many unknown) defects, the most serious of which was a systematic error which evidenced itself in Fourier space (Figure 4). The line of intense diffraction is seen to rotate as the data is rotated. The spurious line is not consistent with the crystalline symmetry and was not found in the optical diffraction pattern of the original negative. A negative prepared by the computer from the scanned data did show these lines when placed in the optical diffractometer (Figure 4). Such "extra" components are not found in the data we get from the scanner that we presently use, a Perkin Elmer PDS scanner.

The PDS scanner, located in the UC Berkeley Astronomy department, is a tremendously useful piece of equipment, especially in the computer environment that it inhabits. Completely controllable by the computer, the scanner can handle up to 8 inch X 10 inch plates. Positional accuracy is claimed to be $\pm 1 \mu m$. The scanner can produce either transmission or optical density readings with 10 bit precision. Scanning apertures range from 5 $\mu m$ up to 50 $\mu m$. 
Evidence of scanner defect found in Fourier space

A) Computed diffraction pattern showing anomalous intensity introduced by "homemade" scanner. Display is the logarithm of the power spectrum.

B) Optical diffraction pattern of a computer-produced scan picture, showing the anomalous diffraction pattern.
circular or square. The position of the table is digitally displayed in \( \mu \text{m} \) relative to a user-defined origin. The transmission or OD reading is also displayed. A light pipe which bypasses the illuminating aperture allows a wide-field view of the object to be seen on a viewing screen while leaving intact the calibrated measurement light path. These features allow a plate to be aligned along the orthogonal scanning axes with great ease. The viewer is marked with a cross-hair ruling which is aligned with the scanning axes. A scribed line, tape edge or scanning mask edge can be clearly seen in the viewing screen. After aligning this cue by eye with one of the table’s movements, exact alignment can be found as follows. First, center the edge of the scribed line, etc. in the viewer. Next, translate the table along this edge so that the relative tilt between edge and table motion can be seen. Rotate the table in the appropriate direction and re-center the edge. Repeat the steps until the edge and table motion are parallel. Not many iterations are needed to achieve this.

The table position readouts, together with the viewing screen, provide an accurate and easy way to measure distances. The centers of scanning masks can also be located. The X-coordinates of the mask sides perpendicular to the X-direction are read and their average calculated; likewise for the Y-coordinates. These average coordinates are those of the center of the scan mask. The available software control of the table position can be used to move the table to
the calculated center.

D. Scanning

The scanning process is more profound than the ease of operation of the scanner might suggest. The real aim of scanning is to present the continuously varying optical density of the micrograph by a set of discretely sampled values. Since the dimensions of the sampling aperture and the sampling grid approach the physical size of the silver grains which make up the photographic image, a clear understanding of all the variables involved is important.

The first understanding to develop is the mathematical relationship between sampled functions and continuous functions. Realistically, how many samples do we need and how close together do we need them in order to represent accurately a continuous function? Since we are concerned with the Fourier transforms of our images, the question could be reworded: how much do we need to know in order to accurately represent the Fourier transform of a function? These questions are more easily answered in Fourier space than in real space. The one-dimensional case will be treated for simplicity.

A continuous function \( g(x) \) has a Fourier transform \( G(s) \) as shown graphically in Figure 5. We are interested in sampling this function on a regular grid, with spacing \( \Delta x \), that is, we want functional values \( g(x_i), \ x_n = n \Delta x \).
Comparison of the continuous and discrete Fourier transforms

a) Continuous object function, \( g(x) \)

b) Its continuous Fourier transform \( G(s) \), band-limited at \( s_{\text{max}} \)

c) and d) A comb function for sampling and its Fourier transform

e) The "discrete" object results from multiplying a) and c)

f) Its continuous transform results from convolving b) with d). \( \Delta x \) was chosen so that the band-limited transforms do not overlap.
Real Space

\[ g(x) \]

Object function

Fourier Space

\[ G(s) \]

\[ S_{\text{max}} \]

Sampling (comb) function

\[ s(x) \]

\[ x_i \]

Sampled function

\[ g(x_i) \]

\[ \frac{-1}{\Delta x} \]

\[ \frac{1}{\Delta x} \]

\[ S(s) \]
n=...-1,0,1,2,... . Mathematically, this is realized by multiplying \( g(x) \) by the comb function (Goodman, 1968)

\[
 s(x) = \sum_{n} \delta(x - n \Delta x) \tag{III-3}
\]

What is the relationship between \( G(s) \) and the Fourier transform of the sampled function, \( g(x) \ast s(x) \)?

\[
 F[g(x) \ast s(x)] = F[g(x)] \ast F[s(x)] \tag{III-4}
\]

where * denotes convolution.

\[
 G_{\text{Samp}}(s) = G(s) \ast S(s) \tag{III-5}
\]

where \( S(s) \) is the Fourier transform of the comb function.

The Fourier transform of a comb function with spacing \( \Delta x \) is another comb function but with spacing \( \frac{1}{\Delta x} \).

\[
 S(s) = \text{comb}(\Delta x \ast s) = \sum_{n} \delta(s - \frac{n}{\Delta x}) \tag{III-6}
\]

Combining equation 6 with equation 5, and making use of the sifting property (Goodman, 1968) of the delta function, we get

\[
 G_{\text{Samp}}(s) = \sum_{n} G(s - \frac{n}{\Delta x}) \tag{III-7}
\]

Let us look at the meaning of this last equation. When \( n=0 \), \( G_{\text{Samp}} \) is identical to \( G(s) \). When \( n=1 \), we have \( G(s) \) shifted to the right by \( \frac{n}{\Delta x} \). When \( n=-1 \), \( G(s) \) is shifted to the left by an equal amount. The right side of equation 7 can most easily be visualized by imagining that the function
G(s) is "plunked down" at the various origins, \( \frac{\Delta x}{n}, \ n = \ldots , -1, 0, 1, \ldots \). These origins correspond to the teeth of the reciprocal comb function \( S(s) \). Since our goal is to approximate \( G(s) \), how can this be done, given the form of \( G_{\text{samp}}(s) \)? One method would be to use that part of \( G_{\text{samp}}(s) \) that was centered at \( n=0 \), since that represented \( G(s) \) directly. Whether it is possible to excise that portion of \( G_{\text{samp}} \) without "contamination" from the other centers depends entirely upon the properties of \( G(s) \) (see Figure 5).

If \( G(s) \) is non-zero over a limited area, separation is possible. Functions whose transforms have this property are aptly named band-limited functions. If \( S_{\text{max}} \) is the frequency of the highest resolution non-zero Fourier component of such a function, we see from Figure 5 that a reciprocal space comb with spacing \( \Delta x < \frac{1}{2S_{\text{max}}} \) will ensure that there is no overlap between the continuous transform laid down at one tooth and that laid down at the next. Hence we can retrieve a single copy of the continuous transform without contamination from neighboring copies.

Functions that are not band-limited have Fourier transforms which are non-zero for all \( s \). No comb separation is sufficient to prevent the overlap of neighboring continuous transforms in the "sampled" transform (Figure 5). This overlap, or aliasing effect, means that the true continuous transform can not be retrieved from the transform of the
Figure 6

Aliasing effects in sampled functions

The continuous transform of a sampled non-band-limited function has a value $G_{\text{samp}}(s)$ which is the sum of many terms of the continuous transform of the unsampled function. There is no finite sampling interval which will avoid overlap.
sampled function. Each coefficient will contain the "correct" value together with other frequency components. These other components are using the "alias" of another frequency, i.e. they appear at frequencies other than their "true" frequencies. As such, these overlapping terms are called "aliasing" terms.

We can rewrite equation 7 as

$$G_{\text{samp}} = G(s) + G(s + \frac{1}{\Delta x}) + G(s + \frac{2}{\Delta x}) + \ldots + G(s - \frac{1}{\Delta x}) + G(s - \frac{2}{\Delta x}) + \ldots$$

where $G(s)$ is the "correct" value and the other terms are "errors".

The extra components are the aliasing terms. Aliasing terms are always present but are only important if the terms $G(s + \frac{n}{\Delta x})$, $n=+1,+2,+3,\ldots$ are non-zero and comparable to $G(s)$ in magnitude. The smaller the $\Delta x$, the more different the frequency of the aliasing terms from the frequency of the "correct" term. This means, in general, the smaller the magnitude of the aliasing terms. The problem of contamination also applies if a band-limited function is not sampled finely enough, ($\Delta x > \frac{1}{2S_{\text{max}}}$). In this case, $G(s + \frac{n}{\Delta x})$ will still be within the band limit or set of non-zero Fourier coefficients.

A spatially-limited function can not be band-limited just as a band-limited function can not be spatially-limited. Since the specimens we view in the microscope are
bounded (i.e. have compact support), will it be impossible to exactly represent the images by a sampled function? Yes, it will be impossible in a strict mathematical sense, but an accurate representation is possible. The transforms of specimens encountered in biological work are pseudo-band-limited in that the magnitude of the transform at large \( s \) values decreases rapidly. The aliasing terms, \( G(s + \frac{n}{\Delta x}) \), will be quite small and can be considered another source of noise, albeit noise which is correlated to the structure.

Since there is no sampling interval that will avoid aliasing, what interval should be used? The aliasing error increases with increasing radius in Fourier space. This occurs because the higher resolution terms of the central transform are aliased with lower resolution (more intense) terms from the neighboring transform (Figure 7). By using a finer sampling, the aliasing terms are shifted to a higher frequency and will be less intense. If \( \Delta x \) was used for the sampling grid spacing, then the largest aliasing term for the \( (2\Delta x)^{-1} \) Fourier component (the highest frequency component that is calculated by the discrete Fourier transform) will be the \( (-2\Delta x)^{-1} \) component (the Freidel symmetry-related component). Frequencies greater than this will have larger (lower frequency) errors. Frequencies smaller than this will have smaller (higher frequency) errors. This leads to a rule of thumb for scanning - if the structural preservation extends to \( x \) \( \AA \), use a sampling grid spacing of less than \( \frac{x}{2} \). Remember that decreasing the sample spacing merely
Figure 7

Effect of the sampling interval on the aliasing error

A) Insufficient sampling of $\Delta x_1$ leads to aliasing at frequencies $S_a$, $S_b$, and $S_c$.

B) Increased sampling $\Delta x_2$ improves the result. $S_a$ is alias free, while $S_b$ and $S_c$ are aliased with higher frequency information (which has lower power).
shifts the aliasing terms to higher frequencies so that decreasing it beyond what is needed to avoid significant aliasing errors in the Fourier components of interest is a literal waste of time. Later, when Fourier transforms are calculated from the over-sampled data, there will be a waste of money. Sampling somewhere between $\frac{X}{2}$ and $\frac{X}{3}$ is sufficient. There is the factor of interpolation "error" to consider (see below) which is related to the fineness of the real space sampling. $\frac{X}{3}$ sampling is sufficient to minimize this error.

There are other factors involved in scanning besides that of aliasing. Rather than sampling the plate's optical density at an infinitesimally small point, the optical density of a finite area is determined. This averaging effect can be approximated as a convolution of the image with an aperture function (DeRosier and Moore, 1970). The Fourier transform of this convolution will be the product of the Fourier transform of the image with the Fourier transform of the aperture. For a circular aperture of radius $r$, the aperture transform would be $A(s) = \frac{2J_1 (2\pi rs)}{2\pi rs}$, where $J_1$ is the Bessel function of order 1. The maximum value $r$ can assume without overlapping sample areas is half the sampling distance, $\frac{\Delta X}{2}$. The highest resolution Fourier component which can be calculated using the discrete Fourier transform would be $\frac{1}{2\Delta X}$. The value of $A(s)$ under these conditions would be $\frac{2J_1 \left( \frac{\pi r}{2} \right)}{\pi r} \approx 0.72$. Making the aperture smaller would
increase the value of \( A(s) \) for a given Fourier term. In the above examples, cutting the aperture in half would increase \( A(s) \) to \( \approx .93 \). In practice, the data is oversampled so that the highest resolution Fourier component used is not \( \frac{1}{2\Delta x} \) but perhaps \( \frac{1}{3\Delta x} \) or \( \frac{1}{4\Delta x} \). With the largest aperture possible and these frequencies, \( A(s) \) is, respectively, \( \approx .88 \) and \( \approx .93 \).

Figure 8 plots \( A(s) \) as a function of the two variables, aperture size and spatial frequency. The advantage of using the largest aperture possible is that the greatest possible amount of information about the object is included. Halving the aperture quarters the area contributing to the sample. The extra area should halve the statistical noise, which more than makes up for the worst-case 28% amplitude loss. The proper amplitudes can be restored without fear of significant noise amplification by dividing by \( A(s) \).

The image has now been reduced to numbers. We have some idea of the relationship that the numbers bear to the original plate. The task of image restoration and three-dimensional reconstruction lies ahead. Off to the computer!

E. The Discrete Fourier Transform

The discrete Fourier transform (DFT) is the backbone of 2-D image restoration and 3-D reconstruction methods. The DFT is an approximation to the analytic continuous Fourier transform. Its virtue is that it can be calculated with a digital computer. An understanding of the properties of the
Figure 8

Amplitude of the Fourier transform versus spatial frequency for several circular sampling apertures.

This approximates the transfer characteristics caused by the use of a finite sampling aperture.
SPATIAL FREQUENCY (in units of \((2 \times \text{sampling interval})^{-1}\))

APERTURE RADIUS (in sampling intervals)
DFT is essential to a good understanding of image processing as applied to electron micrographs.

Equation III-9 defines the DFT (Brigham, 1973)

\[
G\left(\frac{n}{N}\right) = \sum_{k=0}^{N-1} g(kT)e^{-\frac{-i2\pi nk}{N}}, n = 0, 1, \ldots, N-1 \quad (III-9)
\]

where \(g(kT), k=0, 1, \ldots, N-1\) are samples of the continuous function \(g(x)\).

The inverse transformation is defined similarly:

\[
g(kT) = \frac{1}{N} \sum_{n=0}^{N-1} G\left(\frac{n}{N}\right)e^{\frac{i2\pi nk}{N}}, k = 0, 1, \ldots, N-1 \quad (III-10)
\]

This transform pair shares the familiar properties of the continuous Fourier transform pair. The most important properties from the view of the reconstruction problem are linearity, coordinate origin shifting, correlation, convolution, and autocorrelation. Brigham (1973) develops the discrete equations in a lucid manner. Only the results will be considered here.

Linearity assures us that the DFT of a sum of two functions is the sum of the DFT's of each function.

\[
\text{DFT} (f(kT)+g(kT)) = \text{DFT} (f(kT)) + \text{DFT} (g(kT)) \quad (III-11)
\]

The coordinate origin shifting theorem states that the DFT of a translated function can be found by modifying each term of the DFT of the untranslated function by an appropri-
ate phase term:

\[
\text{if } \text{DFT} \left( f(kT) \right) = F \left( \frac{n}{NT} \right) \quad (III-12)
\]

then \( \text{DFT} \left( f(kT-x) \right) = F \left( \frac{n}{NT} \right) \cdot e^{\frac{-i2\pi nx}{NT}} \)

Alternatively, we can translate a function by modifying its DFT and then performing an inverse transform. An analogous relationship holds for DFT's whose frequencies have been translated:

\[
\text{DFT}^{-1} \left[ F \left( \frac{n}{NT} - s \right) \right] = f(kT) \cdot e^{i2\pi kTs} \quad (III-13)
\]

The discrete correlation of two functions is defined as follows:

\[
\sum_{i=0}^{N-1} f(iT)g(kT + iT) = h(kT) \quad (III-14)
\]

If \( \text{DFT}(f) = F \) and \( \text{DFT}(g) = G \), then the discrete correlation theorem states:

\[
\text{DFT} \left[ h(kT) \right] = F^t \left( \frac{n}{NT} \right) \cdot G \left( \frac{n}{NT} \right) \quad (III-15)
\]

where \( t \) denotes complex conjugation.

Stated another way, the discrete correlation of two functions can be found from the inverse DFT of the product of the (modified) DFT's of each function:

\[
\sum_{i=0}^{N-1} f(iT)g(kT + iT) = \text{DFT}^{-1} \left[ F^t \cdot G \right] \quad (III-15)
\]
The auto-correlation theorem is basically the same except that the functions \( f \) and \( g \) are the same so that \( F^t \cdot G \) becomes \( F^t F = \) the modulus of the DFT.

The discrete convolution of two functions is defined as follows:

\[
\sum_{i=0}^{N-1} f(iT) g(kT-iT) = h(kT) \quad (III-17)
\]

Similar to the correlation result, we have a simpler relationship in the Fourier domain:

\[
F\left(\frac{n}{NT}\right) \cdot G\left(\frac{n}{NT}\right) = H\left(\frac{n}{NT}\right) \quad (III-18)
\]

One last topic involving the properties of the DFT should be mentioned. The inversion formula for the DFT (Equation III-9) can be calculated by an alternate formula (Brigham, 1973)

\[
g(kT) = \frac{1}{N} \left[ \sum_{n=0}^{N-1} G^t \left(\frac{n}{NT}\right) e^{-\frac{i2\pi nk}{N}} \right]^t
\]

The power of this formulation is that it expresses the inverse DFT of \( G \) as the complex conjugate of a forward DFT performed on the complex conjugate of \( G \). Thus, only the forward DFT need be programmed, debugged, and optimized.

Definitional equations for the DFT are adequate for purposes of calculation but do not impart a feel for the function and its properties. Continuous transform theory can be used to illustrate the discrete transform as a
special case of the continuous transform, and in so doing make the relationship between the continuous transform and DFT clearer (Brigham, 1973). The pitfalls to be avoided in applying the discrete transform will also be illuminated.

As shown in Figure 9, a continuous function \( f(x) \) is associated with a continuous Fourier transform \( F(s) \). The discrete function \( F(kT) \) is formed by sampling \( f(x) \) (or multiplying \( f(x) \) by a comb of delta functions \( s(x) \), spaced \( T \) apart). The discrete function is further modified by an aperture function \( a(x) \) which limits the samples to a finite number. Thus the input data to a DFT program would be \( f(x) \cdot s(x) \cdot a(x) \). The continuous transform of this composite function would be, by the convolution theorem, \([F(s) \ast S(s)] \ast A(s)\), where \( \ast \) denotes convolution. The first convolution \( F(s) \ast S(s) \) has the effect of "laying down" the transform \( F(s) \) at each of the delta functions comprising \( S(s) \), their being \( \frac{1}{T} \) apart. This first convolution is then convolved with the transform of a step function (in this one-dimensional example) which is a sinc function (Goodman, 1968). The DFT only calculates regularly spaced samples in Fourier space. Again, a comb function in Fourier space, \( S_2(s) \), multiplied to the results from above can represent this situation. The ramification of this sampling is that the sampled transform is really the continuous transform of \( F^{-1} [[[F(s) \ast S(s)] \ast A(s)] \ast S_2(s)] = f_2(x) \) where \( F^{-1} \) represents the continuous inverse Fourier transform. Simplifying, we have:
Figure 9

The discrete Fourier transform pair

a) The continuous function $f(x)$

b) The sampling function $s(x)$ which is similar to the Fourier space sampling function $S_2(x)$

c) The real space aperture function $a(x)$ which delimits the object being subjected to the DFT

d) The continuous Fourier transform of a)

e) The Fourier transform of the real space sampling function, which is similar to the inverse Fourier transform of the Fourier space sampling function

f) Fourier transform of aperture function, a sinc function

g) The continuous Fourier transform of $f(x) \cdot s(x) \cdot a(x)$. The central portion of width $\frac{1}{T}$, sampled at $\frac{1}{NT}$ is the discrete Fourier transform.

h) The continuous inverse Fourier transform of the (sampled) discrete Fourier transform. The Fourier sampling has resulted in the periodic continuation of $f(x) \cdot s(x) \cdot a(x)$. The central portion of width $NT$, sampled at $T$ is the discrete inverse Fourier transform.
The discrete transform

The discrete object
\[ f_2(x) = [f(x) \ast s(x) \ast a(x)] \ast s_2(x) \]

\( s_2(x) \) being a comb function in real space of spacing \( NT = T_0 \). Instead of our original apertured and sampled function, we now have a periodic array of the same functions.

The discrete Fourier pair can be seen as the continuous transform pair of periodic functions in both real and reciprocal spaces. We can use our knowledge of continuous Fourier transforms if we keep in mind that our finite function is only one period of a periodically continued function and our calculated transform is but one period of a periodically continued transform.

How do the various modifications to the continuous transform influence the relationship between the DFT and the CFT (continuous Fourier transform)? We have already seen that real space sampling leads to Fourier domain aliasing. It should be possible to make the sampling interval small enough to avoid practical problems. The finite length of the DFT corresponds to a convolution with a sinc function. The wider the aperture, the narrower the sinc function and the less effect the convolution will have on the discrete transform. The discrete sampling of Fourier space can make little difference (if the continuous transform itself is continuous) or it can be a disaster (if the continuous transform is discrete and you are not sampling the correct
1) The relationship between sample interval, array size and Fourier sampling interval

A great deal of confusion arises among some people when the relationship between array size and "fineness" of the DFT is discussed. There is a naive belief that bigger is better and hence a larger DFT necessarily leads to a more finely sampled Fourier transform. This is not true, as we shall see.

There are two distinct samplings involved in the discrete Fourier transform, real space sampling and Fourier space sampling. The real space data is usually sampled at a known interval, T. The sampling in Fourier space is then equal to \( \frac{1}{NT} \) where N is the number of samples included in the DFT. \( NT = T_0 \) can be considered the length of the object being transformed, or more loosely, the length of the DFT. Decreasing the sampling interval in real space \( (T_2 < T_1) \) will more finely sample the real space function but will lead to coarser sampling in Fourier space \( \left( \frac{1}{NT_2} > \frac{1}{NT_1} \right) \) for the same size DFT, i.e. for the same number of samples. In other words, we are sampling a "shorter" DFT for the same number of sample points. For a scan of a given length of specimen, decreasing the sampling interval and thus increasing the number of samples will only increase the maximum frequency calculated by the DFT, but will keep the Fourier space sampling constant \( (T_0 \text{ constant}) \). This is the reason for the
prohibition against needlessly oversampling a specimen. If the maximum preserved periodicity in the specimen can be calculated without significant aliasing using a certain sampling interval, there is no need to decrease the interval further.

The spacing of Fourier sampling is decreased when a "larger" DFT is calculated, i.e. when NT represents a longer distance. Given that T is usually fixed by the requirement of preventing aliasing, a larger N' must be used. In this case, a bigger DFT does lead to increased sampling in Fourier space. Often, a specimen is of limited spatial extent and it is not possible to extract a large number of samples from it. As detailed below, it is possible to "pad" a larger array with some appropriately chosen constant value in order to increase the apparent "length" of a DFT. The resulting DFT will indeed be more finely sampled than if the specimen data alone were used.

2) The DFT of a periodic specimen

A discussion of the DFT of a periodic specimen is both germane and illuminating in regards to all the topics covered above. Assume we have a periodic function \( f(x) = \cos(2\pi g^* x) \). The wavelength of this function is \( \frac{1}{g} \) and its continuous Fourier transform would have two peaks, one at \( +g \) and the other at \( -g \) (see Figure 10). Assume we sample an integral number of periods, \( m \), of this function with a sample spacing of \( T \), for a total of \( N \) samples. Then the
Figure 10

The discrete Fourier transform of a cosine wave

A) A cosine wave of m periods in N samples

B) The continuous transform of this apertured wave, showing the aliasing effect and the convolved sinc function

C) A close-up of the transform centered at \( \frac{m}{T_0} \). Note that neighboring samples fall at the zeroes of the sinc function.
length of the object which is sampled, $T_0$, will be equal to 
$NT$, which in turn is the same distance as $m(\frac{1}{3})$. We have 
$m(\frac{1}{g}) = NT = T_0$. Fourier coefficients are sampled at 
$\frac{1}{NT} = \frac{1}{T_0} = \frac{g}{m}$, thus the $m'$th coefficient of the DFT will sam-
ple the continuous transform at $g$. The discrete transform 
will be the convolution of delta functions at $g$ and $-g$ with 
a sinc function (truncation effect). We will neglect alias-
ing effects because they will not interfere if we choose $T$ 
correctly. The sinc function which corresponds to an aper-
ture of width $T_0$ is, apart from a scale factor, 
$\frac{\sin(\pi T_0 f)}{\pi T_0 f}$. 
Convolving this with delta functions at $g$ and $-g$ is 
equivalent to laying down $\frac{\sin(\pi T_0 f)}{\pi T_0 f}$ at both $g$ and $-g$. The 
sinc function has zeroes at frequencies $\frac{1}{T_0}$ from its center. 
Since this function is centered at $g$ and $-g$, the zeroes will 
occur at frequencies $g - \frac{i}{T_0}, i = ..., -2, -1, 1, 2, ...$. The 
discrete transform is sampled at $g = \frac{m'}{T_0}$ and hence sam-
ples the maximum value of the convolution peak. The other 
sampled coefficients will all be zero since only the zeroes 
of the convolution $(g - \frac{i}{T_0} = \frac{m'}{T_0} - \frac{i}{T_0} = m' \cdot \frac{1}{T_0} = a$ sample point) 
will be seen. Thus, the DFT of an integral number of 
periods of a periodic function is identical, apart from 
scaling factors, to the continuous transform.

The above discussion neglected aliasing errors. Where 
would the aliasing terms lie? With "neighboring" transforms 
lying $\frac{1}{T} = \frac{N}{T_0}$ apart, the nearest aliasing terms would come
from the first "neighboring" transforms centered at $\frac{N}{T\theta}$. These alias terms would be at $\pm |\frac{N}{T\theta} - \frac{m}{T\theta}| = \pm \frac{|(N-m)|}{T\theta}$ (see Figure 10). The Nyquist sampling rate leads to the requirement that the $N$ samples cover at most $m = \frac{N}{2}$ periods of the function to avoid aliasing. If more periods are sampled, then this frequency will be aliased and will appear as a lower frequency ($N-m < m$). The relationship between $N$, the number of samples; $m$, the number of periods; and $N-m$, the position of the nearest aliasing term, can be put to good use. It is shown below that, by proper choice of $m$ and $N$, it is possible to compute Fourier coefficients of crystalline specimens to a resolution beyond that predicted by the sampling theorem, using the aliasing effect.

The Fourier transform of a crystalline object is characterized by the existence of a reciprocal lattice. For an infinite crystal, the only non-zero Fourier coefficients lie on the reciprocal lattice. In the discrete transform case, the reciprocal lattice concept still applies. If an integral number of unit cells is included in the Fourier transform, again only the reciprocal lattice coefficients will be non-zero. If the function (crystal) is undersampled, however, there will be aliasing components in the transform. We can classify a non-zero coefficient as being either non-aliased or aliased. Since most of the coefficients calculated by the DFT are not on the reciprocal lat-
tice, we can avoid aliasing problems by conspiring to have the aliasing terms end up at calculated frequencies that are not on the reciprocal lattice. Placing aliased reciprocal lattice points off the reciprocal lattice sounds contradictory but this can be obtained through a thoughtful choice for the DFT size, the number of unit cells sampled, and the number of samples per unit cell.

For the sake of simplicity, we will describe the Fourier coefficients by their number rather than their spatial frequency. Since all sampled coefficients are of frequencies $j \cdot \frac{1}{T_0}$, using the $j$ description is simpler and unambiguous. In the crystalline case, $m$ represents the number of repeats of the basic structure which are sampled by the $N$ points. If $m$ is a factor of $N$, then the aliased terms will coincide with the unaliased terms. The aliased terms will be found at $N-i'm$ for the $i$-th order. Since $N = mk$, this becomes $mk - mi = m(k-i)\) which is the unaliased position of the $(k-i)$-th order. If $m$ is not a factor of $N$, $N = m(k+f)$. The aliased terms will fall at $N-i'm = mk + mf - im = m(k-i+f)$ which does not correspond to any members of the unaliased reciprocal lattice. Thus, in the absence of noise (intensity off the reciprocal lattice), it is possible to recover frequencies higher than that predicted by the sampling theorem. The practical implementation of this scheme would be limited by the weakness of the high-frequency components and the strength of the low frequency noise. Since the aliased high-frequency
components appear "off" the reciprocal lattice where noise would show up, the relative intensities of signal and noise would determine whether the crystalline reflection would be detectable and, if so, how much of the apparent signal was caused by noise.

The lack of convolution problems for DFT's of samples containing an integral number of periods is motivation to try to achieve this sampling but with limited knowledge of the specimen and/or limited choices for scanning steps on the microdensitometer, the chances of reaching this goal are nil. An examination of the DFT of a non-integral number of repeats of a periodic function will demonstrate what we can expect.

Using the notation from above, we assume our scanning interval of \( NT = T \) contains \( m + f \) periods of length \( \frac{1}{g} \) of the function \( \cos(2\pi g x) \), where \( 0 < f < 1 \). The DFT again samples reciprocal space at a spacing of \( \frac{1}{T} \), and the continuous transform would consist of delta functions at \( \pm g \), convolved with the sinc function characteristic of the finite length \( T \) which was sampled. The difference between this case with a non-integral number of periods and the previous one is that \( \frac{1}{g} \) is not a sampled Fourier component. The object length, \( T \), is equal to \( (m + f) \cdot \frac{1}{g} \). Solving for \( g \), we get: \( g = (m + f) \cdot \frac{1}{T} \). Since reciprocal space is sampled at integer multiples of \( \frac{1}{T} \), the \( g \) vector would not be sampled. As in the integral period case, the convolution of the delta
function about $+g$ leads to a $\frac{\sin(\pi T_0 s)}{\pi T_0 s}$ function being centered on $+g$. The zeroes of this function still occur at $\Delta f_s = \frac{1}{T_0}$ from the center, but since $+g$ are not sampled, neither will the zeroes of the convolution be sampled. The sinc function will be sampled at $S_1 = \frac{f}{T_0}, S_2 = \frac{1-f}{T_0}$ in the main lobe of the function and $\Delta s = \frac{1}{T_0}$ in the other lobes. Instead of having one sample at the maximum, we now have two sample-points straddling the maximum.

As can be seen from Figure 11, the true maximum cannot be estimated from linear interpolation between the two sampled locations. If $f$, the fraction of a period sampled, were known, the true peak height could be calculated. An estimate for $f$ can be generated by looking at the ratio $\frac{F(s_1)}{F(s_2)}$. Figure 12 is a plot of this ratio versus $f$. The larger of $F(s_1), F(s_2)$ can be corrected by a simple multiplicative factor based on the value of $f$. A plot of this factor versus the $\frac{F(s_1)}{F(s_2)}$ ratio is also shown in Figure 12. This analysis ignores noise and other sources of nearby intense diffraction which, through the convolution operation, will influence the sampled coefficients. Thus, any changes made to the coefficients on the basis of delta function convolution must be treated with respect.

The phases of two neighboring samples should be the same if they are derived from a single delta function. This
Figure 11

Part of the discrete Fourier transform of a non-integral number of periods of a cosine wave. The inclusion of \((m+f)\) periods shifts the central peak of the convolved sinc function off the sampling lattice.
Figure 12

Correction graphs for a mis-sampled delta function

A) Displacement of a delta function from the larger of the two DFT samples straddling its actual (unsampled) location versus the ratio of these two samples.

