

VISUALIZATION AND EXTRACTION OF STRUCTURAL COMPONENTS FROM
RECONSTRUCTED VOLUMES

Edgar Garduno

A DISSERTATION

in

BIOENGINEERING

Presented to the Faculties of the University of Pennsylvania in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

2002

Gabor T. Herman, Ph.D.

Supervisor of Dissertation

Gershon Buchsbaum, Ph.D.

Graduate Group Chairperson

DEDICATION

Para Oliva, Leopoldo y Gustavo quienes siempre han estado . . .

To Oliva, Leopoldo y Gustavo who have always been present. . .

Por todos los ladrones y ángeles que pasarón, y los que pasarán . . .

Con todo el corazón y cariño.

ACKNOWLEDGEMENTS

This has been a long road since that Fall of 1974 when “Kairos” was the first relay. Many years later the “Centro Activo Freire”. Those years in the U.N.A.M. have long passed. Now the University of Pennsylvania. This work would not have been possible without the assistance or support of many individuals and institutions.

First of all I would like to thank the guidance and friendship of Dr. Gabor T. Herman during these years. Without his guidance this project would not have been possible. Dr. Herman is only one of many who have examined this work, I would like to thank Dr. Kenneth R. Foster, Dr. Robert M. Lewitt, Dr. Dimitris N. Metaxas, Dr. Jayaram K. Udupa, and Dr. Felix W. Wehrli, the members of the committee in charge of reviewing this dissertation.

This dissertation could fall into the category of “Nomad Ph.D.s” as I have visited several laboratories in different parts of the U.S. and abroad. I first arrived to the Medical Image Processing Group at the University of Pennsylvania, then to the Biocomputing Unit at the C.N.B. (Spain), later to the Center for Computer Science and Applied Mathematics (Temple University) and finally to the Mathematical Sciences Research Institute (Berkeley), respectively. The final experiments and the final version of my thesis were finished in the Discrete Imaging and Graphics Group at the Graduate Center of the City University of New York. In all these laboratories and institutions I was always welcome and assisted by their members. In particular I want to thank the support from José María Carazo, Gabor T. Herman and Jayaram K. Udupa. I am also very thankful to Samuel Matej, Robert M.

Lewitt, Roberto Marabini, Rasmus Schröder, Carlos Oscar Sánchez, Bruno Carvalho, T. K. Narayan, José Jesús Fernández, and Pedro de Alarcon for their help, guidance and their time.

During these years of graduate school I have received the invaluable financial and economical support from several institutions whose support and trust I will always thank: the Consejo Nacional de Ciencia y Tecnología (México) and the International Institute of Education (México and U.S.), the National Institute of Health through grants HL28438 and HL70472, the National Science Foundation through the grant DMS96122077 and the Consejo Superior de Investigación Científica, (Spain).

Clearly, it would be impossible to “survive” graduate school without feeling like home. In these years I have met amazing people who did exactly that, my gratitude and love to all my dear friends and “relatives” all around the world: Margie, Vasilis, Katerina, Octavio, Daniel, Donna, Lisa, Kristina, Jonas, Teresa, Natalia, Pedro, Paola and Julia.

There has been a group of people who have been there no matter the difficulty, the distance, and always in my heart. Nada de esto sería posible sin Oliva, Leopoldo y Gustavo: ¡¡mil gracias!!.

CONTENTS

1	INTRODUCTION	1
2	BACKGROUND	5
2.1	Macromolecules: An Overview	6
2.2	Structure Analysis of Macromolecules by Transmission Electron Microscopy	9
2.3	The Transmission Electron Microscope	12
2.3.1	Specimen Preparation	14
2.3.2	Image Formation in the TEM	16
2.4	Mathematical Background	19
2.5	Data Collection in TEM	30
2.6	Three-Dimensional Reconstruction from TEM Images	33
2.7	Final Representation	39
3	VISUALIZATION USING KAISER-BESSEL FUNCTIONS	40
3.1	Reconstruction Algorithm	40
3.2	Blob Parameters and Grids	44
3.3	Implicit Surfaces and Visualization	50
3.4	Selection of Blobs for Visualization	52
3.5	Comparison with Explicit Surface Visualization	59
4	FAST AND ACCURATE VISUALIZATION METHODOLOGIES	65
4.1	Method to Speed Up the Raycasting-Blobs Technique	66
4.2	Experiments for the Faster Raycasting-Blobs Technique	68
4.3	Representation of Implicit Surfaces from Voxelized Sets	73
4.3.1	Voxels	73
4.3.2	Adjacencies for Cubic and Dodecahedral Voxels	76
4.3.3	Definition of Boundaries	78
4.3.4	A Boundary Tracking Algorithm	80
4.4	Experiments	81

