TR82-012
GRIP820908

A MAN-MACHINE INTERFACE FOR INTERPRETING ELECTRON DENSITY MAPS

by

Thomas Victor Williams

A MAN-MACHINE INTERFACE FOR INTERPRETING ELECTRON DENSITY MAPS

by

Thomas Victor Williams

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill

1982

Approved by:

_____

Adviser: F. P. Brooks, Jr.

_____

Reader: L. F. TenEyck

_____

Reader: J. S. Richardson

THOMAS VICTOR WILLIAMS.
A Man-Machine Interface for Interpreting Electron Density Maps (Under the direction of F. P. BROOKS, JR.)

ABSTRACT

I have designed and implemented a tool for biochemists to use for interpreting electron density maps of crystallized proteins. My work has been concentrated on the representation for electron density maps and on the man-machine interface.

Interpreting an electron density map is a difficult pattern recognition problem requiring specialized knowledge of protein structure and global views of the map. The tool is an interactive graphics system in which the human makes strategy decisions and does global pattern recognition, naming and pointing to recognized features in the map. These features belong to a hierarchy of objects natural to the problem. The computer does local, anchored pattern recognition for indicated features, displays the map in a ridge line representation using color to encode the map's interpretation, and automatically builds a molecular model as the user identifies residues.

A ridge line representation for maps was chosen because of the close correspondence of ridge lines to stick-figure models of molecules, because of the relatively few line segments required to display them, and because of the ease with which the density threshold can be changed in real time.

Three different sets of people have interpreted electron density maps using this tool. A computer scientist (myself) interpreted a 2.5 A map of Staphyloccal nuclease in 26 hours. This was the first map I had ever interpreted. A highly-skilled professional biochemist interpreted a 3.0 A map of cytochrome b5 in 9 hours. A group of three biochemistry graduate students and post-doctoral fellows interpreted a 2.8 A map of cytochrome c550 in 22 hours. These three successful interpretations done in such relatively short times have shown that the system design is a good and useful one for this application.

The contributions of this work to computer science are 1) a documented example of a good man-machine interface, 2) a detailed discussion of the major design decisions that I made, and 3) a demonstration of the usefulness of a ridge line representation for a scalar function of three variables.

To Julie

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Frederick P. Brooks, Jr., for helping me formulate and organize this project and dissertation.

I also thank Dr. Lynn Ten Eyck and Jane Richardson for helping me to understand the application area and for making useful comments on various features that I proposed.

Jane Richardson also deserves credit for supplying several illustrations and test data.

Jane Richardson, Libby Getzoff, John Tainer, and Duncan McRee spent hours testing the system that I developed.

Mike Pique, Lee Nackman, and Gary Bishop each supplied pieces of software. I had many useful technical discussions with them and with Jim Lipscomb and Geoff Frank.

Most of all, I would like thank my wife for her loving support.

This work was supported by NIH Biotechnology Research Resource grant #RR00898.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter I

INTRODUCTION

This dissertation describes the design of a tool for biochemists to use for interpreting the electron density maps of proteins. The main thrust of this work is the design of a man-machine interface for solving the problem and not the problem itself.

My thesis is that there is a class of problems, especially those that require global perception, for which a man-machine system can be effective where automatic, machine-only systems have been ineffective and manual, man-only systems are too slow.

Chapter 1 introduces the application area, explaining what proteins and electron density maps are, what is meant by interpretation, and why the problem is of interest. This chapter is intended to convey a general understanding of the application area, familiarizing the reader with the task of interpretation and the properties of the data.

Chapter 2 describes the major design decisions that I made, concentrating on the division of labor between man and machine, the representation of the data, and communication between man and machine.

Chapter 3 is a reference manual for the system. It describes the architecture of the system, i.e., the details visible to the user.

Chapter 4 presents highlights of the system implementation. Portions of the system that are peculiar to this application or that are particularly important to the system are described, not in fine detail, but sufficiently to give a general feel for how the system works.

Chapter 5 explains how the system was tested, the results of those tests, and conclusions based on those results. The chapter concludes with some suggestions for further research.

The contributions of this work to computer science are 1) a documented example of a good man-machine interface, 2) a detailed discussion of the major design decisions that

I made, and 3) a demonstration of the usefulness of a ridge line representation for a scalar function of three variables.

1.1 A DESCRIPTION OF THE PROBLEM

1.1.1 Significance of Problem

Proteins are essential for the existence of all living organisms, fulfilling a diversity of roles. Globular proteins are responsible for enzymatic catalysis, regulation, transport and storage, and immune protection. A protein's three dimensional structure determines its function. Much of its activity depends on its ability to recognize other molecules by shape, size, and charge distribution, all properties of their three dimensional structures. The active sites of all globular proteins seem to be in clefts in the protein structure, isolated places with special environments just right for some particular function. A lock and key analogy is often used since the recognized molecule must fit precisely into the active site and have the necessary properties to be bound there at least temporarily.

Since the mechanism by which a protein functions is so closely related to its structure, it is possible to discover or further explain how a protein works from studying its structure. It is the precise structure that is important, not just the general shape. Modifying a protein structure even slightly so that a critical atom is displaced half an Angstrom can make the protein nonfunctional. In the same manner, if a protein structure determination is not precise enough, no hint might remain of how the active site works.

The analysis of X-ray (and to a lesser extent electron and neutron) diffraction data is now the only method of determining precise three-dimensional structures of proteins. Maps produced by electron or neutron diffraction may also be amenable to interpretation using the same methods that work on maps produced from X-ray diffraction data. Raman spectroscopy and especially nuclear magnetic resonance techniques are improving and it may soon be possible to solve protein structures from data acquired from those sources.

1.1.2 Protein Structure

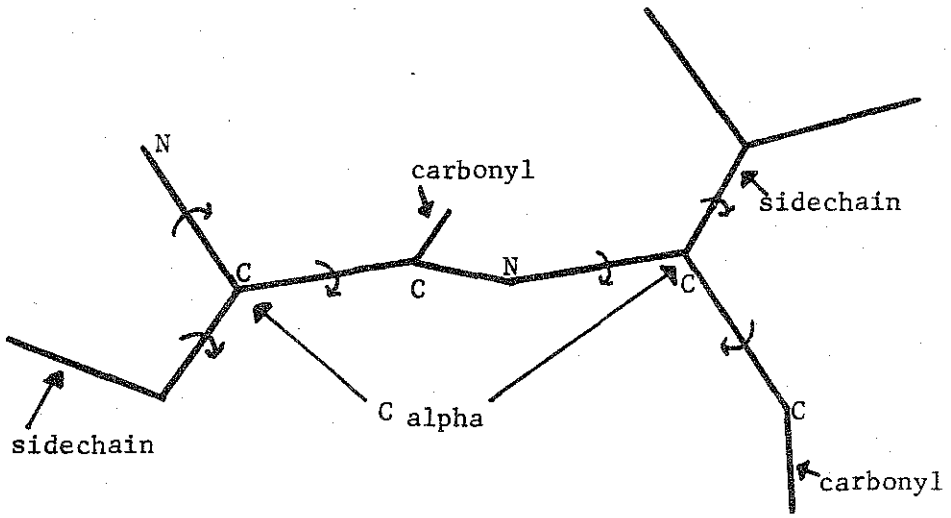

Figure 1.1: A stick figure model of two residues. The arcs indicate twistable bonds.

Proteins are linear polymers of amino acid residues. An amino acid residue is an organic molecule consisting of carbon, nitrogen, oxygen, and hydrogen atoms with occasional occurrences of other types of atoms. The atoms of a residue can be grouped into three overlapping parts: a mainchain segment (N-C-C'), a carbonyl segment (C'-O), and a sidechain segment that may be null, where N, C, and O are nitrogen, carbon, and oxygen atoms (see figure 1.1). By convention, the hydrogen atoms are suppressed in most views of proteins. There are twenty common residue types each with its characteristic sidechain structure. The mainchain and carbonyl segments of all residues are alike. All sidechains are bonded to the C alpha atom of the mainchain. Proline sidechains are also bonded to the N atom of the mainchain. Each residue type has a fixed topology; fifteen have tree struc-

tures and the remaining five have cycles. There are geometric constraints on bond lengths, bond angles, and dihedral angles. As a first approximation, from potential energy considerations, the bond lengths, bond angles, and some dihedral angles may be considered to be fixed, leaving some dihedral angles that can vary by rotation about appropriate bonds.

The residues are connected along their mainchain segments by peptide bonds from the C' atom of one residue to the N atom of the next residue, forming a chain which runs the length of the protein. The group C-C'(-O)-N-C is a rigid body since the peptide bond (C'-N) is not twistable (see figure 1.1).

Each type of protein is built of a fixed sequence of amino acids which completely determines the primary structure of the protein, that is, the topology and much of the geometry of the molecule. Most of the proteins whose structures have been determined have between 50 and 500 residues and are folded into a globular shape. The secondary structure consists of groupings of adjacent residues into features, the most significant of which are the regular features, the alpha-helices and the beta-sheets. Figure 1.2 illustrates the primary and secondary structure of a protein.

The tertiary structure describes the folding of the mainchain and the orientation of the sidechains and carbonyl oxygens of each of the amino acid residues, i.e., the relative positions of all the atoms. Figure 1.3 shows the secondary and tertiary structure a protein.

The ordinary chain structure is augmented by occasional cross-linking by two different types of bonds between nonadjacent residues: the strong, covalent disulfide bonds that can form between cysteine residues and the considerably longer and weaker hydrogen bonds that can form between various parts of residues or between residues and solvent molecules. Covalent bonds have rigid geometric and physical requirements; hydrogen bonds are much more variable. These two types of bonds play important roles in stabilizing the folding of the protein, helping determine the tertiary and secondary structure.

A protein can then be modeled as a graph whose nodes are labeled with atom types and three dimensional coordinates. This graph has some small cycles (sidechains with cycles) and a few large cycles (closed by disulfide bonds).

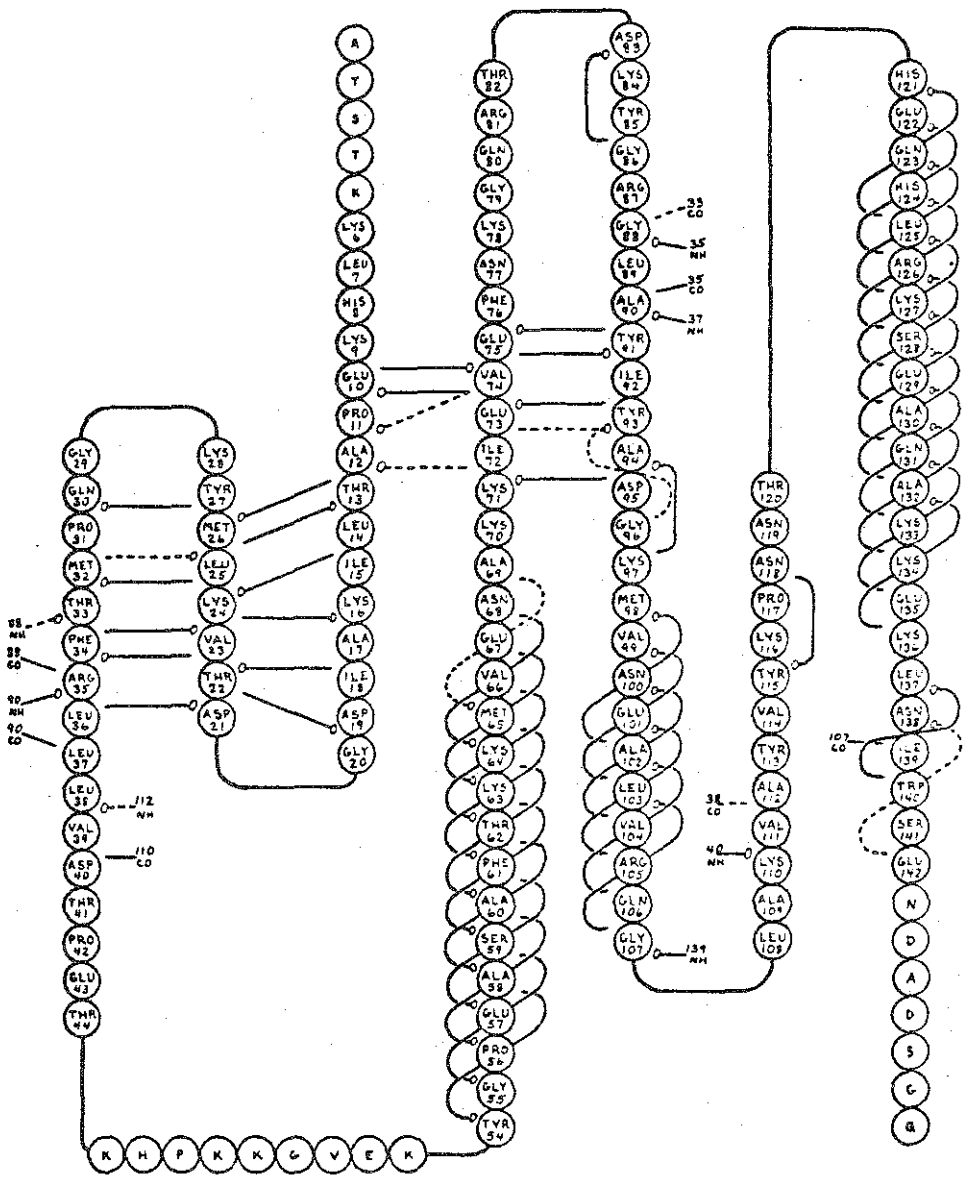

Figure 1.2: Primary and Secondary Structure.

This is a schematic diagram of the protein Staphylococcal nuclease. The circles represent amino acid residues and lines between non-adjacent residues represent hydrogen bonds. The curved short lines are parts of alpha-helices and the straight lines are between strands of beta-sheets.

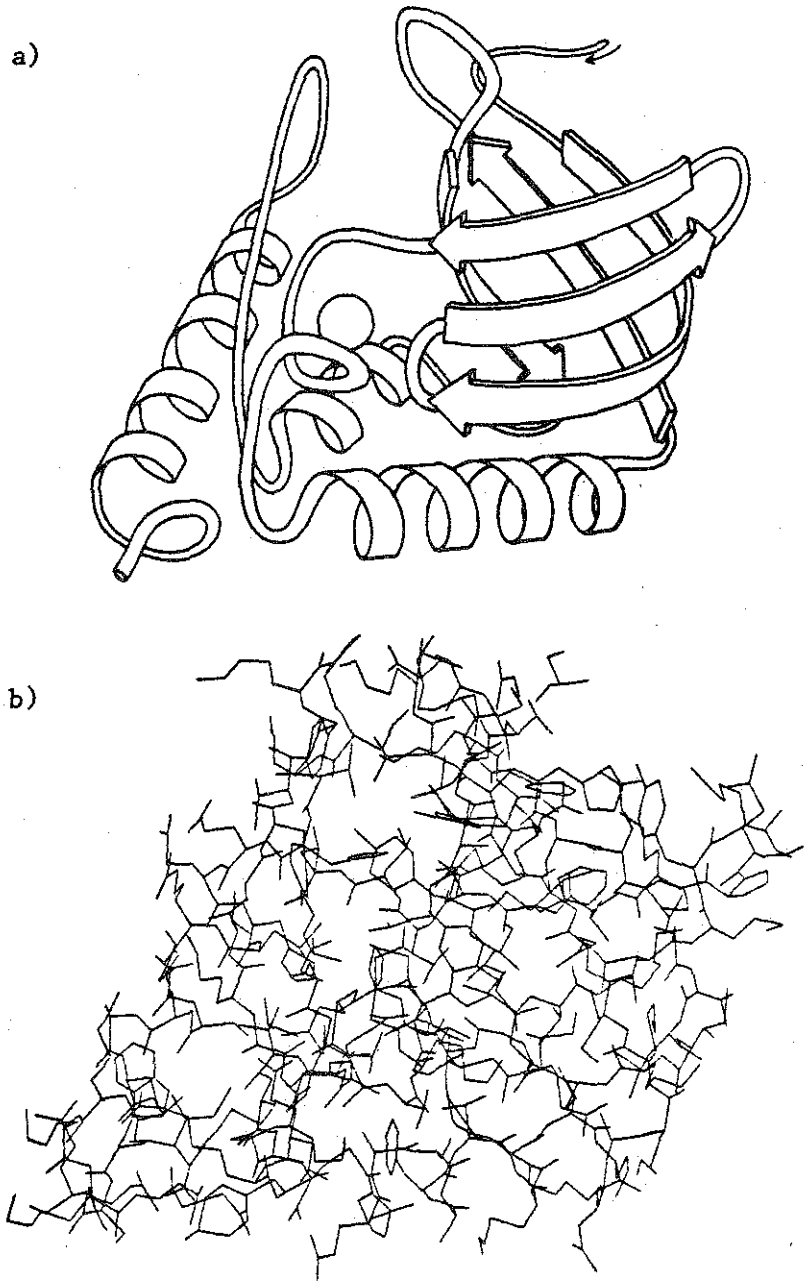

Figure 1.3: Secondary and Tertiary Structure.

a) A ribbon plot of Staphyloccocal nuclease showing only the mainchain. The coils represent helices and the arrows represent beta-sheet. A metal ion is shown by the circle.
b) A stick figure of the molecule from the same view, showing all atoms except hydrogens.