B) Multiplicative correction factor to apply to the larger of the straddling DFT samples to estimate the true delta function amplitude versus the sample ratio.
A

DISPLACEMENT OF THE DELTA FUNCTION

(in Fourier sampling units)

RATIO OF STRADDLING SAMPLES

B

MULTIPLICATIVE AMPLITUDE CORRECTION FACTOR

RATIO OF STRADDLING SAMPLES

XBL793-3311
"phase" criterion can be used when deciding if a presumed split peak is in fact derived from a single delta function. Remember that a split peak might represent a missampled delta function on the reciprocal lattice, or it might represent a missampled noise coefficient. The existence of a constant phase relationship over two adjacent discrete Fourier coefficients does not discriminate between signal and noise. It merely discriminates between a missampled delta function and two juxtaposed (noise) coefficients.

The cases considered above were all one-dimensional. A delta function that is not sampled by the two-dimensional DFT behaves in a similar manner. Rather than involving the convolution of a single sinc function, the two-dimensional case has a convolution of two sinc functions. The sinc function convolutions are independent and their combined effect on a single delta function can be considered in two parts. Figure 13 shows a typical result. Instead of two DFT samples in the main peak, there are now four adjacent samples. The relative strengths of these four sample points can be calculated from successive applications of sinc convolution. Let the Fourier space sampling in one direction be \( \Delta S_x \); the other, \( \Delta S_y \). In one direction, the delta function will lie \( f \Delta S_x \) from two samples and \( (1-f)\Delta S_x \) from the other two. Similarly, the separations in the other direction are \( g \Delta S_y \) and \( (1-g)\Delta S_y \). The resulting amplitude at any of the four points can be predicted on the basis of first applying a sinc function convolution in one direction,
Two-dimensional convolution of a mis-sampled delta function

A) Graphical representation showing the four ($F_1$ to $F_4$) DFT samples obtained from a mis-sampled delta function. Two orthogonal sinc function convolutions lead to the observed values. The displacement of the delta function is expressed as fractions $f$ and $g$ of a sampling unit from one of the four DFT samples.

B) A diagram showing the relationship of the mis-sampled delta function to its neighboring DFT samples.
taking into account the delta function-sample separation in that direction, followed by a convolution correction in the perpendicular direction. It is easy to see that the ratio of two sampled coefficients that lie the same distance in one direction from the delta function should reflect their relative separation from the delta function in the orthogonal direction. To use this information to divine the actual delta function amplitude is straightforward. Although each of the four samples can be used to estimate the correct amplitude, I use the coefficient with the largest amplitude if they differ greatly among themselves (which they will if the delta function is much closer to one sample point than it is to the others, i.e. the fractions \( f \) and \( g \) are small) because the relative contribution of noise will be the least. When all the sampled coefficients are similar (\( f \) and \( g \) close to 0.5), then all four samples should be used to get four estimates for the correct amplitude.

The first step is to calculate the two sampled coefficient ratios, \( \frac{F_1}{F_3} \) and \( \frac{F_2}{F_4} \) (see Figure 13). Both ratios should be identical except for the influence of noise. They should pinpoint the fraction \( f \), as in the one-dimensional case. Calculating \( \frac{F_1}{F_2} \) and \( \frac{F_3}{F_4} \) gives the fraction \( g \). One can now correct the sampled amplitude(s) by first using the \( f \)-based one-dimensional correction factor on \( F_1 \) and \( F_2 \). A \((1-f)\)-based correction could be used on \( F_3 \) and \( F_4 \) also. Next, the
$g$-based correction can be made to $F_1$ and $F_3$, while the $(1-g)$-based correction can be made to $F_2$ and $F_4$. Rather than calculating $f$'s and $g$'s, then getting factors, Figure 12 can be used to find the correction factors directly. In the case of real (noisy) data, one must use judgement as to which corrected samples to use in determining the actual amplitude.

The cases considered above were characterized by an aperture length equal to the length of the sample array. For particles or small patches or perhaps for convenience, the array length might be larger than the data "aperture". Again, assume $N$ samples at a spacing of $T$ for an array length of $NT = T_0$. The aperture now consists of $n$ samples, $T_a = nT < T_0$. Assume a function $\cos(2\pi g^*x)$ which is sampled so that either an integral or non-integral number of repeats is within the data aperture. The delta functions at $\pm g$ will fall on sampled points in the DFT only if an integral number of repeats would have been included in the full length DFT, had the truncated function been extended to the array boundary. If $r$ repeats occur in the data aperture ($T_a = nT$), then each repeat is $\frac{nT}{r}$ long. If each repeat is $\frac{nT}{\bar{r}}$ long, then $g = \frac{r}{nT}$. The DFT is sampled at $\Delta s = \frac{1}{NT}$. Therefore, $\frac{g}{\Delta s} = \frac{N}{n \bar{r}}$. Thus, $\pm g$ will be "correctly" sampled if $\frac{n}{\bar{r}}$ is a factor of $N$.

As will be seen, the question of exact sampling in the DFT at $\pm g$ is not as important in the "small" aperture case
as it was in the full aperture case. The sinc function corresponding to the aperture is 

\[ \sin(\pi T_a s) \]

which has zeroes at \( \Delta s = \frac{1}{T_a} = \frac{N}{n} \cdot \frac{1}{T_0} \). The central lobe of the sinc function thus spans \( 2 \cdot \frac{N}{n} \) sampled Fourier coefficients in the DFT. Regardless of where \( g \) fall with respect to the coefficients sampled at \( i \cdot \frac{1}{T_0} \), the convolution of the aperture sinc function with the delta function at \( g \) will lead to some samples close to the maximum of the convolution. The worst case would have \( g \) half-way between two samples \( i \cdot \frac{1}{T_0} \) and \( (i+1) \cdot \frac{1}{T_0} \).

We have previously developed the expected drop in intensity as a function of the fractional distance of a sample point toward the first zero of the sinc function. What fraction corresponds to our worst case and what is the correction factor needed for this displaced sample point? The sinc repeat is \( \frac{1}{T_a} \). Our worst case sample occurs at \( \frac{1}{2} \cdot \frac{1}{T_0} \) from the maximum. \( \frac{1}{T_0} = \frac{n}{N} \cdot \frac{1}{T_a} \), so we are at most \( \frac{n}{2N} \cdot \frac{1}{T_a} \) from the maximum. The fraction we are looking for is \( \frac{n}{2N} \). Figure 14 is a plot of the correction factors needed for the worst case as a function of \( \frac{n}{N} \). From this we can see that making the array size twice the data window (\( \frac{n}{N} = .5 \)) will lead to a worst case error of only \(-10\%\) compared with a worst case error of \(-36\%\) for a full-width window. Making the array size four times the data window size drops the worst-case error to \(-2\\%).
Graphs for the maximum correction factor and percent error for various data width to array width ratios

A) The maximum multiplicative amplitude correction factor versus the ratio of the number of data samples to the number of samples in the transform.

B) The maximum percent error in the largest uncorrected straddling DFT sample versus the ratio of the number of data samples to the number samples in the transform.
What is the effect of including a great number of repeats in a full-width window? As we saw in the general case, the position of the convolved sinc function with respect to the sampled lattice depends on what fraction of a period is included, not the magnitude of the integral number of periods. So the same multiplicative factors will be needed to get the correct intensity regardless of the number of repeats included in the DFT. The quantity that does improve, as a result of including a greater number of periods, is the separation of diffraction orders in terms of the width of the convolved sinc function. For larger array lengths (greater $T_0$), the sinc function becomes more peaked. Neighboring coefficients have less effect; the convolution is more local. Figure 15 demonstrates the difference between small and large numbers of repeats when another delta function is present in the neighborhood. Thus, no matter how large the transform, if a non-integral number of repeats is included in a full-width DFT, the observed intensities could be as low as $\approx 41\%$ of what they should be. Of course, the observed intensities are not contaminated with (convolved with) as much noise or other spurious intensity. The only way to increase the chance of sampling a delta function closer to its peak value is to use a smaller than full-width aperture, together with a large length (large $T_0$) array. The smaller aperture will smear the delta function out further while the large length array will keep all of this smearing relatively local.
The effect of the number of repeats on the DFT in the case of two waves of nearly identical frequency

A) A large number of repeats assures convolution with a narrow sinc function but the actual DFT samples in each peak still suffer from mis-sampling. There is little overlap (in this case) from one sinc function to another.

B) The same two waves but with a smaller number of repeats. Again, both sinc functions suffer from mis-sampling, but in addition, they overlap each other, leading to more complicated samples.
A

No convolution problems

Large number of repeats

B

Convolution problems

Small number of repeats
Another means exists for calculating the "correct" amplitudes of a periodic function. The original data can be interpolated onto a new sampling grid which will contain an integral number of repeats. As long as an integral number of repeats is included in the full-width DFT, there are no problems. The actual interpolation process is described in more detail below.

Although much has been said about the DFT, nothing has yet been mentioned about the fast Fourier transform (FFT). This was done intentionally because the FFT is merely an algorithm to calculate the DFT quickly, efficiently, and accurately. In actual fact, the DFT is not the backbone of image processing, the FFT is. Without it, none of the work undertaken so routinely could be done either from the point of view of cost or time. In computer terms, the cost of an arithmetical process can be estimated from the number of basic operations involved. Since the cost (time) to do multiplications is usually much greater than that required to add or to subtract, the number of complex multiplications is a good estimator of relative cost. A straightforward evaluation of the DFT formula would require \( N^2 \) complex multiplications for an \( N \)-point transform. The FFT algorithm does the same transform in \( N \log_2 N \) complex multiplications (if \( N \) is a power of 2). Such time savings become enormous when \( N \) reaches \( 2^8 \) (256) or more.
F. Two-Dimensional Processing

As already detailed, the "scanner" data must be processed before it is suitable for use in a three-dimensional reconstruction effort. The influences of the scanner aperture, image defocus, and objective lens spherical aberration must be countered. Other sources of systematic error, as yet not fully documented as to their seriousness (such as projector lens distortions), will need to be corrected. The off-shoot of these corrections is a two-dimensional function which more directly describes the projection of the object than the original image did. This is valuable in itself, apart from its use in the 3-D reconstruction problem. The process of performing such corrections on the image data has been described as "two-dimensional image restoration".

Figure 16 shows the major steps involved in two-dimensional processing. Before delving into the specific steps, the usefulness of relying on visual results should be noted. Although computers and their inflexible directives perform much or most of the work, the human is still needed and, in some cases, indispensable. Our ability to recognize meaningful patterns and perform measurements guided by what we see is exploited at every opportunity in the programs developed in this work.

The first stage of processing is to "see" what has been scanned in order to assess what should be done with the data. The manipulations of contrast that are possible with
Figure 16

Organization of two-dimensional processing
a digital representation of the image make it easier to spot some features in the scanned data than on the original plate. The initial look at the data gives one the chance to look subjectively for scratches in the plate, blobs of dust, orientation of the specimen and sub-areas of interest (the location of a phage tail, for example). It may be possible to see a "ramping" effect in the scanned data, wherein there is a gradual change in the optical density readings in a direction perpendicular to the linear scanning direction i.e. a change in density correlated with the time it took to do the scan. In a scanner such as the PDS, it is possible to have the illuminating and measuring optics change their relative alignment during the progress of a 2-D scan. As the illumination into the measuring aperture changes, so will the readings (a changing calibration effect). There are other ramps (which need not be perpendicular to the scanning directions) which may be caused by non-uniform exposure of the image or non-uniform development of the plate. Unequal staining could also cause a ramp in the scanned data. The detection of such problems and the decision as to what to do about them is best left up to the human.

The next step involves selecting a sub-area with which to work. This optional selection can be used to avoid troublesome areas or to select a particular region of interest. "Boxing", as it is called, can be combined with "padding" to form an array with larger dimensions than just the area of
interest. Several reasons for padding have already been pointed out, such as the need to achieve a certain array size (because of programming constraints or convenience) and the desire to sample Fourier coefficients more finely.

The image may need to be reoriented with respect to the scanning raster. This can be executed by means of an interpolation step. As developed in the previous section on the discrete Fourier transform, a periodic (crystalline) specimen will not have its Fourier coefficients optimally sampled by the DFT unless an integral number of repeats of the basic period (unit cell) is included in the array. This means the sampling grid must be aligned with the axes of the unit cell, if possible. The sampling interval, fixed by the capabilities of the scanner, usually must be modified in order to include an integral number of unit cells in a given array. The angular orientation of the sampling grid, together with the grid spacing, can be altered at will, but some interpolation scheme is needed to supply the unknown value of the projection at the new sampling points. There exist many complicated and powerful interpolation schemes which, for certain classes of objects, can guarantee a correct value of the function at the sampling point. Taking a simpler approach, Aebi et al. (1973) have shown that a bilinear interpolation utilizing only the four previously sampled points surrounding the required sample point is sufficient if the original sampling rate is above the Nyquist limit. The averaging nature of the bilinear interpolation
leads to a decrease in the intensity of the higher frequencies. The degradation of the higher frequencies can be kept to a minimum if the new sampling grid is kept as close to the old grid as possible, i.e. new sample points are close to old sample points. By carefully aligning the crystal axes with the scanning grid doing the scan, the need for large rotation of the interpolation grid is minimized. By choosing the transform size such that a close-to-integer number of repeats of the crystal (unit cells) are included, the sampling rate needs to be changed only slightly. The advantage of being able to use a mixed-radix (array lengths factorable by other than a single number, other than 2 for example) DFT algorithm is clear. Another factor influencing high frequency attenuation is the original data sampling rate. By over-sampling (for example, up to twice the Nyquist rate), the severely attenuated frequencies are mostly beyond the band-limit of the object.

Often in practice, the interpolation, if desired, is not performed on the scanned data during its initial trip through the computer. The large amount of noise and variation present in many specimens precludes the accurate choice of interpolation parameters such as the angle between old and new grid, the new sampling interval, and the number of sample points desired. The data can be preliminarily treated, as described below, to make the specimen easier to see and allow a proper choice to be made. If computer costs are low enough (or money is of no concern!), the use of a
larger array with data padding, as developed above, can be invoked to dispense with the need to interpolate.

A larger array is also necessary when using small, isolated particles. Two abilities are thus included in the 2-D processing stage. It is possible to create larger arrays around smaller amounts of data, and one can fill the excess array elements so created. As DeRosier and Moore (1970) point out, the data must be "floated" in the larger array. The choice of the constant value that is used to fill out the larger array will determine the mismatch occurring at the boundary between "scanning data" and "filler values".

Let us investigate the influence of this mismatch on the Fourier transform. Assume a periodic one-dimensional object \( f(x) \) with an average value \( \bar{A} \) and deviations from this average \( \Delta A \ll \bar{A} \). This latter condition is typical for the specimens encountered in electron microscopy. The Fourier transform of such an object would have a zero-order term of amplitude \( \bar{A} \) and higher-order components, \( A_g \), on the order of \( \frac{\Delta A}{2} \) or less:

\[
F( f(x) ) = \bar{A} \delta(0) + \sum A_g \delta(s+g) \quad (III-19)
\]

If a limited region of this function were masked off and the remainder replaced with a constant value \( A_v \), we would have a one-dimensional situation analogous to the masking and floating operation (see Figure 17). What is the Fourier transform of this modified object, \( f^1(x) \)? The
Figure 17

Effect of the filter value on the floated discrete Fourier transform

a) The continuous floated object, \( f(x) \)

b) Decomposition of \( f(x) \) into the sum of a constant function, \( A_v \), and the product of a masking function \( m(x) \) and a cosine wave

c) Fourier transform of \( f(x) \) for the case \( \bar{A} \sim A_v \). There is little contamination of one order by another.

d) The Fourier transform of \( f(x) \) for the case \( \bar{A} \gg A_v \). The sinc function convolved about the origin invades the higher order diffraction.
object can be decomposed into the sum of a constant function, $A_v$, and a function which is a modification of our original object. This modified function is the original function minus the constant $A_v$, multiplied by a step function $M(x)$:

$$f'(x) = A_v + M(x)(f(x)-A_v) \quad (III-20)$$

The Fourier transform of the decomposition is the sum of the Fourier transforms of the separate parts. Letting the step function be of width $2w$, we have the Fourier transform of $f'(x)$:

$$A_v \delta(0) + 2w \frac{\sin 2\pi sw}{2\pi sw} \ast (\bar{A}-A_v) \delta(0) + \sum A \delta(s+g) \quad (III-21)$$

The key point in the above equation is the convolution of the sinc function with the modified transform of the original object. Each delta function will have a sinc function "placed" on it by the convolution operation. What will the influence of the sinc function, centered on one diffraction spot, be upon its neighbors? How does this influence depend on the padding value, $A_v$, that is used? The latter question is easily answered. The convolutions about the delta functions not at the origin are unaffected by the choice of $A_v$. The convolution centered at the origin is weighted by the amplitude $\bar{A}-A_v$. If $A_v = 0$ is used, the sinc value centered at the origin will have its maximum value, while a choice of $A_v = \bar{A}$ will eliminate the convolution terms about the origin.
The former question is important since the only reason we will spend the money to calculate and use a non-zero $A_v$ is to avoid significantly influencing parts of the transform we wish to investigate. The influence of the non-origin sinc functions on each other is also important. If the width $2w$ spans $N$ repeats of length $R$, we can calculate the value of the sinc function at the neighboring diffraction spots located $\frac{m}{R}$ away, $m = 1, 2, \ldots$. We need to evaluate

$$\frac{\sin 2\pi \frac{mN}{2}}{2\pi \frac{N}{2}} = \frac{\sin 2\pi x}{2\pi x}, x = m \frac{N}{2} \quad (III-22)$$

for some realistic cases. Values for $N$ can range from 10 to 50. Of course, if an integral number of repeats is included with the step function, the nodes of the sinc function will occur at neighboring diffraction spots, in which case there is no problem. To achieve maximum effect at the first neighbor, let's use $N = 10.50$ and $N = 50.50$. For the former case, the sinc function will have values 0.03, 0.0, and $-0.010$ at $m = 1, 2, 3$, while the corresponding values for the latter case are 0.0063, 0.0, and $-0.0021$. Thus, we can see that in the range of typical image processing problems in electron microscopy, the influence of the origin sinc function can be as high as 2% of the origin term $\bar{A} - A_v$ out to the third order of reflection. Since $\bar{A}$ is usually a factor of 100 to 1000 times greater than the most intense diffraction amplitude, a padding value of 0 will be unacceptable. Calculating $\bar{A}$ for the boxed data would be the best option since
it would eliminate the origin term, but more limited schemes, such as taking the average of the boundary values, would seem to promise effective relief. If the difference \( \bar{A} - A_v \) is of the same order as \( \Delta A \), which almost any choice for \( A_v \) based on the actual data will achieve, then the influence on nearby diffraction spots will be limited to a few percent, at most, of the diffraction amplitude. This also sets a limit on the effect of one diffraction amplitude on another.

The data next needs to be Fourier transformed, and after doing so it is advantageous to view the power spectrum. The immense range of values that can be meaningful in this spectrum necessitates the use of a logarithmic representation. Both visual and numeric displays can aid in locating the reciprocal lattice axes and in the assessment of the degree of retained order for crystalline specimens (maximum radius in Fourier space that still demonstrates reciprocal lattice amplitudes above the noise).

A particularly useful operation at this stage of data reduction is the generation of the digital analogue to the optically filtered image that would be obtainable from the optical diffractometer. The reciprocal lattice spots, having been located, are "masked off" by a filter which "passes" only those parts of the Fourier transform which are within a certain radius of a reciprocal lattice point. Only reciprocal lattice points having intensity observable above
the background noise are used. Masking is accomplished by zeroing those parts of the transform which fail the criteria for inclusion. The "holes" can be made sharp, or a weighting function can be used to make the holes "fuzzy". Inverse transformation of the resultant array will yield an "optically filtered" image. The size of the apertures around the reciprocal lattice points will determine the extent to which the filtered image preserves local variations in the structure or exhibits only the average features found throughout the specimen. Large apertures lead to the former case, small apertures to the latter. These effects can be derived from a simple Fourier analysis.

The act of choosing small regions surrounding a reciprocal lattice point to use in an inverse Fourier transform can be described by the following operations: 1) convolution of the Fourier space aperture function with a comb function of the required spacing to form the reciprocal space mask; and 2) multiplication of this masking function with the original Fourier transform to create the reciprocal space function which will be inverted.

Let \( M(s) = C(s) \ast A(s) \) be the mask function and \( F(s) \) be the untouched Fourier transform. We seek

\[
\hat{f}(x) = F^{-1}[F(s) \ast (C(s) \ast A(s))] \tag{III-23}
\]

which is equivalent to

\[
F^{-1} F(s) \ast [F^{-1} (C(s) \ast A(s))].
\]
We thus find that \( f^1(x) = f(x) \ast (c(x) \ast a(x)) \), where \( f(x) \) is the function we would have obtained if no filtering had been done. \( c(x) \) is itself a comb function, of spacing \( P \), while \( a(x) \) is the shape function whose particulars depend on the aperture used. Since \( c(x) \) is a comb function, the product of \( c(x) \) and \( a(x) \) will be a weighted comb function. This product, in turn, is convolved with the unfiltered function so that the exact nature of the convolution depends on the product \( c(x) \ast a(x) \).

For simplicity, let us use a circular aperture around each reciprocal lattice point. The radius of this aperture will be expressed in terms of the reciprocal lattice dimensions. If the separation between adjacent reciprocal lattice points is \( \frac{1}{P} \), let the radius of the aperture be \( \frac{1}{nP} \), where \( n \geq 2 \). This aperture is a circle function whose transform, \( a(x) \), is \( \frac{1}{(nP)^2} \frac{J_1(2\pi u)}{u} \), where \( u = \frac{x}{nP} \). This function has zeroes at \( x = nPu, u = 0, 0.610, 1.12, 1.62, 2.12, 2.62, \text{etc.} \). Figure 18 shows this function plotted in reduced coordinates. We are interested in evaluating this function at \( x = mP, m = \ldots, -2, -1, 0, 1, 2, \ldots \), since this will give us the weights for the new comb function. Figure 19 is a plot of the normalized aperture function \( a'(m) \) versus \( m \) for several choices of aperture size, \( n = 2, 5, 10, 20, 50, 100 \). As a rule of thumb, the weighted comb function extends for approximately \( \frac{n}{2} \) periods on either side of the origin.
The inverse Fourier transform of a circular, reciprocal space mask of radius $l/nP$
Figure 19

Plot of the normalized aperture transform for several circular masks versus distance measured in unit cell repeats.
AMPLITUDE OF THE MASK TRANSFORM

DISTANCE (in # of unit cells)

Mask Radius (in 1/(unit cell))

100
50
20
10
5
2

XBL793-3310
What is the effect of convolving this weighted comb function with the unfiltered object? For such a function, the convolution is easy to visualize. Apart from a scale factor, the value of the convolved function can be obtained by taking the value of the unfiltered image and adding to it the value of the unfiltered image at a distance $P$ away, weighted by the height of the first comb function peak, then adding information from a distance $2P$ away, correctly weighted of course, and so on, until the comb function weighting factor becomes so small that the process need not be continued.

Two facts become apparent. First, if the sampling (comb) function in Fourier space accurately matched the reciprocal lattice of a crystal, then the convolution in real space would coherently add together corresponding parts of adjacent unit cells. Second, the radius of the filtering aperture determines, through its weighting of the real space comb function, the number of adjacent unit cells which add into any given unit cell. In this sense, the size of the filtering aperture limits the extent of the so called "spatial averaging". If the masking lattice is misaligned with the true reciprocal lattice, either rotationally or with respect to size, then "spatial averaging" will add together non-equivalent parts of neighboring unit cells, leading to a filtered image which bears little or no resemblance to the unfiltered structure. A large aperture is inherently safer to use, as the convolution or averaging effect is limited to
the immediate neighborhood. Slight misalignment ranging over a few unit cells will lead to almost equivalent parts being added together. The low resolution details should be preserved.

The limiting case of a small aperture mask is the delta function filter. Impossible to achieve in the optical diffractometer, it is realizable in the computer. The shape transform for this aperture is flat (a constant) and spatial averaging extends over the entire crystal. Thus, each unit cell displays the average structure of all unit cells. Of course, if the sampling lattice is not aligned with the reciprocal lattice, then the "unit cells" seen are the average of non-equivalent parts of all unit cells, i.e. noise. Since we work with finite numbers of unit cells, such a misaligned filtration will exhibit some structure that is apparently more meaningful than random noise, but it will not resemble the actual unit cell contents. Again, a slightly misaligned lattice, given the finite number of unit cells available for averaging, will retain the low-resolution features, but no more.

One last common pitfall of computer filtration will be mentioned. The naive processor has a compulsion to use delta function filtration because (1) it leads to a uniform picture and (2) the diffraction peaks in the transform are often quite sharp, so that a single coefficient can be found for each of the observable reciprocal lattice points. When
these high peaks alone are used for the filtration, the results may look terrible. There is no uniform unit cell appearance across the transformed array. What happened? Simply this – the coefficients used in the inversion do not lie on a regular sampling (comb) function. As we know from above, the DFT does not sample the true peak if the real space sampling grid and crystalline lattice vectors are not aligned. The peaks chosen for the reinversion may be, in one peak, the DFT sample that lies farther from the origin than the actual reciprocal lattice point, and in another may be the sample that lies nearer the origin than the actual lattice point. These one array element shifts lead to the Fourier combination of terms which are slightly shifted in frequency from what they should be, which leads to a "beating" phenomenon. As outlined above, a safer approach would be to use larger apertures around each diffraction "spot". In the event of small mistakes, the lower frequencies will be all right. An alternative viewpoint is that the larger aperture will "include" the unsampled reciprocal lattice point through the intermediary of the neighboring convolved Fourier coefficients.

The "optically" filtered image, i.e. one produced with larger-than-delta-function masks, is of great use. Since it is a (more-or-less) local average, features of the unit cell which might be obscured by noise, lack of statistical definition, or random damage can more easily be seen. Regions in the specimen that are severely disordered will stand out
more readily. Besides demonstrating an increase in visual clarity, the filtered image, in our work, provides the backbone for bilinear interpolation.

G. The Act of Bilinear Interpolation

Mention has been made of the mathematics behind bilinear interpolation, but little has been said about the method used to interpolate an image. Although there is no single "right" way to do an interpolation, experience has shown the following method to be acceptable.

The aim of the interpolation is to align the sampling lattice (points at which the function is known) with the crystal lattice. To satisfy some restrictions we encountered in the DFT, we would like an integral number of unit cells within our image array. The size of this array can be limited by the algorithm used to perform the DFT. The usual "radix-2" FFT requires that all arrays have dimensions that are a power of 2, for example. We also came to the conclusion in our theoretical treatment of interpolation that small deviations from the old lattice were superior to large deviations. This will be another goal of the interpolation. Interpolation can also be used to "rationalize" a plane group (see Section H). These aims are not always independent and we seek a method of interpolation which will satisfy them all with a minimum of work.

There are five unknowns that must be determined for a
rigid rotation of the old sampling grid and six unknowns for a non-rigid rotation. By rigid, I mean that the two sampling axes remain orthogonal while non-rigid allows the sampling axes to change their relative orientation. In a rigid interpolation, we need to determine for the new grid the angular rotation, the \((x,y)\) origin, and the new sampling interval along each axis. For the non-rigid grid, we still need the origin and the sampling intervals, but we also need a relative rotation between the sampling axes. In actual practice, more (redundant) information is sought to enable the user to specify the new grid in an easier manner.

Rotational alignment of the scan direction relative to the crystal axes is relatively straightforward. To a first approximation, the angular displacement can be measured from the digitized image of the specimen (if the crystal lattice can be seen) or from the power spectrum. The power spectrum suffers from being convolved with the shape transform and from being of limited visible extent in the types of display used. Convolution with the shape transform spreads out the diffraction spots making the location of the reciprocal lattice, and hence its rotation relative to the sampling lattice, unclear. The image, on the other hand, suffers from lack of definition because of low exposure or high noise. The preparation of an "optically"-filtered image is a great help at this stage. The image need not be a high resolution filtration. The filtering of the noise provides an easy-to-follow lattice. Using a wide aperture in the filtration
eliminates the need to know the position of the reciprocal lattice points to high precision, thus eliminating the shape transform convolution problem (see Figure 20). The angular misalignment can be measured quite easily from such a picture. For most processing, we have found this first guess of sufficient accuracy that no iterative methods are needed to improve it. There are methods, however, to refine the relative orientation (Frank, 1975).

Non-rigid rotations can be determined in the same way as rigid rotations. Again, the optically filtered image is of great benefit in many specimens.

The choice of the origin for interpolation can be guided by two goals. The interpolation algorithm, as implemented in the PLTPROC system, requires that the new grid fit within the boundaries of the old grid. The origin must be chosen so that the new grid will satisfy this condition. Since all of the variables we are trying to define influence the size of the new grid, the choice of origin must be made taking all the variables into account. The second goal of origin placement is to make the origin coincident with a symmetry axis of the plane group. Such a placement is not necessary but it will make the phase relationships among symmetry-related spots apparent without the need for phase-origin shifting. In origin placement, the optically filtered image again plays an important part by clarifying the location of symmetry axes in a noisy image.
The use of an optically filtered image for the determination of interpolation values

A) The uninterpolated, unfiltered object

B) Its power spectrum, showing a (schematic) mask used for optical filtrations

C) The filtered object. An area of interest is boxed off. The origin of this area, its angle of rotation, and its size can all be measured from the filtered object much more easily than it could have been measured from the original object.
The last variables to fix are the sampling interval (or sampling rate as it is sometimes called) along each sampling axis. Calculations can be performed to determine the sampling interval needed to ensure that an integral number of repeats fits in a conveniently sized array. The use of a mixed-radix FFT is encouraged because the number of "conveniently" sized arrays is greatly expanded. Again, the easiest method uses the optically filtered image. Using an enlarged print of this image, the number of rows and/or columns per repeat of the unit cell can be calculated without difficulty. Next, the maximum number of subunits that can be extracted from the original data, given the rotation present in the data and the possible origin locations, is determined visually.

The actual number of repeats that will be interpolated is determined by several factors. If the allowable array sizes limit the repeats that can be extracted, only the number of repeats is used that will yield approximately the maximum array size. If the number of repeats available is the limiting factor, then the array size to use is adjusted. Basically, we are trying for an integral number of units in a specified array size, both being somewhat variable.

Starting with a new sampling interval identical to the old, the number of rows or columns needed to hold an integral number of repeats is calculated. From the list of possible array sizes, the one closest to this is used. The
sampling interval is then adjusted to yield an exact match between array size and number of repeats. Sometimes it is possible to add or subtract one "row" or "column" of unit cells from the interpolated crystal to obtain even closer correspondence between a good array size and the array size obtained without changing the sampling interval. If this is possible, it is often done in order to keep the new sampling interval as close to the original as possible.

H. Plane Group Rationalization

It is usually the case that two orthogonal vectors in a plane (2-D) projection of a crystal are chosen to describe the crystal with respect to the scanning axes. The ratio of the lengths of these vectors will not be, in general, a small integer fraction. In fact, the ratio might be irrational. If an integral number of repeats in one direction are made to fit into an integer number of array elements, then there is little chance an integral number of crystal repeats will fit into an integer number of array elements in the perpendicular direction. By modifying the sampling interval in one direction to be different than in the other direction, it is possible to fit an integer number of repeats into an integer number of array elements in both directions, regardless of the ratio of perpendicular vectors established by the plane group. The p3/p6 plane groups are an example of this, and these groups were actually used in this work.
Figure 21 demonstrates the problem using a p3 projection. Assuming ten samples along one of the crystallographic unit cell vectors, $\bar{a}$, there would be only 8.66 samples per unit cell along the perpendicular direction. With interpolation, this number could be changed to ten also. If the scanning interval along the perpendicular direction were changed to be 0.866 of the original interval, we would get ten samples where once there had been 8.66. In this way, the ratio of the number of samples in the two perpendicular directions can be manipulated at will.