5	EXTRACTION OF SUBUNITS FROM DENSITY FUNCTIONS	87
5.1	Identification of Classes of Subunits and Their Alignment	88
5.1.1	Segmentation	89
5.1.2	Fuzzy Segmentation	89
5.1.3	Registration Methods	97
5.2	Experiments	106
5.3	Results	109
6	CONCLUSIONS	116
6.1	Summary and Contributions	116
6.2	Future Work	122
A	Fourier Transforms of Trains of Pulses on Various Grids	125
B	Bacteriorhodopsin	129
	BIBLIOGRAPHY	133

LIST OF FIGURES

1.1	Relative size of several objects from the cosmic-scale to the nano-scale together with the instruments used to observe them (image courtesy of the INVSEE project at Arizona State University, Dr. B. L. Ramakrishna, Project Director, INVSEE).	3
2.1	Schematic representation of protein structure: (a) Linear sequence of amino acids forms the primary structure, here forming the porin protein. (b) Local interactions among chains of amino acids form the secondary structure. It is common practice to represent β -pleated sheets as ribbon arrows, α -helices as helical ribbons and random coils as thin lines. (c) The 3D structure formed by the combination of secondary structures forms the tertiary structure, a good example of a single-chain protein is the protein G. (d) The combination of several polypeptide chains results in the quaternary structure. A clear example is the protein GroEL formed by seven polypeptide chains [124].	7
2.2	(a) Actual configuration of a JEOL [®] 2010 F TEM. (b) Schematic operation of a TEM.	13
2.3	(a) A series of TEM images (from [8]) obtained for the study of the protein complex DnaB·DnaC which is shown in (b). The images demonstrate the low signal-to-noise ratio present in a micrograph.	18
2.4	Under specific conditions, a micrograph can be considered a collection of line integrals $\mathcal{P}\mathfrak{v}$ crossing the specimen, denoted by \mathfrak{v} . Every line integral is determined by an orientation vector $\vec{\sigma}$ and a position vector $\vec{x} \in \vec{\sigma}^\perp$	29
2.5	In the <i>series expansion methods</i> a density function \mathfrak{v} is approximated by the linear combination of basis functions. A common choice for basis functions are functions that have a unit value inside a cube and zero outside. To approximate a density function, each instance of the basis functions is weighted by coefficients c_j . Under this model, a line integral $[\mathcal{P}\mathfrak{v}]_i$ will intersect only a few basis functions.	36

3.1	The Fourier transform of a blob. We plot $\log \left(\frac{\hat{b}(2,13.36,2.40;R)}{\hat{b}(2,13.36,2.40;0)} \right)$ as a function of the frequency R	42
3.2	Scheme for the approximation of a density function, represented in blue, by the linear combination of weighted blobs (the center of each blob is represented in the image by a small yellow sphere). The support of a blob is a sphere and we indicate the support of each blob in transparent red.	43
3.3	Points in the simple cubic (a), body-centered cubic (b), and face-centered cubic (c) grids in a $2 \times 2 \times 2$ portion of space (assuming $\Delta = 1$). The rest of the points can be obtained by filling in space by the most natural repetition of the indicated $2 \times 2 \times 2$ portion.	45
3.4	The root mean square (rms) error between a constant-valued function and its approximation by a linear combination of blobs using several values α and $\frac{a}{\Delta}$ (with $\Delta = \frac{1}{\sqrt{2}}$). The continuous heavy curve indicates the locus of the points (a hyperbola) which satisfy (3.8), the broken heavy curves indicate the hyperbolas which correspond to the other zero-crossing of the Fourier transform of b	49
3.5	(a) Surface representation of the macromolecular complex DnaB·DnaC with blob parameters $a = 1.25$, $\alpha = 3.60$ with grid separation $\Delta = \frac{1}{\sqrt{2}}$ [8] (the distance between pixels is equal to $\frac{1}{4}$). (b) A central slice from the reconstruction by ART of the macromolecular complex DnaB·DnaC with the same parameters as for (a) with $\Delta = 1.00$	53
3.6	In all cases the distribution υ is defined by (2.28) using the grid $B_{\frac{1}{\sqrt{2}}}$ with blob coefficient 1 at the points $(0,0,0)$, $(\sqrt{2},0,0)$ and $(0,\sqrt{2},0)$ and blob coefficient 0 at all other points. The displayed surfaces are defined by (3.9) with $t = 0.5$. The values of a and α are (a) 1.25 and 3.60 (same as for Fig. 3.5(a)), (b) 2.40 and 13.36, (c) 3.20 and 18.85 and (d) 1.65 and 0.00. The first three pairs of (a, α) satisfy (3.8) and the fourth is computed with the value of x for the second zero crossing of $J_{\frac{7}{2}}(x)$. (Throughout this dissertation we report on a and α to an accuracy of 0.01, however the values actually used by our programs were always calculated to the accuracy of our computations.)	54
3.7	Visualizations of the implicit surface ($t = 0.5$) of reconstructions of a sphere. For the choices of the parameters in cases (a), (b), (c) and (d), see the caption of Fig. 3.6.	55