1.1.3 Electron Density Maps

An electron density map gives the time-averaged density of a molecule's electrons as a function of position in three space, usually specified in units of electrons per cubic Angstrom. The discipline dealing with the production of electron density maps is called X-ray crystallography since the maps are calculated from the intensities of X-rays diffracted by crystallized proteins. As a result of many different factors the map is only a fuzzy picture of the molecule. The molecule's structure cannot be determined easily or algorithmically from the map for large molecules such as proteins.

An electron density map is computed by a 3-dimensional Fourier transform of complex numbers called structure factors. The amplitudes of the structure factors are experimentally determined by measuring the X-ray diffraction pattern of the crystallized protein, but the phases cannot be measured and must be estimated. The two most common methods for estimating the initial phases are 1) the method of isomorphous replacement and 2) using phases from a map of a structure for which there are already phases and which is expected to be almost the same as the map in question. The map is calculated on a 3-dimensional grid.

Below is a brief description of crystallization, X-ray diffraction, and isomorphous replacement. For more complete information on any of these steps or the physical processes involved see a standard text on protein crystallography [Blundell76].

Crystallization is a complex process in which many factors must be considered. Although many proteins have now been crystallized it is still largely a matter of lengthy experimentation and empirical knowledge. The crystal is never 100 percent pure protein but contains substantial amounts of the solvent of crystallization and some small but non-zero amount of impurities. Many proteins work with cofactors such as metal ions or prosthetic groups like the heme group in hemoglobin. These small molecules are normally crystallized with the protein and thus show their presence in the electron density map. Crystallization and the preparation of heavy-atom derivatives (discussed below) are the rate-limiting steps in producing electron density maps.

To determine the amplitudes for the Fourier transform, the X-rays are normally produced by a fine-focus X-ray tube and the diffracted radiation is measured on photographic

film or by photon counting. For photography, the spots on the developed film are then measured with a densitometer. For any desired resolution, the number of reflections that must be measured is inversely proportional to the cube of the resolution. Here, resolution means the minimum spacing between Bragg planes for which intensities were measured. A protein, such as lysozyme, with about 1000 non-hydrogen atoms has about 8500 reflections for a resolution of 2.0 Angstroms.

Heavy atom derivatives are used in the method of isomorphous replacement. Heavy atoms such as iodine, mercury, or platinum, are substituted into or added to the protein in such a way as not to disturb its three dimensional structure. This process is also highly empirical in nature. The heavy atoms exhibit an electron density so much higher than the normal atoms of the protein that their positions can be determined by direct methods that were developed for solving small molecule structures, or from an autocorrelation function (a process called Patterson synthesis). The intensities due to the heavy atoms alone are calculated by subtracting the intensities for the protein alone from the intensities for the heavy atom derivatives (protein plus heavy atoms). From these intensities pairs of phases can be calculated. To chose between the two possible phases it is necessary to use either another heavy atom derivative or a technique called anomalous scattering. The selected phases are then used to determine the phases for the protein alone. The phases can be estimated more accurately if more heavy atom derivatives are used.

Errors in electron density maps can be placed in two classes: those that result from the physical processes of crystallization and X-ray diffraction and those that result from data collection and processing. Impurities in the crystal are in the first class of errors. Radiation damage causes several sorts of problems: electrons are knocked loose, altering the structure that is being measured, and heating of the molecule causes increased motion of the atoms. Because X-ray diffraction measures a time average, this blurs the electron density map. Normal thermal vibrations in the protein also cause blurring but this is not an error in the same sense as blurring caused by radiation damage. Errors in estimating phases, digitizing the intensities, and representing the map with a grid of points fall in the second class of errors.

1.1.4 Interpretation

The process of determining the structure of a molecule may be separated into two parts: first, interpretation, in which the goal is to determine which atoms account for what electron density; and second, fitting, in which the atoms are placed so as to match the observed electron density most precisely. Interpretation is usually an iterative process, applying increasingly more precise levels of interpretation to the map. The interpretation of an electron density map is a problem of pattern recognition in the presence of high levels of noise. Consequently, a considerable knowledge of protein structure is essential for successful interpretation. Many molecular graphics systems are designed solely for fitting coordinates and require as input the results of a previous interpretation. In the rest of this dissertation the word interpretation will be used only in the technical sense to refer to interpretation of electron density maps unless otherwise noted.

Interpretation is easiest when done top-down, e.g., when a helix has been located and the sidechains are being sought, since then the task is to find more specific patterns with geometric constraints such as distance and orientation relative to some other known feature. It is especially useful to have registered a chain of residues with the known amino acid residue sequence, i.e., to know the positions of particular residues in the sequence, because one can then look for specific sidechains with known topology instead of a generic sidechain pattern. Quite often, however, the high level structures cannot be readily seen and must be recognized by a bottom-up procedure. For example, some sidechains and carbonyl oxygens may need to be found before it is possible to tell whether a beta-sheet strand is continuing straight ahead or is making a sharp turn. Also, the sequence may be only partially known or not known at all.

It is useful to start interpreting a map by looking for clear, easy-to-interpret features and then expanding the interpretation outward from those regions. Helices are often recognizable because of their size and high regularity. Sulphur atoms in sidechains and metal atoms are easily located because of their high electron densities. In moderately good maps, sidechains with rings are good features to search for. In general, the mainchain of the protein will exhibit higher density values than the sidechains so the mainchain can be more easily traced by only looking at higher density values in the map. Long sidechains have more degrees of freedom than short ones; since the map gives a time-averaged picture, long sidechains tend to be blurred, often to the extent that their density is indistinguishable from noise. Sidechains are more blurred

on the exterior of the protein where they are freer to move. The first and last several residues may also move around quite a bit in the crystal, which blurs them in the map.

The major problems with interpreting electron density maps are that high noise levels cause extra or missing peaks and connections between peaks, that some sidechains and residues appear to be missing because of blurring, and that a global view is needed to resolve ambiguities that appear in a local view of the map.

The expected error level in a mainchain tracing can be estimated from the resolution of the electron density map used [Richardson81]. If the map resolution is poorer than (numerically higher than) 3.5 Angstroms, only occasionally will a tracing be correct and then only for simple, helical structures. At finer resolutions than 2.5 Angstroms all chain tracings can be expected to be correct. Between 3.5 and 2.5 Angstroms the amino acid sequence is of great importance. In cases for which the sequence was known, four-fifths of the tracings were correct; but when the sequence was not known, only one-third of the tracings were correct.

Generally, errors in interpretation are more likely in sidechains than in the mainchain, and more likely on the disordered exterior of the molecule than in the more ordered interior.

1.2 PREVIOUS WORK

1.2.1 Manual Interpretation with Mini-maps

A mini-map is a stack of parallel transparent sheets with contours of an electron density map drawn on them. Interpretation is done unaided by a computer except perhaps for the string-matching problem of sequence registration. The mini-map affords the viewer only one orientation of the map which causes some difficulties. Helices whose axes are aligned with the preferred direction can be easily seen but others may appear very different and be hard to find. This problem of a single view is partially compensated for because the mini-map is a real three-dimensional object with parallax depth cues. As interpretation progresses the positions of atoms along the mainchain and prominent sidechains and carbonyls can be marked on the transparent sheets with pieces of colored tape. When the interpretation has been completed as far as desired or possible, the coordinates of the marked atoms are measured by laying the sheets one at a time on graph paper ruled at the appropriate scale. Between three days and four weeks are required to build a mini-map, interpret it, and produce machine-readable coordinates,

depending on the size and quality of the map and the skill and experience of the interpreter. The lower time requirement applies only to good quality, fine resolution maps of small molecules interpreted in laboratories with special set-ups for producing mini-maps. Interpretation with a mini-map normally yields only C alpha coordinates and perhaps some carbonyl and sidechain orientations but not coordinates for all of the atoms in the protein.

1.2.2 Interactive Interpretation and Fitting

The following subsections are a sample of current interactive graphics systems for interpretation and fitting proteins to electron density maps. They each are missing some capabilities listed below that are fundamental to interpretation and will be described mainly in terms of these capabilities:

1. giving a global view of the map
2. editing the sequence - adding, deleting, and changing residues in the molecular model
3. interpreting portions of the map in an arbitrary order
4. interpreting at different levels of abstraction, not just in terms of residues.

1.2.2.1 BILDER at MRC, Cambridge, England (R. Diamond)

BILDER represents the map by contours, making a global view of the map impossible at working contour levels since too many line segments would have to be drawn. Because only residues in an active range are visible it is difficult to get a global view of the molecule - a residue may be far away down the chain and not show but be close spatially. It is possible to place residues arbitrarily and to edit the sequence but the operations required to do so are complicated; the system works best when adding one residue at a time to the end of an existing chain that has been registered with the sequence. The user guide [Diamond81] suggests that a mini-map be used to find a starting place for interpretation, that is, to locate a particular residue in the map so that the interpretation can proceed by adding specific residues, one at a time. BILDER works only in terms of residues.

1.2.2.2 CRYSNET at Texas A&M (S. Swanson)

CRYSNET was a pioneer molecular graphics system and is small and restrictive by today's standards. The region of map to be displayed, selective display of model, and calculation of contours are done statically off-line. A user can specify interactively which sets of precalculated contours to show and which residue to manipulate. CRYSNET was designed for fitting coordinates; static selection of map volume and contour level does not make initial interpretation feasible. Stanley Swanson added to this system the capability to display ridge lines [Swanson79]. They are for display purposes only, and fitting proceeds in the same way as for contours.

1.2.2.3 GRIP75 at UNC-CH (Britton, Brooks, Lipscomb, Pique, Wright)

The GRIP75 system [Tsernoglou77] [Britton77] [Brooks77] represents the map by contours, making a global view of the map impossible at working contour levels. Atomic coordinates from an initial interpretation and a fixed sequence are required as input. The system was designed for fitting and is difficult to use for interpretation. There is a scheme for interpretation by which residues are moved in from a pool of residues kept in an area far away from the visible area of the map. Residues that have not been "interpreted" yet are off the screen and thus are invisible. Although whole residues must be moved by this scheme, the system can be made to show only the mainchain, giving the illusion that only a section of mainchain has been interpreted and nothing more. Although initial interpretation can be done on this system because of its great flexibility, this is not the best way to do interpretation.

1.2.2.4 FRODO (A. Jones)

FRODO [Jones79] also represents the map by contours, making a global view of the map impossible at working contour levels. It is easy to move along the chain in units of one residue but it is hard to move about randomly in the map. This is fine when the interpretation is clear but not satisfactory when it is not clear and various alternatives need to be explored. FRODO has one advantage over the fitting systems described above in that it can deal with arbitrary fragments of molecules instead of only residues. Thus a pool of residue parts can be kept on the screen to select from as needed. This makes it possible to build just a mainchain and then add sidechains and carbonyls later.

One major problem with this idea is that the fragment must be on the screen in order to be selected. This means the parts pool would have to be manually moved along as the center of attention moved through the map.

1.2.3 Automatic Interpretation

1.2.3.1 Columbia Univ. (J. Greer)

The goal of this system is to automatically produce coordinates for the mainchain. The map is represented by ridge lines calculated by an algorithm developed by Greer [Greer74] [Greer76a]. Ad hoc algorithmic methods are used to locate C-alpha atoms, to bridge gaps in the mainchain, and to discard symmetry-related copies of ridge line segments. A significant amount of manual intervention is required in order to use the system because of the high rate of occurrence of false or missing edges.

After extensive manual correction of connectivity problems, prediction of residues from a 2.0 Angstrom ribonuclease S map missed two residues but found ten extraneous ones out of 114 actual residues. This system has not been worked on since 1976.

1.2.3.2 CRYSALIS at Stanford Univ. (Engelmore, Feigenbaum, Nii)

CRYSALIS [Engelmore77] [Engelmore79a] [Feigenbaum77] is a knowledge-based artificial intelligence system, an attempt to build an expert electron density map interpreter. Humans participate in supplying, assembling, and organizing the knowledge at system building time but the computer actually interprets the map on its own.

A collection of knowledge sources such as biochemical facts and heuristics for interpreting maps has been brought together and encoded in a production rule system. There are three levels of rules: the strategy, task, and knowledge source levels. The strategy level contains broad problem solving strategies, most of which deal with opportunistic growth of interpretation outward from previously interpreted areas. An example would be: if a portion of mainchain exists with uninterpreted map attached to one end, then try to extend the mainchain in that direction. The task level knows which knowledge sources to invoke to solve specific tasks such as how to extend the mainchain. These rules generally work by finding initial evidence for some interpretation, proposing that interpretation, and then

attempting to verify it with supporting evidence. The knowledge sources hold specific pieces of knowledge about map interpretation and model building, such as how to evaluate a proposed sidechain by its shape.

Terry's [Terry80] modification of Greer's algorithm is used to compute the map representation. CRYSALIS often uses a higher level of abstraction of ridge lines called segments. The ridge line graph is cut into segment subgraphs by vertices of degree greater than two. A preprocessor labels the segments by functional type (sidechain, mainchain, etc) as an initial guess before CRYSALIS begins interpretation.

The success of the system has been only fair; it could locate only 22 out of the 55 residues [Engelmore79b] in a very high quality rubredoxin map [Watenpaugh79]. The major difficulty is a lack of a global view. The system is using local information to trace the chain and this is often insufficient where ambiguities arise. It is also difficult to find good starting places without the overview capability that humans possess. The knowledge and decision processes required to resolve ambiguities in the map are too complex for the state of the art in artificial intelligence and may be so for some time to come.

Chapter II

DESIGN DECISIONS

The design decisions made for this system fall into three areas:

1. the roles of man and machine,
2. the representations of the electron density map and the molecule, and
3. the communication between man and machine.

The decisions in each of the representation and communication areas are fairly independent of decisions in the other area but are highly dependent on the decisions about the roles of man and machine.