As discussed in Section E above, the restriction of integral unit cells for a "good" DFT made no mention of an integer number of samples in a single unit cell. The aim of the interpolation, then, is to adjust the sampling so that an integer number of unit cells fits into the entire array, not worrying about the number of samples per unit cell.

By sampling at different intervals in two orthogonal directions, the original geometrical symmetry of the crystal (and the diffraction pattern) are lost when they are displayed. The important spatial frequencies, i.e. those on the reciprocal lattice, are still sampled. The change in sampling interval changes amplitudes uniformly and does not affect the phases of the Fourier terms. These statements are easily proven in the next paragraph, using some properties of the DFT.

Assume we have an initial function $f(x,y)$;
An illustration of plane group rationalization

A) With equal sampling distances, an irrational number of samples are needed in a direction perpendicular to $\overline{a}$.

B) By interpolating in this perpendicular direction, a rational (and in this case, equal) number of samples per repeat can be taken.
Unrationalized Plane Group

\[ |\vec{a}| = |\vec{b}| \]

8.66 samples/repeat

Rationalized Plane Group

\[ |\vec{a}| = |\vec{b}| \]

10 samples/repeat

120°
After interpolating, we have a new 2-D function \( f'(x, y) \); \( x = n \Delta t_1 \), \( y = n \Delta t_2 \). If \( F(S_x, S_y) \) is the Fourier transform of the uninterpolated function \( f(x, y) \), what relationship will \( F(S'_x, S'_y) \) bear to the Fourier transform of the interpolated function, \( f'(x, y) \)?

\[
F(f') = F( f(\frac{x}{a}, \frac{y}{b}) ) = \text{scale factor} F(aS_x, bS_y)
\]

This equation shows that the coefficients retain the same phases that they had before interpolating, but move (expand or contract) to other spatial frequencies. In addition, all terms are multiplied by the same scale factor. This movement is the result of a uniform change of size and does not affect symmetry relations if we know which coefficients are symmetry related. We can no longer use a geometrical approach to decide which terms are symmetry related, but need to use a mathematical method which can locate peaks that were previously related by symmetry. The simplest method is to describe diffraction spots by Miller indices. These are assigned on the basis of reciprocal lattice vectors, which can be modified quite easily to account for the change in reciprocal space sampling that is brought about by a change in real space sampling.

A non-mathematical justification for the lack of change in the phase relationships among the coefficients can be advanced. The various Fourier terms can be considered to be standing waves, which are oriented in directions parallel to their reciprocal lattice vectors. The relative phase
between terms (waves) is actually a description of the displacement of the waves at the origin. Each phase term represents where in the 2π-cycle the wave is, at the coordinate origin. Using different sampling intervals is akin to stretching or compressing Fourier space, all such changes being zero at the origin. Such stretching/compression can be seen to lead to a change in frequency and amplitude, but since there is no effect at the origin, the relative phases must stay the same.

I. Coordinate Origin Shifts

The origin of the coordinate system defined by the scanning step usually falls at a place within a unit cell that is outside the practical control of the experimenter. In continuous Fourier transform theory, it is a simple task to relate the Fourier transform calculated with respect to one origin to the Fourier transform one would get using another coordinate origin. One can change the origin of the Fourier transform by modifying the calculated coefficients in an appropriate manner. The relationship between Fourier transforms defined with respect to different origins is given by the Fourier shift theorem:

\[ G'(S) = F[f(x-a)] = F[f(x)] e^{-i2\pi a S} = G(S) e^{-i2\pi a S} \] (III-21)

We can then move the coordinate origin for the Fourier transform at will, after the data has been scanned. In particular, we often wish to have the coordinate origin be
coincident with a symmetry axis of the crystal. In this case, the phases of the Fourier transform will display certain relationships dependent on the exact symmetry of the symmetry axis.

The above is true of the continuous Fourier transform but what of the discrete transform pair that we calculate? The existence of a convolved box function with the Fourier transform of the specimen, an unavoidable result of using the DFT, might cause problems when applying a phase origin shift. Does the calculated DFT behave in the same manner as the continuous Fourier transform under a coordinate origin shift?

Let

\[ C(s) = F(S) * A(S) \]  \hspace{1cm} (III-22) \]

where

\[ C(S) = \text{calculated DFT} \]
\[ F(S) = \text{specimen FT} \]
\[ A(S) = \text{aperture FT} \]

When the phase origin is at the center of the box aperture, \( A(S) \) is a real function. Using the subscripts \( r \) and \( i \) to denote the real and imaginary parts of each complex function, we can rewrite Equation III-22 as:

\[ C_r = F_r * A_r - F_i * A_i \]  \hspace{1cm} (III-23) \]
\[ C_i = F_i * A_r + F_r * A_i \]
With \( A_i(S) = 0 \) for the centered aperture, the convolution is seen to affect \( C_r(S) \) and \( C_i(S) \) in the same manner, therefore not influencing the phase.

Once the phase origin is shifted so that the aperture is no longer centered, \( A(S) \) becomes "truly" complex. At first glance, this would seem to begin to mix some \( F_i(S) \) with \( C_r(S) \) and some \( F_r(S) \) with \( C_i(S) \) through the \( A_i(s) \) term. It is easy to demonstrate, however, that such behavior is exactly what is required for the Fourier shift theorem to work.

Assume we move the origin of the function whose transform is given by Equation III-22:

\[
c(x-x_\theta) = f(x-x_\theta) \ast a(x-x_\theta)
\]

The Fourier shift theorem yields

\[
FT[c(x-x_\theta)] = C' = FT[c(x)] e^{-i2\pi Sx_\theta} = C(S)e^{-i2\pi Sx_\theta}
\]

Likewise,

\[
FT[f(x-x_\theta)] = F(S) e^{-i2\pi Sx_\theta}
\]

\[
FT[a(x-x_\theta)] = A(S) e^{-i2\pi Sx_\theta}
\]

We are interested in knowing if the application of the shift theorem to \( C(S) \) will yield the same phase as the application of the shift theorem to \( F(S) \), i.e. we would like to show that the aperture convolution does not influence the phase.
\[ FT[f(x-x_0) \ast a(x-x_0)] = FT[f(x-x_0)] \ast FT[a(x-x_0)] \]
\[ = F'(S) \ast A'(S) \]
\[ = F(S) e^{-i2\pi Sx_0} \ast A(S) e^{-i2\pi Sx_0} \]

The discrete convolution operation is:

\[ F'(S) \ast A'(S) = \sum_{t=0}^{N-1} F'(T) A'(S-T) \]
\[ = \sum F(T) e^{-i2\pi T x_0} A(S-T) e^{-i2\pi (S-T) x_0} \]
\[ = \sum F(T) A(S-T) (e^{-i2\pi T x_0} + i2\pi T x_0) e^{-i2\pi (S-T) x_0} \]
\[ = \sum F(T) A(S-T) e^{-i2\pi S x_0} \]
\[ = [F(S) \ast A(S)] e^{-i2\pi S x_0} = C(S) e^{-i2\pi S x_0} \]

We see that the convolution operation is quite independent of the phase origin shift. The convolution has added nothing to the phases; the phase change is a function of the origin shift only. We could change the coordinate origin of the two real space functions and then transform their product or merely modify the transform of the product by the Fourier shift theorem. Both courses represent the same action.

The relationship between the phase of \( C(S) \) and that of \( F(S) \) is clear. In the centered aperture case, convolution does nothing to the phase of \( F(S) \) except through the weighted addition of neighboring Fourier coefficients. Granted, this will undoubtedly change the phase of \( F(S) \) but in an expected way. For a delta function in the absence of noise, the phase of the delta function will be unchanged.
We have shown that the phase shift for both $F(S)$ and $C(S)$ is $-i2\pi S\theta$, so the phases of the two shifted functions will retain the same relationship regardless of the value of the coordinate origin shift.

**J. Contrast Transfer Correction**

The stage is now set for applying contrast transfer corrections to the projection data. The sampled Fourier coefficients at the reciprocal lattice positions will be the coefficients we need, after some correction, for three-dimensional reconstruction. As developed above, the Fourier coefficients should be corrected for the influence of the scanning aperture. This correction involves multiplying the terms by a simple factor. A more complicated correction is the one for the contrast transfer function.

For some classes of thin specimens (those using glucose embedding, frozen hydration, or another technique yielding a low-contrast micrograph), the contrast transfer is expected to be that of a weak-phase object and, as we have seen, a correction for $\sin(\gamma)$ is all that is needed. In addition to the low-dose image used for image processing, a high-dose micrograph must be taken for the statistical definition of the contrast transfer rings derived from the support film. The spacing of these rings reveals the defocus of the objective lens (Thon, 1965), a value needed for the calculation of $\sin(\gamma)$. The rings also reveal the direction and magnitude of astigmatism, quantities that are also needed for the
sin(\(Y\)) calculation. Sin(\(Y\)) corrections can then be made for all the available coefficients. Any coefficients lost because they fall near the minima of the contrast transfer function must be garnered from the image of another specimen taken with a different defocus value. If this is not possible, then only the coefficients extending out to the first "lost" coefficients should be used, thus limiting the resolution of any subsequent processing steps to a value lower than that possessed by the physical specimen.

K. Translational, Rotational, and Magnification Alignment

Using several specimens and different defocus values necessitates being able to relate one Fourier transform to another. Four potential problems arise. All are concerned with finding relationships between the separate images. The first is the determination of the relative rotation of the scanned micrographs. The second is finding the relative translation of the micrographs. The third is discovering the relative magnifications of the plates. The fourth involves scaling the plates together in terms of intensity and contrast. All of these problems have solutions which are set out in some detail below.

If the reciprocal lattice is well-defined, determining the relative rotation is not a big problem since the visual assignment of the reciprocal lattice vectors will establish a consistent angular relationship between micrographs. Care must be taken to avoid aligning the reciprocal lattice of
one plate with the Freidel-symmetry related lattice of another plate. This can be avoided by comparing the phases of terms in the two transforms referred to the same symmetry origin. Such a check will show if the terms being compared are equivalent (phases are the same, apart from contrast transfer reversal) or if they are Freidel related (phase of one is the negative of the other, again with the caveat about contrast reversal). Another point of concern would be achieving the correct alignment of crystals with pseudo-symmetry. By this I mean a diffraction pattern which geometrically appears to show symmetry (for example, six diffraction spots appear at the same radius, rotated 60° from each other, as in p3/p6 symmetry) but which does not obey the symmetry rules of the group it mimics. It must be kept in mind that sampling mismatch and image astigmatism, among other effects, will cause the amplitudes of symmetrical peaks to appear non-symmetric. To display symmetry of phase, the coordinate origin of the Fourier transform must be coincident with the appropriate symmetry axis of the crystal. Astigmatic imaging could throw members of a group of symmetry related spots into different contrast transfer zones, thereby changing some phases by 180° while leaving others unchanged. Keeping these points in mind, it should be possible to align correctly two statistically defined lattices.

The above discussion dealt with the "nice" case of a visible reciprocal lattice. If the specimen is too small to
lead to a statistically defined lattice at the exposure levels needed to avoid radiation damage, there is no way that a visual alignment can be carried out. Saxton (1974) and Frank (1973) have been using auto- and cross-correlation function (ACF and CCF respectively) methods to determine relative orientations of objects, non-crystalline as well as crystalline. The basic idea is to compare values of the power spectra of the two images to be aligned along corresponding annuli. These annuli are "unwrapped" to form linear arrays by sampling them in equi-angular increments. The two linearized arrays can then be cross-correlated to yield peaks representing the "linear" displacement needed to achieve best agreement between the values in the two arrays. This linear displacement actually corresponds to an angular displacement, hence the relative rotation can be ascertained. The power spectra are not a function of translational positioning so the two objects do not need to be translationally aligned, which is very convenient.

The power spectrum of a crystalline object should be a discrete series of delta functions, so the annuli chosen for comparison must lie at a radius containing some members of this reciprocal lattice. This presupposes a knowledge of the reciprocal lattice, and such knowledge may or may not be possible to obtain. In some cases, the electron micrograph chosen for processing might have large enough patches to show some optical diffraction, in which case the radius needed could be calculated. In other cases, the lattice
constants of the specimen might be known and a knowledge of the plate magnification could lead to a useful prediction. Of course, theoretical plate magnification can vary in a \( \pm 5\% \) range so some type of internal calibration would seem to be necessary in order to rely on this approach.

Another idea is to adopt a "shotgun" approach. Rather than choose a single radius in reciprocal space, a range of radii could be used. The resulting linearized arrays will be two-dimensional and a two-dimensional cross-correlation could be performed on them to locate the shifts (now both in radius and in angle) needed to align the two power spectra. The advantage here is that magnification differences will not doom the search from the start.

The success of any scheme depends on the statistical definition of the objects which are being compared. Saxton and Frank (1977) have delved into some theoretical calculations relating the electron dose needed in the face of electron noise having a Poisson distribution (shot noise). Their conclusion is that individual particles can be located even under low-dose (1 electron/\( \AA^2 \)) conditions. Another factor governing success is the defocus values of the micrographs. It would be necessary to compare annuli from regions of high contrast transfer and to avoid the minima of the contrast transfer functions. The rapid fall-off in actual contrast transfer as measured by Unwin and Henderson (1975), when they compared electron diffraction amplitudes
(which do not suffer from contrast transfer effects) with computed (image) amplitudes, does not bode well for comparing high resolution annuli (which are needed for a more precise determination of the relative orientation). Some of the high resolution fall-off may be caused by projector lens distortions operating on an extended image. This process would not apply in the case of small patches so it is hard to estimate the outcome of such trials.

The second problem to overcome is the determination of the translational misalignment of the two images. The CCF function has been used in the case of aperiodic specimens but it suffers from severe degradation at some defocus values (Frank and Al-Ali, 1975). The crystalline symmetry can be used in this case to search for a set of transform origins in each image separately that satisfy the symmetry-imposed phase restraints. A common origin in these two sets can then be used as the common origin for the two images. When the phases are adjusted accordingly, the complex transforms can be added. The accuracy of such a procedure in the face of noise remains to be investigated.

L. Three-Dimensional Helical Reconstruction Theory

The introduction to this chapter touched on the essence of the method of three-dimensional Fourier reconstruction from projections, the Fourier projection theorem. For the case of a helical object, the relationship between the object and its projection can be formulated in a more
specific and useful form. A cylindrical polar coordinate system is a natural one to use in describing a helical object. The following treatment is based on helical diffraction theory as found in DeRosier and Moore (1970), who in turn based theirs on the work of Klug, Crick, and Wyckoff (1958).

Let our helical object be \( f(r, \theta, z) \), the \( z \)-axis being the helical axis. The Fourier transform can be expressed as

\[
F(R, \Phi, z_c) = \sum_{n,l} G_{n,l}(R)e^{i \frac{\Phi}{2}}
\]

The helical symmetry "discretizes" the transform in the \( z \) direction. \( c \) is the axial repeat distance of the object \( (f(r, \theta, z) = f(r, \theta, z+c)) \), while \( l \) is called the layerline number. The \( G_{n,l} \) functions, if known, can be used to derive the object function as follows:

\[
f(r, \theta, z) = \sum_{n} \sum_{l} g_{n,l}(r)e^{i \theta} e^{-i 2\pi \frac{l}{c} z}
\]

where

\[
g_{n,l}(r) = \int_{-\infty}^{\infty} G_{n,l}(R) J_n(2\pi Rr) 2\pi RdR
\]

This transformation is the inverse Fourier-Bessel transformation. What are the meanings of \( G_{n,l} \) and \( g_{n,l} \)? The \( G_{n,l} \) are the structure factors for the asymmetric unit with the azimuthal dependence factor explicitly removed (Klug, Crick, and Wyckoff, 1958).
They depend on Bessel functions of order \( n \), and the composition and distribution of atoms in the asymmetric unit. The presence of helical symmetry constrains the values of \( n \) allowable on any layerline \( l \). This constraint is summarized by the selection rule:

\[
l = tn + um
\]

where \( t \) is the number of turns of the continuous (basic) helix per axial repeat, \( u \) is the number of discrete subunits per axial repeat, and \( m \) is any integer which causes the relation to be satisfied (see Figure 22).

Thus, we can see that the Fourier transform on any layerplane \( Z = \frac{1}{c} \) depends on weighted Bessel functions whose allowable orders are determined by the layerplane (line) number, \( l \), and the helical selection rule for the specimen. One property of the Bessel function determines the behavior of the summed \( G_{n,l} \) functions. Bessel functions are effectively zero until their argument approaches \( n-2 \), where \( n \) is their order. From Equation III-28, with \( r_{\text{max}} \) being half the outside diameter of the specimen, we can say that \( G_{n,l} \) will be zero for \( R < \frac{(n-2)}{2w_{R_{\text{max}}}} \). Thus, the near axial portion of a layerplane will depend on only one of the \( G_{n,l} \), the others being zero for such small arguments. Knowing the selection rule, we can define a region on each layerplane in which only one \( G_{n,l} \) contributes to the Fourier transform. In this
A discrete helix illustrating the selection rule 
\[ l = 2n + 15m \]

The basic helix has a period of \( P \), the discrete subunits are spaced \( p \) apart in the \( z \) direction, and the structure repeats in a distance \( c \). Therefore, \( c/P = 2 \) and \( c/p = 15 \). In other words, there are two turns of the basic helix and 15 subunits per repeat.
region, Equation III-25 becomes

\[ F(R, \phi, \frac{1}{C}) = G_{n,1}(R) e^{\text{in}(\phi + \frac{\pi}{2})} \]  (III-29)

The Fourier projection theorem tells us that the two-dimensional Fourier transform of our helical specimen (a projection) is a central section in three-dimensional Fourier space. Our calculated 2-D Fourier transform is thus such a section and the layerlines we see on the two sides of the meridian are \( F(R,\phi,Z=\frac{1}{C}) \) and \( F(R,\phi,Z=\frac{1}{C}) \). Equation III-29 can be solved to yield two estimates of \( G_{n,1}(R) \) in the single Bessel function region of each layerline.

\[ G_{n,1}(R) = F(R,\phi,Z=\frac{1}{C}) e^{-\text{in} \frac{\pi}{2}} \]  (III-30)

\[ = F(R,\phi,Z=\frac{1}{C}) e^{+\text{in} \frac{\pi}{2}} \]

Except for layerlines where there are \( G_{0,1} \) (meridional reflections), one estimate of \( G_{n,1} \) can be considered to arise from one half of the helix (splitting it perpendicular to the projection direction). Assuming that you look at the projection image as if you were at the source of electrons, the half of the helix closer to you is termed the near side, the half farther away, the far side. Each \( n \) value can be considered to arise from a set of continuous helices in the object. For \( +n \) and viewing from the vantage point described above, these helices would rise to the right on the near side and rise to the left (when viewed in projection) on the far side. Thus, the layerline on the left of the meridian
arises from the helical portions on the near side, while the right layerline comes from the far side portions. A -n value reverses this assignment (see Figure 23).

In practice, two separate three-dimensional reconstructions are usually done, one based on the near side data only, the other on the far side data only. Sometimes, an average structure is generated by using the average value of \( G_{n,1} \).

To do the reconstruction, we merely need to extract \( G_{n,1}(R) \) from all the "single-Bessel-function" (near axial) regions of all the definable layerlines, perform the integration of Equation III-27 to get the \( g_{n,1}(r) \)'s, and then use Equation III-26 to generate \( f(r,\theta,z) \). In fact, the \( g_{n,1}(r) \)'s can be stored to produce \( f(r,\theta,z) \) at will, on any set of \((\theta,z)\). With a fine enough sampling in \( r \) of \( g_{n,1}(r) \), an interpolation scheme can be used so that \( f(r,\theta,z) \) can be calculated at any \((r,\theta,z)\) also.

M. The General Three-Dimensional Reconstruction Problem

The reconstruction of helical specimens, outlined above, is convenient for the electron microscopist, since only one image is needed for low resolution retrieval. Higher resolution for this type of specimen can be achieved by using multiple specimens tilted at different angles about the helical axis and combining their Fourier transforms (Amos and Klug, 1975). What of specimens that do not possess
Assignment of near side and far side origin to diffraction spots

a) A set of three continuous helices which have a positive n value (they rise to the right)

b) Diffraction pattern of a two-dimensional projection of a) perpendicular to the helix axis. The spots arise from the structures indicated in a)

c) A set of three continuous helices which have a negative n value (they rise to the left)

d) Diffraction pattern of a two-dimensional projection of c) perpendicular to the helix axis. The spots arise from the structures indicated in c)
- $n$ helix
  - Nearside
  - Farside

$+n$ helix

Diffraction pattern

$-n$ helix
  - Nearside
  - Farside

Diffraction pattern
such a convenient symmetry? In this section, we will outline a more general procedure for 3-D reconstruction and point out some of the trouble spots ahead. The limiting elements in our achievement of a general three-dimensional reconstruction capability have been a combination of time for development of computer programs and a high-angle tilting stage for our microscope. Because of a lack of actual general 3-D experience, this discussion will be limited in its scope.

Again, we return to the Fourier projection theorem. The 2-D Fourier transform of each projection gives us a plane in 3-D Fourier space. On the other hand, we desire a "complete" knowledge of 3-D Fourier space in order to recover the three-dimensional object. By varying the tilt and azimuthal orientation of the specimen, all of Fourier space will be sampled as a succession of 2-D central sections. Unfortunately, the range of available tilts is limited, especially when one considers the increase in thickness attendant upon large tilts. A rule of thumb is to limit the tilts to approximately $60^\circ$ maximum. Isolated particles can adopt a variety of orientations with respect to the supporting film so it is not necessary to tilt any single specimen through a large tilt angle in order to collect data from many specimens over a $180^\circ$ range. For high resolution, low-dose work, however, the extended and thin crystalline patch is the specimen of choice. Such a structure lands either up or down but not in between. In general,
then, we cannot count on getting tilts $> 60^\circ$ and therefore we will not have information about the 3-D Fourier transform in a $30^\circ$ cone about the $z^*$-direction. This "hollow cone" problem has been analyzed from several viewpoints (Crowther et al, 1970b; Klug and Crowther, 1972) and at this time is being investigated in our laboratory for the case of monolayer crystalline patches. For now, we must presume that the data from the "hollow cone" is unavailable and our reconstruction will suffer from poor definition in the $z$-direction.

What are the practical problems in assembling a three-dimensional data set? First, one must determine the tilt axis in the image. By being tilted, the specimen exhibits a defocus ramp perpendicular to the tilt axis. Since the correction for $\sin\gamma(s)$ depends on the defocus (see Equation II-6), the orientation of this tilt ramp with respect to the Fourier transform coordinate system must be known. If the specimen is a 2-D crystal, the tilt axis and the magnitude of the tilt angle can be determined by an examination of the geometry of the diffraction pattern. The determination of tilt sign must be made on the basis of other information.

Once the contrast transfer correction has been made, the assignment of spatial frequencies with respect to a fixed coordinate system must be done. This involves a knowledge of the azimuthal orientation in addition to the previously determined tilt angle. Although the tilt angle
can be determined from either the diffraction pattern geometry or from a goniometer stage tilt readout, the azimuthal angle is harder to determine. For crystals in the $0^\circ$ projection, the symmetry of the Fourier transform makes the azimuthal assignment trivial. Once tilted, however, the central section plane in Fourier space will not, in general, intersect symmetry related coefficients. The geometry of the diffraction pattern is still of some help since it limits the possible values for the azimuthal orientation. Unlike the $0^\circ$ projection in which there is a symmetry generated equivalence, the tilted view has no degeneracy.

Another problem facing the experimenter is ensuring that all the Fourier transforms are referred to the same coordinate origin. Again, the symmetry found in the $0^\circ$ projection made phase origin refinement easy. Each plate can have its phase origin coincident with a certain symmetry axis. All plates will have the same phase origin in this case. Other tilt and azimuthal orientations remove the ability to refine each projection independently by removing the symmetry of the diffraction pattern. Different plates must be related to one another directly. The only points in common between two intersecting planes all lie on a common line. The phase origins of the two plates must be moved so that the phases along the common line are the same. Since we will be working with crystalline materials, it is conceivable that the line of intersection may not, in turn, intersect any reciprocal lattice points. In this case, you
would be out of luck.

In the case of two-dimensional crystals, if the tilt angle between data sets is not great, then one can appeal to the smooth nature in the \( z^* \)-direction of the delta function rods for help in orienting the transform. The uncertainty in azimuth can be resolved by comparing the amplitudes between two plates at slightly different tilts. The orientation that leads to the closest fit is the correct one. Once the projections have been aligned, one needs to derive equi-spaced samples in Fourier space in order to calculate the correct three-dimensional object. One can use bilinear interpolation to get these samples if the data are closely spaced or sinc function interpolation if they are more sparse. Unfortunately, tilt angle limitations will enforce a gap in our knowledge of 3-D Fourier space which cannot be spanned by interpolation (Klug and Crowther, 1972). Research on methods that might be used to "fill in" the gap is ongoing in many fields with similar problems. At this time, however, one must accept the loss of information (resolution) for certain specimens.

The three-dimensional object is recovered by inverse 3-D Fourier transforming the equally spaced Fourier space samples. The results need to be displayed in a variety of forms so that a feeling for the 3-D character of the object can be formed. Two-dimensional sections and 3-D perspective plots of single contour sections can impart this feel. As
more sophisticated graphic devices become available (at rea-
sonable prices) to the electron microscopist, the often
painful step of interpreting a three-dimensional density map
should be greatly improved.

N. Conclusion

The previous sections have attempted to spotlight some
of the theory and methods involved in image processing and
three-dimensional reconstruction as practiced in electron
microscopy. There is still much to be done in both 2-D and
3-D to utilize fully the information locked up in each
micrograph. Apart from improving the processing, great
strides can be made in producing even better micrographs.
Although computers are powerful, they still do their best
work on great specimens.

Although 2-D image processing has advanced the
farthest, the increasing number of workers doing 3-D recon-
structions will surely affect that field. Even though the
currently used methods are not fully optimized, a tremendous
amount of biologically significant information is waiting to
be extracted from many specimens. Although ultra-high reso-
lution is the goal, the high resolution information obtain-
able with all the present limitations can help many biologi-
cal studies.

The image processing and reconstruction programs which
are needed and have been implemented are the subject of the
next chapter.
IV. Implementation of Image Processing and Three-Dimensional Reconstruction Algorithms

A. Introduction

The preceding chapter has detailed techniques involved in image processing and 3-D reconstruction from a theoretical viewpoint. There are a series of tasks which must be performed on an electron micrograph before meaningful results can be achieved. The practical implementation of these tasks, as realized at our laboratory, is the purview of this chapter. The step from mathematical to practical is not a trivial one and the actual results can vary from marginally useful to significantly helpful.

The overall principle guiding the development of programs for processing and reconstruction was a desire to supply the electron microscopist with practical tools useful in analyzing micrographs. The involvement of the computer is a necessary evil, so one aim in developing the system of programs was to avoid involving the user in the intricacies of computers and computer programming. The emphasis is on useful tools that can be invoked with a minimum of knowledge about the methods used to achieve the results. The programs are not so arcane, however, that the curious user can not understand details of the process. The motivation has therefore also been to provide the curious with details but not to burden the uninterested with same.
In accord with this philosophy are the actual capabilities of many of the image processing programs. There are so-called "standard" options which are in effect but, for the investigator who would like to custom tailor the programs, many specialized options exist. There is also an easy-to-use flexibility in the programs. By use of different commands, even the beginning computer user can change the flow of the tasks to fit a special need. Naturally, the programs are not foolproof nor can they be used without some effort and thought. There is room for improvement and the incorporation of new ideas and better methods. At present, however, the programs have demonstrated themselves to be useful, flexible, and relatively efficient of one's time.

Another aspect of our approach is in the presentation of results. The first property of image processing that comes to mind is size of scale. The arrays are very large (even when they seem small by comparison with even larger arrays) and the number of numbers generated by the programs range upwards into the millions. Lurking in this sea of numerals lie the few values of interest. Even when identified as important, the numerical representation of a result is, at best, a limited measure of its worth to the investigator. After all, the ultimate goal in image processing and 3-D reconstruction is a structure, not a set of numbers. Thus the visual has been stressed whenever appropriate. Greyscale plots and isodensity contour plots are available to visualize results. The production of greyscale pictures
has been optimized so that they can be produced with little cost and can be used extensively to analyze data.

The present capabilities of the system include all major steps needed to process individual electron micrographs in two dimensions. Three-dimensional reconstruction has been limited, to date, to helical specimens for want of three-dimensional data (tilted projections) from planar, crystalline structures. Extension of the tools to include combining small patches (crystalline or not), correcting for the transfer function in tilted specimens, correcting for projector lens distortions in the image, and overcoming maximum tilt limitations are the only additions needed to handle the full gamut of electron microscopy image processing at high resolution.

B. Large Interpolation

One of the requirements for high resolution and low dose imaging of crystalline specimens is the use of large numbers of unit cells in the processing of the images. This requirement leads to large arrays in the computer. Since the need to handle arrays of dimensions 5000 x 5000 or even larger might arise in the future, programs have been written to handle such very large arrays. Two functions were singled out as especially important: interpolation of the data and calculation of the Fourier transform.

A bilinear interpolation routine has been written which
can interpolate an arbitrarily large array. The interpolation is broken down into pieces, each of which can fit into the memory made available for the routine's use. The interpolation task can be envisioned as superimposing a new sampling net over an existing (old) one. The value of the function at each node (point) of the new net is determined by the values of the sampled function at each of the four surrounding points on the old net. Under this scheme, the new net is only defined within the boundaries of the old net (interpolation versus extrapolation).

Since bilinear interpolation is a local operation, only the old data in the immediate vicinity of the new net point need to be in the computer at any one time. Efficiency demands that, at a minimum, the old data needed to produce at least one new column be in core memory. Since the new net is often rotated with respect to the old net, several (and perhaps many) old columns are required for each new column. Consecutive new columns make use of many of the same old columns for their interpolated values so, again, efficiency leads one to interpolate a contiguous block of new columns. The present algorithm calculates the size of the largest block of new columns consistent with the core space available for storage of old net data values. The required old values are read, the block of new net values are calculated and stored, and the set of old values needed for the next block are read to begin the cycle once again. The larger the core space available for storage, the fewer
are the cycles needed to interpolate the entire new net.

The size of the array which can be interpolated is limited by the requirement that at least one new column be interpolated at a time. The number of columns in the old array is not important, only the number of rows. The maximum usable memory is \(8^6 \approx 2^{20}\) for this program. If the relative rotation of the nets is 5°, an array of approximately 1700 rows would be the maximum that could be handled by this algorithm. If the rotational misalignment is only 0.5°, an array of approximately 5400 rows can be accommodated. Of course, an algorithm which extracts less than a full new column at one time could extend the usable range as far as is wished.