3.8 The rms error between analytic normals to a sphere and normals to the implicit surface of its reconstruction at each display point for which the ray casted crosses both surfaces (see Fig. 3.7). The continuous heavy curve indicates the locus of the points (a hyperbola) which satisfy (3.8). The broken heavy curves indicate the hyperbolas which correspond to the other zero-crossing of the Fourier transform of b 56

3.9 Representations of the implicit surface at level $t = 0.5$ for the combination of two blobs whose centers are immediate neighbors in the bcc grid $B_{\frac{1}{\sqrt{2}}}$ and whose coefficients are 1. The parameters of the blobs match those in Figs.3.6 and 3.7. 58

3.10 Representations of the implicit surface for the macromolecular complex DnaB·DnaC. The reconstructions using ART and visual representations of the DnaB·DnaC were created with (a) parameters $\Delta = \frac{1}{\sqrt{2}}$, $a = 1.25$ and $\alpha = 3.60$ as used in [8], (b) parameters $\Delta = \frac{1}{\sqrt{2}}$, $a = 2.40$ and $\alpha = 13.36$. . . 59

3.11 Central section of the reconstructions of the macromolecular complex DnaB·DnaC by ART with (a) parameters $\Delta = \frac{1}{\sqrt{2}}$, $a = 1.25$ and $\alpha = 3.60$, (b) parameters $\Delta = \frac{1}{\sqrt{2}}$, $a = 2.40$ and $\alpha = 13.36$. Both images are central slices of the discretized density function v using the sc grid with $\Delta = 1.00$ 60

3.12 Representations of the explicit surfaces obtained from the reconstructions of the macromolecular complex DnaB·DnaC using OpenDX with a $128 \times 128 \times 128$ voxelized distribution over the sc grid ($\Delta = 1.00$). Parameters used for reconstruction were: (a) $\Delta = \frac{1}{\sqrt{2}}$, $a = 1.25$ and $\alpha = 3.60$ and (b) $\Delta = \frac{1}{\sqrt{2}}$, $a = 2.40$ and $\alpha = 13.36$. It is clear that for this example the general visualization software hides some important details; for a comparison with the corresponding output of the raycasting method, see Fig. 3.10. 62

3.13 Representation of the explicit surface obtained from the reconstructions of the macromolecular complex DnaB·DnaC using OpenDX with a $400 \times 400 \times 400$ voxelized distribution over the sc grid ($\Delta = 0.25$). (a) Parameters used for reconstruction were $\Delta = \frac{1}{\sqrt{2}}$, $a = 1.25$ and $\alpha = 3.60$. (b) Parameters used for reconstruction were $\Delta = \frac{1}{\sqrt{2}}$, $a = 2.40$ and $\alpha = 13.36$. For this representation it is necessary to increase four times the memory allocated by OpenDX. 63

4.1 (a) Four micrographs of specimens of the DnaB·DnaC [8]. (b) Projections (*above*) without noise and (*below*) with noise from the macromolecular complex bacteriorhodopsin created with the assistance of atomic descriptions [10] and simulation programs [30, 32, 75]. 69

4.2 Central slices corresponding to the DnaB·DnaC reconstructed with ART with blob parameters $a = 2.40$, $\alpha = 13.36$ and $\Delta = \frac{1}{\sqrt{2}}$: (a) the set $\{c_j\}$, (b) the discretized density function υ at the locations of $\{\bar{p}_j\}$ 70

4.3 Three-dimensional representations of the macromolecular complexes (a) DnaB·DnaC and (b) bacteriorhodopsin produced by the modified raycasting technique described in Section 4.1. They are identical to those produced by the original implementation of the raycasting-blobs method show in (c) DnaB·DnaC and (d) bacteriorhodopsin. While the real shape of the DnaB·DnaC is unknown, we can compare (b) and (d) with the “real” shape of the protein in Fig. B.2 (at an arbitrary threshold that allows the visualization of the seven α -helices). Both images (b) and (d) represent the surface that encloses the same volume as that represented in Fig. B.2. 72