2.1 ROLES OF MAN AND MACHINE

Five plausible combinations of roles for man and machine have been identified:

1. artificial intelligence - machine alone
2. computer uses human as subroutine for specific tasks
3. coroutines
4. human uses computer as subroutine for specific tasks
5. manual - man alone

All of these approaches except coroutines have been tried in the past and examples of these approaches are discussed in chapter 1. The CRYSALIS system [Engelmore79a] is an artificial intelligence approach; Greer's system [Greer74] is essentially a computer-driven system with human intervention required to fix errors; interactive molecular graphics systems [Tsernoglou77] are human-driven systems where the computer is requested to do specific tasks; the mini-map method is a manual system.

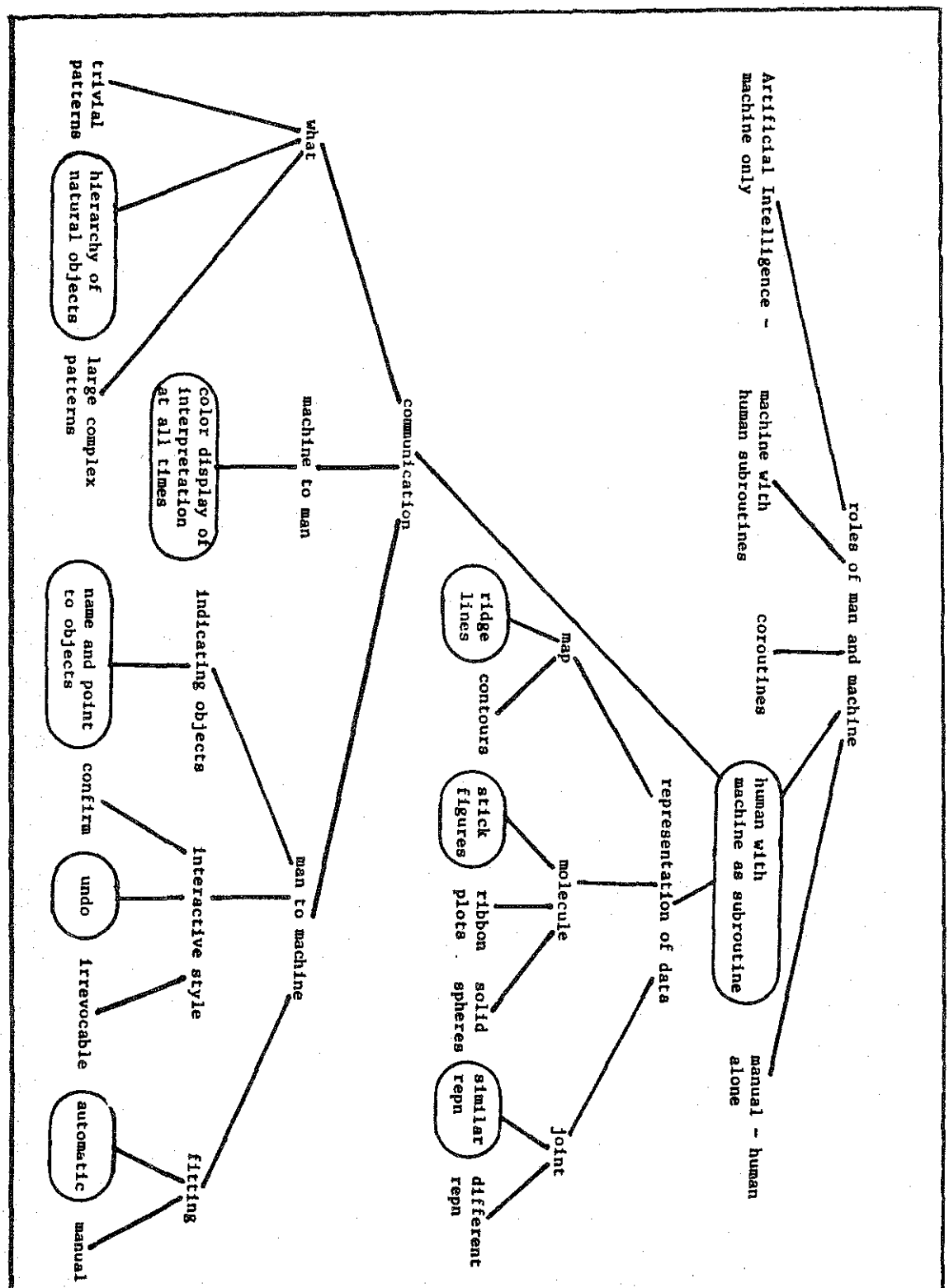

Figure 2.1: Tree of Design Decisions.

The circled options were chosen.

The system that I have designed is a human controlled interactive system of the fourth type. The user of the system provides strategy decisions and does the pattern recognition tasks necessary to interpret the map. The computer stores and displays the map and its interpretation. It does limited pattern recognition so that the man-machine communication can be at a reasonably high level.

The thesis demonstrated by this dissertation is that for some problems, especially those that require global perception, a man-machine system can be effective where automatic, machine-only systems have been ineffective and manual, man-only systems are too slow.

In an artificial intelligence approach the computer attempts to interpret the map unaided. Human experts supply the computer's knowledge base in advance, but they give no help during interpretation. Some of the top researchers in the artificial intelligence field have been working on a solution for six years with no real success - only 40% of a refined, very high resolution map can be interpreted [Engelmore79b]. I think that the pattern recognition problem of map interpretation is so hard that there will be no successes with this approach for some time to come. The basic difficulty is a lack of the global view of the problem which is needed both to resolve ambiguities that appear in a local view and to correct errors that appear to be quite reasonable interpretations from a local view.

Both the approach where the computer uses a human as a subroutine to do specific tasks and the coroutine approach allow human interaction but only on demand from the computer. If the computer makes a mistake it may build quite a bit of interpretation on that mistake and perhaps never discover what the mistake was. In the coroutine approach the human can at least correct the error even though the cost may be high. (An interesting modification to the coroutine approach is one where the human may interrupt the computer if it appears to be making a mistake.) The problem with these two approaches is again that the entity driving the interpretation is not the one with a global view of the data.

Interactive molecular graphics systems are examples of the approach where a human uses the computer as subroutine to do specific tasks. The GRIP75 system [Tsernoglou77] [Britton77] has demonstrated well that a system of this type can be effective for adjusting protein structures to fit electron density maps, yielding an estimated two- to sixteen-times speedup over the use of traditional manual methods. It is reasonable to assume that a similar choice of man-machine roles could be effective for solving the similar problem of initial interpretation, and it is this approach that I have adopted.

The manual mini-map method certainly works and is the only method in use for initial interpretation. It is slow and tedious, however, and improvements similar to those made for fitting would be welcomed.

2.2 REPRESENTATIONS FOR MAPS AND MOLECULES

Representations must be chosen for both the input electron density map and the molecule whose structure is being determined. For an interactive system there are also questions about the perception of both representations together.

2.2.1 Representations of Electron Density Maps

2.2.1.1 Contours

Contour representations of electron density maps delineate surfaces in the map defined by constant values of the electron density. The contours are computed on planes and a series of parallel planes is needed to show the surface in three dimensions. Where the contours are drawn on only one series of parallel sheets of a clear medium, the map is often contoured at multiple levels to give additional information on the gradient of the density. Computer displays of contours often draw three series of planes: those perpendicular to the x, y, and z axes. These are customarily done at a single contour level because of the difficulty of understanding more. The regions of high density appear to be surrounded by wire cages and, because of the tree structure of the underlying molecules, these cages form a network of tubes. Almost all molecular graphics systems use contour representations exclusively. The most informative spacing between planes is half the map resolution. A closer spacing may appear to be more useful for viewing; contouring on parallel transparent sheets seems to need a spacing of 1/3 to 1/4 of the resolution. When contours are drawn in three series of planes and can be viewed from any angle, a spacing of 2/5 of the resolution suffices.

A major difficulty with using contours is the many line segments required to represent the map adequately. This is of special concern during interpretation when a large overview of the map is needed. A large portion of the surface of an object must be known to estimate well the distribution of density in its vicinity. Two problems arise from this. First, the number of line segments in a volume of map needed for interpretation is so high high that most line-drawing displays cannot draw them without flicker. Second, the image becomes complex and it becomes hard for a viewer to

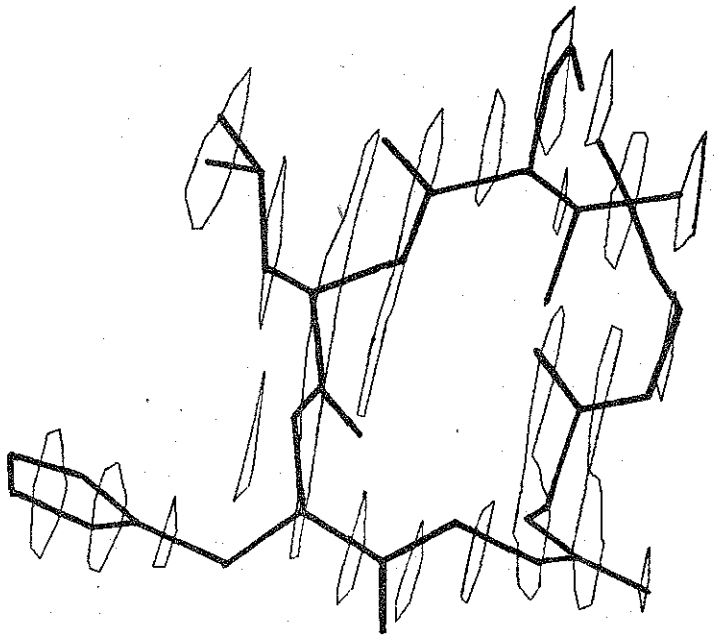

Figure 2.2: Contours and Molecular Model.

A portion of an electron density map of a protein contoured at a single level. The thicker model lines generally pass through the cycles of contour lines.

assign lines to the front or back surfaces of an object. This latter problem is not alleviated by faster hardware; it is inherent in the contour representation.

In molecular graphics systems the most common representation used for the molecular model is a stick figure where the lines and their intersections represent bonds and atoms respectively. Because the presentations of the map and model are different, confusion between them is minimal. However, the major task is to match the model to the map. The atoms of the molecule are presumably located at peaks in the electron density map. A viewer of a contour representation must interpolate to estimate the positions of the local maxima of the map. The value at a maximum must be estimated from its distance from the surface. For both the position

and the value of a maximum, assumptions must be made about the distribution of density within the region of high density surrounded by the contours.

Changing the contour level is not a trivial task, since a completely different set of line segments must be displayed for a different contour level. Either one computes contours for many levels in advance, recomputes them from scratch on demand, or uses the concept of locality to help recompute contours at a level near the level last used. These approaches are expensive in space, time, and complexity respectively. Usually only the second approach is taken, which means that the contour level cannot be changed rapidly. Much extra information about the distribution of electron density can be seen if the contour level can be changed rapidly.

2.2.1.2 Ridge Lines

Ridge lines in electron density maps are named in analogy to geographic terrain. Terrain can be described by a height function h(x,y). Three types of critical points are possible for a function of two variables: local maxima, saddle points, and local minima. These critical points in terrain are called <u>peaks</u>, <u>passes</u>, and <u>pits</u> (see figure 2.3). Ridge lines are paths of steepest ascent from passes to peaks. At any point on the ridge line except the passes or peaks the maximum gradient points the way up the ridge to the peak. Along the direction perpendicular to the maximum gradient, the point is a local maximum.

An electron density function can be given as d(x,y,z). With the added dimension there comes a second type of saddle point which Carroll Johnson [Johnson78] has named a <u>pale</u>. The four classes of critical points may be distinguished by examining the signs of the eigenvalues of the Hessian matrix at the point as shown in table 2.1.
The definition of ridge lines remains the same as for the two-dimensional case.

If contours may be said to describe tubes surrounding volumes of high density, then ridge lines run through the centers of these tubes. The peaks correspond roughly to atoms and the ridge lines to bonds as shown in figure 2.4.

One advantage of a ridge line representation of d(x,y,z) over a contour representation is this close correspondence between the representation of the map and a familiar representation of the molecule. If one is looking for peaks in the density and how these peaks are connected, the information is immediately available. Another advantage is the few

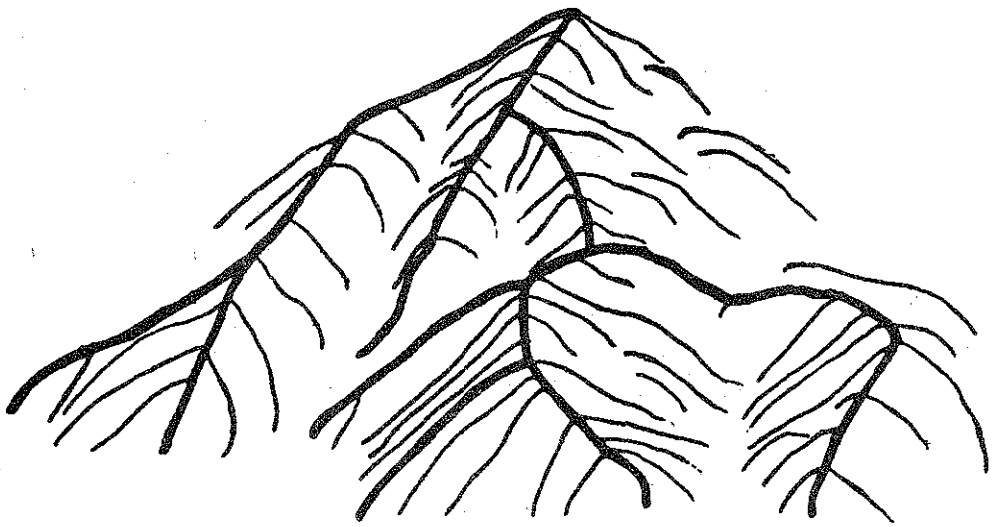

Figure 2.3: Peaks, Passes, and Ridge Lines in Terrain.

The thick lines are ridge lines.

line segments required to display the ridge lines. This means that with a line-segment-limited display one can view a much larger volume of the map and get a better overview.

By representing each curved ridge line by a series of short straight line segments and by assigning to each line segment an average density, it is easy to display ridge lines selectively based on density value, or to color them based on density value. A particularly useful way of doing this is to show only those segments whose density is above some threshold. With a computer graphics display it is possible to change this threshold dynamically, thus providing the equivalent of a dynamically changeable contour level. This capability is quite valuable for interpreting electron density maps.

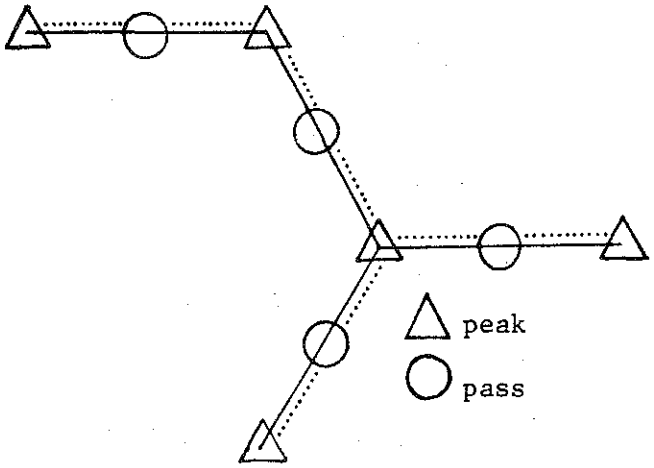

Figure 2.4: Critical Points for a Molecule.