C. Large Discrete Transforms

Methods exist for calculating the full DFT of externally stored data, but none of these has been implemented to date at our laboratory. Since our objects are crystalline, diffraction is limited to the reciprocal lattice. Thus, only small regions in Fourier space need to be calculated. It has been common practice to calculate transform values in a small (9 x 9 coefficients, for example) neighborhood of the expected lattice location to account for errors in the location of the lattice. Unwin and Henderson (1975) have used a hybrid approach to calculate the transform values in these small neighborhoods. The 2-D transforms can be decomposed into two consecutive 1-D transforms, one over the
rows, the other over the columns. For an array of size N x M, M 1-D transforms, one for each column, can be done. If these "transformed" columns are written out to disk, the data can then be read into core by rows, and the total 2-D DFT can be calculated by performing N 1-D transforms using this data.

The obstacle to performing such a transform lies in the physical organization of the disk storage device. The data can not be easily read as rows, once it has been stored as columns. Unlike core memory, the disk is not a random access device (a device in which the access time for any storage element is virtually the same for all elements). The disk is divided into concentric tracks. The greatest amount of time involved in accessing a computer word stored on the disk is the positioning of the reading head at the track which contains the word to be read. Once at the track, consecutive computer words can be transferred in rapid succession. Hence, data is most easily retrieved in the same groupings in which it was stored. The transformed columns that are written out to the disk can most easily be read as columns. In order to read this data by rows, it is necessary to extract one element from consecutive columns until a row has been finished. Since consecutive columns will not be on the same disk track, in general, such an accessing order will mean that the disk will spend most of its time searching for one track after the next. Naturally in an environment where time is money, such a procedure is
expensive.

Algorithms to compute the full DFT of externally stored data need not spend all their time in track searches. The array to be transformed can be subdivided into small blocks. Groups of these blocks can be read as either a vertical stack, as it were, to form a sub-array of full length columns or they can be read as a horizontal stack to form a sub-array of full width rows (see Figure 24).

For the special case of calculating a limited number of coefficients, there is another approach. Every column of the image that is transformed via the FFT becomes a hybrid function of both real and reciprocal space coordinates:

\[ H(x, S_i) = \sum_{k=1}^{\text{nrows}} f(x, y_k) e^{2\pi i (S_i \cdot y_k)} \]  

(IV-1)

The required 2-D Fourier coefficients are merely a linear combination of these hybrid terms:

\[ F(S_j, S_i) = \sum_{k=1}^{\text{ncols}} H(x_k, S_i) e^{-2\pi i x_k \cdot S_j} \]  

(IV-2)

This sum can be evaluated directly as written, without employing an FFT algorithm and, consequently, evaluated term by term. Thus, if the frequencies of the desired terms are known beforehand, a running sum can be kept for each coefficient. As each 1-D hybrid column transform is produced by an FFT, the required terms \((S_i \text{'th for example})\) are multiplied by their respective complex factors and the results
Figure 24

Subdivision of a large array into smaller blocks for FFT calculation
A large array...

Split into smaller blocks on disk...

or read into core as a series of rows

can be read into core as a series of columns...
summed with the results of previous columns. There must be storage available for one complex column of hybrid data. Each coefficient requires one complex memory location for its building sum and one or more (or if one is clever, less) memory locations for storage of its reciprocal space coordinates. This memory requirement is small compared to the vast number of data values that go into the calculations of the DFT coefficients. The 1-D hybrid transforms are written to disk and saved for future calculations of other selected coefficients.

The expense involved in the hybrid approach can be analyzed with respect to performing a comparably sized complete in-core FFT. The time to do an FFT varies as \( N \cdot M \log_2 N^M \). For simplicity, assume \( N=M \). We then have

\[
\text{Time} = N^2 \log_2 N^2 = 2N^2 \log_2 N
\]

The time to perform an FFT of length \( N \) would be \( N \log_2 N \). Since the hybrid function involves \( N \) of these transforms, the time to calculate the hybrid function is proportional to \( N^2 \log_2 N \). It can be seen that calculating the hybrid function takes \( \frac{1}{2} \) as long as calculating the entire transform. Added to this fixed cost is the expense involved in carrying out the single sum (non-FFT) to get each 2-D Fourier coefficient. Each term involves \( N \) complex multiplications. The total cost is proportional to \( N^2 \log_2 N + \) (number of spots times \( N \)). Thus, the hybrid approach becomes more expensive
when \( N \log_2 N \) spots need to be calculated. Based on calculating a \( K \times K \) group of neighbors, this leads to cost equivalency when \( \frac{N \log_2 N}{K^2} \) diffraction spots are needed. For a large array, say \( N=2048 \) and a \( 9 \times 9 \) block, this point is reached after (about) 256 diffraction spots (with neighbors) are calculated. For a half-plane of diffraction data, this means about 11 orders of diffraction, assuming an orthorhombic unit cell. Of course, if a mistake were made in positioning the reciprocal lattice, some of the coefficients calculated on the basis of the incorrect lattice will be of no use and new coefficients will have to be calculated. This will lead to a further narrowing of the cost gap between selected coefficient calculation and calculating the full DFT. At some point, then, it is better to calculate all \( N^2 \) diffraction amplitudes using an externally based DFT algorithm than it is to use the hybrid method. Unwin and Henderson (1975) made use of a minicomputer to perform half of the hybrid process. It could be possible to perform the Fourier transform as the data is coming off the scanner, if the controlling computer is fast enough.

D. Piecewise Transform

Another method has been investigated to calculate the discrete transform of the reciprocal lattice points. If an integral number of unit cells fits into an \( N \times M \) array, it is possible to break the larger array into blocks of size \( L = \frac{N}{I} \) by \( K = \frac{M}{J} \), where \( I \) and \( J \) are factors of the number of
unit cell repeats in the row and column direction respectively. Thus, there would be \( I \cdot J \) of these blocks, each block containing an integral number of unit cells. If a 2-D transform is done of each block, reciprocal space is sampled at spacings \( \vec{S} = \frac{n}{L} \vec{S}_x + \frac{m}{K} \vec{S}_y \). But since \( L = \frac{N}{I} \) and \( K = \frac{M}{J} \), we see that \( \vec{S} = \frac{nI}{N} \vec{S}_x + \frac{mJ}{M} \vec{S}_y = nI(\frac{1}{N}) \vec{S}_x + mJ(\frac{1}{M}) \vec{S}_y \), which correspond to reciprocal space points which would be sampled if the larger array of size \( N \times M \) had been transformed. Since an integral number of unit cells is enclosed by the smaller blocks, the reciprocal lattice of the crystal is sampled correctly. The number of diffraction orders (resolution) found in the small transform is the same as what would be found in the full-size transform, being determined by the sampling interval, not the array size.

When we calculate the smaller DFT, we find the same spatial frequencies we need from the larger DFT (those that fall on the reciprocal lattice) with less "in between" frequencies. Of course, the smaller array does not include as many unit cells as the larger one. This is not a problem since we can combine many smaller transforms by phase shifting them prior to combination. Since the larger array is broken down into well-defined blocks, each of which is separately transformed, the relative coordinate origin shifts are known (see Figure 25). All of the separate DFT's can be related to a common origin (origin of block 1, for example) merely by using the Fourier shift theorem to move the coordinate origins of the separate blocks. By
Figure 25

Subdivision of a large array for the piecewise transform
performing this shift, we are rigorously combining the separate transforms, irrespective of the contents of each real space block. This procedure can be used on any large object which is divided into smaller blocks; it need not be crystalline. The stated requirement of including integral numbers of unit cells in each block "forces" the DFT to be calculated at the spatial frequencies we need (those on the reciprocal lattice); it is not required for the proper combining of separate block transforms. If in fact there are exactly an integral number of unit cells in each block, we could dispense with the phase shifts for each block. The coordinate origin of each transform would occupy the same location with respect to the unit cell's contents and thus the phases of the diffraction orders should be the same in each transform. Unfortunately, exact alignment is impossible and the analytic phase shifting is required (for peace of mind, at least).

As mentioned above, the disadvantage of this procedure lies in the coarser sampling of Fourier space. If the large array is decomposed into I blocks in a particular direction, the resultant sampling in Fourier space in that direction will be every $i^{th}$ coefficient when compared to the DFT of the full array ($L = \frac{N}{I}$ leads to $\frac{1}{L} = I\left(\frac{1}{N}\right) = I$ times full-size DFT sampling). A further disadvantage is that the shape transform will not be that of a block L wide but rather that of the full array. A diffraction peak that is not sampled will be convolved by a narrow shape transform, affecting, to
a first approximation, nearest neighbors (of the full-sized array that is) only, while Fourier space sampling is limited to every \(1^{\text{st}}\) coefficient. Thus, such diffraction orders could be easily missed. The usual situation for small transforms is quite different, the larger shape transform serving to spread a missampled diffraction order to the nearest sampled coefficient. This scheme of using blocks to calculate a subset of the large array coefficients will only be useful if the crystalline data is interpolated such that each block contains an integral number of unit cells.

Of course, it is a practical impossibility that an exact number of unit cells be interpolated into each block. What are the errors which can be tolerated and still leave the method useful? The convolved shape transform corresponding to the full-size array leads to an error bound. Assume that instead of \(P\) unit cells in a block of size \(K\) (\(K = \frac{N}{I}\)), there are \(P+x\), \(0<x<1\). The first order of diffraction will occur at a spatial frequency \((P+x) \cdot \frac{1}{K}\). Assume all orders out to the \(L^{\text{th}}\) are desired. The \(L^{\text{th}}\) order will fall at \(L(P+x) \cdot \frac{1}{K}\). The nearest sampled coefficient will be \(LP \cdot \frac{1}{K}\), assuming the simple case of \(x < \frac{1}{2L}\), which corresponds to an interpolation error of less than half a Fourier sampling interval by the \(L^{\text{th}}\) order. Larger errors may lead to coincidental alignment of some diffraction orders with the sampling grid in Fourier space but, by necessity, will cause other orders to have larger errors.
In order for the sampled coefficient at $LP \cdot \frac{1}{K}$ to detect the $L$th order, the shape transform characterized by the full array width, $N$, must be sampled before it dies out. To sample in the main lobe of the sinc function, the true peak must be within $\frac{1}{N}$ of the sampled peak $LP \cdot \frac{1}{K}$. Since $K = \frac{N}{I}$, our true peak is at:

$$L(P+x) \cdot \frac{1}{K} = L(P+x) \cdot \frac{I}{N} = \frac{LPI}{N} + \frac{LxI}{N}$$

Therefore we require that $LxI < 1$. To be assured of strongly sampling the sinc function shape transform, the inequality should be made:

$$LxI < 0.5 \text{ or } x < \frac{1}{2LI}$$

That is to say, the fraction of a unit cell error allowed in interpolation is the reciprocal of twice the maximum diffraction order times the number of blocks in the full array. For a high-resolution study, $L$ might be 15 and $I = 64$. This leads to the restrictive maximum error of $\leq 0.0005$ unit cells interpolation error. Reducing $L$ and $I$ to 10 and 8 respectively would lead to a manageable error of 0.6%.

Considerations of the computation time are also important. Assume once again that we are working with an $N \times N$ array, which we want to transform in a piece-wise manner as $K \times K$ blocks. Let $I = \frac{N}{K}$. Each small (block) DFT will take time proportional to $K^2 \log_2 K^2 = 2K^2 \log_2 K$. Since there are $I^2 \left(= \frac{N^2}{K^2}\right)$ of these blocks, total DFT time will be:
DFT time $2N^2 \log_2 \frac{N}{I}$

In addition, we must phase-shift each member of all blocks except the first, so we need to add $N^2 - K^2$ complex multiplications.

Total time $2N^2 \log_2 \frac{N}{I} + N^2 - \frac{N^2}{I^2}$

$\approx N^2 \left( \log_2 \left( \frac{N}{I} \right)^2 + 1 \right)$

How does this compare with the times required for the hybrid method or complete FFT? The complete FFT would take $N^2 \log_2 N^2$ compared to $N^2 \left( \log_2 N^2 - (2 \log_2 I - 1) \right)$. Again, for the case $N = 2048$ and $I = 64$, the time to complete the piece-wise transform as a fraction of full DFT time would be $T_{\text{complete}} - T_{\text{piece}} \cdot 100\% = 50\%$. This is a dubious time saver unless the scanning lattice is very precisely positioned with respect to the crystal lattice.

The comparison with the hybrid scheme is more advantageous to the piece-wise transform. From above, the hybrid time is $N^2 \left( \log_2 N + \frac{\# \text{ spots}}{N} \right)$. Hybrid and piece-wise times will be equal when $N \left( \log_2 2N - \log_2 I^2 \right) = \text{number of spots}$. When the left side of this equation is smaller than the right side, the piece-wise transform is faster. If $2N \leq I^2$, then the piece-wise transform is faster, even with no spots being calculated for the hybrid approach. For example, with $N=2048$, $I=64$ (corresponding to a block size of $K=32$), the piece-wise method is faster. With $N=4096$, $I=128$
(block size of K=32), the piece-wise method is still faster. If the block size were to be doubled to K=64 in the above examples, the piece-wise method becomes cheaper only after 4096 Fourier coefficients have been calculated by the hybrid algorithm.

Thus, we see that the piece-wise transform would be most useful in cases where the crystalline lattice is very accurately sampled. The savings over a full DFT or a hybrid DFT are not great and will not exceed 50%. The method is of more interest for what it shows us about the nature of the DFT than for its usefulness as a computational algorithm.

E. Ordering of the Fourier Coefficients in the DFT

The output of the 2-D Fourier transform routine looks a bit unusual to someone familiar with optical transforms of images. The computed transform, using any of the common FFT algorithms, has its origin at array element 1,1. Spatial frequencies in the FFT array increase with row or column number until the "middle" row and/or column is reached, whereupon negative spatial frequencies, in reverse order, are found (see Figure 26). In any dimension, the "middle" element is $N - \text{int}(\frac{N}{2}) + 1$, where $\text{int}(x)$ is the largest integer $\leq x$. In keeping with the PLTPROC system's philosophy of presenting results in an intuitive and recognizable manner, such arrays are "flipped" before being displayed. This transformation, as diagrammed in Figure 26, places the negative and positive frequencies in a more usual relationship.
Figure 26

Frequency ordering in the FFT, before and after array flipping

A) After taking the FFT, the origin and frequencies are as shown. A circular pattern would appear broken into four arcs. The array can be subdivided into four blocks: A, B, C, and D

B) By rearranging these blocks, a "normal looking" diffraction pattern is obtained. In this orientation, $+S_x$ is toward the bottom, $+S_y$ is toward the right.
to one another and makes interpretation more straightforward.

Prior to inverse Fourier transformation, the array must be flipped back to its initial (FFT-compatible) configuration. A simple method to accomplish the flipping task would be to store half the array, switch the remaining half, then switch the stored half. This method suffers from having to store, then retrieve half of the array (or else reserve extra memory equal to half the array size). The large size of the arrays transformed in core (up to 360 x 360 complex) makes this an expensive method.

In arrays with even numbers of rows and columns, array elements can be swapped in order to accomplish array flipping. The flipping transformation would bring $A(I_1, J_1, K_1)$ to $A(I_2, J_2, K_2)$ and $A(I_2, J_2, K_2)$ to $A(I_1, J_1, K_1)$. This means that one merely needs to systematically switch element pairs until one half of the array has been switched with the other half. The pairs to be switched are in mutually exclusive blocks so that no fancy calculations are needed to decide which pairs to flip next. Starting with the first array element, which is to be flipped with the "middle" array element, one can go through the array, element by element, until the "middle" element is reached. A second fortuitous circumstance is that the linear index of the switching partner can be calculated with a single Boolean operation on the index of one member of the switch-
ing pair. For an array of dimension \( L \) rows by \( M \) columns by \( N \) planes, the linear index for an element \( A(I,J,K) \) is:

\[
\text{index} = (K-1)L'M + (J-1)L + (I-1)
\]

The swapping partner's linear index is generated by performing a Boolean exclusive or with a "magic number" and the linear index of the array element in question. The "magic number" is the linear index of the "middle" array element, \( A(\frac{L+1}{2},\frac{M+1}{2},\frac{N+1}{2}) \). While this may not sound like a great advantage, the time savings can add up when one is flipping and inverse flipping many 300 by 300 arrays. A machine language routine was written to take full advantage of the speed of such a flipping method.

For multi-dimensional arrays with one or more odd length indices, pairs of elements do not interchange, so another algorithm is needed. The first step to achieving this algorithm is to decompose the flipping task into a series of operations on the various dimensions of the array. For example, a three-dimensional array is said to be composed of rows, columns, and planes. Using these divisions, flipping can be accomplished by rearranging full planes first, followed by rearranging column positions with each (moved) plane and ending with the rearranging of the contents of each (moved) column. The order of operation is not important in this case. Columns could have been done first, for example. In an abstract sense, there is no difference between rearranging planes, columns, or rows, so a single
algorithm will serve to accomplish all three tasks. If the dimension we are rearranging is even, then a pair swap can be used, just as in the case for an all-even array. In this case, though, individual elements are not necessarily switched, but rather entire planes or columns can be switched. For example, if a 33 row by 17 column by 100 plane array is to be flipped, planes can be swapped, but not rows or columns. Plane 1 will become plane 51 and plane 51 will become plane 1. A generalization of an array element swapping algorithm is used to rearrange the members of any even length dimension.

What about odd length dimensions? Let us investigate the frequency assignments in an odd length linear DFT. As usual, element 1 is the zero frequency term. There follows an equal number of positive and negative frequencies, grouped and ordered much like the even length transform. The difference between even and odd length DFT's is that in the even there is one fewer positive frequency than negative; while in the odd, there are equal numbers of positive and negative terms. Upon switching a 100 element (even) DFT, element 1 would switch with 51, 2 with 52, ..., 50 with 100. In a 101 element array, on the other hand, 1 would go to 51, 51 to 101, 101 to 50, 50 to 100, 100 to 49, etc.. The disadvantage of this behavior is that the rule to follow is more complicated than a simple pair-wise exchange. In principal though, this is not a handicap. To flip, we could save element 51, replace it by element 1,
save element 101, replace it by element 51, etc. until all elements were flipped. In practice, however, such an approach could become expensive, especially when the elements to be saved were columns or worse yet, planes of data. It would be better to move elements directly from their unflipped location to their flipped location, eliminating the need for intermediate storage in another location in memory. This can be accomplished by storing just one element (or 1 column or 1 plane). In our example above, if element 51 is stored, then element 1 can be moved to 51. Element 52 can go to element 1. Element 2 goes to element 52. Element 53 goes to element 2, etc.. Finally, element 51 is retrieved from storage and is moved to element 101 to complete the flip. Rather than 101 intermediate stores, only one has been needed. Therefore, when an odd dimension in a multi-dimensional array is encountered, an algorithm based on the above ideas is used to flip that dimension.

The disadvantage of this segmented approach, as opposed to an approach which would move individual array elements regardless of the dimensionality of the array, is that every array element is moved N times for an N-dimensional array (and some elements even more times if they are temporarily stored). An array element flipped by the unified approach need only be moved twice, once to a temporary storage location and once to its final "home". Since almost all our flipping is done on two-dimensional data, this is a slight problem, at worst. Even in the 3-D case, the avoidance of a
destination calculation for every element could save as much time, if not more, than is lost by the extra data movement.

F. Picture Generation

The ability to display visual information is emphasized in my approach to image processing. Of course, there are many ways to implement visual output. The use of computer overprinting is an easy-to-program technique that yields quick results, the results of which have the added benefit of being physically combined with the rest of the program's printed output, thus forming a permanent part of the physical record. An overprinting subroutine has been made a part of the image processing system. The routine is based on a greyscale advanced by Henderson and Tanimoto (1974) and characterized by a choice of overprinting symbols that avoid directional bias. This is an important consideration when displaying large areas of constant density. The existence of patterns in the basic grey-level symbols will cause the human eye-brain combination to integrate these patterns over a large area and make a supposedly uniform region appear patterned. This problem is not limited to overprinted pictures, as can be seen by Figure 27, which is an example of a poor choice of grey-level patterns for a dot picture (to be explained below).

The inability to display adequately many shades of grey and the rather coarse resolution (in our case 10 pixels/inch horizontally and 6 pixels/inch vertically) are factors
Figure 27

Patterning in a greyscale dot picture

A) Each pixel's dot pattern is determined by a pseudo-random number generator

B) The pixel dot patterns in this picture were generated by a variety of strategies. Large areas have the same dot pattern and show the effects of poor pattern choice.
working against the overprinting method. In addition, large pictures (256 x 256 pixels, a common size of array) cannot be printed on a single width of computer paper, making cut and paste a possibility.

In the PLTPROC system, the printer pictures are available for use as a debugging aid or in any application where speed and/or large pixel size is needed. Large pixel size is very useful when finding the boundaries of particles, because exact row or column number information can be found for any feature in the image. With this in mind, each row in the printer picture is labeled with the complete row number, and each column is labeled with the last digit of the column number, the complete column number range being printed as a heading to the picture.

The mainstay of the visual output for the PLTPROC system was originally a CRT device connected to a 35mm camera, the Control Data 250 Series Display system. This device had a 1024 x 1024 raster and the ability to write a spot of constant size and intensity at any one of the raster points. Based on a routine implemented at the Lawrence Berkeley Laboratory by Dr. Ivy Kuo, a program was written to produce pictures on this device in an efficient manner.

A brief description of the algorithm for producing greyscale pictures on this device follows. The raster space is divided into a quilt of pixels of size N x N raster points. Each pixel then has the potential of containing
from 0 to $N^2$ spots. An increased number of spots placed in a pixel leads to a more exposed negative at that pixel location. Rather than the $N^2 + 1$ variations in transmission possible, a more modest $\frac{N^2 + 1}{2}$ different grey levels was used. The fact that each spot, when recorded on the film, has a diameter greater than the inter-raster distance, coupled with errors in CRT beam positioning, lead to the need to use less than the maximum number of spots allowed by the pixel size. There are choices when deciding how to fill up the pixel with spots. Two general approaches have been used. The first is a pattern approach, the second a random one.

In the pattern approach, each grey level is achieved in a preset way so that all pixels with six spots, for example, will appear identical. Pattern choice is important in this case because large picture areas filled with the same pattern can disclose a "super" pattern. The individual pixels work in concert to produce annoying large-scale patterns in a supposedly uniform area. To avoid directional bias in my patterns and to avoid having to store a great number of patterns (one for each grey level and pixel size combination), I chose to use a universal spiral pattern for all grey levels and pixel sizes. A single table of 32 entries containing x and y raster displacements is used to predict the location for any grey level pattern, up to a pixel size of 8 by 8. This maximum pixel size could be increased by enlarging the table, but an 8 by 8 pixel is about the largest
pixel size that can be tolerated in terms of the physical size of the pixel in the finished picture. Larger pixels look "block-like" and are too obvious to the unaided eye. In the spiral pattern approach, the center of the pixel (defined as \( M, M \), where \( 1, 1 \) is the lower left corner of an \( N \times N \) pixel and \( M = \text{largest integer} \leq \frac{N}{2} \)) serves as the origin. The first spot for any pattern is placed at the origin. As further spots are needed, their raster coordinates are calculated by adding \( x \)- and \( y \)-offsets to the origin location. The table is constructed so that the spots spiral outward counterclockwise from this origin, leaving open the raster positions at adjacent raster locations, and approaching other filled raster points only as diagonal neighbors. This \( \sqrt{2} \) increase in inter-spot distance gets around the enlarged size of the spot so that overlap is cut to a minimum. Truncating the number of allowed spots at \( \frac{N}{2} + 1 \) guarantees that raster locations belonging to neighboring pixels will not be filled. The single table of offsets is applicable for a square pixel of any size. Indexing the correct entries is trivial, being tied directly to the number of spots needed. If \( S \) spots are needed, the first \( S \) entries are used.

The random scheme for spot generation is the other alternative to placing spots within the pixel. This method insures that any two pixels of the same spot density will bear only a pseudo-random relationship to one another. The \( x \)- and \( y \)-offsets for any spot are determined by using a
random number generator. The pixel origin is taken as 1,1 and the offsets are constrained to the 0 to N-1 range. The process used emulates the statistical nature of photographic recording and eliminates problems of unwanted patterning in large areas of uniform density (see Figure 27). One of the drawbacks to the implementation of this scheme is the need to use a random number generator twice for every spot created. Since a 256 x 256 picture having an average spot density of half the useable maximum (four spots/pixel) would need over 260,000 spots, 520,000 calls to a random number generator would be needed at a minimum. This represents a sizeable computational overhead since calling a function involves the storage of all internal registers in the computer (15 18-bit registers and 8 60-bit registers in the CDC 6600 and 7600 computers) prior to the call. To remove this overhead, a machine language subroutine was written which incorporated a pseudo-random number generator resident in the operating registers of the computer, eliminating the need to transfer to an external random number generator. The generator used was based on the CDC function RANF. Economies in output buffer management and compression of the device-specific spot writing instructions were also incorporated at this stage to decrease computer costs further. The resulting subroutine is called once per array column reducing the 520,000+ functions calls, in our example from above, to 256 subroutine calls per picture. This condensation of function was undertaken because of the anticipated
high volume of pictures that would be produced in the course of image processing. Typical CPU (Central Processor Unit) times for an average density 256 x 256 picture on the CDC 7600 are approximately 1 second. This corresponds to an approximate cost of 18 cents per picture.

The most annoying problem involved in the production of pictures is the inability to capture adequately the dynamic range of the data. As a first effort, the minimum and maximum data values found in the displayed data were used to construct a linear mapping between the data values and the available grey levels (typically only eight grey levels are available for a 256 x 256 picture). Unfortunately, the distribution of data values is not uniform over the minimum/maximum interval, tending to lump about a mean and then tail off. Thus, many of the available grey levels were wasted on relatively few data points, leaving the majority of the picture area represented by only two or three levels.

A method was needed to adjust, automatically, the linear mapping to give the best-looking pictures possible, consistent with the least amount of extra computer time needed to achieve this standard. The following adjustment is used. A 1000-bin histogram is made of the data in the picture area. The display minimum is moved higher and/or the maximum is moved lower bin-by-bin, until a user-defined percentage of the data points has been excluded as being above the new maximum and/or below the new minimum. The
intent is to "throw out" that small percentage of data points which are exceedingly dark or light (high or low values). When converting the actual data to picture values, array elements with values outside the new limits can be detected and treated differently from those inside those limits to produce, for example, pictures which display only data in a predetermined range. This "windowing" is most easily achieved by letting all data values outside the window map into blank pixels. To provide "normal"-looking pictures, data values below the picture minimum are mapped into blank pixels and those above the picture maximum are mapped into the most filled (darkest) pixels. Adding this minimum and maximum adjustment subroutine proved to be the difference between consistently good pictures and consistently terrible ones.

G. General Computer Considerations

The actual programs written to implement the analysis of electron micrographs will now be described. Since the exact data requirements and formats are not of general or lasting interest, they will not be described here. Comments embedded in the actual computer code contain these details. When such comments are lacking, an examination of the Fortran code yields these details. What will be highlighted is the purpose of the various programs, their interrelationship, and their place in the overall image analysis procedure.
A few words about computer programs and computer organization is needed to prevent misunderstandings and drawn-out explanations later in this chapter. Although parts of this discussion apply to the specific characteristics of the LBL BKY computer system, many of the concepts are of a general nature. Programs, or instructions for the computer, written by us are called software. Hardware, on the other hand, refers to machines, devices, and equipment. Often hardware and software are considered to be independent, but in reality they interact. The procedure outlined by the software must be carried out by the hardware, and the specific physical organization of a machine can have a bearing on what is attempted and how it should be done, even though such considerations are not ideally part of software. In the code descriptions to follow, mention will be made when hardware considerations have been important.

One last word about the general organization of the programs developed to implement EM image processing. Some programs are quite complicated and perform a set of operations on the data that is presented to the program. Others are quite simple and perform only one task. The structure of the computer operating system at LBL allows a series of programs to be invoked sequentially so that it is possible to string several programs together to perform more complicated sets of tasks. Such stringing is made possible by attempting to have the output of one program be in the correct form so that it can be used as input for another.
Unfortunately, this is not always the case, so that some sequential operations that can be envisioned are not possible at this time. In addition, the need for human interaction still exists at many of the junctures of the image processing road and thus prevents the mechanical linking of operations.

H. Scan Conversion Program

As we have seen, the first stage in processing is to reduce the photographic image to a computer tape filled with numbers. This "reduction" is undertaken with the scanning densitometer. The exact bit patterns used to represent numbers in the minicomputer controlling the scan are different from those used in the CDC computers. Thus the scanning data, composed of 10-bit integers, appears to be nonsense if directly interpreted as numbers by the CDC computer. The first computer task is to translate the raw scanning tapes to a form recognizable by the CDC computer. Besides performing this translation, the program provides several other services. In order to identify the origin of a set of numbers (what image they represent), I have chosen to insert a set of identifying numbers at the beginning of all arrays which I use in image processing. Since the aim of my programs is to provide flexibility for others who might use them, the addition of this ID (identification) record, or logically distinct piece of information, is optional.
Another option describes the manner in which the scanning data is stored. As mentioned above, the scanner produces 10-bit integers. Since the standard CDC computer word has a length of 60-bits, it is possible to fit six scanning values into every CDC word. This necessitates taking these "packed" words apart in order to perform any mathematical operations on them, but it does lead to an $83\frac{1}{3}\%$ reduction in the amount of storage needed for the scanner data since all sixty bits are filled with data rather than just ten. The computer time saved by reading fewer words compensates for the extra time used to unpack them, so the process is not only space efficient but also time efficient. Again, changing the scanner data to "packed" form is strictly optional.

Finally, since the Perkin Elmer scanner can operate in either of two modes, producing numbers proportional to optical density or ones proportional to percentage transmission, an optional conversion can be invoked while translating the scanner tape to change transmission values into values proportional to optical density. There are certain scanning conditions which necessitate the use of transmission output even though optical density values are desired. Since OD is achieved by electrically integrating the scanner output over time, the values thus obtained will be in error if a scan is attempted on a high-OD plate, using a small aperture and a fast scanning speed. In this case, the electronics do not have time to integrate the signal accurately before another reading is required. Transmission mode, which produces an
instantaneous reading, is preferable for such specimens. The optional transmission-to-OD-like conversion in the translation program nullifies this OD problem. The low-dose, low-OD images used in high resolution electron microscopy can be scanned in OD-mode with no problems.

I. The Histogram Program

One method of analyzing the results of a scan is to produce a histogram of the optical density or transmission spectrum. A separate program produces histograms from scanner data. At present, the output must be displayed on either a plotter or a film writing device; no printer produced histogram has been implemented. The histograms are composed of 1024 bins, one for each possible scanner output value. Through the use of options, the user can produce one plot which spans the entire spectrum, five plots each covering one-fifth the spectrum, or a variable number of plots covering user-defined portions of the spectrum. Figure 28 is an example of the histograms produced.
Figure 28

An example of scan histogram program output

A) One option produces a histogram over the entire 0-1023 scanner range

B) A user-specified range can also be plotted to blow-up interesting areas
J. Lattice Definition and Viewing Program

Locating and looking at the reciprocal lattice of a crystalline specimen is one of the important steps in image processing for electron microscopy. The logarithmic pictures and maps of the diffraction pattern, as produced by PLTPROC for example, go far in fulfilling this need, but the amount of human labor required to view and quantify the lattice justifies a computer approach to the mundane aspects of this task.