4.4 We present the voxels associated with (a) the simple cubic grid (which are cubes whose centers are points of the grid), (b) the body-centered cubic (which are truncated octahedra), and (c) the face-centered cubic grid (which are rhombic dodecahedra). 74

4.5 Computer graphic display of a sphere using different grids. (a) is the display based on a simple cubic grid with voxels of the same volume as the display based on the *fcc* grid used for (b). The image (c) corresponds to a display based on a simple cubic grid with voxels of volume equal to one eighth of the voxel volume in the other two images (reproduced from [51]). 75

4.6 The cubic and rhombic dodecahedral voxels, identified by a point \bar{c} in F_Δ . (a) The edge (\bar{c}, \bar{d}) of the cubic voxel is a diagonal of the face (\bar{c}, \bar{d}) of the rhombic dodecahedral voxel. (b) The face (\bar{c}, \bar{e}) of a cubic voxel is produced when the two voxels \bar{c} and \bar{e} in G_Δ are ω -adjacent. (c) The face (\bar{c}, \bar{d}) of a rhombic dodecahedral voxel is produced when the two voxels \bar{c} and \bar{d} in G_Δ are $(\varpi - \omega)$ -adjacent. 77

4.7 Superimposition of directions on voxel faces. 78

4.8 Definition of bel-adjacencies for cubic voxels. (The light faces are bel-adjacent from the dark faces.) 79

4.9 Three-dimensional display of the DnaB·DnaC approximated as a collection of rhombic faces from rhombic dodecahedral voxels at points of $F_{1.68}$ 83

4.10 Three-dimensional display of the DnaB·DnaC (reconstructed by ART using the same parameters as in Fig. 3.10(b)) approximated as (a) a collection of 82,310 faces of cubic voxels at points of $G_{0.25}$ and (b) a collection of 63,769 faces of rhombic dodecahedral voxels at points of $F_{0.42}$. These images were created with the same threshold value used in Fig. 4.3(a). . . . 84

4.11	Three-dimensional display of the bR (reconstructed by ART using the same parameters as in Fig. 3.10(b)) approximated as (a) a collection of 82,310 faces of cubic voxels at points of $G_{0.25}$ and (b) a collection of 63,769 faces of rhombic dodecahedral voxels at points of $F_{0.42}$. These images were created with the same threshold value used in Fig. 4.3(b).	85
5.1	Configuration of six subcomponents (ellipsoids) to form a simulated complex macrostructure. The simulated complex macrostructure is formed from two different repeating units: S_f and S_t . We show in (a) a view perpendicular to the plane formed by the six major axes of the subunits and in (b) a view almost parallel to such a plane, respectively.	106
5.2	Sequences of images showing the central slices for (i) a discretized reconstructed density function and (ii) segmentation using the fuzzy segmentation described in Section 5.1.2 of the reconstructed density function. For the fuzzy segmentation results, the hue indicates the object to which the voxel belongs (i.e., the m such that $\sigma_m^c = \sigma_0^c > 0$), in (ii) the red is for the macromolecule and the green is for the background. The intensity indicates the grade of membership (it is proportional to σ_0^c). We present results for (a) a mathematical phantom and (b) the bacteriorhodopsin.	108
5.3	(a) Segmentation using simple thresholding based on the volume of the mathematical phantom and (b) result of finding subunits, creating models for them and substituting the models in place of the subunits.	111
5.4	We show in (a) the separation of the bacteriorhodopsin's segmentation into three subunits by the subunit-averaging method. We show in (b) and (c) the result of the method with different views.	111
5.5	Inaccuracies of representations of density functions by (dotted line) thresholding based on the volume of the density function and (continuous line) subunit-model approach. We show in (a) the results after 40 experiments and in (b) after 160 experiments.	112
5.6	(a) Segmentation using simple thresholding based on the volume of the bacteriorhodopsin and (b) result of averaging the three subunits.	114
B.1	(a) Scheme of the seven α -helices forming the bacteriorhodopsin, see text, (reprinted with permission from the author and Nature [66] copyright 2000 Macmillan Publishers Ltd.) (b) View of the arrangement of three Bacteriorhodopsins as they would appear from within the cytoplasm. (c) View of the three bacteriorhodopsins, represented by ribbons, using the model of atoms available in the Protein Data Bank [10].	130