The dotted lines are ridge lines.

eigenvalue signs	type of critical point	name
- - -	local maximum	peak
+ - -	saddle point	pass
+ + -	saddle point	pale
+ + +	local minimum	pit

TABLE 2.1

Types of Critical Points

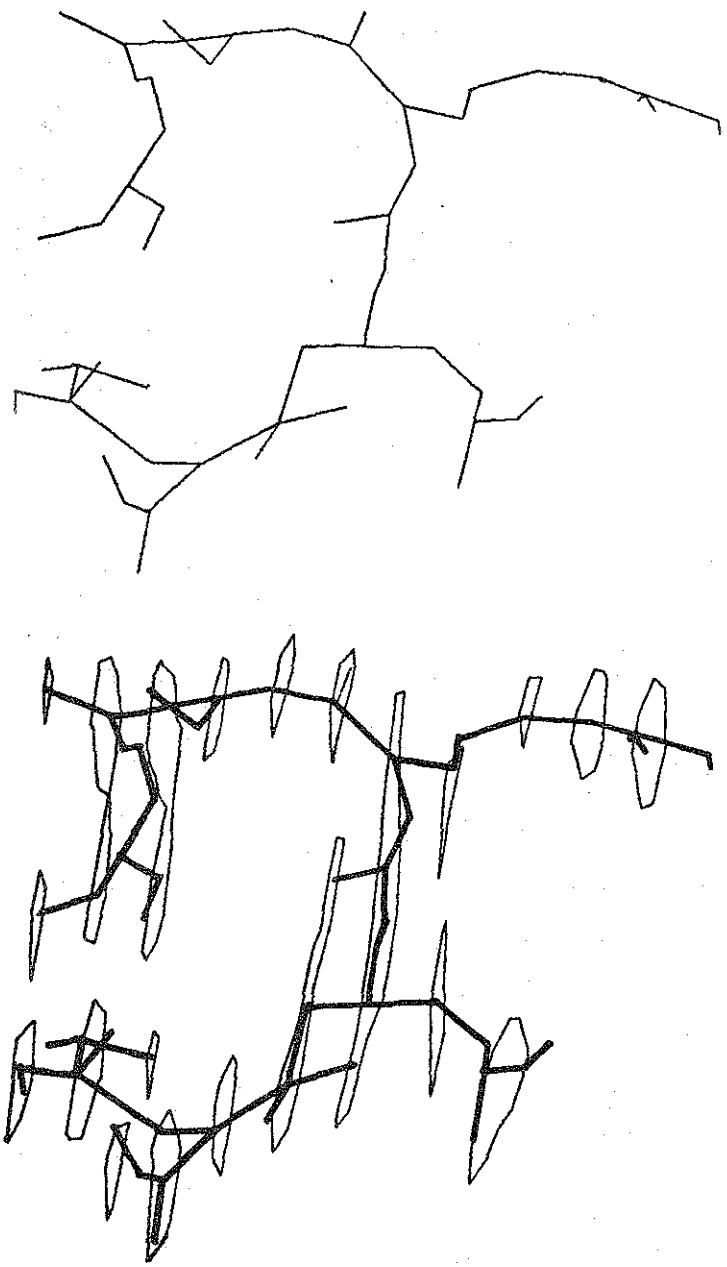

Figure 2.5: Ridge Lines.

a) ridge lines for a portion of a map.
b) a comparison of ridge lines and contours.

2.2.2 Representation of the Molecule

There are three main possibilities for representing the molecule:

1. a stick figure with line segments for bond and joints for atoms.

2. a surface, using either spheres for atoms or a surface accessible to a probe sphere.

3. higher level abstractions such as ribbon plots

Stick figures have several advantages: they are familiar to biochemists, inexpensive to compute and display, and can easily show an abstraction such as only the mainchain over a large volume.

A surface or solid sphere model takes much longer to display and rules out smooth motion if large numbers of atoms must be shown.

A representation such as a ribbon plot might be useful during the first stage of interpretation but could not be used throughout the interpretation because all the atoms of the molecule are not represented. Sidechain atoms could not be positioned because the model does show that level of detail.

2.2.3 Joint questions of Representation

A question to be considered in addition to the independent choices of map and molecule representations is whether those two representations should be similar in appearance or not. The advantage of dissimilar representations is that the map and molecule are easily distinguished. One advantage of similar representations is that the user need think of only one type of representation. If, for instance, a stick figure molecule representation was used with a contour map representation, then one must mentally match these two representations - either imagining the volume occupied for each atom in the stick figure representation or imagining where the local maxima are inside the contours. Using similar representations such as stick figures and ridge lines avoids this mental effort. Some other advantages are (1) the map may be easily transformed into the molecule as the interpretation progresses, and (2) the map and molecule representations can be managed by the same computer procedures.

The stick-figure and ridge-line representations were chosen because of their individual merits as well as the advantages of using similar representations for map and molecule. An attractive bonus for these particular representations is that they are both graphs and graphs can be easily manipulated by computers.

Once similar representations have been chosen it is essential that the user be able to distinguish between the map and molecule. The two main choices here are color and line quality or texture. Of these two, color is much superior and should be used to distinguish between the two representations if at all possible.

2.3 COMMUNICATION

There seem to be few reasonable options in this area once decisions have been made in the other areas, hence this section will, for the most part, present only the methods adopted.

2.3.1 What to Communicate

As the goal of this system is to recognize the correspondence between parts of the map and some model of a molecule it seems natural that the communication should be in terms of this molecular model. A model of a protein naturally divides into a hierarchy of objects such as amino acid residues and sidechains (see chapter 1). These objects can be used both to describe parts of the model being built and to show what parts of the map correspond to them. Although it might occasionally be fruitful to bypass levels in the object hierarchy, its presence adds useful structure to the process of interpretation.

If the communication is to be in terms of these objects, then the computer must be able to recognize patterns for these objects. However, the difficult job of locating the object in a noisy environment taking global information into account will have been done by the user; the computer need only match a relatively simple pattern at a location known to contain an object that matches the pattern.

The complexity of the patterns matched by the computer during this object finding affects the character of the system. If the patterns are trivial, matching only individual edges in the map, then progress in interpreting the map will be slow, not only because many interactions are required, but also because the user would be forced to think

and work on a lower level than is natural to the problem. At the other end of the spectrum, the computer-sought pattern could be complex enough to match the whole protein molecule, effectively removing the user from the interpretation process. Between these two extremes there is a more reasonable middle ground in which some tradeoffs exist. Computer matching of more complex patterns will allow the user to interpret larger portions of the map with each interaction, but in the presence of noise, more complex patterns are less likely to match exactly what the user has in mind. Simpler patterns may require fewer interactions because each interaction is more likely to have the desired effect and not force additional interactions to make corrections. Making these corrections also interrupts the user's thought process and may reduce his confidence in the system. A range of patterns over this spectrum gives the user flexibility in specifying objects to the computer. At times, low-level corrections to the interpretation may be required and at other times it may be reasonable to specify high-level objects.

2.3.2 Man to Machine Communication

The user always determines what object is to be communicated to the machine, both what kind of object and where the object is. In this system these two kinds of information are each always communicated in exactly uniform ways, first naming the type of object by selecting from a menu of names and then pointing to the object with the map. Using a menu relieves the user of memorizing a list of commands and their spellings. The menus are tree structured so that at any state all then legal commands and operands are visible.

Selection of both menu items and map objects uses a visible cursor on the graphics screen controlled by a physical cursor on a data tablet. When selecting a menu item, the motion of the visible cursor is one-dimensional and discrete, the cursor always resting on a menu item. Pushing the button on the physical cursor selects that item. When selecting a map object, the motion of the visible cursor is two-dimensional and continuous. The cursor is not three dimensional because it is difficult both to see and to manipulate a cursor in three dimensions. Pushing the cursor button selects the map object that is closest to the cursor in the plane of the screen. Because there is visual and tactile continuity, and because the visible cursor behaves naturally in the menu and map object spaces, the user can concentrate on viewing and interpreting the data without the concern of having to determine what actions are necessary to make the system behave in the desired manner.

The same principle is applied to controlling the density threshold and viewing angle of the map. The density threshold is controlled through a slide potentiometer such that the threshold value increases as the control is moved up. The numerical value of the threshold is displayed so that the threshold can be accurately reproduced by the user. An analog joystick with three nested potentiometers is used to control the viewing angle so that the user can smoothly rotate the picture. The computer rotates the picture in the same direction and the same amount as the user rotates the joystick, so that it feels to the user as if the joystick is actually physically attached to the map and he is actually rotating a real object.

Options for the interactive style of this system are:

1. irrevocable commands,
2. tentative commands requiring confirmation before execution, and
3. immediately executed commands with an "undo" capability.

The problem solved with this system is a difficult one in which many ambiguities arise. It is very useful to be able to try alternatives, exploring the ramifications of each alternative to help determine the correctness of that choice. Based on this observation, the first alternative is immediately rejected. The third choice is chosen over the second because the "undo" capability is useful even if no obvious mistakes are made and because fewer interactions are required - every command does not demand a bothersome confirmation. If a mistake is made, it is trivial to backup. The current implementation allows the user to "undo" up to the last twenty commands and to reverse the effect of any immediately preceding "undo" with a "redo" command.

The user only points to parts of the map that correspond to the molecular model and never to the molecule itself - there is no manual manipulation of the model. The position of the molecular model is determined by the computer and can be expected to be close to the desired position. Fine positioning of the model is not appropriate to initial interpretation, where some parts of the interpretation are often quite tentative until the interpretation is nearly complete. Also, there are many existing systems for fitting which are specifically designed for fine positioning, making it unnecessary in this system.

I would contend that it is not only unnecessary but even undesirable to put positioning capabilities into an interpretation system, because:

1. Initial interpretation and fine-adjustment fitting are separable tasks. Interpretation determines which atoms account for what electron density; fitting determines the atom positions that most precisely match the observed electron density. Separating these tasks allows two systems each simpler than a combined system. Simple systems have the benefits of ease of learning and use, ease of implementation, and a higher probability that the implementation is correct and robust.

2. Much of the interpretation is tentative during much of the process. It is counterproductive to make fine adjustments to an interpretation that is incorrect and will have to be discarded.

3. I have observed on the GRIP-75 interactive graphics system that the mere presence of fine adjustment facilities does in fact lure users to unproductive use of those facilities even after repeated advice to the contrary. If the function of adjustment is not necessary it should not be provided.

2.3.3 Machine to Man Communication

The computer displays the molecule and map as a stick figure model and ridge lines. As the map is interpreted, the color in which edges of the ridge line graph are displayed is used to show to what type of object that part of the map corresponds, e.g. mainchain or sidechain. When a new object is identified to the computer by the user, the appropriate edges of the map are recolored to reflect the new interpretation of that part of the map, giving immediate feedback to the user. Only visible edges of the map are ever changed; edges that are invisible because of scale or selective display options cannot be changed, ensuring that the feedback is accurate as well as immediate. The entire interpretation of the map is contained in the color-encoded assignments of object to the map and in the molecular model that has been built. Thus for the volume of map displayed, the current interpretation is always visible.

Of the many techniques for displaying a three-dimensional object on a two-dimensional device, the single most effective aid to 3-D perception is the kinetic depth effect derived from smooth motion of the object [Lipscomb79]. It was the cheapest method to implement on the available display hardware, costing nothing in terms of the number of line segments that could be displayed, and is the only 3-D perception aid used in this system.

Chapter III

SYSTEM ARCHITECTURE

3.1 OVERVIEW

The problem of interpreting an electron density map is one of pattern recognition, trying to see the molecule by looking at a picture of the electron distribution. In this system the molecule is represented by a stick figure model and the map is represented by ridge lines. The ridge lines are computed by an offline process prior to interpretation; the molecular model is the goal of the system and is built during interpretation.

The map is interpreted by locating objects in the ridge lines. There is a hierarchy of object types, natural to the problem area: chains, residues, and segments. Each object is constructed of objects at the next lower level in the hierarchy, except for the lowest level objects which are constructed directly from edges of the ridge line graph. These objects are not parts of the molecular model itself, but are parts of the ridge line representation of the map which correspond to parts of the molecule. Segments correspond to parts of an amino acid residue such as sidechains or carbonyl oxygens and are built directly from edges of the map's ridge line graph. A residue is built of a sidechain segment, a carbonyl segment, and part of a mainchain segment. When a residue object is identified and an amino acid type is assigned, the computer automatically supplies a stick figure model of that amino acid type, positioned by a least squares fit to the residue object. No manual manipulation of the molecular model is required or allowed. A chain is a series of adjacent residue objects that have been registered with the known amino acid sequence.

The human user of this system controls the interpretation, supplying the strategy decisions of what region of the map to examine next and what type of object to look for. The user is also responsible for locating the objects and pointing them out to the computer. Since the user only names and points to objects, the computer must also do pattern recognition but this involves only local, anchored pattern matching; the computer is given the exact type of object that must be matched and a lower-level object that

must be part of the object being sought. Initially the computer displays the ridge line graph as white line segments. When it locates an object, the line segments that belong to the object are colored according to the type of the object. Thus the current interpretation is always highly visible. The user gradually builds up the interpretation iterating the sequence of viewing the map and its current interpretation, recognizing objects in the map, and informing the computer of the type and location of the object.

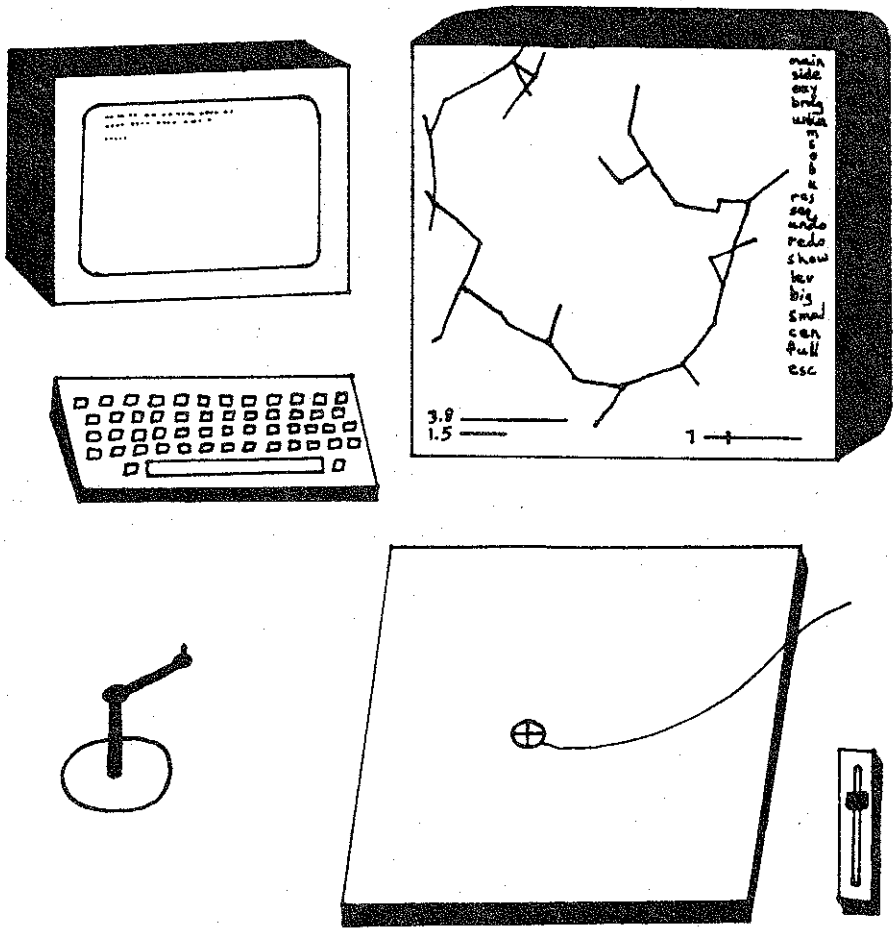

Figure 3.1: Physical Components of the System

3.2 COMPONENTS

The physical components of the present system are a graphics screen, a keyboard and text screen, a joystick, a data tablet cursor, and a potentiometer. The last three components are controls that affect the system in real time, that is, potentially every 1/30 of a second. These controls are said to be dynamic, as opposed to interactive events that happen only after explicit discrete actions by the user.