The "lattice look" (LTCLOOK) program performs several useful functions. Given a calculated diffraction pattern stored on disk and a set of data cards indicating both the Miller indices and the row and column locations of a number of diffraction orders, the program calculates a least-squares fitted reciprocal lattice to the input data. A useful abstract of the information found in the diffraction pattern is printed on the basis of the calculated lattice vectors. Depending on the planar symmetry of the crystal, symmetry related diffraction orders are collected and printed together. The output for each diffraction order consists of two two-dimensional displays of a portion of the Fourier transform around each calculated reciprocal lattice point. One display presents the scaled amplitudes, the other the phases in degrees (see Figure 29). The number of each row and column is displayed and the location of both the theoretical reciprocal lattice point and location of the
Figure 29

An example of LTCLOOK output

Each diffraction spot found in the array produces similar amplitude and phase maps
\[ H = -2 \quad K = 1 \]

LOCATION OF LARGEST PEAK IN BLOCK = (127, 68)

EXPECTED LOCATION = (127.500, 68.000)

AMPLITUDE = 0.2067091E+05, PHASE = 110.75367

SCALE = .1E-01, MIDDLE = (127, 68)

<table>
<thead>
<tr>
<th>AMPLITUDES</th>
<th>PHASES</th>
</tr>
</thead>
<tbody>
<tr>
<td>COL</td>
<td>64 65 66 67 68 69 70 71 72</td>
</tr>
<tr>
<td>ROW</td>
<td>123 44 39 11 24 45 11 34 22 51</td>
</tr>
<tr>
<td>ROW</td>
<td>124 28 8 29 53 12 27 37 2 48</td>
</tr>
<tr>
<td>ROW</td>
<td>125 36 12 36 58 31 22 43 16 4</td>
</tr>
<tr>
<td>ROW</td>
<td>126 7 38 3 53 83 48 42 45 46</td>
</tr>
<tr>
<td>ROW</td>
<td>127 53 21 53 28 207 68 18 35 47</td>
</tr>
<tr>
<td>ROW</td>
<td>128 33 32 42 22 131 70 32 22 36</td>
</tr>
<tr>
<td>ROW</td>
<td>129 53 23 37 29 51 22 39 41 60</td>
</tr>
<tr>
<td>ROW</td>
<td>130 51 63 25 25 50 16 56 16 45</td>
</tr>
<tr>
<td>ROW</td>
<td>131 17 41 23 32 38 36 40 12 64</td>
</tr>
</tbody>
</table>
largest amplitude term within the displayed area is printed along with the Miller index of the spot. This type of display is a slight modification of that used by Unwin and Henderson (1975). Among other functions that could be incorporated in the future is a statistical analysis of the contents of each displayed area to determine if the highest amplitude is more likely to be noise or signal. An algorithm for choosing a most likely diffraction spot, then calculating a weighted average phase and sinc function corrected amplitude for this peak, could also be incorporated. Data cards containing frequency, amplitude, and phase information are written for use in the PLTPROC filtration program.

The presentation of this type of summary of the diffraction pattern is a boon to image processing for electron microscopy. Locating spots that are readily recognizable from log maps or pictures takes a long time. Searching for harder-to-see peaks requires even more time. The log displays do not impart a good idea of the surrounding noise and the search for symmetry-related spots is frustrating. By requiring only the locations of a few spots, the lattice viewing program reduces handwork. More of human thought can be applied to analyzing the results at hand. The least-square lattice is more accurate than hand-drawn lattices so that candidates for diffraction spots can be screened with more confidence and finality. The enormity of the number of calculated coefficients for even a small \((128^2)\) image is
inspiring (almost 17,000 amplitudes and 17,000 phases). Eliminating the need even to see the vast majority of these numbers is important.

This program is presently separate from the PLTPROC system, operating on arrays saved on disk by PLTPROC. In an interactive environment, the input data to LTCLOOK could be generated by light-pen hits on a log map of the PLTPROC-generated diffraction pattern. In addition to a paper listing of the area around each spot, a two-dimensional perspective plot could be made, helping us decide if noise or signal lurks near a given reciprocal lattice point.

K. Phase Origin Refinement Program

One means of translationally aligning images of two crystals is to move the coordinate origin of the two Fourier transforms onto a common symmetry axis. For crystals, rotational alignment can be achieved by consistent assignment of Miller indices. This movement of the so-called "phase origin" is accomplished by monitoring the phases of symmetry-related diffraction orders. Each type of symmetry-axis leads to a phase relationship that must hold between certain diffraction spots if the coordinate origin is coincident with that symmetry axis. The easiest way to find a suitable phase origin is to conduct a two-dimensional stepwise search for a minimum in some function which mirrors the discrepancy between theoretical and actual phase relationships. Two-dimensional maps of this function can be used to assess
visually the location and "depth" of the minimum. Since the crystal is a repeating structure, there will be an infinity of minima, the spatial disposition of these minima mimicking the symmetry of the axis. Since there is no need to use any minima farther than one unit cell from the original (randomly chosen) origin, this periodicity causes no practical problems.

Rather than blindly searching for a minimum, a system of linear equations, solved by a least-squares method, could determine the "best" minimum. A complication arises because each group of symmetry-related diffraction spots possesses a periodic array of minima. The system of equations must be constructed to echo this fact. The 2-D maps offer an advantage over the least-squares method in that more control is exerted over the final choice of origin, with the experimenter having a feel as to the relative influence of the various inputs to the final output. If control is not desired, the computer can be programmed to search the maps for the global minimum. In fact, the maps do not even need to be first generated, then searched. Rather, the maps can be searched as they are generated, obviating the need to store the finished map.

Besides calculating the minima map, the program can be given a (tentative) phase origin and instructed to recalculate the phases at this new origin. The output of this step compares symmetry-related spots at this origin, showing each
group's contribution to the overall RMS error. On this basis, one might want to recheck the input data or discard some diffraction orders from further consideration because of unacceptable error in it.

An interactive phase origin manipulating program has also been written which makes no assumptions about symmetry. The diffraction data is provided by the user and the phase origin shifted at will. Phases are recalculated at each origin and displayed under user control. Although this program is a simple-minded predecessor to the current batch-oriented program, the interactive features could be wed to a more powerful algorithm for origin determination to yield a useful and fast program.

L. The PLTPROC Program

The bulk of the routine image processing has been collected into a large program named PLTPROC (PLaTe PROCessor). This program can accomplish many tasks, depending upon the user to supply instructions via data cards as to what to do. Effective use of the available features depend on not only being aware of the options but also having some idea of the organization of the program. I first outline the structure of PLTPROC, then describe the currently available options. Finally, some suggestions for changes are made.

As shown in Figure 30, there are four major activities in the PLTPROC program: numerical data input, program com-
Figure 30

Organization of the PLTPROC program
mand input, data processing, and display and storage of the results. The processing steps are the best place to begin an explanation.

The heart of the PLTPROC program lies in the processing steps. Processing is arbitrarily divided into four main parts, each of which corresponds to a form that the data takes as it passes through the program. The first stage in processing I term the "spatial stage". Normally, real space data is input (spatial domain), hence the name. The second stage is termed the "transform stage" because the Fourier transform and related functions are performed at this point. The third stage is the "modulus stage" and the fourth, the "logarithm stage". The last two stages are potential candidates for inclusion in the transform stage but remain distinct to keep the display of results sections of the program uncluttered.

Each processing stage is composed of an operation section and an output section. The allowable operations vary from stage to stage, while the output options remain constant, except for changes in the form of the output. First, I will discuss the operations, then the output. Currently, there is only one operation allowed in the spatial stage of processing. A simple bilinear interpolation can be used to produce an output array which is written to disk in a form suitable for input by subsequent instructions to the program. The interpolation routine assumes that the data is in
complex form and thus interpolates both real and imaginary parts. The interpolated data must lie wholly within the original data area and only rigid rotations of orthogonal sampling axes are allowed. There exists a more general interpolation program (LRGINT) which operates separately of the PLTPROC program (Section B, above). The PLTPROC interpolator is usually exercised to modify (or re-establish) the spacing or geometry of a specimen in order that it look correct in various display types. For example, an orthogonalized hexagonal specimen (III-H above) can be returned to hexagonal form prior to producing a contour plot. Another use is to distort the sampling to counter the distortions of sampling inherent in a printer picture (ten pixels per inch horizontally versus only six pixels per inch vertically). What of other operations that could be done to spatial (scanner) data such as selection of sub-areas, masking, floating, or data averaging? These modifications to the input data are handled in the data input stage of the PLTPROC program.

The transform stage of the program carries the largest number of operation options. Data in the transform stage might represent spatial data about to be transformed, transformed data, reciprocal space data before inverse Fourier transformation, or inverse-transformed transform data (spatial data again!). As can be seen, the interpretation of the data at this stage (and at any stage for that matter) is in the mind of the beholder. The only solid
interpretations are that the spatial and transform stages expect to work with data in complex form, while the modulus and logarithm's stages work with data in real form. Of course, each stage can only perform certain operations but will blindly apply these operations to whatever occupies the data storage region in the computer.

One can perform either a forward or an inverse DFT using Singleton's (1969) mixed radix algorithm. Cards can be read which protect user specified regions of the transform from being zeroed, thus effecting a mask generator. The array under consideration can be flipped (see Section E, above) either before and/or after transformation. In the language used in the program, there are two directions of flipping, forward and reverse. A forward flip takes the array element 1,1 to array element \( \frac{N}{2} + 1, \frac{M}{2} + 1 \) for an \( N \) by \( M \) array, while a reverse flip reverses this process. Flipping done before the Fourier transform is "reverse" flipping, while that done after taking the DFT is "forward". The latter takes a computed DFT and makes it presentable for display by placing the undiffracted term in the center of the array. The former returns an array so flipped to a form amenable to inverse Fourier transformation. Again, the user is free to use these operations on any data with complex form, not just DFT's. If a spatial array is flipped prior to transformation, the effect is to change the location of the phase origin to the center of the box delimiting the data, thereby eliminating the alternating
$180^\circ$ phase shift caused by the convolution of a displaced box function (see III-I, above). Post-flipping after an inverse DFT can be used to put the phase origin so moved back to its original position.

The modulus stage accommodates the calculation of the modulus of the complex array residing in the computer, together with a possible flipping operation (forward flip). The flip duplicates that which can be done after the DFT but is included because it is cheaper to flip a real array (post-modulus) than it is to flip a complex array (post-DFT) since complex arrays contain twice as many computer words. Both abilities are needed since there are occasions when the DFT needs to be flipped before the modulus is taken, as in the production of logarithm maps (see below). The modulus stage also includes the option of replacing the undiffracted term ($F(00)$) with the next most intense term in the array. In most cases, the $F(00)$ term is $10^2$ to $10^3$ times the amplitude of any other term in a diffraction pattern. This disparity in amplitude makes displays of the diffraction pattern difficult and nothing is lost if the $F(00)$ term is thrown out for display purposes.

The last stage is the simplest. In the logarithm stage, the $\log_{10}$ can be taken of the real array stored in the computer. This array is usually the modulus of the diffraction pattern. The log is taken in order to produce an array which can be effectively displayed by the routines.
available (see below). Of course, the PLTPROC program will take the $\log_{10}$ of any real array placed in core, regardless of its "meaning".

Much of the work performed by the PLTPROC program is not in the implementation of the operations mentioned above but rather in the effective communication and storage of the results of these operations. It is this ability which makes the PLTPROC program useful to the experimenter since interaction with the results and reuse of them is greatly facilitated. In general, three forms of output are employed. Numbers can be printed, represented by greyscale pictures, or written to disk for later retrieval and re-use. For historical reasons (i.e. that is the way I first decided to do it), other types of output such as histograms, contour plots, and super-imposed contour-grey scale plots have been implemented outside the PLTPROC program. Each processing stage customizes these three output types so that its form is appropriate for the data found in that stage. One of the features of PLTPROC is that picture or print output can be easily limited to a user-specified portion of the data array. In this manner, only data of interest is presented to the user. The mechanics of this choice will be detailed below in the section on program instructions. The storage of the data on disk is currently all-or-none. If the user requests disk storage, the entire array (with ID record) is written to disk.
How are the print and picture displays customized? For spatial arrays, a print command will cause the printing of only the real part of each complex number. Numbers are printed column by column, each column number being identified. Row numbers are not identified. A spatial picture will generate a greyscale picture of the real part of each complex number. Higher numerical values lead to darker regions in a print made from the negative produced by the program. A picture option, available for any picture type, produces histograms of the data within the area to be displayed and adjusts picture digitization minimum and maximum to yield a picture which linearly spans the central portion of the histogram, avoiding the wings. This adjustment uses the limited number of grey levels to good advantage by not wasting them trying to span the relatively sparsely populated histogram wings. A title is generated for each picture produced, containing comments and identification supplied by the user at the time of the scan translation (or changed through the use of PLTPROC options). The title contains the name of the processing stage that generated the picture, in this case spatial.

The transform stage contains a few more output options. Array printing is of two types. A straightforward print yields an output of both real and imaginary parts of the complex array. As in a spatial print, column numbers but not row numbers are printed. A second print option is the production of an amplitude and/or phase map, similar to that
made by the MRC programs (see below). To get amplitudes and phases, the complex transform is converted from "real and imaginary part" form to "amplitude and phase" form. The phases are in radians. This conversion is done "in place" so that the results "destroy" the original transform in core. This action is in contrast to all the other display functions which leave the original array contents unchanged. Because of this property, this transform print option can (at the user's discretion) be used solely to convert from real and imaginary form to amplitude and phase form. In this case, the printing portion of this routine is suppressed. After conversion, the amplitude and phase terms alternate in core.

These two intercalated arrays can be optionally output as maps. The phase map is simple, the amplitude map a bit more abstruse. Both maps establish a one-to-one mapping of array location to location on the printed page so that the geometry of the array is roughly preserved (remember the unequal pixel sizes for horizontal and vertical spacings). The phases are digitized into the range $-90$ to $+90$, representing the domain $-\pi$ to $+\pi$ radians. Thus each digitization level represents two degrees in angle. Since the amplitudes in a diffraction pattern can vary dramatically in scale, a linear mapping onto the DFT will lump the majority of points into only a few values. For the same reason that the logarithmic transformation is useful in producing grey-scale pictures, it is also useful in displaying amplitudes
as multi-digit integers. A choice was made to use integers in the range 0 to 99 to represent the amplitudes, as well as to cover four orders of magnitude in the diffraction pattern. This four-order coverage is adequate to capture the most dynamic diffraction patterns encountered, that of negatively stained specimens.

To produce a $\log_{10}$ map of the diffraction amplitudes which spans four orders of magnitude, the following is done. First, the $F(00)$ term is replaced by the term following it (which is always orders of magnitude less strong if the array was not floated. This is done to avoid having the $F(00)$ term potentially distort the mapping with its exceptionally high value. Then, the common logarithm is taken of the area to be mapped. The largest term found in this area is used as the digitization maximum. The minimum is set to 4.0 less than the maximum. All terms are digitized to map the integers 0-99 linearly to this range. All terms less than $10^{-4}$ of the maximum amplitude in the area have a digitized value of zero. This mapping yields a scale wherein each unit in the map represents a multiplier whose value is the $99^{th}$ root of 10,000 (approximately 1.1). A difference of 7 units between two values corresponds to an amplitude ratio of approximately 2:1 (1.9:1). A difference of 4 units represents an intensity ratio of approximately 2:1 (2.1:1).

The choice of two-digit maps coupled with one "separating space" was made to increase the amount of information
contained in each map. The single digit map, as used by the MRC group, coupled with the base 2 logarithm, leads to an ability to capture a range of 1 to 512 on a single map, while producing a resolution of only ten levels. Given the range of amplitudes in some diffraction patterns, I felt that this limitation would require additional maps to be made in order to span the useful range of diffraction amplitudes. The increase of resolution from ten levels to 100 levels would also be useful in investigating the behavior around diffraction peaks. The phase mapping of $2^0$ per unit in the range $-180^\circ$ to $+180^\circ$ is more useful than the mapping of $0-360^\circ$ into the character range $0-9$, A-Z in ten degree intervals used by the MRC algorithms. Again, the increase in resolution is useful and the more direct connection between digital representation and actual value does away with the need to consult a character-phase conversion chart (phase wheel). A ten-fold increase in resolution is achieved with no extra computational cost. The use of a $+180^\circ$ range is more in tune with crystallographic symmetry, where phase relations of the form "a and \( -a \)" are often found.

The price paid for these increases in usefulness is an increase in the area of computer paper needed to represent a given array size. A cramped single-digit map can be made by juxtaposing the digits with no intervening space. A slightly more "airy" display would include a space between each value. The two-digit display requires three spaces in
the horizontal direction for each value as a space is mandatory between each value (at least for anything that is useful for a human being). The worst case comparison leads to the two-digit map using three times as much paper as a single-digit map. Each page of standard computer paper (11 inches by 14 inches) can still hold approximately 60 rows by 40 columns of the map. The increased legibility and the added capability of being able to label the rows completely and the columns to two-digit accuracy makes the larger display more useful for quantitative use.

Picture making at the transform stage is also different than it is at the spatial stage. Since the data is complex with both the real and the imaginary parts carrying useful information, a single picture can not suffice. Two approaches can be used to display the complex array. In one, two pictures are produced, one of the real parts only and one of the imaginary parts only. The second option produces four pictures. Again, there is a separation into real and imaginary parts. Each of these is further divided into two pictures. One, done in normal contrast, covers the positive values only. The other covers the negative values only and is produced in reverse contrast so that the most negative values lead to the darkest areas on a print of the picture. Within either the two or the four picture option, any of the possible pictures can be suppressed if not desired. All pictures are labelled as transform (or inverse transform if appropriate). The specific picture type (all
real, imaginary positive, etc.) is also placed in the picture label. As with the spatial pictures, the optional "exposure" control afforded by the histogram adjustment is available.

Modulus stage outputs include simple array printing and a picture of the real array. The picture carries a label identifying it as a modulus picture. The logarithm stage similarly offers simple printing but the pictures that are made are more complicated. A request for a logarithm picture actually leads to the generation of three pictures. The range of log values found in the data is distributed into four ranges, of which three are pictured. The largest logarithm value in the displayed area establishes the maximum for the first picture. The picture's minimum is set to 1.0 less than the maximum. Since the data are base 10 logs, the minimum corresponds to diffraction peaks that are a power of ten less than the maximum. A second picture is produced using the minimum of the first picture as its maximum. The minimum for the second picture is taken to be 0.5 less than the maximum. The third picture uses this minimum as its maximum, with the third picture's minimum again being 0.5 less than its maximum. In this way, the second and third pictures together span another factor of ten in the diffraction amplitudes. Any diffraction less than 0.01 of the array maximum (first picture's maximum) is not included in any picture.
How are array elements with values outside the window of a picture handled? For spatial, transform, and modulus pictures, values that are below the minimum are treated as if they were the minimum and values above the maximum as if they were the maximum. This produces the type of effect normally found in optical photographs with overexposed and underexposed areas. The logarithm pictures handle such values in two ways, either selectable by the user. The first choice is to treat out-of-range values as the other pictures do. Thus, information that is included in the first picture is carried into the second as maximally dark areas. Array elements covered by the first two pictures appear dark in the third picture of the series. In practice, this leads to a gradual saturation of the entire negative. The second way of dealing with out-of-limit values is to treat those below the minimum as being the minimum (leading to blanks for them) while treating those above the maximum as if they also were the minimum. Thus each picture contains only information found in its window. Array elements that were represented in a prior picture drop out of the following ones. This helps prevent confusion as to which features are new to a picture and which are carried over from a previous one. The separate pictures can also be used to prepare pseudo-color pictures by providing distinct images for each log window. This exclusive picture approach is the one I usually use to view the diffraction patterns from crystalline specimens.
The decision to use one order of magnitude in the first picture but only a "half" order in each of the two others is a result of experience. With the zero frequency term eliminated in a diffraction pattern, the first window will usually contain a few strong diffraction spots. The remaining diffraction spots that will be visible above the noise usually fall within the next order of magnitude. Two pictures are used to cover this range to provide an expanded scale. The three picture approach is used over a one-shot attempt in order to avoid having to analyze the diffraction pattern closely to determine what display limits to use so that one picture shows diffraction but not noise. There is not enough dynamic range in the greyscale output to allow a single picture to produce a useful picture consistently. Too often such attempts have led to the pattern being lost in the noise by being lumped in the same (or a similar) grey level. The lack of dynamic range of the display is the reason for going to a logarithmic representation in the first place. What we seek is a logarithmicly scaled picture rather than a linearly scaled one.

This capsule summary has presented the operations and display options available in the PLTPROC program. The source of input data, along with the means to direct the program must still be explained.

Input data can come from three sources. The first is integer (scanner) data. Provisions have been included to
preprocess this data before it is placed in the computer. Any integer data can be fed into the program in this manner; it need not come from the scanner. The most common operation performed on this type of data (termed "RAW DATA" by the program) is to convert the integers to complex form so that they may be Fourier transformed using the implemented algorithm. These constructed complex numbers have their imaginary parts set to zero, while the real parts are equal to the integer data. Beyond this operation, many other tasks can be accomplished. Since scans are often much larger than the object under study, it is possible to declare a small, rectangular sub-section of the raw scan as the data which is to be read into core. Often the data has been scanned too finely for the task at hand or perhaps initial (small array) processing needs to be done on a more coarsely sampled version of a larger array. For these situations, an array averaging feature is included, the data placed in core being the average of N by M scanning data points. In some cases, it might be desirable to delimit an object and place it in a larger array for further processing. This is possible, using several choices for a filler value in the larger array, such as 0.0, the array average, or the array perimeter average. Options are included to choose automatically an array size conducive to efficient transformation by Singleton's (1969) FFT algorithm. This algorithm works best on array sizes that are composed of prime factors no larger than five. Procedures did exist at
one time to remove opaque areas and replace them with a value derived from the non-opaque portions of the array. This feature was developed when transmittance data was being used and hence the routine searched for low (transmission) values. Since OD data is being used and opaque regions have high (OD) values, the code no longer works but we have found no real need for this lost ability. Careful scanning or computer masking removes large opaque areas, which would be poor candidates for processing anyway. Gross dirt is not found and "small" dirt can not be distinguished from large silver grains when using small apertures.

A second form of input data is the disk storage form ("TAPE DATA"), presumably the result of a disk store from a previous PLTPROC run (or dataset, see below). Again, the data can be from anything as long as it has the correct form. This data is of four types: spatial, transform, modulus, or logarithm, corresponding to the four stages where it may have been saved. Apart from this label attached to it, there are really only two types of stored data, real or complex numbers. Spatial and transform types are complex data while modulus and logarithm types are real data. The only method the program has to identify the data is the ID record so deception can be used (and often is) to input any data as long as the real or complex distinction is preserved. Once read, the label attached to the data has no significance whatsoever so that data labelled spatial could actually be a Fourier transform, data labelled logarithm
could be displayed as if it were a modulus, etc. The labels exist only as a means to keep track of what data type is stored on tape or disk.

The last source of input data is numbers which have already been put into core storage ("AN ARRAY" data). This allows the PLTPROC program to be called as a procedure for array manipulation from another program, without having to store the array to be manipulated on disk as either integers, real numbers, or complex numbers so that PLTPROC can read it. The full complement of array storage, printing, picturing, transforming, etc. operations are available. Besides tying into other programs, this input option allows data to be read into core through one of the input options, then have its identification record changed before further processing. In some cases, more operations are required on an array than can be accomplished in one passage through the PLTPROC program. Rather than reread the data, it can be declared to be "AN ARRAY" and the program will continue processing it.

The last area to cover in the PLTPROC program is that of instructing the program. Instructions are grouped into datasets. One dataset instructs the program to proceed through all its decision making points once. There is no intrinsic limit to the number of datasets that can be used to instruct the program although there is a current limit of 99. A dataset can do as little as redefine a comment field
or as much as exercising every consistent (or inconsistent if you prefer adventure) operation and display type in all processing stages. Within each dataset there are three parts: instructions dealing with the location of the data for that dataset, instructions defining the size of printed or picture displays, and instructions detailing which operations with what options are to be performed.

Data location instructions inform the program to either read and preprocess integer data, read disk storage data, or use the data already in core. The first two data types must have attached ID records in order to inform the program of array size, as well as to supply title information. The third data type requires the user either to supply ID data or to indicate that the program already knows it (from a previous dataset, for example). By supplying new ID information, it is possible to change the ID associated with a particular set of data.

Display delimiter instructions are next in the dataset. PLTPROC operates by invoking a set of default conditions for displays (as well as for operations). These defaults could be changed in the actual code to effect new default conditions. Usually, however, the defaults need only be temporarily overridden (or the range of variation is so great in the job mix that the concept of default is meaningless). The program allows card input to change temporarily the default display limits. First the user informs the program
whether or not the default values are to be changed. If they are not to be changed, no other instructions are needed in the display section. If they are to be changed, cards indicating the changes to be made are included. When all the changes needed have been indicated, a terminating card is included to enable the program to detect the end of the changes. Each change card lists the processing stage to which the change applies and whether it affects the printing and/or picture making display. Two methods exist for defining the limits of the display. In one, the user explicitly states which rows and columns (inclusive) are to be used. In the other, the user declares normalized fractions of the array size to use. For example, you want to print from 0.0 to 0.5 of the rows and 0.75 to 1.0 of the columns. This latter method is especially suited for multiple datasets dealing with unequally sized arrays. There is a provision for reusing previously changed defaults (see below) and the fractional approach will give the same relative results regardless of array size. The explicit approach, on the other hand, has the advantage of letting you know exactly which row and column numbers will be in the display.

Operation and option instructions compose the last portion of a dataset. Each instruction card in this section contains one field that identifies the processing stage to which the instruction applies and up to seven fields holding keywords which control the operations and options. As many instruction cards as are needed to hold all the keywords
necessary can be used. Keyword fields can be left blank which is useful when editing previous datasets (no need to move all the remaining keyword fields when one or more are removed). There is only one restriction governing the use of multiple instruction cards for a single processing stage. There are four keywords which behave differently from all the others and these must appear on the last instruction card for a given processing stage in a given dataset, if they are to appear at all. These special keywords are the commands to initiate disk storage (ISAVE), result printing (IPRINT), picture generation (IPIC), and one or more operations per stage (IPERF). The operation(s) under IPERF control are 1) bilinear interpolation for the spatial stage; 2) Fourier transformation and flipping (both pre- and post-transform) for the transform stage; 3) finding the modulus and flipping for the modulus stage; and 4) taking the common logarithm for the logarithm stage. Any other operations are controlled by keywords besides IPERF.

The reason for the special treatment for these four keywords is that the processing stage identification field on the instruction card triggers PLTPROC to choose the "no action" route for all four keywords applying to that stage until told differently by an instruction field. This allows a processing stage to be skipped by merely including an instruction card with blank instruction fields. For a processing run with many datasets which invoke very few of the processing stages, this feature can save a lot of typing.
This feature also allows the user to assume a clean slate when setting up custom processing. It is not necessary to worry about what the default instructions have the program doing since mentioning the stage in the identification field will tell it to do nothing for that stage. In the case of multiple instruction cards for a single processing stage, this property of the instruction cards can be very frustrating. The last card in a series of instructions dealing with a single processing stage must be the one to specify which of the automatically negated keywords (INOSAVE, INOPRN, INOPIC, INOPERF) is (are) to be invoked. If this choice is made on any instruction card other than the last dealing with a particular processing stage, the last card will automatically negate any choices previously made for the four keywords mentioned above. As with the display boundary change cards, the instruction change cards are terminated with a special card. The default processing instruction can be permanently changed by modifying the defining computer code so that change cards would not be needed for often-used sequences.

The last (optional) section in a dataset is the data section. Some operations require the user to supply information, which is done in the data section. The individual operations determine the format of these cards and their order in the deck is determined by the order in which the operations are called. The end of the dataset is signalled to the program by another special card.
Often, one wants to design a specific processing procedure, then apply it to a series of images. The present syntax does not allow the construction of "DO loops" as in a Fortran program but it does allow the display and instruction sections of any dataset to be reused by the following dataset. Thus, one custom dataset can be used for any number of following datasets without excessive retyping.

Since the PLTPROC program developed slowly without a master plan to guide its growth, there is obviously room for improvement. A rewrite of the program could yield a smaller, more modular program which would be more amenable to future revisions. The proliferation of labelled common blocks for inter-routine communication could be cut back to a few, larger common blocks. Consistency of nomenclature could be improved. Many of the subroutines were written to be of general use and could be replaced by slimmed down, more specific versions if computer space were a problem. If a different computer environment were available, the most important change that should be made is to implement a truly interactive image processing system. There are several areas where the ability to interact with the program (such as the choosing of a reciprocal lattice or selection of a sub-area with which to work) would speed the work along.

M. 3-D Helical Reconstruction Programs from the MRC

The reconstruction program for helical objects that has been implemented on the LBL computer system has at its core
the Medical Research Council routines kindly supplied by Dr. Linda Amos. The output of a PLTPROC run can be processed by routines I have written in order to prepare the data for use by the IBM-based MRC routines. Changes have had to be made to some of the input and output instructions to make the final code compatible with the CDC way of operation. For the most part, these changes have no effect on the user. I have added code that saves the reconstructions, with identification, on tape for later use by powerful graphic routines.

In order to accomplish the conversion and have confidence that the correct changes were made, it was necessary to analyze the workings of the MRC code step-by-step. When the function of a section of code was deciphered, the need for changes (if any) became apparent and they were made. This "nuts and bolts" dismantling of the programs made me very confident that the joining of the two processing systems (and two computer systems) would not introduce any errors that might be impossible to detect in a black-box, trial-and-error approach to changing the programs. The analysis of the MRC code revealed several examples of incorrect coding (which were corrected) and some "tricks" that had little theoretical justification.

Not all of the programs available at the MRC were sent, and, of those sent, only those dealing with helical reconstruction from a single view using a single specimen have
been converted. Programs to align and combine data from more than one specimen are available, but these still are limited to the cylindrically symmetric portion of the Fourier transform. Rotational Fourier transforms and rotational filtering programs have also been sent but not converted.

Before delving into the implementation of the MRC programs, one aspect of the differences between the IBM-based MRC routines and the CDC-based programs I have written must be highlighted. There are two methods for storing two-dimensional (or higher) arrays in a linear form, as must be done to place them on linear recording media such as magnetic tape or disk. These schemes are termed row-major and column-major order.

In row-major order, used by IBM-compatible software, the two-dimensional array is split into rows (see Figure 31), each row being output (input) as a unit before the next row is started. In FORTRAN notation, arrays are stored (read) as \( (A(I,J), J=1,N), I=1,M \). For arrays of higher dimension, the process is similar and can be succinctly described by saying the outermost index varies the fastest.

Column-major order, used by CDC-compatible software, breaks two-dimensional arrays into columns which are written (read) consecutively. In FORTRAN, this means \( (A(I,J), I=1,M), J=1,N \). The innermost index varies the fastest.
Figure 31

Comparison of linear array storage of two-dimensional arrays using row-major and column-major ordering
Row-Major Order

Two-dimensional Array

Column-Major Order

Two-dimensional Array
The starting point for the MRC programs I converted is a half-plane complex DFT. A patching program (FTHDFT) was written to convert the full-plane DFT's produced by PLTPROC to the needed form. The EM4OUT3 routine produces log₂ maps of the half-plane diffraction amplitudes and encoded phase maps. These maps allow the layerlines associated with helical diffraction to be located. The maps are also used to delimit regions of each layerline which will be useful in helical reconstruction. The EM4OUT3 routine includes an option to use a different phase origin than the default of the meridian when producing the phase maps. This is useful when monitoring the results of a phase origin shift designed to place the phase origin on the axis of the helix. The EMSRCH program, using data extracted from the layerlines by the user, attempts to find this phase origin, a necessary condition for using the diffraction data in the helical reconstruction program. Since only a small amount of data, that supplied by the user, is used by the phase search program, the ability to recalculate all the phases in the half-plane is an excellent check on the results of this step.