- B.2 Rendering of the bacteriorhodopsin at the desired threshold from perfect projections at 9 Å resolution. This display has the same orientation and volume as that shown in Fig. 4.3(b). This image will serve as the representation of *truth*. 131

Chapter 1

INTRODUCTION

Medical imaging has provided physicians with new tools to diagnose or plan treatment on patients with non-invasive or almost non-invasive techniques. Nowadays the imaging techniques in use in medicine are capable of producing two-dimensional or three-dimensional images from the human body. These devices measure the interaction of organs and tissues with some type of energy, e.g., Computerized Tomography uses X-rays, Positron Emission Tomography uses positrons. The measurements are treated as line integrals, of some spatially-varying physical parameter which is related to the local interactions of the tissue with the energy, through the body from the source to the detector. The distribution of the spatially-varying physical parameter is recovered from these measurements. Medicine and biology also rely on other imaging techniques to visualize microorganisms, pathogens and some organelles. Technology has even provided tools for the dissection of objects at the micro-scale.

At the micro-scale and the body-scale humankind has acquired sufficient knowledge

to recognize structures and shapes so that it is possible to distinguish between a “normal” structure and an “irregular” one. Thus, it is possible to distinguish between a healthy bone and osteoporotic bone, or between a *staphylococcus* and a *vibro cholerae* bacterium from acquired images.

There are several mechanisms by which cells interact with each other, microorganisms and pathogens attack cells, cells protect the body, or organelles produce substances. Microorganisms, pathogens, organelles and the substances they produce are made of molecules that cannot be observed with optical devices. Therefore, it has been necessary to invent other instruments to obtain information on objects at a such small scale, in particular on macromolecules. However, many of the instruments capable of yielding images of nano-scale objects produce only two-dimensional images. For reasons that are exposed later in this dissertation, it is important to obtain the three-dimensional structure of objects at the nano-scale and consequently it is necessary to apply image processing techniques to the two-dimensional images to produce three-dimensional representations. Whereas at the body-scale and micro-scale it is possible to compare the images with the objects imaged, at the nano-scale it is impossible with the current technology to make such a comparison, see Fig. 1.1.

Images produced by a device are usually post-processed. There are two main reasons for the further processing of the images, the first being that imaging devices are not perfect and they introduce noise into the images. Another reason for processing is to compen-

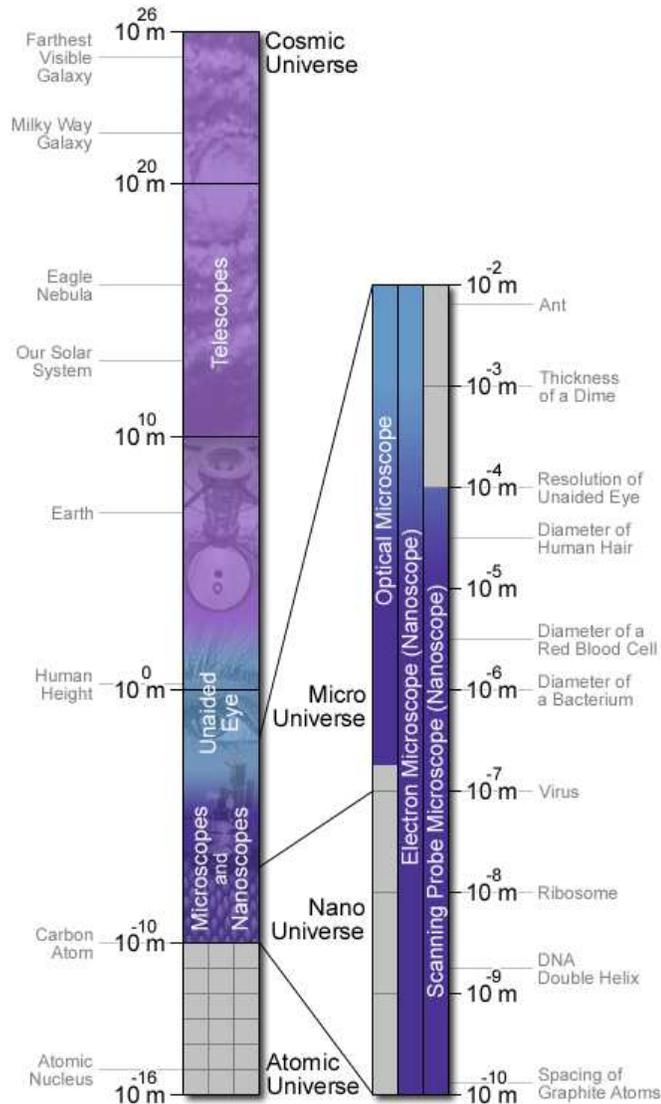


Fig.1.1: Relative size of several objects from the cosmic-scale to the nano-scale together with the instruments used to observe them (image courtesy of the INVSEE project at Arizona State University, Dr. B. L. Ramakrishna, Project Director, INVSEE).

sate for artifacts in computer-produced three-dimensional images. Finally, the user of the images may want to obtain higher level knowledge from the images, such as extracting meaningful portions of the objects or visualizing the objects.