3.2.1 Graphics Screen

The graphics screen is divided into three areas: graph, menu, and auxiliary information, as seen in figure 3.1. Only for purposes of selection from the screen is there a distinct boundary between the graph and menu areas. Some edges of the graphs may spill over into the menu area, but those edges may not be selected.

3.2.1.1 Graphs

The edges of the stick-figure model of the molecule and the ridge line representation of the electron density map are displayed in the graph area. The volume of the map space that is visible is always a cube and will be referred to as the view cube. The same right-handed coordinate system is used for the map and model. The coordinates of the vertices of the ridge line graph are fixed when the ridge lines are initially computed and these coordinates never change. The color in which an edge is drawn depends on its interpretation, i.e., the type of object (see section on object selection) to which it belongs. The user may vary the center and size of the visible cubic volume of the map space and the minimum density level which an edge may represent and still be visible. Edges from the graphs can be selected as operands to some of the commands.

3.2.1.2 Menus

There is one menu of commands and several menus of operands. After selecting a command from the menu, the appropriate menu of operands is made available for selection. Commands for specifying objects are displayed in the colors for those objects. User-supplied file names and coordinates are not on menus and must be entered on the keyboard.

3.2.1.3 Visible Screen Cursor

The visible screen cursor echoes the position of the data-tablet cursor for selecting items from menus or edges from graphs. When the cursor is in the graph area of the screen, the visible cursor appears as a small square which can be moved continuously in two dimensions. When the cursor is in the menu area of the screen, the visible cursor appears as a small diamond which can be moved discretely in one dimension among the vertical positions of menu items. The visible cursor changes to a small triangle for a short time after a menu item or graph edge is selected. The cursor is normally colored white. Between the times when a command is selected and when the command has been completed, the cursor is displayed in the color in which the command appears on the menu.

3.2.1.4 Auxiliary Information

At the bottom left of the screen are two horizontal reference lines: one that is 1.54 Angstroms long (the length of a C-C bond in a sidechain) in map space and one that is 3.8 Angstroms long (the distance between C alpha atoms of adjacent residues). At the bottom right is a numerical value and a scale with a marker that show the current density threshold for the map. Warning and error messages are displayed along the bottom of the graphics screen.

3.2.2 Viewpointer Joystick

The user can control the orientation of the graphs by manipulating a 3-dimensional joystick [Pique80] [Britton78]. The joystick is operational at all times except during the time when a command is being executed. The kinetic depth effect produced by rotating the graphs is the only depth cue provided by this system.

3.2.3 Cursor Movement and Selection

Menu items and graph edges are selected by moving the physical cursor so that the visible cursor is correctly positioned and then pushing the button on the cursor. When in the menu area, the cursor is always correctly positioned to select a menu item since the visible cursor is discrete in the menu area. In the graph area, pushing the selection button will select the edge whose midpoint is closest to the visible cursor and that is within 0.70 Angstroms (in map space) of cursor.

3.2.4 Density Level Control

The user can dynamically change the density threshold so that only map edges with associated electron density at or above that threshold are displayed. The threshold is controlled by moving a slide potentiometer, with the limitation that the threshold can be made no lower than a value set by the LEVEL command (this limitation exists only for implementation performance reasons).

The current threshold is displayed in the auxiliary information area at the bottom right of the graphics screen.

3.2.5 Text screen and Keyboard

Any of the commands and operands that can be selected from a menu can also be entered on the keyboard. User-supplied file names and coordinates are not on menus and must be entered on the keyboard. Warning messages are printed in full on the text screen in addition to the abbreviated messages displayed on the graphics screen. Other information too voluminous for or inappropriate for the graphics screen is also displayed on the text screen.

3.3 COMMANDS

The commands can be divided into four classes: control flow, selective display, object specification, and miscellaneous. Only the object specification and miscellaneous commands are particular to this system. The rest are commands that would be found, in one form or another, in any interactive graphics system.

The following notation is used to describe the syntax of the commands. An asterisk (*) after an item means zero or more of the items may occur. A plus (+) after an item means that any number, but at least one, of the items may occur. A list of items in between square brackets [] means that one item must be selected. Upper case items are fixed keywords. Lower case items refer to user-supplied names and numbers. Items with the word edge in them must be supplied by selecting edges from the ridge line graph.

Control Flow	Object Specification
LOAD	BRIDGE
SAVE	B
UNDO	MAINCHAIN
REDO	M
QUIT	OXYGEN
	O
Selective Display	SIDECHAIN
WIDTH	S
BIGGER	UNKNOWN
SMALLER	U
CENTER	RESIDUE
WINDOW	SEQ
SHOW	
LEVEL	Miscellaneous
	ADD
	DELETE
	DISTANCE

TABLE 3.1

Commands

3.3.1 Control Flow

LOAD file [YES NO]

The system is initialized with the contents of the named file. A file contains the map graph and its interpretation, the model graph, and display information (i.e., center, scale, density threshold). An initial file containing an uninterpreted map, a null model, and default display information is prepared by an off-line batch process. If the interpretation has been changed since the last SAVE command, a warning is given and a confirmation of yes or no must be given.

SAVE file

The current map, model, and display information is stored in the named file. If the file already exists, its contents are overwritten. Otherwise a new file is created.

UNDO

This command reverses the effect of the last command issued so that the map, model, and display information is the same as it was before the last command was issued. If another UNDO command is immediately given it does not "undo" the first UNDO command but operates on the command two commands before the first UNDO. LOAD and SAVE commands cannot be "undone". Up to twenty consecutive UNDO commands may be given. In the following examples of command sequences "=" means "is semantically equivalent to" and A, B, C are commands.

```
A UNDO B UNDO C = C
A B UNDO UNDO C = C
A UNDO B C UNDO = B
```

REDO

The REDO command reverses the effect of an immediately preceding UNDO, making everything as if the UNDO had not been issued. If another REDO command is given it operates on the UNDO before the last UNDO.

```
A B UNDO REDO = A B
A B UNDO UNDO REDO REDO = A B
A B UNDO C UNDO REDO = A C
A UNDO B REDO = error
```

The last sequence of commands is not valid because REDO commands operate only on immediately preceding UNDO commands. There is no way to go back and "redo" the "undone" A command after the B command has been issued.

QUIT [YES NO]

The QUIT command stops the system. If the interpretation has been changed since the last SAVE command, a warning is given and a confirmation of yes or no must be given.

3.3.2 Selective Display

WIDTH number

The volume visible of the map space is always a cube and will be referred to as the view cube. The width of the view cube is set to the given number, which is in

Angstroms. The width refers to the length of an edge of the view cube. If the width is given as 10.00 then the view cube will be 10x10x10 cubic Angstroms.

BIGGER
The map and model graphs are made to appear bigger by decreasing the width by 20 percent.

SMALLER
The map and model graphs are made to appear smaller by increasing the width by 25 percent. Note that a BIGGER SMALLER or a SMALLER BIGGER sequence leaves the width unchanged.

CENTER edge
The center of the view cube is set to the midpoint of the selected edge.

CENTER number number number
The three numbers are the x, y, and z coordinates in Angstroms of the point that becomes the center of the view cube.

WINDOW
The center and width of the view cube is set such that all the map and model is in the view cube and is displayed as large as possible.

SHOW [ALL MODEL FIT UNFIT UNKNOWN COLORED MAINCHAIN]+
ENDSHOW
Any number, but at least one, of the options between the brackets may be selected. The effects of the options are additive, each specifying a portion of the map or model from the current view cube for display.

ALL - show all the map and model

MODEL - show all the model

FIT - show map edges that belong to residue objects

UNFIT - show map edges that do not belong to residue objects

UNKNOWN - show map edges that do not belong to any object

COLORED - show map edges that belong to any object

MAINCHAIN - show mainchain model and map edges

FIT and UNFIT are a complementary pair of options that specify disjoint portions of the map and that together specify the whole map. UNKNOWN and COLORED are another such pair of options. The MAINCHAIN option is different from all the options except ALL in that it specifies portions of both the map and the model. This option is provided as a convenience for viewing the mainchain tracing only.

LEVEL number

The minimum density level threshold is set to the specified number. Map edges that are in the view cube but have an associated electron density value less than the threshold are not displayed. The dynamic level control can make the effective threshold larger than but not smaller than this threshold.

3.3.3 Object Specification

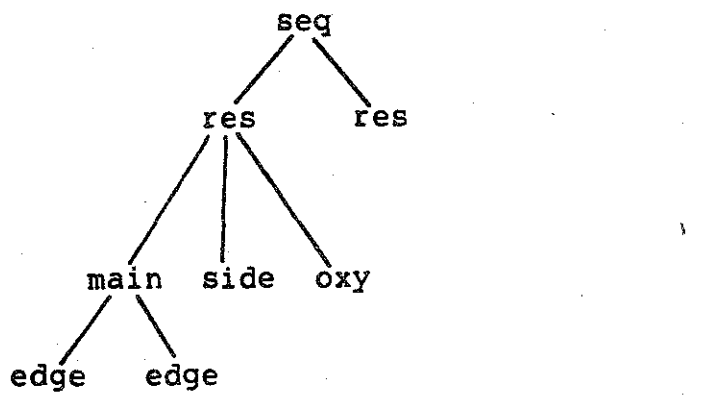

Figure 3.2: Hierarchy of Objects

There is a three-level hierarchy of objects: sequence, residue, and segment. Sequences are constructed of residues, residues are constructed of segments, and segments are constructed directly from map edges. Each object is ultimately a connected graph of map edges. Edges that belong to no object are said to be of type "unknown".

Each object specification command instructs the computer to find an object of the specified type containing the selected edge or edges. The computer finds the object by matching a local pattern in the map graph. Only edges visible to the user may be matched by the computer. The edges matched by the pattern are made to belong to an object of the specified type and are displayed in the color of the type of segment to which they belong. If a newly created segment is adjacent to (shares a node in the map graph with) another segment of the same type, the two segments are fused into a single segment. The colors of the segment types are given in table 3.2. The model is displayed in blue.

The single letter commands operate on lists of single edges, treating each as an object, and the multi-letter commands match more complex patterns. Since adjacent

SEGMENT TYPE	COLOR
mainchain	green
sidechain	violet
oxygen	red
bridge	brown
unknown	white

TABLE 3.2

Segment Colors

objects of the same type are fused, the single-letter commands can be used to add edges to segments by specifying edges adjacent to the segment. They can be used to remove edges from a segment by specifying another segment type for those edges. It may be convenient to specify whole segments by enumerating the edges of which they are composed. The multi-letter objects always specify entire objects.

MAINCHAIN map_edge

This command extends an existing mainchain segment. The mainchain object matched is the shortest path of unknown edges from the selected edge to a visible end of an existing mainchain object. If no path is found then the pattern matches just the selected edge.

M map_edge+

The selected edges are made members of mainchain segments.

SIDECHAIN map_edge

The largest connected subgraph of edges of type unknown that contains the specified edge is found and treated as a tentative pattern. If the tentative pattern touches only one mainchain segment (see figure 3.3) then the pattern is a sidechain segment and is defined

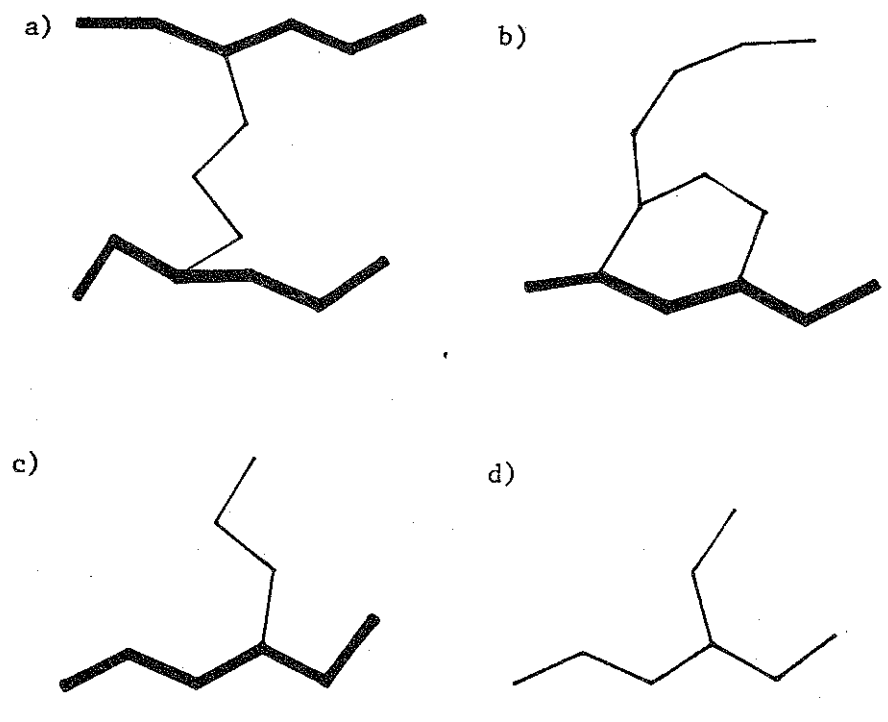

Figure 3.3: Examples of Sidechain Objects.

The thick lines represent mainchain.
In a) and b) the sidechain objects touch mainchain twice,
in c) once, and in d) not at all.

and displayed as such. If tentative pattern touches mainchain segments in two places the pattern is rejected and an error message is displayed. If the pattern does not touch any mainchain segment, the pattern is cut into two subgraphs at the selected edge, and only the selected edge and the smaller subgraph are defined and displayed as a sidechain segment. The pattern is rejected if a path of length greater than 14 Angstroms exists in the pattern.

S map_edge+
The selected edges are made members of sidechain segments.

OXYGEN map_edge
This command works just as the SIDECHAIN command except that the segment becomes a oxygen segment and oxygen segments may be up to 4 Angstroms long.

O map_edge+
The selected edges are made members of oxygen segments.

UNKNOWN map_edge
The segment (but not the edges of which it is composed) that contains the selected edge is deleted and all the edges that belonged to it are then said to be of type unknown.

U map_edge+
The selected edges are removed from membership to any objects to which they belong.

BRIDGE map_edge
The selected edge (that must be of type unknown) is made a member of a bridge segment.

B map_edge+
The selected edges are made members of bridge segments.

RESIDUE residuetype map_edge1 map_edge2
A residue object is composed of a sidechain segment, an oxygen segment, and a portion of a mainchain segment. Map edge1 designates the sidechain segment and normally belongs to a sidechain segment. Map edge2 designates the oxygen segment and normally belongs to an oxygen segment. The mainchain segment that connects the sidechain segment and the oxygen segment is inferred by the

computer. If the sidechain segment is missing then map edge1 must belong to the mainchain segment. The end of map edge1 that is closest to map edge2 is taken to designate where the sidechain segment would have attached to the mainchain. If the oxygen segment is missing then map edge2 must belong to the mainchain segment. The end of map edge2 that is closest to map edge1 is taken to designate where the oxygen segment would have attached to the mainchain.

A molecular model residue of type residuetype is fit to the map residue object by a least-squares procedure and is linked to existing adjacent model residues. Ideal geometry is only partially maintained in the model. See chapter 4 for details of the implementation.