The EM4HLX and EM4NTR routines are used to extract layerline information in a form usable by the 3-D Fourier-Bessel inversion program. EM4HLX has provisions for the geometrical correction of tilted and skewed layerlines. Figure 32 demonstrates some of the parameters used by this program. The code used to correct for tilt had to be
Figure 32

Geometry used for tilt and skew correction by the MRC routine EM4HLX
changed since it seemed to employ two definitions for the angle of tilt, one using the correct value for the number of degrees per radian, the other using a value which seems to suffer from a typing error. This undocumented "feature" could have been a mistake or an attempt to avoid trigonometry problems when the angle of tilt reached 90°. Since it lead to incorrect results at all angles, albeit not severe, I removed the "incorrect" code.

Negatively stained specimens of helices often give diffraction which displays a shearing effect (see Figure 32). The centers of the layerlines lie on a line which is not perpendicular to the layerlines. The MRC routines compensate for the shear by moving the origin of each layerline. Rather than defining the origin of the layerline by the shear angle, then interpolating data values from this origin, the MRC chose to interpolate the layerline assuming that there was no shear, then shifting the interpolated values an integral number of array elements left or right, establishing an approximation to the shear-shifted origin after the fact. The fact that such a scheme could leave the layerlines off-center by up to half a sampling distance apparently was of no concern. The purely geometric correction for shear does not spring from a theoretical basis. What causes shear and how does it affect amplitudes and phases of the diffraction pattern? The correction used may assign the correct spatial frequency to the components but it does nothing to correct for any other problems that may
have affected the value of the component. Presumably, shear is caused by deformation of the helical lattice during drying in negative stain but such changes would seem to do more than merely shift the origin of the layerlines. The unmentioned assumption is that the effect of shear on the low resolution portion of the layerline is purely geometric. This question should be investigated before high resolution helical reconstructions are attempted with sheared transforms.

The EM4NTR routine is used to interpolate the calculated DFT to produce regularly spaced samples of the cylindrically symmetric portion of the layerlines for use by the reconstruction program. The MRC version of this program was heavily dependent on the MRC-IBM method of data storage, a random access, disk-file structure. The entire program was rewritten to reflect our way of storing transforms.

The interpolation used by EM4NTR contains some fine points that are not mentioned in the literature. Since helical specimens are of limited spatial extent in directions perpendicular to the helical axis and are usually apertured close to their natural boundaries, the calculated DFT is the convolution of the helix transform F(s) with the aperture transform A(s). Since the origin of the DFT is usually not the center of symmetry of the aperture, the aperture transform is complex. In the bilinear interpolation scheme used in the program, the four known values surrounding an
unknown value are used for the calculation of that value. If these known points represent a convolution and are used without correction, the interpolation can not be carried out in a straightforward manner. As detailed earlier, the convolution of $F(s)$ with a complex $A(s)$ yields output values whose real and imaginary parts are each functions of both the real and imaginary parts of $F(s)$. We want to interpolate the real and imaginary parts of $F(s)$ separately. To accomplish this, the MRC program moves the phase origin of the four transform values to the center of the aperture. The now centrosymmetric aperture has a real transform and the convolution of $F(s)$ with this real $A(s)$ will not mix the real and imaginary parts of $F$. Simple bilinear interpolation can be used on real and imaginary parts separately. To complete the interpolation, the phase of the interpolated point is changed to move the phase origin back to the original transform origin. Since the role of EM4NTR is to produce helical diffraction output for reconstruction, the phase origin is moved once again, this time to the location determined by EMSRCH to be on the axis of the helix. The general lesson taught by this routine is that bilinear interpolation should be applied only to complex transforms whose phase origin is at the center of the convolved aperture function. This location is at the center of the array for objects without an explicit aperture.

Another view of this problem follows. We wish values of the convolved function $C(s) = F(s) \cdot A(s)$ at spatial
frequencies other than those sampled by the DFT. If F(s) is a slow-varying function, then C(s) also will be when A(s) is a real function. When the aperture is not centered, especially when the origin is at the corner of the aperture, C(s) will be a rapidly oscillating complex function. For the origin in the corner, the phase will shift by 180° from sampled point to sampled point. Bilinear interpolation is not appropriate for such a function.

The output from the extraction routine is input for the actual reconstruction program EMHLFOR. Rather than reconstruct, one time, the three-dimensional object on a 3-D cartesian grid over a predetermined number of repeats of the structure, then prepare specialized views of the object by bilinear interpolation, the MRC programs chose to use a different method. The specialized views that are desired are used to generate locations at which density values are needed. The reconstruction program is then invoked to calculate the object density at those locations. Another choice of desired sections elicits another round of reconstruction. Given the large number of views needed to investigate an unknown or poorly known structure, it seems inefficient to recalculate a reconstruction so many times, especially when the reconstruction is accomplished through a relatively slow and expensive algorithm (Fourier-Bessel inversion). It would appear that calculating a finely sampled cartesian net, once, and then using bilinear interpolation to generate specific sections would be more cost
effective in the generation of many sections.

Modifications have been made to the EMHLFOR program to generalize its graphic output. Formerly, two-dimensional maps of density values were produced on printer paper, density values being scaled to the 0-99 range. On some types of sections, sampling in one direction is calculated from that in the other so that sections displayed on computer paper would not show geometrical distortion. This correction was preset for printers with 10 characters/inch in the horizontal and 6 characters/inch in the vertical direction. To allow undistorted displays on other devices (such as printers with 10 characters/inch horizontally and 8/inch vertically or CRT devices with equal spacings horizontally and vertically), the routines were changed so that a single variable could be used to signal the ratio between horizontal and vertical pixel spacing for the output device to be used.

Code was added to allow the disk storage of all computed sections together with the basis functions used to produce the sections, \( g_{n,l}(r) \) (see III-L above). The sections were stored to allow a more sophisticated display program (EM3DISP) written by me to produce contour and grey-scale plots of the sections. The \( g_{n,l} \) values can be used to produce new sections at will without rerunning the entire calculation.

The EMHLFOR program was designed to produce five types
of sections. Figure 33 illustrates the section types which can be calculated together with some of the variables used to describe the sections. The choice of horizontal sections (Z-sections) is accompanied by the production of a single side projection perpendicular to the helix axis, parallel to the y-axis. This side projection only includes the first forty Z-sections produced because of an array size limitation in the program. Cylindrical sections are accompanied by an angular projection or cylindrical average wherein the density is projected over $\phi$, the azimuthal angle. Vertical sections through the helix axis and vertical sections parallel to the x-axis are straightforward and without accompanying "extra sections". The production of the z-axis projection is subject to some manipulation. Since the average value (DC offset) of the reconstruction may be meaningless because it depends on equatorial data, which is unreliable and often not used, negative reconstruction values should not be allowed in a projection where they might cancel positive values. While producing the z-axis projection it is also possible to suppress all negative values and it is possible to use only negative values. Code should be added to allow the floating of the reconstruction so that both positive and negative values could contribute (but not cancel) to a projection. One merely needs to add a constant to all density values equal to the largest negative density value found in the data to be projected.

The EM3DISP program produces a contour plot and three
Figure 33

Section types available from the MRC helical reconstruction program

The figures on the left show a perspective view of a helical object and the geometry of the sections. The figures in the middle and on the right show the arrays produced and their relationship to the perspective views.

A) Horizontal z-sections and side projection (MRC designation - IHFH)

1, 2, or 4 quadrants can be reconstructed at each z value. Array sizes limit the z-sections to 75 X 81. The sampling distance in the y-direction is tied to the sampling in the x-direction. The side projection is limited to data from the first 40 z-sections.

B) Cylindrical sections and cylindrical average (IHFC)

The cylindrical sections are limited to 100 X 181 by array sizes. A maximum of 51 sections can be produced. The cylindrical average is taken over $\phi$. 
Figure 33

C) Vertical-through-the-axis sections (IHFV)

The section size is limited to 100 X 51 by array sizes

D) Vertical-parallel sections (IHFP)

The parallel sections can be oriented by \( \phi_{\text{min}} \). The sections are limited to 100 X 101 by array sizes. The sampling distance in y is tied to that in z.

E) Z-projection (IHFZP)

The object is reconstructed on horizontal z-sections spaced delzed apart, then projected parallel to the z-axis. Rotation of the projection can be achieved by varying \( \phi_{\text{min}} \).
Vertical-thru-Axis Sections

Vertical-Parallel Sections

Project

Z-Projection

\phi = \phi_{\text{min}} + n \cdot \text{Delphi Section}

X = X_{\text{min}} + n \cdot \text{Delx Section}
greyscale pictures for each section saved on the disk or tape. One of the greyscale pictures displays positive values only, another displays negative values only (in reverse contrast), and the third scales the display from array minimum to array maximum (floating zero display). Sample output is shown in Figure 34. Using a graphics terminal capable of producing the plots, it is possible to run only the contour plot portion of the program. The values of the contour levels can be defined by the user and the results seen on the screen, allowing interactive selection of contour levels. This is useful when trying to discriminate the reconstructed object from spurious noise features. Since the equatorial diffraction is not often used in the reconstruction process, the cutoff value to use in differentiating the object from the background is best found by trial and error.

N. Radial Projection Program

The MRC-based helical reconstruction programs did not include one type of sectional view of the object that is important in studying tubular structures, especially those derived from sheets. A radial projection in which the density is integrated along lines of constant \( z \) and \( \phi \) is helpful in determining what a structure would look like if it were unrolled (intersected by a half-plane terminating on the helical axis and flattened out perpendicular to this plane). The radial projection should yield a structure
Figure 34

Examples of EM3DISP output

A) Greyscale picture of a section showing the positive values only

B) Greyscale picture in reversed contrast of the negative values only

C) Greyscale picture scaled from the negatives to the positives

D) Contour plot of the section
similar to the sheet structure for tubular structures with sheet precursors, since only small alterations of the bonds are presumably involved.

A program has been written to reconstruct a helical object on a radial net, then project it subject to the same type of manipulations available for the z-axis projection (see above). The $g_{n,1}$ functions, stored from a previous reconstruction run using EMHLFOR, are used to produce the required density values. Figure 35 shows the geometry of this section.

O. Other Programs

The programs mentioned above are not the only ones which were written during the course of this thesis. Many programs and subroutines have been written to perform a variety of useful tasks, many having to do with image manipulation or generation. Rather than waste space and time detailing them, a brief description of all the routines available (including those mentioned above) in connection with image processing are listed in Appendix A in alphabetical order.

P. Programs That Are Needed

Apart from fancy frills, the programs that would be extremely useful are few in number, most functions in image processing already being available. There is a need to align and combine the diffraction patterns of small,
Figure 35

Geometry of the radial projection

A) Perspective view, showing the location of data points used in the radial projection

B) The radial projection array is generated by projecting along lines of constant $z$ and $\phi$
Radial Projection

The Object

\( \Delta Z \)

\( R_{\text{max}} \)

\( R_{\text{min}} \)

\( Z_{\text{max}} \)

\( Z_{\text{min}} \)

\( \Delta R \)

\( \phi_{\text{max}} \)

\( \phi_{\text{min}} \)

The Array

XBL793-3305
statistically-poorly-defined crystalline patches. Techniques for achieving this in principle exist (Saxton and Frank, 1977) but have not yet been implemented at LBL. Programs correcting for spiral, pincushion, and barrel distortions in the image are needed before higher resolutions can be reached. The rotational transformation and filtration routines from the MRC should be activated, as should the code to combine coherently the transforms from several helical specimens. Programs to use several views from a helical object in order to separate out different Bessel orders on a layerline are needed to push helical diffraction to higher resolution. Related programs to separate the Bessel functions found on the layerlines of cylindrical objects could extend reconstruction techniques to another class of objects. In planar 3-D processing, we need to implement code to correct for the transfer function characteristics found in tilted views. Programs are also needed for the handling and indexing of many tilted datasets. Programs to derive electron diffraction amplitudes from computer scanned ED patterns would speed up the reconstruction work and improve the accuracy of the results.

These are some of the basic additional programs needed to produce a full-potential image processing system. As has been stated, improvements in the image processing software would be helpful as would more hardware to improve the interactive aspects of the programs. The shear mass of numerical results demands that the computer free the
investigator from as many "number burdens" as possible. Even so simple a task as keeping track of results gets to be a chore when one is dealing with multiple images at different defocus values taken at different tilts, etc. No system should ever be considered complete, but neither should it be considered useless because of this incompleteness.

Q. Conclusion

The present collection of image processing programs at LBL is versatile and useful in a wide variety of image processing applications. The generality of the programs allows them to be used in applications other than image processing. Benefiting from hindsight, the organization of the programs could be greatly improved but in their present state they are nonetheless expandable and understandable. All the programs are stored in computer-readable format on on-line storage devices, which allows changes and exchanges to be made rapidly. Standard computer card output can be produced, of course, so that the programs are available for shipment to others. Since the LBL computer system is on two computer networks (ARPANET and TYMNET), programs could be run at LBL from terminals located around the country. For the short term, I think any programming effort should go into developing the necessary programs mentioned above. Any extra time could go into condensing, standardizing and more fully documenting the existing code.
V. Cylindrical Diffraction Theory

A. Introduction

In order to understand more fully the diffraction patterns arising from the rolled-up sheets of hexagonally packed subunits found in *S. serpens*, a model structure was used to develop analytical relationships between object structure and the diffraction pattern. Although so-called "zero-pitch" (Kiselev and Klug, 1969) helices can be understood in terms of regular helical diffraction theory, a more detailed and specific treatment was used so that there would be an explicit demonstration of the properties of the diffraction patterns of such objects. Helical theory tells us that all layerlines (layerplanes) contain mixtures of Bessel function terms. By having the analytic expression for this diffraction, it was hoped that a method could be developed to pull out a low-resolution reconstruction from a single view of a cylindrical object. Unfortunately, a single view can not be used but the expressions derived for the general cylindrical object can be used as the basis for a method that uses multiple views to achieve a reconstruction.

B. The Delta Function Model

A hexagonal net can be "rolled-up" to form a cylindrical tube, $T(r, \theta, z)$, as shown in Figure 36. In order to form a cylindrical tube with no discontinuities, one has to use a sheet which is an integral number of repeats wide. This
Figure 36

The transition from a hexagonal net to a tubular structure

A) A cylindrical tube of repeat distance $c$, composed of delta functions

B) The hexagonal net of delta functions which results from unrolling object A)
A tubular structure can be decomposed into a stack of units, each unit having two annuli, one rotationally displaced from the other but otherwise identical, as shown in Figure 37. There are $N$ delta functions on each annulus, spaced $\theta_{\theta} = \frac{2\pi}{N}$ apart. The second annulus is rotated by half this angle with respect to the first annulus, and is translated half the repeat distance, $\frac{c}{2}$, in the $+z$ direction. I call this unit $U(r,\theta,z)$. It is useful to describe this unit as having $N$-fold rotational symmetry. Figure 37 also shows the variables which define the orientation of the unit with respect to a fixed cartesian coordinate system.

The complete tube can be generated by convolving the basic unit of length $c$, $U(r,\theta,z)$, with an infinite set of delta functions on the $z$-axis, spaced $c$ apart.

$$T(r,\theta,z) = U(r,\theta,z) * \sum_{l=-\infty}^{\infty} \delta(z-1c)\delta(r)\delta(\theta) \quad (V-1)$$

Expressing $U(r,\theta,z)$ mathematically,

$$U(r,\theta,z) = \delta(r-a)\delta(z-z_{p}) \sum_{n=1}^{N} \delta(\theta-n\theta - \theta_{d})$$

$$+ \delta(r-a)\delta(z-\frac{c}{2}z_{p}) \sum_{n=1}^{N} \delta(\theta - n\theta - \frac{\theta_{d}}{2} - \theta_{d}) \quad (V-2)$$

The $z_{p}$ and $\theta_{d}$ terms are retained in the expression above to maintain generality, which will be of use when we consider the transform of a real object.
Figure 37

A single unit of the cylindrical tube

A) One repeat of the tubular structure

B) Variables used to describe the general orientation of this unit
$U(r, \theta, z)$

N 8-functions around each ring

Bottom ring of $U(r, \theta, z)$

Orientation Variables
C. The Fourier Transform of the Model

Since we are interested in the Fourier transform of the tube, we have:

\[ F[ T(r, \theta, z)] = F[U(r, \theta, z)] = F[\sum \delta(z - l\phi) \delta(r) \delta(\theta)] \]  \hspace{1cm} (V-3)

\[ = G(R, \phi, S_z) \cdot H(R, \phi, S_z) \]

The Fourier transform of the delta functions on the z-axis consists of a set of parallel planes, perpendicular to the z-axis and spaced \( \frac{1}{c} \) apart.

\[ H(R, \phi, S_z) = \sum_{l=-\infty}^{\infty} \delta(S_z - \frac{1}{c}) 1(R) 1(\phi) \]  \hspace{1cm} (V-4)

The Fourier transform in cylindrical coordinates of the function \( U \) is:

\[ F[U(r, \theta, z)] = G(R, \phi, S_z) = \int_{\mathbb{R}^3} U(r, \theta, z) e^{-2\pi i R \cos(\theta - \phi) + z S_z} \, r \, dr \, d\theta \, dz \]  \hspace{1cm} (V-5)

Treating the two summations in the definition of \( U(r, \theta, z) \) separately and making use of the properties of the delta function, this transform becomes:

\[ G(R, \phi, S_z) = \sum_{n=1}^{N} e^{i 2\pi \frac{S_z}{2} n} \frac{1}{\sqrt{n \pi}} e^{i 2\pi a \cos(n \theta + \frac{\theta_0 + \phi - \phi}{2})} \]  \hspace{1cm} (V-6)

Using the identity \( \cos x = \sin \left( \frac{\pi}{2} - x \right) \), we get:


\[
G(ae^{i\frac{\pi}{2}}) \sum_{k=-\infty}^{\infty} J_k(2\pi Ra) \left(1+e^{i\frac{\pi}{2}}\right).
\]

Substituting for \( \theta \) in the last factor, it becomes

\[
N \sum_{n=1}^{N-1} e^{-i kn \frac{2\pi}{N}}. \quad \text{This sum has the following useful property:}
\]

\[
N \sum_{n=1}^{N} e^{-i kn \frac{2\pi}{N}} = 0 \quad \text{if } k \neq mN
\]

\[
= N \quad \text{if } k = mN
\]

This results in:

**Selection Rule #1:** \( k = mN \)

Thus, the orders of the Bessel functions comprising \( G(R,\Phi, S_z) \) are restricted to being multiples of the cylindrical symmetry of the unit.

Substituting \( k = mN \) and \( \theta = \frac{2\pi}{N} \) in Equation V-7, we have finally:

\[
G(R,\Phi, S_z) = a N e^{i2\pi S_z \Phi} \sum_{m=-\infty}^{\infty} \frac{\sin(\frac{m}{2} - \frac{\pi}{2}) + \Phi}{(1+e^{i\pi/2})} J_m(2\pi Ra) e^{i\pi(cS_z \Pi - m)} (1+e^{-i\pi/2})
\]

What we want is the transform of \( T(r, \theta, z) \) which, from Equation V-2, is the product of \( G(R,\Phi, S_z) \) and \( H(R,\Phi, S_z) \). Since \( H(R,\Phi, S_z) \) is only non-zero on planes with \( S_z = \frac{1}{c} \), \( l \) an integer, the product becomes:
\[ F(T(r, \theta, z)) = \sum_{l=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} a_N e^{\frac{i 2\pi l z}{c}} J_m(2\pi R_e) e^{imN(\frac{\pi}{Z} \theta - \phi)} (1 + e^{i\pi(l-m)}) \]  

The transform is restricted to layerplanes with \( S_z = \frac{1}{c} \).

Examining the \( 1 + e^{i\pi(l-m)} \) factor, we get:

\[
1 + e^{i\pi(l-m)} = 0 \quad \text{if } l+m \text{ odd} \\
1 + e^{i\pi(l-m)} = 2 \quad \text{if } l+m \text{ even}
\]

From this we derive:

**Selection Rule #2**: \( l+m \) must be even

Thus on even numbered layerplanes, the multiplicative factor, \( m \), used to determine the Bessel orders allowed (Selection Rule #1) is even; on odd planes, it is odd.

Our final result for the tube transform, in a form which explicitly shows both positive and negative Bessel orders is:

\[
F(T) = 2aN \sum_{l=-\infty}^{\infty} e^{\frac{i 2\pi l z}{c}} \sum_{m=-\infty}^{\infty} J_m(2\pi R_e) e^{imN(\frac{\pi}{Z} \theta + \phi)} \quad \text{if } l+m \text{ even} \quad (V-12)
\]

To form a better idea of what one gets when calculating the Fourier transform of a cylindrical specimen, let us make use of a relationship between positive and negative orders of Bessel functions. Substituting \( J_{-p} = (-1)^{p} J_{p} \) into Equation V-12, we get:
D. Consequences of the Model

The two equations, $V_{-12}$ and $V_{-13}$, lead to several conclusions which are summarized here.

1. Only Bessel functions which are multiples of the rotation symmetry of the tube are allowed; even multiples on even-numbered layerplanes and odd multiples on odd-numbered layerplanes.

2. Positive and negative Bessel orders are allowed on the same layerplane for all layerplanes.

3. The three-dimensional Fourier transform has no region of cylindrical symmetry (except for the on-axis term $J_0$), the amplitude being modulated by $\cos mN(\tilde{\phi} - \theta_d)$.

E. Properties of the Diffraction Pattern as a Function of $N$, the Rotational Symmetry

It is possible to group all values of rotational symmetry into one of four classes, classes which have noticeable differences in their diffraction patterns. By working out the phase and amplitude relationships for these classes, it might be possible to deduce the rotational symmetry of a
cylindrical tube from a single diffraction pattern, given some knowledge of the approximate dimensions of the tube and its component building blocks.

The four cases for N that are considered are:

1. N even \( N/2 \) even
2. N even \( N/2 \) odd
3. N odd \( N \text{ mod}(u_l) 4=1 \) ((N-1)/2 even)
4. N odd \( N \text{ mod} 4=3 \) ((N-1)/2 odd)

For these four cases, we will look at the phases on the layerlines and values of the transform at \( \omega^1 \) and \( \omega^1 + \omega \) (on opposite sides of the meridian). For simplicity, we will place the phase origin in the middle of a subunit on the bottom "ring" of subunits. This origin leads to \( z_e = 0 \) and \( \Theta_d = 0 \).

1. Case 1 - N even and \( N/2 \) even

We will let \( N=4k \). The phase term from Equation V-13, \( \text{im} \frac{N}{2} e^{-i\omega} \), becomes 1 for all \( m \) and \( k \). Therefore, all the layerlines are real and have the same sign. The amplitude modulation term from Equation V-13 is \( \cos N(\phi - \Theta_d) \). This term is the same for \( \phi' = \phi \) or \( \phi' = \phi + \pi \). Therefore, the layerlines are symmetrical with respect to the meridian.

2. Case 2 - N even and \( N/2 \) odd

In this case, we let \( N=2(2k+1) \). The phase term alternates between -1 for \( m=\text{odd} \) and +1 for \( m=\text{even} \). Through
selection rule #2, we can say that even-numbered layerlines have zero phase, odd layerlines have \( \bar{\phi} \) phase. The amplitude modulation term is the same for \( \phi \) or \( \phi + \bar{\phi} \), so the layerlines are symmetric in the meridian, again.

3. Case 3 - \( N \) odd and \( (N-1)/2 \) even

Here we can express \( N \) as \( N=4k+1 \). The phase term \( \text{im} \frac{\bar{\phi}}{2} \) reduces to \( e^{i \frac{\pi}{2}} \). For even-numbered layerlines, \( m=0,2,4,\ldots,2n \) so that the phase alternates \( 0, \bar{\phi}, 0, \phi \) for each successive Bessel order (all real). Odd-numbered layerlines have Bessel orders whose phase alternates \( \frac{\pi}{2}, \frac{3\pi}{2} \), etc. (all phases are pure imaginary). The amplitude modulation term yields a +1 for even \( m \), a -1 for odd \( m \). Hence, the phases are symmetric in the meridian for even layerlines and differ by \( \bar{\phi} \) for odd layerlines.

4. Case 4 - \( N \) odd and \( (N-1)/2 \) odd

Let \( N=4k+3 \). The phase term becomes \( e^{i \frac{3\pi}{2}} \) which translates to the even layerlines having Bessel orders whose phase alternates \( 0, \bar{\phi}, 0, \phi \) (all real). Odd-numbered layerlines are pure imaginary, phases alternating \( \frac{3\pi}{2}, \frac{\pi}{2}, \frac{3\pi}{2} \). The results for the modulation factor are the same as in case 3: even-numbered layerlines are symmetric about the meridian, odd layerlines differ by \( \bar{\phi} \) on the two sides of the meridian.
What is the influence of allowing $\theta_q$ and $z_B$ to be non-zero? The magnitude of the modulation factor changes but its behavior on either side of the meridian stays the same. The $z_B$ term does introduce another phase term, one which is constant along each layerline. This term can be removed, effectively moving the phase origin back onto the first annulus.

The discussion has considered a rather artificial object, the delta function cylindrical net. A real object can be described in terms of this net and nets derived from it. If we place a spherically symmetric atom at each net point, we can construct an object from a sum of rotated, elevated and (de)magnified nets (vary $\theta_q$, $z_B$, and $a$), each containing one atom type.

$$\text{Object} = \sum_i f_i T(r_i, \theta_i, z_i)$$  \hspace{1cm} (V-14)

What can we say about the transform of this real object based on our study of the delta function model? If we combine equations V-13 and V-14 in a form which is useful for looking at the amplitude and phase along each layerline, we get:
$$F(\text{object}) = \sum_{l=1}^{\infty} \sum_{i} \sum_{m} \frac{i2\pi}{c} z^m_{\cdot i} + (V-15)$$

$$\sum_{m} \sum_{i} \frac{4Nf_i a_i \cos N(\phi-\Theta_q)}{2} J_{\frac{mN}{2}}(2\pi R_{\cdot i}) e^{imN\pi}$$

$$m+l=\text{even} \quad J_0 \text{ term on } l=\text{even only}$$

In comparison with the delta function net, we can see the following differences: 1) the $f_i$'s weight the contributions from various atom types; 2) the $a_i$'s also weight the contribution from each atom and cause a scale change in the Bessel function (this can be visualized as a stretching or contracting of the Bessel functions); 3) the $\Theta_q$'s change the cosine modulation term for a given $\phi$; and 4) the $z^m_{\cdot i}$'s affect the complex phase of each Bessel order term. The first three factors can certainly produce amplitudes which are not interpretable in a straightforward manner but this is typical of any diffraction pattern. The factors $a_i$, $\Theta_q$, and $f_i$ will affect the phase only through a change of sign. The conclusions reached in the single atom-type cases 1-4 apply with one modification. No prediction can be made about the sign of any term, so $\phi$ and $\pi$ are equally probable, $\frac{\pi}{2}$ and $\frac{3\pi}{2}$ likewise. In this case, the only discrimination possible is between even and odd orders of $N$. The $z^m_{\cdot i}$ factor, however, does influence the phase in a manner totally dependent on the structure of the tube subunit. When all four factors are considered together, the resulting
complex transform has amplitudes and phases which depend on the subunit structure and are not constrained by the symmetry of the underlying cylindrical nets. The lessons learned in cases 1-4 above are, in general, useless for the real object case, except for the behavior of the phases across the meridian, which remains the same. This is the same behavior that is seen in true helices - even and odd Bessel functions have different symmetry with respect to the meridian.

Even though the general tube subunit leads to little information about the transform, a subunit with symmetry can affect the transform. We will look at four cases of subunit symmetry which are (or may be) of interest in biological specimens.

The first case is a subunit whose radial projection has a mirror plane perpendicular to the cylinder axis. Further, the tubular form is assumed to be "thin"-walled, wherein the thickness of the tube wall is (much) smaller than the tube radius. The second case is a subunit with a two-fold symmetry axis perpendicular to the cylinder axis. The third is a subunit whose radial projection is 2-fold symmetric, again assuming thin walls. Lastly, we will look at a subunit from a thin-walled tube whose radial projection has a mirror plane parallel to the cylinder axis. These cases are not the only possible cases of interest but they are related to the cylindrical tubes of *Spirillum serpens*. 
The basic symmetry relationship that characterizes the first case is a pairing of atoms, one at \( \Theta_{d_i}, z_{\beta_i} \), another at \( \Theta_{d_i}, -z_{\beta_i} \). The thin wall approximation leads to \( a_i \sim a_i' \). We can see from equation V-15 that the latter approximation, especially for the lower order Bessel functions, leads to approximate equality for the Bessel function factors for the two "paired" atoms. Simplifying Equation V-15 under these approximations:

\[
F(T) = \sum_{l} \sum_{i \neq i'} 4Nf_i a_i \cos\left(2\pi \frac{1}{c} z_{\beta_i}\right) J_0(2\pi R_{a_i}) \cos(\Theta_{d_i}) e^{\text{imN}_l \frac{\pi}{2}} \\
+ \sum_{m} 2 \cos mN(\Theta_{d_i}) J_m(2\pi R_{a_i}) e^{\text{imN}_l \frac{\pi}{2}}
\]

where the sum is over the asymmetric half of the subunit.

Although the amplitudes will be structure dependent, the phases are not. For \( N \) even, the phase of any term will be 0 or \( \pi \) while for \( N \) odd, the phase will be \( \frac{\pi}{2} \) or \( \frac{3\pi}{2} \).

The second case requires that \( z_{\beta_i} = -z_{\beta_i}, \Theta_{d_i} = -\Theta_{d_i} \), and \( a_i = a_i' \), for symmetry related atoms. For this type of subunit, Equation V-15 becomes:

\[
F(\text{object}) = \sum_{l} \sum_{i \neq i'} 4Nf_i a_i \cos\left(2\pi \frac{1}{c} z_{\beta_i}\right) J_0(2\pi R_{a_i}) \cos(\Theta_{d_i}) e^{\text{imN}_l \frac{\pi}{2}} \\
+ \sum_{m} 8Nf_i a_i J_m(2\pi R_{a_i}) e^{\text{imN}_l \frac{\pi}{2}} \left(\cos mN_0 \cos mN_0 \Theta_{d_i} \cos 2\pi \frac{1}{c} z_{\beta_i} + \sin mN_0 \sin mN_0 \sin 2\pi \frac{1}{c} z_{\beta_i}\right)
\]

Again, the amplitude is structure dependent but the phase follows the same pattern as case 1.
The third case is similar to the second except that $a_i \sim a_i'$. Following the reasoning used in case 1, this thin-walled tube should give similar results to case 2. The only difference will be the expansion/contraction of the Bessel order terms.

The last case, mirror symmetry in the radial projection parallel to the cylinder axis, forces $\theta_{1i} = -\theta_{1i}'$ and $z_{B1} = z_{B1}'$, for pairs of symmetry related atoms. The assumption of a thin-walled tube, lets us state $a_i \sim a_i'$. In this case, Equation V-15 becomes:

$$F(\text{object}) = \sum_{l} \sum_{i \neq i'} 4Nf_i a_i J_0(2\pi Ra_i)e^{im\pi \frac{1}{2}} z_{B_i} e^{imN\pi} i2^{\frac{1}{2}} z_{B_i}$$

where $m \neq 0$.