In Chapter 2 we present a summary of the importance of studying macromolecules and proteins, and the methods and techniques used to obtain three-dimensional density functions, together with some of their limitations. In Chapter 3 we propose a method to improve the visualization of the computer representation of macromolecules and in Chapter 4 we propose methods to produce computer representations of the three-dimensional density functions by improving the raycasting method introduced in Chapter 3 and by an approximation, by polygons, to the surface of the density function. Finally, in Chapter 5 we propose to take advantage of the repetition of subunits that form some macromolecular complexes to improve the accuracy of their representation.

Chapter 2

BACKGROUND

Biological macromolecules are the main structural components from which living matter is built. This group of molecules consists of four broad subclasses: *proteins*, *nucleic acids*, *polysaccharides*, and *lipids*. All of these macromolecules exhibit a complex structure that in high degree determines their interaction with themselves or with others. In particular, proteins combine with lipids, nucleic acids, or polysaccharides to form essential compounds for most living matter. Proteins are different from the other three macromolecules in that they perform multiple functions such as transporters, storage elements, enzymatic catalyzers, structural elements, immunity elements, hormonal or genetic regulators. Proteins are capable of such a wide range of functions due to the uniqueness of their three-dimensional structure (*conformation*) and to their chemical properties [63, 68, 111, 124]. In fact, the appropriate operation of proteins depends fully on their specific conformations. Hence, the interest in knowing the structure of a protein is ultimately linked to the need of knowing its function.

2.1 Macromolecules: An Overview

Proteins can be analyzed at four hierarchical levels of complexity [63, 68, 111, 124]. Proteins are polymers composed of a sequence of any of 20 amino acids [63, 124]. Such a sequence, or chain, folds in itself in a very specific fashion to give the protein its conformation. The specific amino acid sequence forming the polypeptide chain is referred to as the *primary structure* of the protein. It is believed that the conformation of a protein is determined by its primary structure. The *secondary structure* refers to the spatial relation of neighboring amino acids that produce four types of local conformations: α -helices, β -strands, β -turns and random coils. The *tertiary structure* describes the spatial relation of more distant amino acids and hence describes the full three-dimensional (3D) structure of a single polypeptide chain. This conformation is the result of associations between segments of secondary structures. Sometimes proteins, called oligomeric, possess more than one polypeptide chain. The *quaternary structure* describes the spatial relationship between individual polypeptide chains in an oligomer, see Fig. 2.1.

The size of a single-chain protein can vary from 350 daltons (one dalton, denoted Da, corresponds to one atomic mass unit and is equivalent to $1.66053873 \times 10^{-24}$ g) to 1000 kDa, but the typical size of this type of protein is 10-50 kDa (because amino acids have molecular weights of the same order of magnitude, it is possible to determine the size of a protein by its molecular weight). In the case of oligomeric proteins it is frequent to find oligomers greater than 200 kDa.