SEQ res1 res2

This command requests the computer to match a sequence of residues with the known residue sequence. Res1 and res2 must be connected through a sequence of residues. Either res1 or res2 may be the amino end of the specified sequence. The computer constructs a trial matrix T with one column for every specified residue and one row for each residue type. An element t(i,k) of T gives an estimate of the probability that the i'th residue guessed really is of residue type k. Matrix T is presented to the user through the vi [Joy80] text editor so that the user may customize the matrix for the specific environment of the residues that were guessed. The resulting matrix is used to rate the match of the specified sequence to the known sequence. A sum of the appropriate matrix elements is computed for each possible registration of the specified sequence to the known sequence, trying both directions. The user is presented with the top ten rated registrations from which to select a winner. In the present implementation, no further action is taken by the computer; the user must make any changes required to make his guess match the registration that he chooses.

3.3.4 Miscellaneous

ADD mapedge mapedge

An edge of type "unknown" is added between one endpoint of the first edge and one endpoint of the second edge. The endpoints are chosen to make the new edge as short as possible.

DISTANCE edge edge

The shortest distance between the endpoints of the first edge and the endpoints of the second edge is displayed.

Chapter IV

IMPLEMENTATION

This chapter presents highlights of the system implementation. Portions of the system that are peculiar to this application or that are particularly important to the system are described, not in fine detail, but sufficiently to give a general feel for how the system works.

Most of the software is written in the high level language C and executes on a VAX 11/780 computer under the UNIX(1) operating system. The rest is written in GIA2 [Bishop82], a locally developed compiler for an AMD 2900 based microprocessor.

4.1 GRAPHICS SYSTEM

Most of the graphics functions are performed on an IKONAS RDS-3000 graphics system. The VAX computer constructs display lists for drawing the graphs and reads the joystick, slide potentiometer, and data tablet. The display lists, transformation matrices, and effective density threshold are sent to the IKONAS. The IKONAS has a frame buffer, a crossbar, a color lookup table, a general-purpose AMD 2900 microprocessor with a 200 nanosecond instruction time, private memory for the microprocessor, and an MA1024 special-purpose transformation processor. The menu, auxiliary information, and graphs are stored in separate sets of bit planes in the frame buffer. The MA1024 processor multiplies a 3x3 rotation matrix times each endpoint in the display list. The IKONAS BMP processor generates lines in the frame buffer from the transformed endpoints using a modified Bresenham line-drawing algorithm and taking into account the electron density threshold. The line display is double buffered and two 60hz refresh times are used to compute the lines, yielding a 30hz update rate. About 1000 line segments can be drawn during the 1/30 sec update. The crossbar selects bit planes from the frame buffer for the color lookup table so that the appropriate line drawing buffer is displayed.

---

(1) UNIX is a trademark of Bell Laboratories.

4.2 COMMAND LANGUAGE SCANNING AND PARSING

The scanner and parser for the command language were written using the LEX [Lesk75] scanner generator and the YACC [Johnson75] parser generator. LEX takes regular-expression descriptions of tokens and generates a scanner. YACC takes a context-free grammar and semantic actions for the grammar productions and generates a parser which executes the appropriate semantic actions whenever a production is used in parsing. The scanner uses a special input function which acquires tokens from menu and edge selections, keyboard entries, and REDO and UNDO command operations. All of these inputs are transformed into a common format and are treated uniformly by the scanner.

The language accepted by the parser is formally defined and thus precise statements can be made about the language, much more so than if the commands were processed by a hand coded procedure. Using a context-free grammar for the command language encourages the design of a simple and consistent language. Also, the language is easy to change. For example, an early implementation of the command language had postfix operators. Later, the language was easily changed to have prefix operators. This was easy because the complete definition of the syntax was written in a formal, simple language and was contained in a single source file.

4.3 ALGORITHM FOR COMPUTING RIDGE LINES

Three different approaches to finding ridge lines were examined. The straightforward approach is to approximate the electron density function, find the critical points, and connect them into a graph. The other two approaches do not exactly find the ridge lines. They do determine graphs that are similar to and for the most part contain the ridge lines.

To implement the first approach, Carroll Johnson has used a linear-blended quintic polynomial with thirty-two terms to approximate the function [Johnson76]. Eric Groose has developed a tensor product B-spline approximation which is calculated piecewise on cubes of volume within the map [Groose80]. To determine the connectivity, a graph is produced with peaks for vertices and passes for edges. A forest of minimal spanning trees is derived from this graph using a weighted cost function of the distance between peaks, the density at the pass, and the collinearity of the vector between peaks and the uphill eigenvector of the Hessian matrix at a pass.

The second approach [Swanson79] finds peaks, passes, and points on the ridge lines with a single procedure. The map is represented as a three dimensional grid of discrete points. All the points below a threshold are rejected and each of the remaining points is tested to see if it is a local maximum along the lines that pass through that point and are parallel to the x, y, and z axes. Any point that is locally maximal in all three directions is retained as a peak. Any point that is locally maximal in any two of the directions is retained as a point on a ridge line. The other points are rejected. Points are connected to points adjacent(2) in the grid and the positions of the points are improved by quadratic interpolation. Heuristics are used to prune the graph of some classes of construction artifacts.

The third approach [Greer74] [Greer76a] also selects points from a grid, connects them into a graph, and prunes the graph. The selection process is much different, although the other steps are much the same as in the second approach. The points that are above the threshold value are grouped into a few density levels. For each level, beginning at the lowest level, the points are tested for removal. Heuristics based on the values of the twenty-six adjacent points in the grid detect when

1. a "hole" in the remaining density would be created,

2. an isolated point would be removed, or

3. a tip or end of a thin chain of density would be removed.

If none of these situations is detected then the point is removed. Alan Terry [Terry80] has modified this algorithm to work top-down, beginning at the highest levels and working down to the lowest.

For the system described in this dissertation the ridge line representation of the map is computed once, off-line, using Swanson's algorithm with the following modifications. The positions of the ridge line nodes are more closely approximated by using a center-of-mass calculation. For points that are local maxima in a plane, the point and the four closest grid points in the plane are used. For 3-D local maxima, the 18 closest grid points are used. Edges are added from each node to all nodes within the 3x3x3 cube of grid points centered at the node. Cycles with less than four edges on a side are broken by removing the most

---

(2) A point is adjacent to each of the other twenty-six points in the 3x3x3 cube of grid points centered on the point.

expensive edge, where a small Euclidean distance between nodes decreases the cost and a small electron density at the nodes increases the cost.

The extent of the map to be used must be guessed and might not contain a whole protein molecule. The size of a unit cell and the symmetry of the crystal help in making this guess. It is easy to tell from an examination of the ridge lines at a high density threshold whether a complete protein is present. A new estimate of the correct bounds for the map can be made and new ridge lines calculated.

On a VAX 11/780 it takes about five minutes to calculate the ridge lines for a map with 40x40x50 grid points.

4.4 LOCAL PATTERN MATCHING BY THE COMPUTER

Each of the map object patterns is encoded in a high level language procedure. Although each procedure was programmed individually, some graph traversal aids were constructed to aid in programming. The most useful of these are procedures for depth-first and breadth-first searches of the map graph, each of which takes as a parameter a procedure to be executed at each edge in the graph to determine if the search should proceed past that edge.

4.5 LEAST SQUARES FITTING OF MODEL RESIDUES TO MAP

This automatic procedure determines the coordinates for all atoms of the model residue corresponding to a map residue object. No facility for adjusting the coordinates is provided. The procedure produces a compromise between ideal model geometry and a precise match to the electron density map.

The system has an ideal geometry dictionary of residue types containing one instance of each residue type. As a first approximation based on energy considerations, the bond lengths, bond angles, and some dihedral angles of a residue are fixed. The bonds that can be twisted (thereby changing a dihedral angle) can be used to partition a residue into rigid bodies. The computer finds target nodes in the map's residue object for each rigid body in the model by local pattern matching. A rigid body with ideal geometry is translated and rotated to minimize the mean square error between the rigid body coordinates and the target coordinates [Mclachlan79]. The results of least squares fits are immediately available to constrain subsequent searchs for target nodes of other rigid bodies. Since rigid bodies

overlap on twistable bonds, the atoms at the ends of the twistable bonds will have multiple estimates for their coordinates. Averaging these estimates yields a single value but distorts the ideal geometry near twistable bonds.

4.6 SEQUENCE REGISTRATION OF CHAINS OF RESIDUES

The problem is to register a sequence of residue types guessed by the user with the known sequence of the protein. Because proteins are linear polymers, the problem is the same as finding a character string within a longer string where the string may be reversed in direction and may have errors. The errors however are not expected to be random. The probability of a guess being correct improves with map quality, but it can be expected that the true identity of the guessed residue is a residue whose structure is similar to that of the guessed residue type.

Jane Richardson has provided a matrix M such that an element m(i,k) gives the probability that for a guess of residue type *i*, the actual residue type is *k*. A new matrix N is formed from the matrix M and the guessed sequence of residue types, where n(i,k) gives the probability that the *i*'th guess really should be residue type *k*. Since a better matrix can be formed by taking into account the environment of the residues in the map and how certain the user is of the correctness of the guess, the user may edit this matrix N before the sequence registration begins. Currently the system creates a file containing the matrix and invokes the *vi* [Joy80] text editor the file. The user edits the file, posts changes by writing the file, and exits the editor. The new matrix N' is used to compute a rating for registering the guessed sequence at every possible position in the known sequence, trying both directions. The user is presented with the top ten rated registrations from which to select a winner. This is done, as was the editing of the matrix, to let the user supply more of his special knowledge to the solution process. The user can, of course, easily accept the default matrix and the top rated registration.

Chapter V

RESULTS AND CONCLUSIONS

5.1 RESULTS

Three different maps have been interpreted by three different sets of people with this system. The maps were of different quality and resolution, and the users varied in expertise in interpreting electron density maps and in experience in using this system. These variations make it difficult to make accurate comparisons between the results, but general conclusions may be drawn. Several other electron density maps have been examined but not fully interpreted using this system.

5.1.1 Staphylococcal nuclease map

The first map fit with this system was a 2.5 A Staphylococcal nuclease map [Arnone71], computed from structure factors obtained from Jane and David Richardson. The 142-residue sequence was used during the interpretation.

I interpreted the map by myself with the goal of correctly locating all of the residues in a minimum amount of time. I am a computer scientist and not a protein crystallographer, and thus strictly an amateur at interpreting electron density maps. This was also the first map I had ever interpreted and the first map ever interpreted with this system.

I took 7 sessions and 26 total hours to interpret the map but some of that time was lost recovering from system failures. The system is an experimental prototype and not a production system. I succeeded in placing all of the residues in density, but the last few residues at either end of the chain were very difficult to interpret. Other than at those ends, no significant errors were made in placing the mainchain. This can be seen in figure 5.1 which shows the mainchain from the model generated by this system along with the mainchain from the published coordinates.
The coordinates of the generated model were compared numerically to the published coordinates [Bernstein77] for the

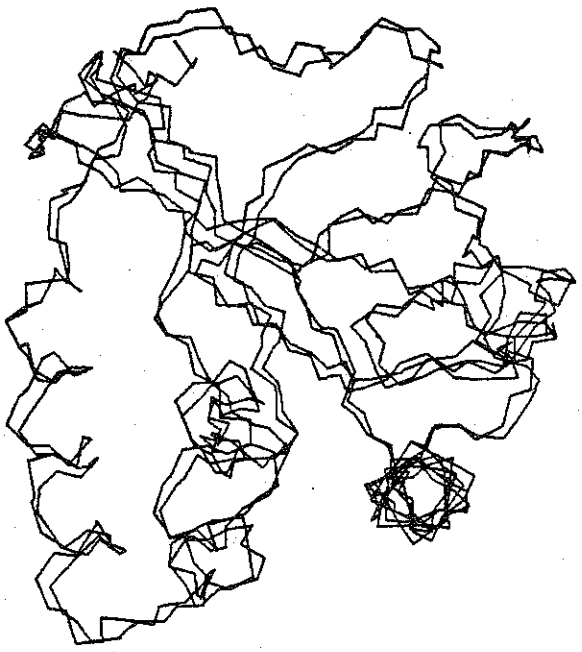

Figure 5.1: Generated and Published Mainchains

molecule. These results are summarized in table 5.1 and figure 5.2.

Figure 5.2 shows in two different ways the distribution of distances between corresponding atoms of the two interpretations, once sorted by residue number and once sorted by difference from the published coordinates. It can be seen that there were significant differences in the first several and last several residues, where the protein has considerable freedom of movement. There are some other clusters of large differences between the two interpretations, corresponding to long sidechains which were placed in density far from the density used in the published interpretation. The coordinates produced with this system were also idealized by an offline procedure [Hermans74] [Ferro80] under geometric and energetic constraints. The idealization improved the internal geometry considerably, moving the

	mean distance in Angstroms	rms distance in Angstroms
all coordinates	1.33	1.73
mainchain only	0.86	0.97
C alphas only	0.86	0.95

TABLE 5.1

Comparison of Staph Nuclease to Published Coordinates

atoms an average of 0.21 A, but did not appreciably improve the agreement with the published coordinates.

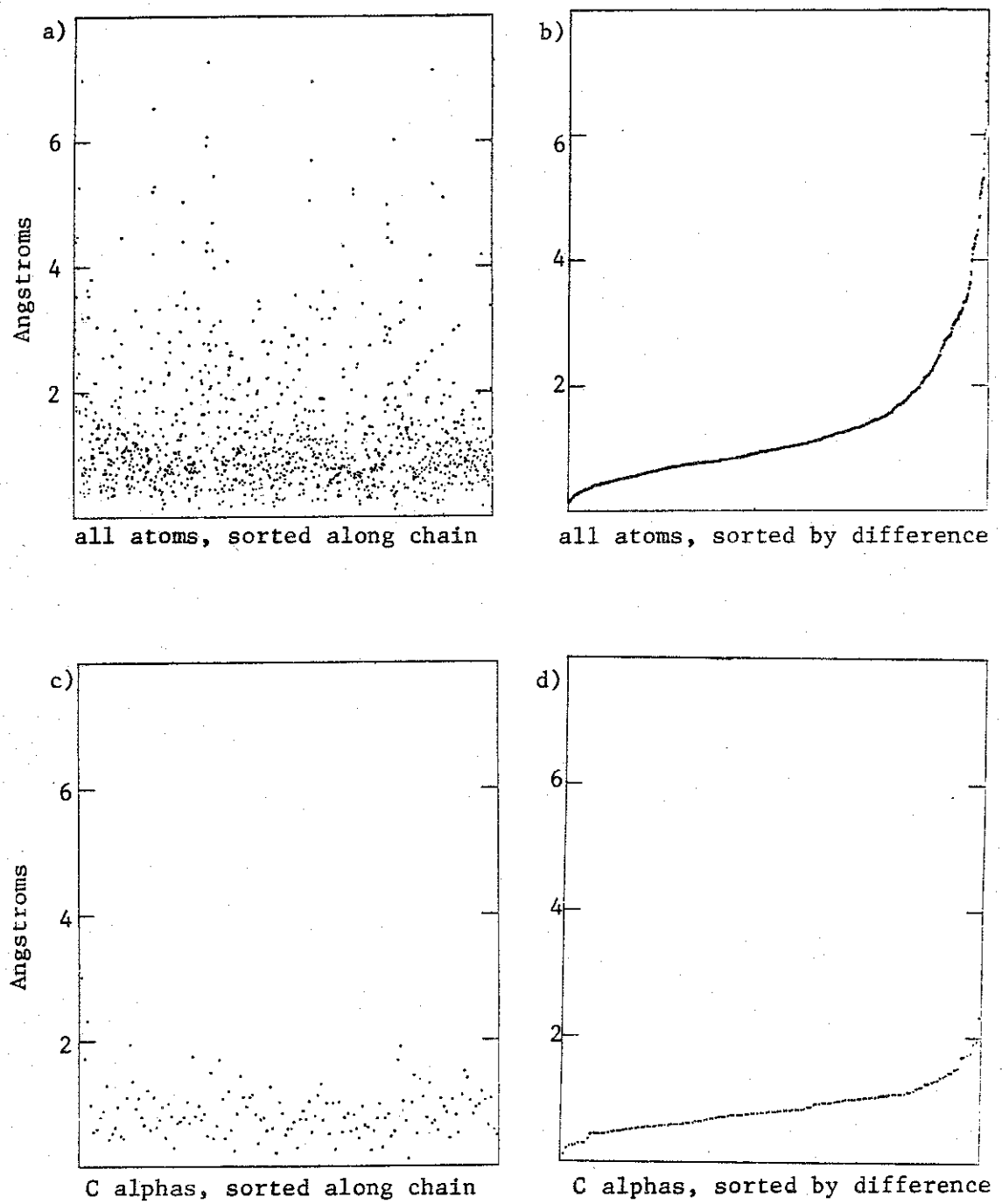

Figure 5.2: Error Distributions for Staph Nuclease.