Once again, the amplitude is structure dependent but this time the phase is too. Evidently, this subunit symmetry does not manifest itself in the cylindrical case.

In summary, the first three cases have an influence in the Fourier transform, the fourth does not. We can not differentiate between the first three cases on the basis of the phases in the transform since all three led to a requirement that even-ordered Bessel functions be pure real and odd-ordered be pure imaginary. In all cases, amplitudes depend on the distribution of atoms in the subunit. Symmetry does
not limit the values the amplitudes may take.

F. 3-D Reconstruction from Cylindrical Objects

The previous development has shown that cylindrical objects possess a very limited "cylindrically" symmetric portion in their Fourier transforms! The simultaneous presence on a layerplane of $J_n$ and $J_{-n}$ where $J_n$ is a Bessel function of order $n$, means that all portions of the Fourier transform will contain contributions from at least two Bessel functions. Thus it will be necessary to obtain more than one projection of the structure. How these projections can be combined to yield a three-dimensional structure will be presented in this section.

There are at least two methods to use when collecting multiply tilted data. One could align the cylinder axis of the specimen with the tilt axis of the tilting stage and take a series of micrographs at known relative tilt. Practical problems of radiation damage and contamination buildup are present, but for the lower resolutions found in negatively stained specimens, it should be possible to obtain useful micrographs (preserving structure in the last micrograph close to that in the first micrograph of a series). The other way to collect data is to take micrographs of different specimens lying on an untilted support grid. There should be some distribution of relative azimuthal rotations for the tubes found in one picture. This method has the advantage of a lower dose to the specimen, no tilt axis to
align, and fewer contamination problems. The drawback is that the relative rotations are unknown as well as the relative translations of the tubes.

What information do we need in order to combine data from these multiple views? In general, three factors can be different for the various micrographs. The origin of the coordinate system along the tube axis may differ (therefore we need to know $z_{\text{shift}}$), the azimuthal origin may differ (need to know $\theta_{\text{shift}}$), and the magnification may change (need to know the magnification ratio). Related to the latter point is the idea of variable radial scaling (Amos and Klug, 1975), produced by an assumed uniform radial shrinking or expansion of the object. Undoubtedly causing structural changes and rearrangement of mass, this radial change can, nonetheless, be treated as an unencumbered change in scale, albeit at an uncertain risk.

An analysis of the calculated Fourier transforms of the various tubes can be used to determine some of these scaling values. Let us refer back to our analytical expression V-15 to see how each of the unknown relationships between tubes evidences itself in the Fourier transform. Assume we have the following correspondences between the coordinate systems used to describe a reference tube (unprimed) and another tube (primed):

$$r' = r/M$$

$$z' = z - z_{\text{shift}}$$
\[ \theta' = \theta_{\text{shift}} \]

The least encumbered term in V-15 is the \( J_0 \) term found on even numbered layerlines. If we substitute \( z' \) in the \( J_0 \) term, we get for the meridional reflection on any even-numbered layerline \( l \):

\[
F(z') = e^{-i2\pi \frac{1}{c} z_{\text{shift}}} \sum_{i} 2N_{i} a_{i} J_{i} (2\pi R_{i} a_{i}) e^{-i2\pi \frac{1}{c} z_{\text{shift}}} F(z)
\]

In the limit that \( M=1 \) (no radial scale factor). This is a good result because the \( \theta_{\text{shift}} \) factor does not influence the result. By comparing phases for the meridional reflections, one can deduce \( z_{\text{shift}} \). The influence of \( M\neq1 \) is also clear. In this case, \( a_i' = \frac{a_i}{M} \). From this we get:

\[
F(z', a') = e^{-i2\pi \frac{1}{c} z_{\text{shift}}} \frac{1}{M} F(z'_{\text{R/M}})
\]

That is to say, there is a change in the peak heights and a compression/expansion of the \( R \)-axis in reciprocal space. More simply put, we need to look in different places in Fourier space to get comparable values. Again, the cylindrical symmetry of the modulus of the meridional reflections allows one to deduce the relative radial scaling factors quite easily. As will be pointed out later, it is possible to deduce this scaling factor from reflections away from the meridian although one has to be more careful.

If the experiment has been done with a tilting stage,
then $\theta_{\text{shift}}$ is known. If not, can $\theta_{\text{shift}}$ be determined from the Fourier transforms of the two tubes? No. As stated above, the $J_0$ term does not depend on $\theta_{\text{shift}}$. If we look at the layerplanes in the near axial region where only two Bessel functions interfere, $J_{mN}$ and $J_{-mN}$, we see from V-15 that the Fourier transform depends on:

$$F(R,\Phi, \frac{1}{c}) = \sum_i 4N_f a_i J_{mN}(2\pi Ra_i) e^{imN \frac{\pi}{2}}$$

$$= \sum_i A_i(R, \frac{1}{c}) \cos mN(\Phi - \theta_{\text{q}_i})$$

$$\cos mN \sum_i A_i(R, \frac{1}{c}) \cos mN(\Phi - \theta_{\text{q}_i})$$

$$= \cos mN \sum_i A_i(R, \frac{1}{c}) \cos mN\Phi - \sin mN \sum_i A_i(R, \frac{1}{c}) \sin mN\Phi$$

The sums over $i$ are structure dependent and will be unknowns. Our data will consist of a set of amplitudes and phases as a function of $R$ and unknown $\theta_{\text{shift}}$ (which corresponds to an unknown $\Phi$). Unfortunately, there are two more unknowns than knowns in any set of linear equations one can construct for these data points. Thus, the unknown $\theta_{\text{shift}}$ can not be determined from an analysis of the transforms. This circumstance is unlike that found in helices with single Bessel function regions in the layerplanes (see Amos and Klug, 1975). In these cases, the relative orientation can be deduced from the phases in the calculated Fourier transform.

Using known relative rotations and deduced axial translations and radial scaling, it should be possible to
separate the various contributing Bessel functions on the layerplanes by using enough different projections to determine (or better yet, over determine) the resulting system of linear equations patterned after V-19.

So, for the moment, cylindrical reconstruction remains an unattained goal. Although the methods applied to helical specimens can not be directly used, the two problems are similar enough that Fourier-Bessel inversion can be used to reconstruct an object once the interfering Bessel contributions are sorted out in Fourier space. The analysis of cylindrical diffraction developed in this chapter has pointed out the difficulties in performing this separation. The information that can be deduced from the diffraction patterns of single tubes has also been discussed. Further research and programming is needed before cylindrical objects can join helical objects as suitable specimens for 3-D reconstruction using Fourier-Bessel methods.
VI. Experiments

A. Introduction

Experimental work has been undertaken along with the development of the computer software, both to test the usefulness of the programs and to answer some biological questions. In addition to the work that will be described below, computer experiments were done to test each computer procedure for correctness. Since these tests are not of general interest and are merely a standard, albeit individualized, part of good programming practice, they will not be elaborated. The experiments are described in chronological order which also correlates with the increasing capabilities of the developing computer system.

B. Rat Liver Gap Junction

The gap junction found in rat liver cells (Goodenough and Revel, 1970) was the first biological problem to be studied using our fledgling image analysis programs. Negatively stained specimens had been imaged and the existence of a centrally located staining pit had been seen in micrographs of such specimens. The rather variable images displayed by individual subunits made the existence of the staining pit open to question. Since the gap junction specimen is ordered as determined by optical diffraction of the micrographs, we felt that a computer analysis of an image would yield a more reliable estimate of the projected
structure of a single subunit.

The micrograph chosen for analysis (Figure 38) was kindly provided by Drs. B. Jap and T. Forte. The specimen is from a partially purified preparation of gap junction processed according to the method of Goodenough and Stoeckenius (1972). Stained in 1% PTA, the micrograph was taken in an AEI-EM802 electron microscope at 40,000X electron magnification. The plate was scanned with a scanning densitometer constructed by and located in the Remote Sensing Laboratory, Space Sciences Building, UC Berkeley. Unlike the presently used scanner, this machine made it difficult to preview the area to be scanned so scans were made of the general area of interest. Data was sampled on a square grid of sampling distance equal to 37.5 μm, corresponding to an aliasing frequency of \( \frac{1}{18.6} \) Å\(^{-1}\). Scans of 512 x 512 sample points were collected from several areas.

Diffractograms of the image, generated by both a laser and a computer, are shown in Figure 39. The two patterns are similar, attesting to the usefulness of the computer display routines. The scanned areas displaying the best computed diffraction were spatially averaged at low resolution by masking their Fourier transforms with square apertures of width \( \frac{1}{600} \) Å\(^{-1}\). The "before and after" results of this procedure are shown in Figure 40. The location of the subunits is much clearer in the filtered image (which we call an "optical" filtration since the mask sizes are finite
Figure 38

PTA-stained rat liver gap junction
Figure 39

Optical and computed diffractograms of gap junction

A) Optical diffractogram - short exposure

B) A computed diffractogram similar to A) created by windowing the $\log_{10}$ of the power spectrum - larger values

C) Optical diffractogram - long exposure

D) A computed diffractogram similar to C) created by windowing the $\log_{10}$ of the power spectrum - smaller values
Figure 40

Gap junction image, before and after computer "optical" filtration

A) A greyscale computer picture of the original image

B) A computer "optical" filtration, including the neighbors of the zero order term

C) The computed diffraction pattern of A)

D) An "optical" filtration similar to B) but made without the neighbors of the zero order term. Notice the improved visibility of the subunits.
and correspond to the real optical filtration case of finite sized mask holes. Both the orientation of the sampling lattice with respect to the crystal and the sampling intervals were corrected by bilinear interpolation in order to include an integral number of "unit cells" in a new data array. The Fourier transform of this interpolated data is shown in Figure 41. By matching sampled coefficients with the reciprocal lattice, the calculated amplitudes are stronger, easier to find (see), and less contaminated by noise. It could be possible to determine orientation and sampling corrections from a careful examination of the original Fourier transform, but I feel the use of the filtered image lets one avoid using image data that contains polycrystalline material or distorted crystals. The transform contains this type of information in a form not readily recognized by the observer (phase shifts and convolutions).

The amplitudes and phases calculated from the interpolated area are used to reconstruct the average unit cell. Since only a single coefficient is used at each reciprocal lattice location, the resulting inverse Fourier transform can be considered a delta function filtration, in contrast to the previously used "optical" filtration. To distinguish the unique ability of the computer to pass a single term, the delta function filtration is also called a "computer" filtration. For this specimen, no attempt was made to correct for the contrast transfer characteristics and the Fourier data was not averaged or symmetrized. The delta
Figure 41

Power spectrum of an interpolated gap junction image
function filtration of the best preserved area is shown in Figure 42. The presence of a staining pit is confirmed by the computer filtration.

C. T4 Bacteriophage Tail Reconstruction

As mentioned before, Dr. Amos at the MRC Laboratory in Cambridge kindly supplied our group with their IBM-based helical reconstruction programs. It was necessary to undertake the three-dimensional reconstruction of a previously determined helical biological structure in order to test our understanding of the technique and our conversion of the IBM-compatible programs to CDC-compatible form. The most convenient specimen for this was the uncontracted T4 phage tail. First reconstructed by DeRosier and Klug (1968), higher resolution for this specimen was achieved by Amos and Klug (1975) and Smith et al (1976) using multiple views and specimens. Our aim was to use a single view for a low resolution reconstruction and check the results with the known structure. Until the need for higher resolution had been shown by a new specimen, the extra work needed to institute multiple image registration was not felt to be justified.

The phage tails were negatively stained and deposited on thin (40-60Å) carbon film grids. A 1% solution of uranyl acetate was used as the stain. Microscopy was performed on the AEI-EMB02, with no attempt being made to use minimal beam damage techniques. Specimens were imaged at a magnification of 63,000X. The preservation of structure was
Figure 42

Delta function filtration of the gap junction
assessed by optical diffraction and those plates with the clearest helical diffraction were chosen for computer scanning. Criteria for "good" diffraction include mirror symmetry about the meridian (indicating identical imaging and preservation of near side and far side structure [Klug and Berger, 1964]), minimization of shear in the diffraction pattern, qualitatively assessed signal-to-noise ratio of the specimen, and resolution of the ordered diffraction. In order to visualize the optical diffraction pattern of such a small structure it is necessary to use an aperture on the illuminating beam that is only slightly larger than the object. Including too much of the support film will cause a swamping of the relatively weak ordered diffraction by amorphous scattering from the film. Naturally, the very small apertures lead to much longer than normal exposure times for these patterns (partially caused by reciprocity failure of the film). Figure 43 shows the image and optical diffraction pattern from one of the T4 tails used for subsequent processing.

The tails were scanned on a 25 X 25 µm raster using a 25 µm square aperture. 320 X 128 points were sampled. Figure 44 shows a typical scanned area, the vertical scale being compressed in the display by a factor of two because of the limited number of pixels available. That portion of the scan holding the tail is boxed off from the rest and floated in a 512 X 256 data array. In order to choose accurately the location of the masking box, a computer overprint
Figure 43

Image and optical diffraction pattern of a T4 bacteriophage tail

A) Negatively stained T4 virus used for processing

B) Optical diffraction pattern, showing the layerline structure
Figure 44

PLTPROC-produced greyscale picture of a scanned T4 tail.

Vertical scale is compressed by a factor of two.
picture was used in conjunction with a CRT greyscale picture (such as Figure 44). The CRT picture is used as a reference guide when searching the overprint pictures for landmarks. The extremely large magnification achieved by the printer picture (100 vertically by 167 horizontally for this specimen) makes the context of a local scene difficult to discern from a viewing position of an arm's length away or less. An alternate procedure to use if a CRT picture is not available is to change the viewing distance by stepping back from the picture (about 5m in this case), sighting the feature and keeping it in view as you move back in. It pays to keep office furniture out of your path! The labelling of rows and columns on the overprint picture makes choosing the box coordinates easy and the large magnification makes the choice accurate. The exact magnification achieved by the printer picture in our system (which prints 6 characters/inch vertically and 10/inch horizontally) varies with the sampling interval used to collect the data as follows: vertical magnification is \( \frac{2500}{(\text{inter-column sampling interval in } \mu\text{m})} \); horizontal magnification is \( \frac{4167}{(\text{inter-row sampling interval in } \mu\text{m})} \).

The array containing the boxed tail was padded with the average of the perimeter of the tail data. The complex, full 512 x 256 DFT's were performed in core and converted to half-plane form (suppression of all Freidel-related coefficients except for the inclusion of the complete zero-frequency column). Time and money could be saved by
computing only the half-plane DFT. Since the object is real, an \( \frac{N}{2} \) by \( M \) complex DFT can be used to calculate the unique coefficients of an \( N \) by \( M \) real array. Algorithms for this purpose are available and have been implemented in our system. The PLTPROC system, however, is presently oriented toward complex real space arrays and full-plane DFT's. Since PLTPROC was used for boxing, floating, transforming, saving, and displaying the data, the full-plane DFT was calculated, then converted to half-plane data for compatibility with the MRC routines. PLTPROC can be modified without too much difficulty to use real arrays and half-plane data since many of the constituent subroutines were written to use either real or complex data at the user's discretion.

The half-plane DFT's were also displayed using the MRC programs, producing a single digit, base 2 log map of the intensity and a one character map of the phases. Layerlines were located and the helical selection rule for the T4 tail \((l=-2n+7m)\) used to assign Bessel function orders to the layerlines. To avoid low resolution noise, minimum radii were used on each layerline. Data was not used until the first noticeable rise toward the first maximum on any layerline. An MRC program which graphs layerline amplitudes and phases was very useful in deciding where to truncate the layerline data. The maximum radius used for each layerline was chosen on the basis of the next higher order Bessel function on that layerline and on the maximum radius of the tail. Bessel functions have approximately zero amplitude.
until the argument \((2\pi r R)\) reaches \(n-2\), where \(n\) is the order of the Bessel function. The maximum radius of the tail will yield the minimum radius at which each Bessel order will start to contribute to the layerline.

The delimited and indexed layerline data is abstracted from the half-plane DFT in three forms. The extraction program outputs either near side data only, far side data only, or both. The format of this output is designed to match with the input requirements of the MRC helical reconstruction program. If near side data is used, the reconstruction will be consistent with the diffraction arising from the near side of the specimen. Far side data will yield a far side consistent reconstruction. If both sets of data are used, the MRC program adds the two sets together thereby producing a reconstruction which represents an average of the near side and far side compatible structures.

In order to visualize the reconstruction, all the section types available (IV-M, N) were produced. These were viewed as the two-dimensional number plots which the MRC programs provide, and as our greyscale and contour plots. The Z-section display, produced at 1Å intervals along the tail axis for one repeat of the structure, was found to be especially useful for becoming acquainted with the structure. A perspective contour plot program was used to produce "stereo pairs" from stacked Z-sections. Figure 45 shows examples of some of these different graphic displays.
Figure 45

Examples of the different displays available for three-dimensional helical reconstructions. T4 tail is shown.

A) Greyscale picture of a horizontal z-section

B) Contour plot of a horizontal section

C) Greyscale picture of a side projection

D) Perspective contour plot of stacked z-sections

E) Stereo pair of stacked horizontal z-sections
A comparison was made with the higher resolution model of Amos and Klug (1975) by matching published Z-sections with our data. The results of this comparison are shown in Figure 46. From the good degree of correspondence, we deemed the conversion of the programs a success and our understanding of the methods adequate.

D. Image Restoration from Frozen Spirillum serpens Patches

Our increased sophistication in two-dimensional processing was used to analyze micrographs of frozen, hydrated *S. serpens* cell wall material. The specimens were provided by Dr. W. Chiu, and the micrographs by Dr. K. Taylor. The cell wall of *S. serpens* has been previously studied in negative stain and thin section (Buckmire and Murray, 1970, 1976; Chiu and Grano, 1977) and the negative stain preparation (Figure 47) shows doughnut-shaped units in an hexagonal array with thin linking "arms" forming three-fold contacts with adjacent units. The frozen hydrated specimen (Figure 47) is similar but exhibits reversed contrast (Taylor, Grano, and Chiu, 1976). The presence of linking arms is evident from a direct viewing of the micrograph but the variation from unit to unit is greater than in the negatively stained specimen. Image restoration procedures were applied to this specimen to suppress the contribution of noise and synthesize a spatially averaged structure.

The frozen hydrated specimens were imaged at 40,000X magnification. Scanning was done on a 17 by 17 μm raster,
Figure 46

Comparison of low-resolution T4 tail reconstruction with the published data of Amos and Klug
3-D RECONSTRUCTION OF A T4 BACTERIOPHAGE TAIL

SECTIONS ⊥ TO HELICAL AXIS

(Modified from L.A. Amos and A. Klug, J. Mol. Biol. (1975) 99)
Cell walls from *Spirillum serpens*

A) Negatively stained specimen

B) Frozen hydrated specimen – note the reversed contrast compared to A)
using array sizes large enough to cover the small patches, typically greater than 512 samples in each dimension. The scanned image and computed diffraction pattern for the best area is shown in Figure 48.

The Fourier coefficients were phase shifted until the coordinate origin of the Fourier transform was on a six-fold axis. The amplitudes and phases were symmetrized (averaged over the three-fold (six-fold) related diffraction orders).

Figure 49 shows the results of inverse transformation after symmetrizing and filtering. Corrections were not made for the contrast transfer function because no high dose micrograph was available for this image.

E. The Cylindrical Tubes of *S. serpens*

Negatively stained preparations of *S. serpens* cell wall display a varied morphology (Buckmire and Murray, 1970). One of the forms seen is a tubular one (Figure 50). Optical diffraction patterns of these tubes (Figure 50) have characteristics common to helical diffraction - a layerline structure with "smeared out" diffraction spots. There is a range of tube diameters but individual tubes have a near constant diameter. There is great variability in the lengths, specimens ranging from 0.1 to 1 μm. All of the tubes are associated with an "inner" core of backing layer (Taylor, Grano, and Chiu, 1976) around which the tubes are organized. The preserved order, as evidenced by the optical diffraction
Figure 48

Frozen hydrated specimen of *Spirillum serpens* cell wall

A) PLTPROC picture of the best patch

B) Computed diffraction pattern from A)
Figure 49

Symmetrized and delta-function-filtered image of a frozen hydrated cell wall from *Spirillum serpens*
Images and optical diffraction patterns of negatively stained *Spirillum serpens* cell wall tubes. The diffraction patterns exhibit layerlines.
HP LAYER FROM SPRLILLUM SPINOS IN THE HELICAL CONFIGURATION. PTA STAINED.

XBB762-1129
pattern, provided the impetus to try to determine the three-dimensional structure of the tubes using the methods of helical reconstruction.

The indexing of the diffraction pattern is the first step in undertaking the reconstruction. The computed diffraction pattern of one tube is shown in Figure 51, with a hexagonal net sketched on top. The layerline spacing is \( \frac{1}{145\,\text{Å}} \) and the pitch angle is 30° degrees. If the hexagonally packed surface layer were to roll up as diagrammed in Figure 52, pitch and layerline spacing would be as observed. The diffraction pattern can be indexed by the selection rule \( l = n' + 2m \), where \( n = n'R \), \( R \) being the rotational symmetry of the specimen. Figure 53 shows an \( n,l \) plot for this tube. The first meridional reflection is weak.

Since the symmetry of the tubes is cylindrical, the results of chapter V apply. There is no possibility of retrieving near side or far side structure by constructing masks which exclude certain orders since both near and far sides contribute positive and negative Bessel functions of the same order, which superimpose. We also saw in chapter V that the prospects for 3-D helical reconstruction using a single projection were nil. We can analyze the diffraction data to determine (perhaps) the rotational symmetry of the tubes.

As mentioned above, the position of the first maximum for a Bessel function depends on its order and the value of
Figure 51

Computed diffraction pattern of *Spirillum serpens* cell wall tube, with a superimposed hexagonal net
Figure 52

Diagram illustrating the formation of tubes from sheets
Figure 53

$n, l$ plot for *S. serpens* cell wall tubes.

$n = Rn'$, where $R$ is the rotational symmetry of the tube.
its argument. From the location of the first large diffraction peak, the order of a contributing Bessel function can be deduced if the maximum diameter of the tube is known. Unfortunately, the first maximum on a layerline arises from the collective actions of many atoms and of specimen noise and therefore does not represent a peak from a sharply localized atom. Hence, the radius to use in the Bessel function argument for the determination of the order may be less than the maximum radius of the tube. If the difference in Bessel orders is small, the first maximum locations do not vary much from order to order. These two factors can lead to an inability to distinguish between orders based on the location of the first maximum on a layerline. One does know, however, that the largest radius in an object will predict the smallest radius in Fourier space at which a given Bessel function can begin to contribute to the diffraction intensity. This lower limit can be used to weed out potential orders. Another characteristic of the Fourier transform which can be used to determine the Bessel function orders is the phase behavior on odd layerlines. If the tube has odd rotational symmetry, phases on opposite sides of the meridian will differ by 180°, if the phase origin is on the cylinder axis.

It might be thought that a knowledge of the tube outside diameter would be sufficient to determine the rotational symmetry of the tubes if they are assumed to be formed from rolled up sheets. The subunit to subunit
contacts will not be severely distorted and this continuous girdle of contacts should place a limit on tube sizes for each value of rotational symmetry. A complication arises because the plane of contact in the sheet form that remains in the tubes is not known. The extremes of contact are the bottom of the sheet (that side of the sheet which is in contact with the backing layer) and the top of the sheet. Table I shows the various tube inner and outer diameters expected for various rotational symmetries from three cases for the plane of contact: top, bottom and middle of the sheet. The calculations leading to these values were based on a sheet thickness of 155Å. This analysis assumes that the contact between subunits which establishes the 145Å center-to-center spacing in the planar sheets uniformly arcs after being bent into a tube. Although a more likely mechanism is to have a rotation about the point of contact, the slightly larger inner and outer diameters predicted on the basis of this latter mechanism is insufficient reason to get too "realistic" in the model for tube formation. The calculated diameters from the geometrically simple model will be used for a comparison of the tubes.

The tube diameter and diffraction maxima location methods have been used on two tube images and their transforms to investigate the rotational symmetry of the tubes. The first tube studied (Figure 54) has an outer diameter of approximately 530Å and an inner diameter of 200Å. From Table I, the rotational symmetry could range
TABLE I

Expected dimensions of *Spirillum serpens* tubes under three different assumptions

1) The sheets retain their center-to-center spacing at the inner surface of the tubes

2) The sheets retain their center-to-center spacing at the midwall of the tubes

3) The sheets retain their center-to-center spacing at the outer surface of the tubes
<table>
<thead>
<tr>
<th>Rotational Symmetry</th>
<th>OUTSIDE DIAMETER</th>
<th>INSIDE DIAMETER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Connection Location</td>
<td>Connection Location</td>
</tr>
<tr>
<td>N</td>
<td>Outside</td>
<td>Middle</td>
</tr>
<tr>
<td>3</td>
<td>240</td>
<td>395</td>
</tr>
<tr>
<td>4</td>
<td>320</td>
<td>475</td>
</tr>
<tr>
<td>5</td>
<td>400</td>
<td>555</td>
</tr>
<tr>
<td>6</td>
<td>480</td>
<td>635</td>
</tr>
<tr>
<td>7</td>
<td>560</td>
<td>715</td>
</tr>
<tr>
<td>8</td>
<td>640</td>
<td>795</td>
</tr>
<tr>
<td>9</td>
<td>720</td>
<td>875</td>
</tr>
<tr>
<td>10</td>
<td>800</td>
<td>955</td>
</tr>
<tr>
<td>11</td>
<td>880</td>
<td>1035</td>
</tr>
<tr>
<td>12</td>
<td>960</td>
<td>1115</td>
</tr>
<tr>
<td>13</td>
<td>1040</td>
<td>1195</td>
</tr>
<tr>
<td>14</td>
<td>1120</td>
<td>1275</td>
</tr>
<tr>
<td>15</td>
<td>1200</td>
<td>1355</td>
</tr>
<tr>
<td>16</td>
<td>1280</td>
<td>1435</td>
</tr>
<tr>
<td>17</td>
<td>1360</td>
<td>1515</td>
</tr>
<tr>
<td>18</td>
<td>1440</td>
<td>1595</td>
</tr>
<tr>
<td>19</td>
<td>1520</td>
<td>1675</td>
</tr>
<tr>
<td>20</td>
<td>1600</td>
<td>1755</td>
</tr>
</tbody>
</table>
Figure 54

Computer generated picture of a cell wall tube - case 1
from 3, for an inner-connected tube, to 7 for an outer-connected tube. The projection image has a glide plane along the cylinder axis, which is indicative of odd rotational symmetry (Moody, 1967). One must be cautious when using the projection image since unequal staining or preservation of near side and far side structures can make even-rotationally symmetric objects have a glide plane and odd-rotationally symmetric objects have a mirror plane. The near axial rise on the first maximum on the first layerline occurs at an argument of \(2\pi r_{\text{near}} r_{\text{max}} = 6.0\), using \(r_{\text{max}} = 265\AA\). From the selection rule, the contributing Bessel order at this position should be \(J_N\) where \(N\) is the rotational symmetry of the object. Only Bessel orders below 8 have a greater than .1 amplitude for this argument so all orders in question are allowed by this fact. The first off-axis maximum on the second layerline should come from \(J_{2N}\). The near-rise argument for this peak is 9.9. Only Bessel functions with orders 8 and 10 have a greater than .1 amplitude at this argument. The analysis of the first layerline phases showed them to differ by 180° across the meridian so we can conclude that Figure 54 is a tube with five-fold rotational symmetry whose "connection radius" lies between the outer radius of the tube and a mid-wall radius (see Table I).

Another tube (Figure 55) displays quite a different appearance than the last image but its outer diameter of 515Å and its inner diameter of 220Å are quite similar to the
Figure 55

Computer generated picture of a cell wall tube - case 2
other tube. Again, there is a strong glide plane in the image. The location of the first maxima on the first and second layerlines leads to arguments of 5 and 9 respectively, using the maximum tube radius in the argument. The phases on the first layerline indicate an odd Bessel function order. Again, a five-fold rotational symmetry is indicated, with the "connection radius" lying toward the outer diameter. The differences in the image are not hard to believe, given the results of chapter V.

What one actually wants to know is the 3-D distribution of material in the tubes. Unfortunately, the cylindrical symmetry they display precludes using simple helical reconstruction for this task. As indicated in chapter V, further experiments, with a tilting stage and new computer programs, would be needed to accomplish this task.

The work accomplished to date shows several things. The tubes are consistent with rolled-up sheets. The sheets roll without twisting or shearing along the tubular axis. An integral number of repeats of the sheet structure are included in the tubes. The tube outer and inner diameters are indicative of subunit to subunit contacts existing somewhere between the middle of the tube wall and the tube outer diameter, although no evidence was obtained to preclude contacts at other planes in the sheet form. Such contacts could break during tube formation and thus not constrain the tube sizes. The placement of the subunit to subunit con-
tacts nearer the extracellular side of the sheets is consistent with and an independent verification of the sheet model published by Glaeser et al (1979). Although both tubes subjected to computer analysis were of the same diameter, tubes do display different diameters. Tubes of various diameters presumably are composed of differing widths of sheet material which have rolled up into the tubular form.
VII. An Overview

The preceding chapters have taken us from the theoretical underpinnings of 2-D and 3-D reconstruction to the practical implementation of these methods and their use. Although 2-D and 3-D image processing programs have been used on biological problems at this laboratory, it is important to consider the question of how generally applicable such methods are. Will the working electron microscopist be able to use image processing programs? Will the biologist who uses the electron microscope as a research tool be able to use them? Lastly, is the cost prohibitive and are the results believable?

First, reconstruction methods will be of tremendous advantage in the solution of biological problems through the use of the electron microscope. Careful attention to the preparation and imaging of biological specimens has resulted in major improvements in obtaining useful information. Unfortunately, the image, being a projection of the structure, may not be interpretable at the level of resolution found in the structure. Sometimes, unavoidable contamination in the preparation obscures the specimen under study. The least intrusive preparation techniques, such as glucose embedding or the frozen hydration method, require the use of a low electron dose to avoid severe radiation damage. This leads to images with almost no visible features because of the low signal-to-noise ratio. Computer processing of these
micrographs is the best way to extract the specimen signal. By combining images recorded at different tilt angles, the 3-D distribution of the Coulomb potential can be determined. This 3-D work may be undertaken at low resolution to answer questions about the spatial orientation of subunits. High resolution work can answer questions about tertiary, and in favorable cases, secondary protein structure. The future may see the attainment of near atomic resolution. Reconstruction methods are not applicable to all biological problems investigated with the electron microscope, but the number of systems under study by these techniques is growing steadily and already represents more work than can be handled by the small number of workers in this field.

Can these techniques be useful to workers other than the developers? Again the answer is yes but it is a more cautious yes. The methods are powerful but subject to abuse. There is a strong subjective or interpretive element to the processing. Experience in this field can develop one's abilities in this regard. The main pitfall is the tendency to give more weight to a particular result than it deserves. The various sources of error in the analysis tend to be obscured as the number of steps between original image and final result increases. A lack of familiarity with the details and theory of the methods can lead to an excess of subjective input. The methods are not so arcane, however, that they are hopelessly beyond the reach of anyone outside the immediate field. To the contrary, the methods are
useable and useful to anyone interested in structure determination by electron microscopy.