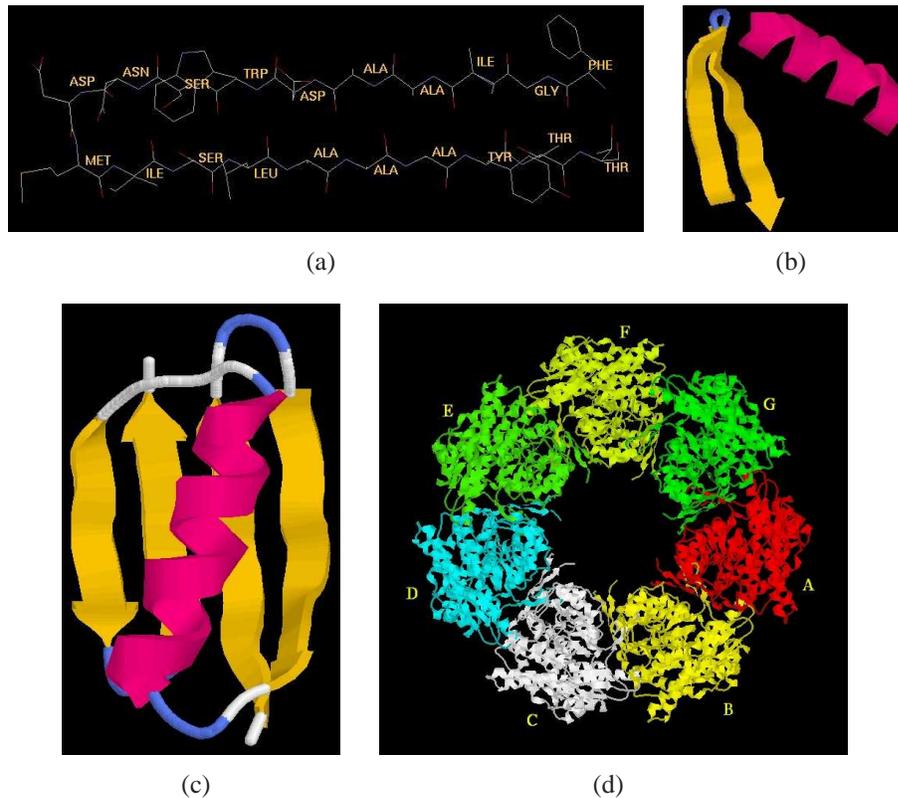


Fig.2.1: Schematic representation of protein structure: (a) Linear sequence of amino acids forms the primary structure, here forming the porin protein. (b) Local interactions among chains of amino acids form the secondary structure. It is common practice to represent β -pleated sheets as ribbon arrows, α -helices as helical ribbons and random coils as thin lines. (c) The 3D structure formed by the combination of secondary structures forms the tertiary structure, a good example of a single-chain protein is the protein G. (d) The combination of several polypeptide chains results in the quaternary structure. A clear example is the protein GroEL formed by seven polypeptide chains [124].

While the complex functions a protein should perform determine the synthesis of large-sized proteins, the synthesis of long polypeptide chains is prone to errors (e.g., about one in seven proteins with 3000 amino acids will not be translated in full in a prokaryotic cell [38]). Nature, however, determines the characteristics of a protein based on functional

requirements. In other words, whenever a function is required nature creates a protein, or a protein complex, to accomplish such a task. Thus, large proteins exist because they possess specific morphological and cooperative functions. It is important to notice that whenever a large protein is needed, nature favors its construction from several small chains (*heterooligomers*) or from identical chains (*homooligomers*). Constructing large proteins from small chains offers advantages in error control, coding efficiency, and regulation of assembly [38].

Interestingly, nature has favored the construction of highly symmetrical oligomeric proteins. Again, symmetry is a characteristic determined by the functional role of these proteins (e.g., the construction of structural elements such as containers, filaments or rulers, or the mechanism whereby the function of a protein is inhibited or stimulated by a molecule that binds to a specific site and causes a conformational change in the protein, called *allosteric regulation*). The symmetries found in proteins fall into the following categories:

- a. Cyclic groups. Those which contain a single axis of rotational symmetry.
- b. Dihedral groups. Those which contain an axis of rotational symmetry and a perpendicular axis of two-fold symmetry.
- c. Cubic groups. Those which contain three-fold symmetry combined with another, non-perpendicular rotational axis, with three possibilities: tetrahedral, octahedral, and icosahedral.

- d. Line, plane and space groups. Those which contain translational symmetry in addition to rotational symmetries.

In spite of the tendency of nature to create symmetrical oligomeric proteins, symmetry is broken or limited in many circumstances due to functional needs such as interaction with membranes or with DNA, or directional motion. However, nature attempts to preserve symmetrical arrangements by using either quasisymmetry (in which identical subunits adopt similar but different positions), pleomorphic symmetry (in which identical subunits are used to build several different structures), pseudosymmetry (in which similar but different subunits perform identical roles), or symmetry mismatch (in which oligomers of different symmetries interact along their respective symmetry axes).

2.2 Structure Analysis of Macromolecules by Transmission Electron Microscopy

Currently there are several techniques in biology to obtain the sequence of amino acids forming a polypeptide chain [63, 124]. Although it is recognized that the conformation of a protein is determined by its sequence of amino acids, it is still impossible to know accurately the form an amino acid sequence will take when it folds. Therefore, it has been necessary to develop techniques to obtain information regarding the higher structures of a macromolecule. In this respect, the development of X-ray diffraction brought the first attempts to determine the structure of biological molecules. However, it has been the invention of the electron microscope which made possible direct imaging of biological

structures at a macromolecular level, and more recently at atomic resolution, although the latter only under very specific circumstances [31, 67].