Along the vertical axis is the distance between corresponding atoms of the published and test interpretations.

5.1.2 Cytochrome b5 Map

The second map interpreted with the system was a good quality 3.0 A map of cytochrome b5 [Matthews72] computed from structure factors obtained from the Brookhaven Protein Data Bank [Bernstein77]. The 93-residue sequence was used during interpretation. The crystal (and thus the map) contains a heme group and associated iron atom in addition to the protein.

Jane Richardson, a protein crystallographer from Duke University, interpreted the map with some minor assistance from me. I helped with the operation of the system and occasionally pointed out features that I could recognize because of my experience in viewing ridge lines but that she did not recognize. Her goals were to interpret as much of the map as could be done with confidence and also to learn about ridge line representations of electron density maps.

Residues 4 through 82 were located and fit to the map in 9 hours. (In the published interpretation only residues 3 through 87 were reported.) In the first 3-hour session, she did some general viewing as well as locating essentially all of the mainchain that was reported. The iron atom had the highest peak in the map and was easily identified. The surrounding heme group was then found and marked, although the plane of the heme had to be changed later. No mistakes were made in the mainchain identification and essentially all subsequent, detailed interpretation could proceed locally along the chain. In a second 6-hour session the known sequence of amino acid residues was registered with the map and the molecular model was built. The direction of the mainchain was obvious from the orientation of the side-chains on several helices. The first attempt to register with the sequence went amazingly well. Jane Richardson guessed the residue types for a chain of 10 residues and I guessed the types of two more residues at the head of the chain, resulting in the following match:

```
GUESS  : TYR TYR THR LEU GLU GLU ILE GLU LYS LEU ASP SER
ACTUAL : TYR TYR THR LEU GLU GLN ILE GLU LYS HIS ASN ASN
```

After the registration, interpretation proceeded along the mainchain, usually sequentially, one residue at a time, but occasionally jumping ahead a small distance and working backward over some difficult spots.

The coordinates of the generated model were compared to the published coordinates [Bernstein77] for the molecule. The results are summarized in table 5.2 and figure 5.3.

	mean distance in Angstroms	rms distance in Angstroms
all coordinates	1.18	1.64
mainchain only	0.66	0.73
C alphas only	0.63	0.70

TABLE 5.2

Comparison of Cyt b5 to Published Coordinates

The results for cytochrome b5 agree more closely with the published coordinates than the results for Staphylococcal nuclease even though the cytochrome b5 map was of poorer resolution. Two explanations are that the interpreter was more skilled and that the disordered ends of the molecule were not interpreted.

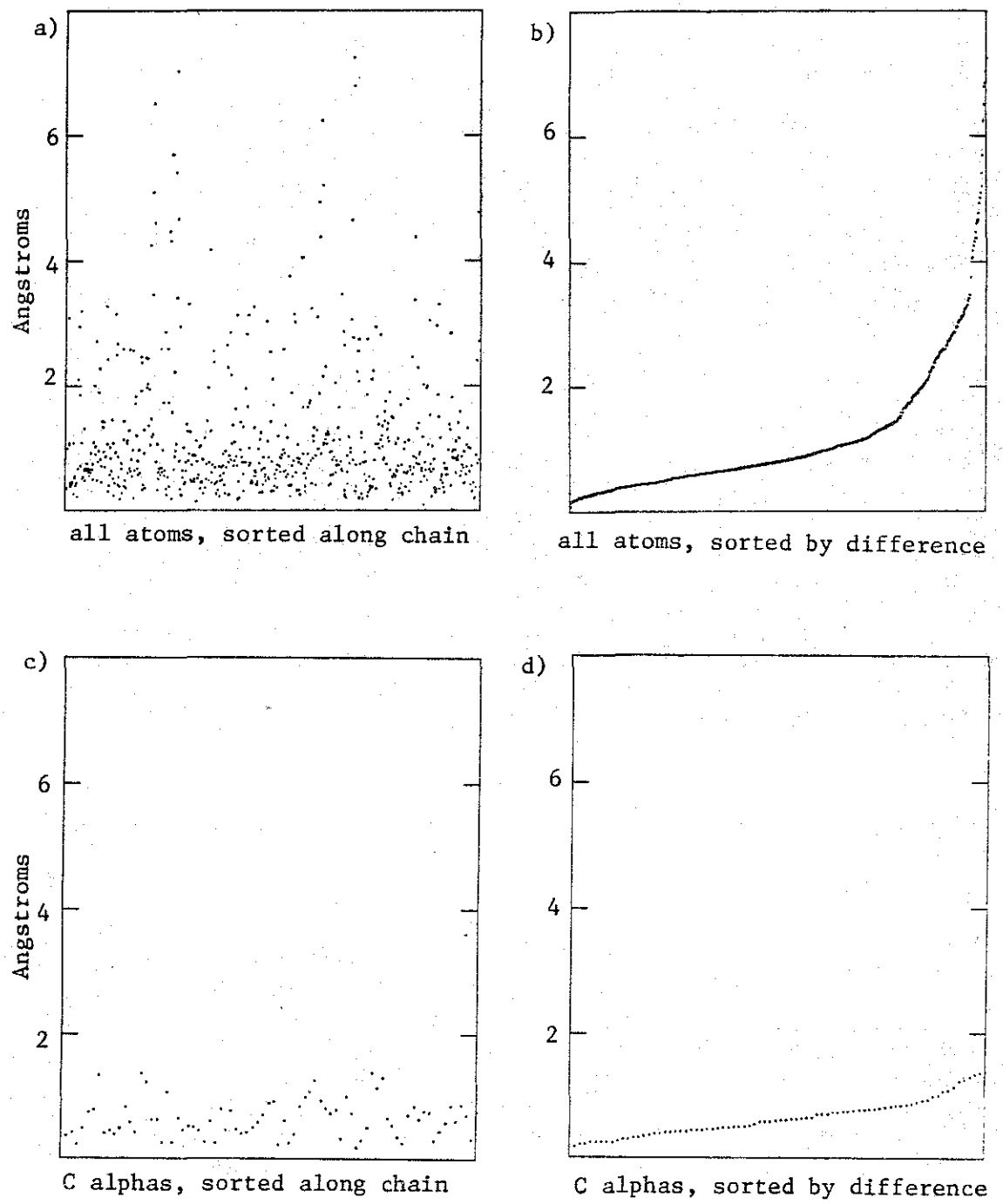

Figure 5.3: Error Distributions for Cyt b5.

Along the vertical axis is the distance between corresponding atoms of the published and test interpretations.

5.1.3 Cytochrome c550 Map

The third map interpreted with the system was a 2.8 A map of cytochrome c550 [Timkovich73] computed from structure factors obtained from the Brookhaven Protein Data Bank [Bernstein77]. A sequence of 134 residues was given but, since the last 13 residues were all listed as glycines, only 121 residues were interpreted. The crystal (and thus the map) contains a heme group and associated iron atom in addition to the protein.

Libby Getzoff, John Tainer, and Duncan McRee, a group of graduate students and post-doctoral fellows from Duke University, interpreted the map with assistance from me. I took a much more active role in this interpretation than in the previous one. Their goals were to interpret as much of the map as could be done with confidence and also to learn about ridge line representations of electron density maps.

We took 6 sessions and 22 total hours to interpret the map. No significant errors were made in placing the mainchain. The coordinates of the generated model were compared to the published coordinates [Bernstein77] for the molecule. The results are summarized in table 5.3 and figure 5.4.

	mean distance in Angstroms	rms distance in Angstroms
all coordinates	1.52	1.94
mainchain only	1.04	1.34
C alphas only	1.07	1.29

TABLE 5.3

Comparison of Cytochrome c550 to Published Coordinates

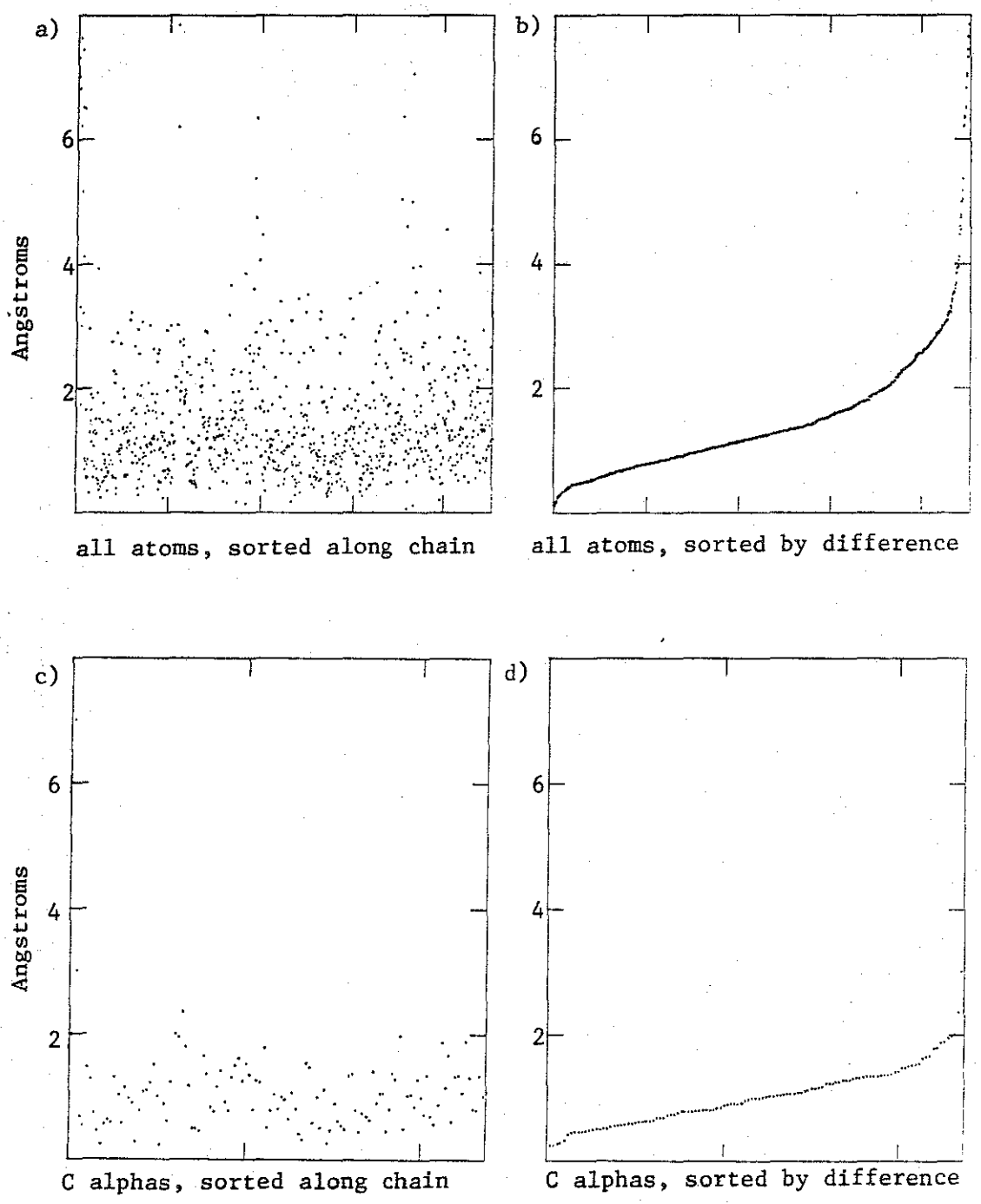

Figure 5.4: Error Distributions for Cyt c550.

Along the vertical axis is the distance between corresponding atoms of the published and test interpretations.

These results for a 2.8 Angstrom map are worse than those for the 2.5 Angstrom Staphylococcal nuclease map and for the 3.0 Angstrom cytochrome b5 map. This may be explained by the better resolution of the Staphylococcal nuclease map and the more skilled interpreter for cytochrome b5 map. I have no measure of the relative amounts of noise in the three maps.

5.1.4 Locating Molecular Boundaries

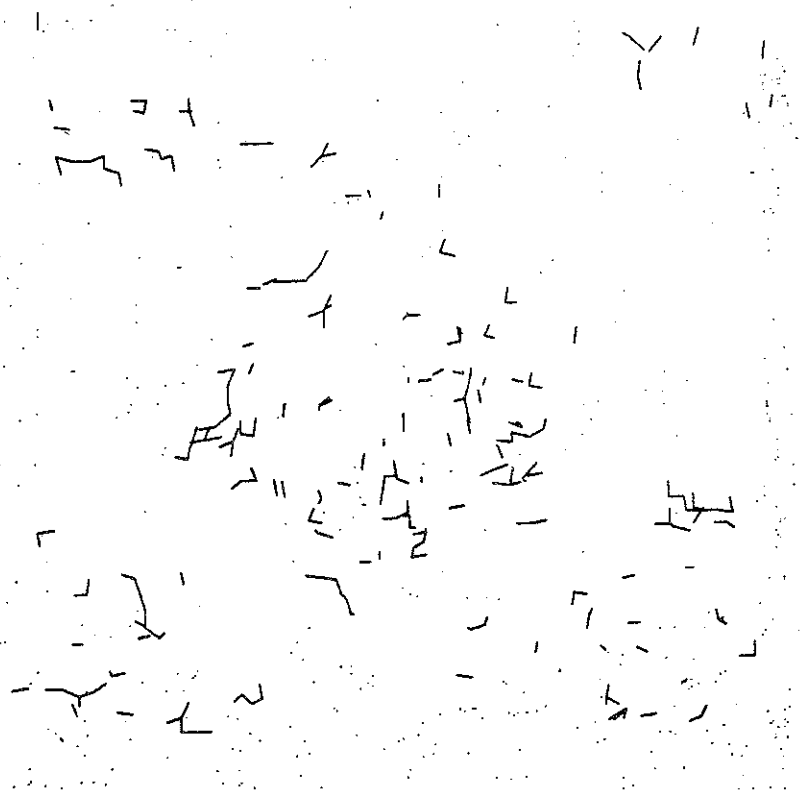

Figure 5.5: Ridge Lines at a High Density Level

An unexpected result is that for all maps that I have looked at with this system, the molecular boundaries have been clearly visible in the ridge line representations. Interpretation is much easier if one whole molecule is

present in the portion of the map that is represented. It is possible to work with part of one copy of the molecule and a symmetry-related copy of the other part, but this is much harder. It is best to have a complete copy of the molecule.

If the density threshold for the ridge lines is very high, the regions of the map between the molecules show as blank areas, as shown in figure 5.5. By rotating the ridge lines on the graphics screen, the boundary for a single molecule can be found. The map is then recalculated based on the molecule's position. Often no single molecule will be completely visible, but one can be selected and a new map can be calculated based on an estimate of where the whole molecule is. Another iteration of viewing and recalculating the map may be necessary. An iteration takes about 15 minutes.