These techniques do require computer facilities and money but not excessive amounts of either. A disk-based minicomputer system can handle the computer processing if the machine is available, at times, for the large blocks of time required to do the Fourier transforms of large data arrays which are required in low-dose work. Outside of this bottleneck, most of the processing is less CPU intensive. Of course, access to some type of image scanner (digitizer) is necessary. The highest resolution work requires a scanning microdensitometer capable of 5 μm sampling, while lower resolution studies could be accomplished with a drum scanner. The costs involved in a structural study are highly variable and depend greatly on the monetary policies of the owners of the computer. On the LBL computer (which runs on a full recharge basis), typical 2-D processing may run $50 to $100 per image. High-resolution 3-D processing could run to $3,000 per structure.

In conclusion, the programs developed in this thesis are applicable to an ever increasing range of biological structures. The structural secrets of nature will yield a bit faster by the use of the new and powerful technique of structural analysis by electron microscopy. Many questions that are outside the ability of X-ray diffraction to answer, and some within it, will be answered by electron microscopy.
Acknowledgments

I would like to thank Professor Robert M. Glaeser both for introducing me to the fields of electron microscopy and 3-D reconstruction and for his generous advice and support during the course of this work. I appreciate the careful reading and criticism of the thesis by Professors Thomas L. Hayes and Wayne L. Hubbell. Special thanks to Drs. Ken Taylor and Wah Chiu for many discussions and for providing the micrographs used in the analysis of *Spirillum serpens* images; to Dr. Bing Jap for testing my understanding and my patience through his questions; to Dr. Linda Amos of the MRC Laboratory for Molecular Biology for her help in providing the MRC helical reconstruction programs; and to Dr. Ivy Kuo, Mr. William Johnston, and Mr. Tom Strong for many fruitful discussions about the arts of computing and computer graphics.

Above all, I wish to thank Helen Johnson for the steadfast support, understanding, and motivation that she has given me throughout my graduate education.
Appendix A - Listing of Computer Routines

Entry Format

Routine name  Field 1  Field 2
              Field 3

Field 1  Program type
P = program
S = subroutine
F = function

Field 2  Language
F = FORTRAN
C = COMPASS (Assembly language)

Field 3  (Optional) Source of code

Abbreviations and Definitions

BKY  LBL computer center
CDC  Control Data Corporation
CPU  Central Processing Unit
CRT  Cathode ray tube display device
DEC  Digital Equipment Corporation
DOS  DEC disk operating system
FILES-11  File format for DEC's
          RSX-11 operating system
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFT</td>
<td>Fast Fourier transform</td>
</tr>
<tr>
<td>IDDS</td>
<td>Integrated data display system (BKY graphics package)</td>
</tr>
<tr>
<td>LBL</td>
<td>Lawrence Berkeley Laboratory</td>
</tr>
<tr>
<td>LCM</td>
<td>Large core memory (CDC 7600 only)</td>
</tr>
<tr>
<td>LLL</td>
<td>Lawrence Livermore Laboratory</td>
</tr>
<tr>
<td><strong>modified code</strong> - a routine which has been adapted for use on the BKY computer system. Some corrections and/or extensions may have been made to the code.</td>
<td></td>
</tr>
<tr>
<td>MRC</td>
<td>Medical Research Council Laboratory for Molecular Biology, Cambridge</td>
</tr>
<tr>
<td>OD</td>
<td>Optical density</td>
</tr>
<tr>
<td>PDS</td>
<td>Perkin Elmer flat-bed scanning microdensitometer</td>
</tr>
<tr>
<td>PRU</td>
<td>Physical record unit</td>
</tr>
<tr>
<td>s</td>
<td>a vector in reciprocal space</td>
</tr>
<tr>
<td>SCM</td>
<td>Small core memory (standard memory)</td>
</tr>
<tr>
<td>(sub-)array - a specified portion (including all) of an array which is the range of an operation.</td>
<td></td>
</tr>
<tr>
<td>UCB</td>
<td>University of California at Berkeley</td>
</tr>
<tr>
<td>Y</td>
<td>contrast transfer phase distortion term for electron micrographs</td>
</tr>
</tbody>
</table>
ABSOLOG S C
Takes the modulus, then log_{10} of a complex LCM array.
Returns the minimum and maximum logarithms generated.

ABSOLUT S C
Takes the modulus of an LCM complex array and returns
the minimum and maximum moduli encountered.

ARYAVG S F
Finds the average of an LCM (sub-)array (complex or
real). Optionally ignores 0.0 values in computation.

ASTIG1 P F
Calculates sin\gamma and \overline{s} for p3/p6 triplets given astigma-
tism and center-to-center spacing.

AVERGR S F
Reads scanner data (integers), optionally averages,
floats, pads, and/or replaces zero values, then converts
numbers to complex form in core.

AVGR S C
Averages integer data and places the result into a com-
plex LCM array.
BIGG P F

Takes scanned data or 1-D FFT's and calculates the 2-D Fourier transform of selected frequencies. Output includes the 1-D FFT's (if not input), the calculated coefficients, and block maps of amplitude and phase.

BIGTRAN P F

Performs a "piecewise" Fourier transform on integer data (which is bilinearly interpolated by this program).

BILIN S F

Interpolates LCM array values using a bilinear interpolation scheme.

BLCKOUT S F

Reads blocks of Fourier coefficients from memory (calculated by BIGG, for example), shifts the phase origin to the center of the transformed area, then prints out amplitude and phase block maps (identified by HK value).

BSL0 F F MRC code

Calculates $J_0$ of an argument.

BSL1 F F MRC code

Calculates $J_1$ of an argument.
BUBBLE S F

Used by routine HEAPSRT to "bubble" higher values to the beginning of an array.
CAMPKRK S F

Converts DOS-format PDS scanner tapes (from the UCB Department of Astronomy scanner) into CDC-format.

CARVER P F

Prepares an input file for the phase origin search program PSRC2D, using the maximum amplitude coefficient from each block of an LTCLOOK run.

CF FT S F (E2 BK Y CFFT)

Performs a radix-2 FFT on complex LCM arrays.

CNV S F

Takes a file with row, column, H and K values (1 set to a card) and converts to row and column sets, eight to a card (the input format which is required by GNEIGH).

COMPACT P F

Creates one input deck for routine PSRC2D for each triplet (p3/p6-symmetry related diffraction orders) in a PSRC2D deck containing many triplets. Allows each triplet to be studied individually.

CONMAP S F (J5 BK Y CONMAP)

Produces a contour plot.
CONPLO  S  F  (J5 BKY CONMAP)
Part of the CONMAP contour routines.

CONVOD   P  F
Converts a percentage transmission PDS-scan into an OD-like scan.

CPFT    S  C  LLL-modified R.C. Singleton routine
Calculates a 1-D complex FFT for radix-2 array length.

CPFTL   S  C  LBL-, LLL-modified R.C. Singleton routine
Calculates a 1-D FFT on radix-2 LCM arrays.

CRTDISP  S  C
Converts an array of integers into a random pixel CRT greyscale picture.

CRTPAT   S  C
Converts an array of integers into a patterned pixel CRT greyscale picture.

CULEFT  F  F
A dummy (do nothing) function.

CUTNGET  S  F
Reads into core (LCM) a specified portion of a disk-stored integer array. Convert to a complex form.

CVRTIVY  S  F
Converts GNEIGH format row and column locations to left half-plane (Freidel-equivalent) positions.
DECOBRD  S  F
Sets-up (sub-)array borders for PLTPROC-called display routines.

DECODER  S  F
Converts coded instructions to an easily usable form.

DECODE1  S  F
Prepares directions for subroutine FOURIER from alphanumeric input.

DECODE2  S  F
Converts input choice arguments to internal flags used by various PLTPROC-called routines.

DEC7T6  S  C
Unpack 16-bit data stored in DOS 7-track format.

DESIGN  S  F
Constructs an array filled with a user-defined motif.

DIFPOS  S  F
Calculates position in a DFT of a given spatial frequency.

DIGITYZ  S  C
Linearly scales an LCM array (real or complex) into a given integer range and puts results into an SCM array.
DIGZ  S  F
   Maps either part of a complex LCM (sub-)array into a given integer array.

DMPREG  S  F  (N2 NYU REGDMP)
   Prints the CPU register contents saved by a register dump.

DOBILIN  S  F
   Prepares a call to the "BILIN"ear interpolator.

DOCAMP  P  F
   Performs a conversion of DOS-format PDS scan tapes to CDC format by calling subroutine CAMPKRK.

DOCAMPI  P  F
   Calls the subroutine NEWCAMP to convert FILES-ll PDS scan tapes to CDC format.

DOCON  S  F
   Produces contour plots, customized for the sections produced by the 3-D helical reconstruction program EMHLFOU.

DOFTH  P  F
   Converts full-plane DFT's to half-plane DFT's and puts them into a form usable by the MRC routines.
DOHLX P F
Reads a half-plane DFT into core and calls routine EM4HLX to extract layerline data for the MRC 3-D reconstruction programs.

DOLRGNT P F
Calls the LRGINT subroutine to perform a disk-based bilinear interpolation.

DOMGMP S F
Sets up a call to subroutine MPOUT which produces two digit maps of array contents.

DOPLATE P F
Sets up a call to PLTPROC, the plate processing subroutine.

DOPRPIC S F
Sets up a call to PRPICT, the greyscale overprint picture subroutine.

DORADPR P F
Calculates the radial projection of a helical reconstruction from its $g_{n,l}$ functions.

DOSRCH P F
Calls subroutine EMSRCH to search for the helical axis origin and tilt.
DOSTUFF  S  F
        Prepares pictures, prints array contents, and/or saves arrays on disk.

DOTRANS  P  F
        Sets up a call to subroutine TRANSLT to convert "UCB Space Science Scanner" format scans to CDC format.

DO3PIC  S  F
        Produces three greyscale pictures of an MRC-LBL helical reconstruction section.

DO4HLX  P  F
        Calls subroutine EM4HLX to prepare data for 3-D helical reconstruction.
EBCDIC  P  F
Converts card image blocks in EBCDIC code to CDC internal display code and creates 80 character card images.

EMBES2  S  F  modified MRC code
Generates positive order Bessel function values for batches of 2000 arguments.

EMCURV  S  F  modified MRC code
Produces a printer plot of $g_{n,1}$ functions.

EMHFC2  S  F  modified MRC code
Produces a Fourier synthesis on cylindrical sections.

EMHFP2  S  F  modified MRC code
Produces a Fourier synthesis in vertical, parallel sections.

EMHFRP  S  F
Performs a radial projection Fourier synthesis from $g_{n,1}$ functions.

EMHFW2  S  F  modified MRC code
Produces a Fourier synthesis on vertical (through-the-axis) sections.
EMHFZP  S  F  modified MRC code
    Produces a down-the-z-axis projection from a Fourier synthesis on horizontal z-sections.

EMHLFOU  S  F  modified MRC code
    Performs a 3-D helical reconstruction using Fourier-Bessel techniques. Produces user-specified section types and saves all results on disk.

EMIDSV   S  F
    Saves helical reconstruction section identification data on disk.

EMLTLG   S  F  modified MRC code
    Calculates and saves the $g_{n,l}$ functions for helical objects.

EMOUTDO  P  F
    Reads into core a column-major order half-plane DFT and produces magnitude and phase maps in MRC format.

EMSCAL   S  F  MRC code
    Scales the $g_{n,l}$ functions for the output routines.

EMSRCH   S  F  modified MRC code
    Searches for the tilt and phase origin of the helical axis in a helical transform.
EMTRFN  S  F  modified MRC code

Applies the correction for the contrast transfer function (CTF) to 3-D helical reconstruction data.

EMUCLA  P  F

Calls subroutine EMHLFOU to perform a helical reconstruction.

EM3DISP  P  F

Produces greyscale and/or contour plots of helical reconstruction sections saved on disk by program EMUCLA.

EM3DIS2  P  F

Produces contour plots of helical reconstruction sections saved from a 3-D run.

EM4HLX  S  F  modified MRC code

Extracts layerline data from a helical transform for the 3-D helical reconstruction programs.

EM4NTR  S  F  modified MRC code

Interpolates Fourier coefficients along layerlines by using bilinear interpolation.

EM4OUT3  S  F  modified MRC code

 Takes a quasi-row major order half-plane DFT in real and imaginary form and produces maps of the phase and the logarithm of the amplitudes.
ERRAB  S  F

Calculates the error between the actual and expected locations of a diffraction spot. Expresses the error in terms of the lattice vector components.

EXPOSE  S  F

Calculates minimum and maximum values for an array so that a specified percentage of the array elements will lie outside the calculated limits.
-F-

**FFT** S F modified from R.C. Singleton
Performs a mixed-radix FFT on LCM data.

**FFTSCM** S F R.C. Singleton
Calculates the mixed-radix FFT on SCM data.

**FILLER2** S F
Replaces real or complex array elements in a given SCM (sub-)array.

**FILLER3** S F
Fills a real or complex LCM (sub-)array with a given value.

**FIXFET** S C
Fixes the File Environment Table (FET) of the film file after routines CRTPAT or CRTDISP are called so that the FET can be used by RUN76 I/O routines.

**FLAT1** F F
Returns the floated (real) form of an integer.

**FLIPEV** S F
Flips an even-dimensioned LCM array to/from "natural" frequency ordering.
FLIPFT S F
Flips a 1-, 2-, or 3-D even-dimensioned SCM array using a "magic number" algorithm.

FLIPLCM S C
Flips a 1-, 2-, or 3-D even-dimensioned real or complex LCM array to/from "natural" order.

FLIPSCM S C
Flips a 1-, 2-, or 3-D even-dimensioned SCM array to/from "natural" order.

FLOAT1 S F
Calculates the average value of a (sub-)array boundary, then floats the (sub-)array by subtracting the boundary average from all (sub-)array elements.

FLOATER1 S F
Takes an integer array and puts its real number equivalent into another array.

FLP1DE S F
Flips an even length LCM array to/from "natural" frequency ordering.

FLP1DO S F
Flips an odd length LCM array to/from "natural" frequency ordering.
FLP2D S F
Flips a 2-D LCM array with at least one odd dimension to/from "natural" frequency ordering.

FLP2DC S F
Flips a real or complex 2-D LCM array to/from "natural" frequency ordering.

FOURIER S F
Performs Fourier transformations and related functions and displays results (all under user control).

FRACKRC S F
Converts fractions of total rows and columns to actual row and column limits.

FTHDFT S F
Converts full-plane DFT's to half-plane DFT's.

F11CDC S C
Converts the contents of a FILES-11 format PRU as read by CDC into one 16-bit DEC word per CDC word.
GAMMAI F F F
Calculates $\gamma$ at specified $s$ given the astigmatism magnitude and direction.

GAPLANE S F F
Constructs an array of 20 unit cells (4 vertically by 5 horizontally) whose motif is determined by card input.

GENFFT S F F
Constructs a complex FFT from user input structure factors.

GENMSKI P F F
Generates subroutine PICSPOT input from user directions.

GENNEY P F F
Generates card output representing array locations neighboring given input locations.

GETDATA S F F
Moves indicated data type from disk to LCM. Data is assumed to be in PLTPROC-compatible form.

GETDATS S F F
Moves indicated data type from disk to SCM. Data is assumed to be in PLTPROC-compatible form.
GETIT S F
Reads an array from disk to SCM.

GETTER S F
Extracts a 2-D LCM array from a disk file after optionally skipping some records.

GETTER1 S F
Extracts a 2-D SCM array from a disk file after optionally skipping some records.

GNEIGH S F
Generates array locations in a box surrounding a given array location.

GRAPHG S F modified BKY IDDS routine
Draws a graph.

GRAPH1 S F
Graphs a function on the printer.

GRAPH2 S F
Graphs a function on the printer.

GSAVE S F
Saves the $g_{n,1}$ functions for a helical reconstruction.
HARM S F MRC code
Computes the FFT for radix-2 3-D complex arrays, stored in column major order.

HEAPSRT S F
Sorts an M by N array in order of decreasing values of A(1,N), carrying along the M-1 by N other elements. Uses a heap sort algorithm.

HERMFT S F MRC code
Calculates the Hermitian symmetric Fourier transform.

HEXDRIV P F
Generates a p3 reciprocal plane lattice based on the array location of one spot.

HEXGEN S F
Generates the array locations of p3 reciprocal plane lattice points, given the location of one spot.

HISTO P F
Creates histograms of scanner data.

HKLIMT S F
From a knowledge of the reciprocal lattice vectors, calculates the largest and smallest H and K that will occur in a given size array.
HKVAL  S  F

Calculates the closest HK value for a given array location using user supplied reciprocal lattice vectors. Optionally, finds the actual array location of the HK value and the error between the two locations.

HLFLIP  S  F

Flips a half-plane DFT so that the zero-frequency column is in the middle of the array.
ICHECK   F  F

Determines if a given diffraction order is within the array bounds or not.

ICKBRD   F  F

Given the array size, reciprocal lattice vectors, and a "border" width, determines if a given diffraction order falls inward of the array's border region.

ICOMPAF  S  F

Determines if the modulus of a DFT at one given frequency is greater than that at another given frequency. Used to determine the "flip state" of a DFT.

ICOMPAR  S  F

Compares the magnitude of two array elements.

IDRITER P  F

Prints the identification records on scanner and PLTPROC-saved files.

INBOUND F  F

Given the array size and x,y coordinates of a location, determines which coordinate(s), if any, are outside an array.
INDFRDL S F
Returns the location of a Freidel-related coefficient to a given DFT coefficient.

INPUTT S F
Reads a (sub-)array of (packed or unpacked) integer data from disk into LCM.

INPUT1 S F
Reads a (sub-)array of (packed or unpacked) integer data from disk into LCM and converts data to complex form.

INTENSE S C
Calculates the modulus of a complex array and returns the largest and smallest modulii encountered.

INTLOG2 F C
For an integer N, returns \( \log_2 \) of the largest integer \( \leq N \).

INT08 S F
Reads HK values from a common block and writes them in sets of 8 on cards (routine PICSPOT format).

INTRPL1 S F
Performs linear interpolation on a 1-D array or on a row or column from a 2-D array. Optionally, stores the interpolated values in place or in a separate storage array.
IP S F (E2 BKY LSQS)

Part of the LBL least-squares linear equation solver.

ITRUNC2 F C

For an integer N, returns the largest integer ≤ N which is a power of 2.

IVY1 P F

Generates randomly distributed Fourier coefficients that are common to two differently sized DFT's.
KRAC  S  F

Converts linear array indices to I,J,K indices.
LAYRMK  P  F

Produces layerline masks from measurements made of power spectra pictures.

LCMCIRC  S  F

Allows one to construct arbitrarily located and sized circles (in either a real or complex LCM array) whose inside or outside value is chosen by the user.

LCMFFT  S  F

Prepares calls for 1-, 2-, or 3-D forward or inverse FFT's on complex LCM arrays.

LCMLG1F  S  F

Performs the $\log_{10}$ on a real or complex LCM array.

LCMLG10  S  C

Performs the $\log_{10}$ on a real or complex 1- or 2-D LCM array.

LIPCHK  S  F

Determines whether a DFT is in "natural" frequency order or not.

LLLFFT  S  F

Performs a radix-2 FFT on real or complex data using a COMPASS-coded FFT routine written at LLL.
LOG10 S F

Takes the $\log_{10}$ of the amplitude part of a complex LCM (sub-)array.

LOKSPTL S F

From a file of Fourier transform data (such as that produced by QWIKPIC), produces block maps of amplitude and phase, both on disk and on paper.

LOOKSPT P F

Produces block amplitude and phase maps from the output of the PICSPOT masking subroutine.

LRCINT S F

Performs a bilinear interpolation on integer (packed or unpacked) disk data.

LSQS S F (E2 BKY LSQS)

Solves an overdetermined system of linear equations $AX=B$.

LTCLOOK P F

Given a Fourier transform on disk and the locations of $\geq 2$ known diffraction orders: 1) produces amplitude and phase maps of the neighborhood around each diffraction order, 2) produces a file of all these Fourier coefficients, and 3) produces a file usable by PLTPROC for optical filtration.
LTCMKR    P  F

Finds the location of all lattice points in the right half-plane, given the reciprocal lattice vectors and (optionally) extinction conditions. The locations are written to disk in subroutine GNEIGH format.

LTCPT    S  F

Given the reciprocal lattice vectors and array size, returns the exact location and nearest array element for a given diffraction order. Indicates whether the array element is within the array bounds or not.

LTO1ST    P  F

Converts a disk file produced by the masking subroutine PICSPOT to one that can be read by the LOOKSPT program.
MAKGRID $SF$

Constructs a box around a CRT picture and optionally puts tick marks at every pixel.

MAKHEAP $SF$

Used by subroutine HEAPSRT to construct a tree.

MAK3D $SF$

Inserts a given 2-D array into a 3-D array at a specified plane.

MDFTKD MRC code

Multi-dimensional complex Fourier transform kernel driver.

MINIMAX $SF$

Finds the minimum and maximum values in a real or complex LCM (sub-)array. Zero values can be ignored.

MOVE $SF$

Moves the contents of one LCM array "column" to another "column".

MPOUT $SF$

Converts a complex LCM array to amplitude and phase form. Optionally produces maps of the amplitudes and/or phases.
MSG       S F

Prints data about the sections produced by EMUCLA (helical reconstruction program).

MUV       S F

Moves an array from one common block to another.
NEWCAMP S F

Converts raw scanner data in DEC FILES-11 format to CDC format. Includes a transmission-to-OD conversion and output data packing, among other options.
ORDER S F

Indirectly orders an array by creating a list of indices which point to the array elements in the order of increasing value.
PACK660 S C

Packs an array of 60-bit words each holding one 10-bit data word into an array containing 60-bit words holding six 10-bit data words.

PATINT S F

Initializes the patterned CRT greyscale picture routine CRTPAT.

PHASE F F

Returns the phase in radians of a complex number. The phase lies in the range $\frac{-\pi}{2}$ to $\frac{\pi}{2}$.

PHASEY F F

Returns the phase of a complex number in the $0^\circ$-$360^\circ$ range.

PHASORG S F

Moves the phase origin of a complex FFT to a user-supplied location.

PHAVG1 S F

Calculates the mean of two arrays (part of the P3PSRH program).

PICSPOT S F

 Masks a DFT according to card input.
PIECETR S F
Performs a "piecewise" DFT in the forward direction on integer disk data.

PLTPROC S F
Processes scanned images, disk data, or array data.

PRINTER S F
Prints the contents of a real or complex LCM (sub-)array.

PRINTID S F
Prints the contents of the identification record found on scanner or PLTPROC disk files.

PRNPIC S C
Given an array of integers, sets up the printing lines needed to produce a line of greyscale overprinting.

PRPICT S F
Produces an overprint greyscale picture of a complex or real LCM (sub-)array.

PSEARCH P F
An interactive phase origin shifting program.
PSRC2D  P  F

Searches for p3/p6 symmetry axes by producing 2-D maps of the RMS phase error as a function of the phase origin location for groups of symmetry related diffraction spots. Also, recalculates phases and errors for a given phase origin location.

PSRPGG  P  F

Searches for PGG symmetry axes by producing 2-D maps of the RMS phase error as a function of the phase origin location for groups of symmetry related diffraction spots. Also, recalculates phases and errors for a given phase origin location.

P3DIGS  S  F

Produces two-digit integer maps of either part of a complex LCM array filled with integers.

P3FILL  S  F

Calculates HK values, expected locations, and nearest array locations for all p3/p6 triplets which fall within a specified array size.

P3OR6  P  F

Calculates "structure factors" for p3 or p6 plane group models given "atomic" coordinates.
P3PSRH  P  F

Calculates the phase origin shift needed to move the phase origin to a 3-fold axis.

P36L1  P  F

From >2 user supplied diffraction spot positions, produces card input for the BIGG program, containing locations of all the triplets within a specified array size.
QUADFR  S  F
Determines in which DFT quadrant a given array location lies.

QUAD2  S  F  (J5 BKY CONMAP)
Part of the CONMAP contour routines.

QUIKSPT  P  F
Facilitates the preparation of input data for the MRC helix axis refinement program EMSRCH.

QWIKGN  S  F
Calculates the locations of neighbors to a set of diffraction spots (part of the LTCLOOK program).

QWIKPIC  S  F
From a file of array locations, creates another file containing array locations and the DFT values found there. This file is suitable for input to the transform masking portion of the PLTPROC routine.
REALF S F
   Converts the results of an N x M complex DFT on 2N x M real data into a N x M complex half-plane DFT.

REALFT S F MRC code
   Performs the Fourier transform of real data.

RECFILL S F
   Generates HK values, expected locations, and nearest array locations for all diffraction orders in the half-plane of an array of given size.

RECKON S F (J5 BKY CONMAP)
   Part of the CONMAP routines.

REGDMP S C (N2 NYU REGDMP)
   Dumps the contents of the X, A, and B CPU registers.

REPLAC S F
   Finds the F(00) term and replaces it with the next largest term in the DFT (if F(00) was the largest). Returns the values found, reports the action taken, and reports the "flip state" of the array.
REPHSBG  P  F

Changes the phase origin of Fourier coefficients stored from a routine BIGG run and recreates the amplitude and phase block output using the new origin.

RERANGE  S  F

Rotates the contents of an array 90° clockwise while transferring it from one common block to another.

RERANG2  S  F

Rotates the contents of an array 90° clockwise while transferring it from one common block to another.

RESOL  F  F

Returns the resolution of a given orthogonal HKL diffraction spot.

RESTOR  S  C  (N2 NYU REGDMP)

Restores the contents of the X, A, and B CPU registers (that were saved by subroutine SAVE).

RINGP  P  F

Traces the path of constant reciprocal radius in a DFT with (un)equal sample spacing by printing the column locations of each row intercept.
**RMSFGG S F**

For PGG plane groups, given a phase origin shift and a set of symmetry-related diffraction spots with their spatial frequencies and phases, calculates the phases at the new origin, the symmetry-consistent phase, and the RMS phase error between the two.

**RMSF36 S F**

For p3/p6 plane groups, given a phase origin shift and a set of triplet diffraction spots with their spatial frequencies and phases, calculates the phases at the new origin, the symmetry-consistent phase, and the RMS phase error between the two.

**ROT2D S F**

Rotates the contents of a real, 2-D array about an arbitrary origin by using bilinear interpolation.

**ROT2DR S F**

Rotates the contents of a real LCM 2-D array about an arbitrary origin by using bilinear interpolation.

**ROWMAJ S F**

Converts a column-major order half-plane DFT to "quasi" row-major order complex form (real and imaginary parts alternate along a row).
RPFTI2  S C

Scrambles the Fourier coefficients of 2 real sequences so that one complex FFT can generate both sequences after transformation.

RPFT2  S C

Unscrambles the FFT of two real sequences performed simultaneously with one complex FFT.

RST  S F

Moves a generalized array "column" from SCM to LCM.

RTHETA  S F

Converts a specified portion of a complex LCM array to amplitude and phase form.

RZTMX  S F

Replaces the zero-order term in a DFT with the next largest term.

R2CFTK  MRC code

Radix-2 multi-dimensional complex Fourier transform kernel.

R3CFTK  MRC code

Radix-3 multi-dimensional complex Fourier transform kernel.
R4CFTK   MRC code
        Radix-4 multi-dimensional complex Fourier transform kernel.

R5CFTK   MRC code
        Radix-5 multi-dimensional complex Fourier transform kernel.

R8CFTK   MRC code
        Radix-8 multi-dimensional complex Fourier transform kernel.
SAVE   S  C  (N2 NYU REGDMP)
Saves the contents of the X, A, and B CPU registers.

SAVER   S  F
Stores an LCM array (real or complex) on disk in the format of one array column per logical record.

SAVFST   S  F
Saves alternate members of a portion of a 2-D LCM array in SCM.

SCRCH5   S  F
Given an integer <100,000, returns the closest integer with no prime factor >5.

SETUP   S  F
Establishes row boundaries for circular areas used in subroutine LCMCIRC.

SFRQS1   S  F
Calculates either |s| or s for a user specified set of diffraction orders and arbitrary reciprocal lattice vectors.

SHEARZ   S  F  modified MRC code
Corrects for shear in layerline data to be used in 3-D helical reconstruction.
SHOWME S F

Breaks apart CRT spot writing instructions to show raster locations.

SHOWPK S C

Extracts raster locations from CRT spot writing instructions.

SINGAMA P F

Calculates $\sin\gamma(s)$ (no astigmatism correction) for user chosen HK diffraction orders for orthogonal reciprocal axes.

SINGAM2 P F

Calculates resolution and $\sin\gamma(s)$ (no astigmatism correction) for user chosen HK diffraction orders in the p3/p6 plane group.

SKIPFIL S F

Skips a given number of logical files on a disk file.

SKIPREC S F

Skips a given number of logical records on a disk file; clears an EOF if encountered (terminates record skipping).

SQRAPT S F

Places rectangular areas of arbitrary size at arbitrary locations in a real or complex LCM array. Either the inside or the outside of the area can be set to a given value.
SPOT    S F
Determine a least-squares fitted reciprocal lattice from user supplied diffraction locations and then calculates the location of all orders which fall within a user defined array size.

SRFP    S F MRC code
Symmetrized reordering factoring program.

STANRD  S F
Establishes standard display options.

STREDIS  S C
Stores the addresses of LCM and SCM arrays for use by DIGITYZ and/or CRTDISP.

STRIPE  S F
Constructs an alternating bar array of variable width and contrast.

STRTAVG S C
Initializes the AVGR routine.

STRTDL  S F
Stores an LCM array on disk in a form compatible with the image processing programs.

SRTTDS  S F
Stores an SCM array on disk in a form compatible with the image processing programs.
SUMVAL  S  F
    Calculates the complex sum in a box around a given array location.

SVE  S  F
    Transfers a general LCM "array" to SCM.

SWITCH  S  F
    Interchanges the contents of two generalized LCM array "columns".

SYNCON  P  F
    Produces greyscale and/or contour plots under user control.

SYNTHC  P  F
    Synthesizes an object from its Fourier transform using subroutine PLTPROC.
TDESIGN  P  F
   Constructs a "design" in an array.

TILT    S  F  modified MRC code
   Corrects layerline data for a helix axis tilted out of the plane normal to the beam direction.

TRANSF  S  F
   Transfers alternate array elements from one column to another.

TRANSLT S  F
   Converts a "Space Science Lab" format scan to CDC format.

TRPLST  P  F
   Produces and prints a list of 3-fold related diffraction spots (HK0) ordered by increasing resolution, given the center-to-center spacing.

TVPICT  S  F
   Creates a greyscale and/or printer picture of an LCM array.

TVSIZER S  F
   Returns some facts for each CRT lettering size.
TWOTIMR  S  F

Converts a complex array with alternating real and imaginary parts to two separate arrays, one real and the other imaginary.
UNPACK S C

Unpacks 16-bit data packed 15 16-bit words in 4 60-bit CDC words.

UNPC660 S C

Expands integer data packed 6 10-bit words per 60-bit CDC word into one 10-bit word per CDC word.

UNSCRMB S F

Takes a complex array column and puts real parts into first half and imaginary parts into second half of the column.
WCARD S F

Converts 80 EBCDIC characters to 80 CDC internal display code characters and writes them out as a card image.
ZEROER  S  C
       Zeroes a complex LCM (sub-)array.

ZFILL  S  F
       Replaces 0.0 values encountered in an LCM array (real or complex, both or either part) with a given value.
References


This report was done with support from the Department of Energy. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the Department of Energy.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.