5.2 CONCLUSIONS

My thesis is that there is a class of problems, especially those that require global perception, for which a man-machine system can be effective where automatic, machine-only systems have been ineffective and manual, man-only systems are too slow.

I believe that the most important aspect of a user interface, after sufficiency, is how easy it is to use. The results reported above have shown this system to be very effective for interpreting electron density maps. It works on multiple sets of data for multiple sets of users. It is significant that the two sets of first-time users were able to interpret maps efficiently with little start-up time. None of the users made errors in the connectivity of the mainchain and all of the interpretations agreed well with the published interpretations.

This system has a number of of substantial advantages over the mini-map method of interpretation:

1. The interpretation takes a biochemist only three days instead of from four days to four weeks. Only two days were needed to interpret even a 3.0 Angstrom map with this system but the time of four days for interpreting a mini-map is only for good maps of small proteins.

2. About 15 minutes is needed to prepare the ridge lines from structure factors instead of a minimum of one day to prepare contours on transparent sheets (which is attained only with a special equipment set-up).

This makes the ridge line representation useful for general map viewing as well as for interpretation.

3. The coordinates are produced in machine readable form.

4. Coordinates are produced for every non-hydrogen atom, not just for C alpha atoms.

Some good aspects of the man-machine interface are:

1. The communication between man and machine is in terms of objects natural to the application area.

2. The tasks were assigned according to the capabilities of the resources. In particular, the difficult, global perception problem was assigned to the human user who is much better at it than is the computer.

3. A single, uniform method is used to communicate similar pieces of information.

4. The actions of the commands are simple and predictable. This minimizes surprises for the user.

5. Any command which changes the stored interpretation of the map gives immediate feedback by changing the displayed interpretation. Changes are made only to visible parts of the map.

6. All commands are executed immediately without need for confirmation, but there is a multi-level UNDO command which will reverse the effect of any command (with the exception of LOAD and SAVE).

7. The system is restricted to operations proper for the goals of the system. A serious attempt was made to implement only necessary operations. This makes the system more efficient to operate since time cannot be wasted doing unnecessary operations (such as manual manipulation in this system). This also makes the system smaller and thus easier to learn.

The ridge line representation is a very good one for interpreting electron density maps. A large volume of a map can be displayed and comprehended because few lines are required for the representation, roughly an order of magni-

tude fewer than with with contouring in three sets of planes. This overview capability is essential for interpreting electron density maps. The similarity of the ridge lines to the stick-figure representation of molecules facilitates matching the model to the map. The similarity, however, makes it imperative to use color to make the map and molecule distinguishable. An accurate ridge line representation of a perfect map would in fact be the proper model. One can think of the interpretation done on this system as the step-by-step transformation of the given, approximate map representation into a model satisfying molecular constraints.

The current implementation of this system uses a color raster display with a 512x512 resolution. No anti-aliasing of the lines is done, so the lines are quite jagged. However, since the picture is constantly being rotated and since the underlying data is uneven at a larger scale, the jaggedness of the lines is not misleading and is rarely noticed. Displaying the lines without anti-aliasing allows many more lines to be drawn and does not apparently degrade the usefulness of the display at all.

A particularly useful attribute of a ridge line representation is that the density threshold can be changed smoothly in real time. This threshold is the value of the lowest density that will be displayed. Changing the threshold is roughly equivalent to changing the contour level for a contour representation. I believe the ability to smoothly change the threshold was essential to the successful interpretations of maps described earlier in this chapter. This system, with its ridge line representation and smoothly changeable density threshold, is more effective for tracing the mainchain of a protein than any other existing method.

The contributions of this work to computer science are 1) a documented example of a good man-machine interface, 2) a detailed discussion of the major design decisions that I made, and 3) a demonstration of the usefulness of a ridge line representation for a scalar function of three variables.

5.3 DIRECTIONS FOR FUTURE RESEARCH

Some of these ideas deal with the implementation, some deal with the architecture, and some deal with more basic research.

5.3.1 Pattern Matching by Computer

The computer's model of proteins is very simple and is applied in an ad hoc manner. The model includes both connectivity information and geometric constraints, but the formal geometric constraints are used only for fitting residues to the ridge lines. Some notions of maximum lengths and connectivity of objects are scattered throughout the code but these are independent and potentially conflicting. Some commands can alter the interpretation of the ridge line in a way inconsistent with the connectivity rules.

For completeness, the model should include hydrogen bonds, disulphide bonds, metal ions, and cofactors such as heme groups. The entire model should be applied consistently by a single method. This does not, however, preclude controlled, temporary violations of the rules to facilitate modifying the interpretation.

I think in the long run the best solution to these problems is to use graph grammars [Claus79] to describe the protein model. Ordinary graph grammars are excellent for describing the connectivity of the model but are not sufficient for describing the geometric constraints. Either a new, more powerful syntax needs to be developed or the constraint will have to be included as semantic actions.

The best way to use such a modified graph grammar is with a parser generator. Once a parser generator has been designed and constructed, the entire model can reside in the graph grammar. A new model can be easily be tried by editing the graph grammar and running the parser generator. Different grammars could be produced that are tuned for different resolutions of maps. It should also be just as easy to write a grammar for nucleic acids as for proteins. The grammar describes not only the structure of the protein but also how ridge lines correspond to the protein. The grammar productions for a sidechain, for example, will describe what ridge line subgraphs can be recognized as a sidechain. The actions that the computer performs when a pattern is matched would be largely subsumed into the productions of the graph grammar. The current code for recognizing patterns and modifying the interpretation would become declarative and formal instead of procedural and ad hoc.

5.3.2 Make Use of Symmetry in Map

It would be useful to have a new type of object called "symmetric" to identify edges that belong to neighboring, symmetry-related molecules in the map. This would be helpful when trying to interpret the map near boundaries between molecules. It would be much less confusing if the map belonging to other molecules was clearly distinguished as such. It is much more useful to be able to make decisions about the boundaries from a global viewpoint than to have to continualy make local decisions when working near a boundary. The positions of symmetrically equivalent edges can be easily calculated since the symmetry relationships are known before the map is interpreted; the problem is determining which edges belong to which copy of the molecule. A useful rule about a set of symmetry-related objects would be that exactly one of the members of the set would have a type other than "symmetric". One possible set of commands to use with "symmetric" type objects is:

1. Use Greer's technique [Greer74] with the midpoint of a selected edge serving as the center of the molecule. This technique finds all sets of symmetry-related segments and selects from each set the member closest to the center of the molecule. This command would be used at the beginning of an interpretation to find an initial boundary for a molecule.

2. Exchange the type of the selected object with the type of the member of its symmetry-related set which is not of type "symmetric". This command would be useful for making adjustments to the molecular boundaries.

3. Do the same for each of a list of edges instead of an object.

4. Temporarily highlight the set of symmetry-related objects which contains the selected edge.

5.3.3 Estimate Map Quality from Graph Properties

There is currently no single measure of the quality of an electron density map; two measures are generally used, a precise resolution and a rough estimate of the signal-to-noise ratio. A "very good" 3.0 A map may be easier to interpret than a "fair" 2.5 A map. Some graph properties of a map's ridge lines might be a good estimate of map quality. Some appropriate graph properties might be the number of edges, the number of vertices, the number of disjoint connected sub-graphs, the number of cycles, the average path

length of cycles, etc. The problem is finding a function of these properties that gives useful results.

5.3.4 Use Intensity of Display to Encode More Informa

The density value associated with a displayed edge in the ridge lines can be determined by sliding up the density threshold control until that edge disappears. It would also be useful to be able to tell the density value without having to take some action. One possible solution is to encode the density value as the saturation or as the intensity of the edge's color.

Intensity could be used instead as a depth cue. At each rotation of the ridge lines each edge would be assigned a single depth value. This value would be used as an address to look up the intensity of the edge's color for that depth. With hardware support, the intensities could be varied smoothly along the edges.

5.3.5 Determine Power of the System

The ease with which the 3.0 Angstrom map was interpreted suggests that there may be more power in this system than in other existing interpretation or fitting systems. Maps which cannot be interpreted on other systems may be interpretable by this system. If this extra power exists, it probably derives from the ridge line representation. The ridge lines have less information than the contour representation used by other systems, but it appears that more information is transfered to the user by the ridge line representation. The information contained in the ridge lines appears to be more appropriate for interpretating maps. This conjecture needs to be verified by careful experimentation. The observed effect may not be due to any perceptual property of the ridge lines at all, but due instead to the associated easy of dynamically changing displayed density levels.

5.3.6 Investigate Bypassing Fitting System

The coordinates produced by this system are a very good starting point for a manual fitting system. They are of much higher quality than those produced by the mini-map method. After the coordinates are idealized, they may be good enough to bypass the fitting system and feed an algorithmic refinement procedure directly.

One result of the refinement procedure is a better quality map. Iteration of this fit-refine cycle may be possible because a better map can be expected to yield better quality ridge lines.

A method for fitting the model manually would have to be added to this system to make the results usable for refinement. The most appropriate method for fitting would be for the user to specify target vertices in the ridge line graph for specific atoms in the molecular model. These targets could be used directly in the weighted least-squares procedure that is currently used for fitting residues automatically.

BIBLIOGRAPHY

Arnone, A.; et al "A High Resolution Structure of an Inhibitor Complex of the Extracellular Nuclease of Staphylococcus aureus," Journal of Biological Chemistry 246 (1971): 2302-16.

Bernstein, F.C.; et al "The Protein Data Bank: A Computer-based Archival File for Macromolecular Structures," Journal of Molecular Biology 112 (1977): 535-42.

Bishop, G. "Gary's Ikonas Assembler Version 2," unpublished documentation, Department of Computer Science, University of North Carolina, Chapel Hill, 1982

Blundell, T.L. and Johnson, L.H. Protein Crystallography New York: Academic Press, 1976.

Britton, E.G. "A Methodology for the Ergonomic Design of Interactive Computer Graphic Systems, and its Application to Crystallography," Ph.D. dissertation, University of North Carolina, Chapel Hill, 1977.

Britton, E.G.; Lipscomb, J.S.; and Pique, M.E. "Making Nested Rotations Convenient for the User," Proceedings of the 1978 ACM SIGGRAPH Conference, Computer Graphics 12, no. 3 (August 1978): 222-27.

Brooks, F.P., Jr. "The Computer 'Scientist' as Toolsmith: Studies in Interactive Computer Graphics," Proc. 1977 IFIP : 625-34

Claus, Volker; Ehrig, Hartmut; and Rozenberg, Grzegorz; eds. Graph-Grammars and their Application to Computer Science and Biology New York: Springer-Verlag, 1979.

Diamond, R. "A Tablet-side Guide to BILDER (v1.11): An Interactive Graphics Program for Biopolymers," ed. Ladner, R.C. MRC Laboratory of Molecular Biology, Cambridge, England, 1981

Dickerson, R.E. and Geis, I. The Structure and Action of Proteins Menlo Park, Ca.: Benjamin/Cummings Publishing Company, 1969.

Engelmore, R.S. and Nii, H.P. A Knowledge-Based System for the Interpretation of Protein X-Ray Crystallographic Data. Stanford Heuristic Programming Project report HPP-77-2, 1977. (also STAN-CS-77-569)

Engelmore, R.S.; Terry, A. Structure and Function of the Crysalis System. Stanford Heuristic Programming Project report HPP-79-16, 1979.

Engelmore, R.S.; Terry, A. separately published appendix to Structure and Function of the Crysalis System. Stanford Heuristic Programming Project report HPP-79-16, 1979.

Feigenbaum, E.A.; Engelmore, R.S.; and Johnson, C.K. "A Correlation between Crystallographic Computing and Artificial Intelligence Research," Acta Crystallographica A33 (1977): 13-18.

Ferro, D.R.; McQueen, J.E.; McCown, J.T.; and Hermans, J. "Energy Minimization of Rubredoxin," Journal of Molecular Biology 136 (1980): 1-18.

Greer, J. "Three-dimensional Pattern Recognition : An Approach to Automated Interpretation of Electron Density Maps of Proteins," Journal of Molecular Biology 82 (1974): 279-301.

Greer, J. "Automated Interpretation of Electron Density Maps of Proteins: Derivation of Atomic Co-ordinates for the Main Chain," Journal of Molecular Biology 100 (1976): 427-58.

Greer, J. "Application of the Automated Interpretation of Electron Density Maps to Bence-Jones Protein Rhe," Journal of Molecular Biology 104 (1976): 371-86.

Groose, E. "Approximation and Optimization of Electron Density Maps," Ph.D. dissertation, Stanford University, 1980.

Hermans, J. and McQueen, J.E. "Computer Manipulation of Macromolecules with the Method of Local Change," Acta Crystallographica A(30) (1974): 730-39.

Johnson, S.C. "Yacc: Yet Another Compiler Compiler," Computing Science Technical Report No. 32, Murray Hill, New Jersey: Bell Laboratories, 1975

Johnson, C.K. and Groose, E. "Interpolation Polynomials, Minimal Spanning Trees, and Ridge Line Analysis in Density Map Interpretation," American Crystallographic Association Program and Abstracts (Aug 1976): 48.

Johnson, C.K. "Interactive Analysis of Critical Point Networks in Macromolecule Density Maps," Acta Crystallographica 34(S) (1978): S353.

Jones, T.A. "A Graphics Model Building and Refinement System for Macromolecules," Journal of Applied Crystallography 11 (1979): 268-72.

Joy, W. "An Introduction to Display Editing with Vi, revised by M. Horton, UNIX Programmer's Manual Volume 2c - Supplementary Documents, Seventh Edition, Virtual VAX-11 Version, Nov, 1980

Lesk, M.E. "Lex - A Lexical Analyzer Generator," Computing Science Technical Report No. 39, Murray Hill, New Jersey: Bell Laboatories, 1975

Lipscomb, J.S. "Three-Dimensional Cues for a Molecular Computer Graphics System," Ph.D. dissertation, University of North Carolina, Chapel Hill, 1979.

Matthews, F.S.; Levine, M.; and Argos, P. "Three-Dimensional Fourier Synthesis of Calf Liver Cytochrome b5 at 2.8 A Resolution," Journal of Molecular Biology 64 (1972): 449-64.

McLachlan, A.D. "Gene Duplications in the Structural Evolution of Chymotrysin," Journal of Molecular Biology 128 (1979): 49-79.

Pique, M.E. "Nested Dynamic Rotations for Computer Graphics," M.S. thesis, University of North Carolina, Chapel Hill, 1980.

Richardson, J.S. "The Anatomy and Taxonomy of Protein Structure"," Advances in Protein Chemistry 34 (1981): 167-339.

Swanson, S.M. "Alternate Electron Density Models for Structural Biochemistry," Journal of Molecular Biology 129 (1979): 637-42.

Terry, A. Department of Information and Computer Science, University of Irvine, California, personal communication, May 1980

Timkovich, R. and Dickerson, R.E. "Recurrence of the Cytochrome Fold in a Nitrate-respiring Bacterium," Journal of Molecular Biology 79 (1973): 39-56.

Tsernoglou, D.; Petsko, G.A.; McQueen, J.E. Jr; and Hermans, J. Jr. "Molecular Graphics: Application to the Structure Determination of a Snake Venom Neurotoxin," Science 197 (1977): 1378-81.

Watenpaugh, K.D.; Sieker, L.C.; and Jensen, L.H. "The Structure of Rubredoxin at 1.2 A Resolution," Journal of Molecular Biology 131 (1979): 509-22.

INDEX TO REFERENCES