Traditionally, there have been three technologies used to obtain information about the structure of a macromolecule: X-ray crystallography, nuclear magnetic resonance (*NMR*) spectroscopy and electron microscopy (*EM*), the latter being a technique that complements X-ray crystallography and NMR spectroscopy [24].

X-ray crystallography was the first method applied to analyze protein structure [64, 96]. X-ray crystallography takes advantage of the diffraction of X-rays caused by crystals. A crystallized sample is irradiated by X-rays producing a pattern of spots whose intensity and position are easily determined. An important limitation of X-ray crystallography is the lack of phase information, such information must be obtained either from the diffraction data or from supplementing techniques [24, 101]. A 3D map of the electron cloud of the molecule (*electron density map*) is produced by combining both the amplitude and phase information. The exact position of every non-hydrogen atom in the molecule is obtained by fitting the sequence of amino acids forming the macromolecule, information that has to be obtained independently, to the electron density map [101]. Another important limitation is the need to crystallize the samples and to produce singular crystals (not forming stacks) and of high quality, a task that is difficult and frequently impossible to achieve. Nevertheless, this technique continues to be the predominant tool for structure analysis of proteins. In the past X-ray crystallography was restricted to small oligomers, but now it is also employed

in the analysis of large proteins.

NMR is an important method for structural analysis of proteins in solution. NMR spectroscopy measures the spectra produced by the resonance of protons in a molecule. In order to determine the structure of a macromolecule by NMR spectroscopy, resonances must be assigned to specific amino acids in the primary structure of a macromolecule. This produces a two-dimensional spectrum whose peaks are correlated with the atomic groups of the macromolecule. The method does not generate a unique solution based on the given constraints; however, some configurations can be discarded by physical or chemical limitations. As a final result, NMR provides the structure of a protein under study (NMR provides a detailed structural and conformational information regarding the protein). Generally, NMR is limited to macromolecules of low molecular weight, ≈ 30 kDa [19], mainly due to the complexity of the spectrum generated by large proteins.

Nowadays new technologies with quasi atomic resolution, such as atomic force microscopy (*AFM*) or scanning tunnel microscopy (*STM*), provide scientists with a greater set of tools for the analysis of protein structure. Yet, the transmission electron microscope (*TEM*) offers several advantages compared to the techniques mentioned before: it provides information on the phase and amplitude of the Fourier transform, it does not require crystallized specimens and it does not require large amounts of proteins for their analysis [24]. Furthermore, the TEM can be used for a wide range of resolutions, from low resolution, in the thousands of angstroms, to atomic resolution, ~ 2 angstroms, [47, 67, 88] (an angstrom

is denoted by Å and is equivalent to 10^{-10} meters). Hence TEM is an important complementary instrument to the high resolution techniques. It is worth mentioning that recently there have been efforts to combine information from the different technologies mentioned above [6].

2.3 The Transmission Electron Microscope

It is possible to think of a TEM as a large light microscope turned upside down that uses an electron beam instead of a photon beam. In a TEM the specimens are irradiated with an electron beam of uniform current (the electron energy is in the range 60-150 keV, or 200 keV - 3 MeV in the case of high voltage electron microscope [100]), such a beam is produced by the *electron gun* located in the top region of the microscope, at about two meters high, see Fig. 2.2.

A condenser lens system permits the variation of the area of illumination and, together with the electron gun, produces a beam of almost parallel electron rays. The electrons continue in their paths until they hit the specimen and interact with it (we explain below the interaction of the specimen with the electron radiation), but most of the electrons continue in their paths passing through a three- or four-stage electromagnetic lens system that focus the electron beam on a fluorescent screen at the bottom of the microscope. Such a screen produces an optical signal, the final image or *micrograph*, that is captured by either a photographic film or, more recently, a cooled charge-coupled device (*CCD*) camera (in modern

microscopes with resolution of 2048×2048 pixels).

The specimen is mounted on a circular support with typically 3 mm of diameter. The specimen must be held very stationary, since any drift or vibration of the specimen will appear in the final image. In spite of the need to keep a stable support, it is necessary to have access to all the area of the support and to introduce the specimen into the TEM without altering the vacuum inside the microscope. Furthermore, specimen tilting is needed in order to collect images at different orientations. The device that allows these operations is called the *specimen stage*, and together with the *airlock* (a small chamber into which the specimen initially enters and which can be evacuated before the specimen enters the TEM) are the most mechanically complex and precision-demanding parts of a TEM. Typically, the specimen stage consists of a rod-like holder, supporting the specimen at the end of it,

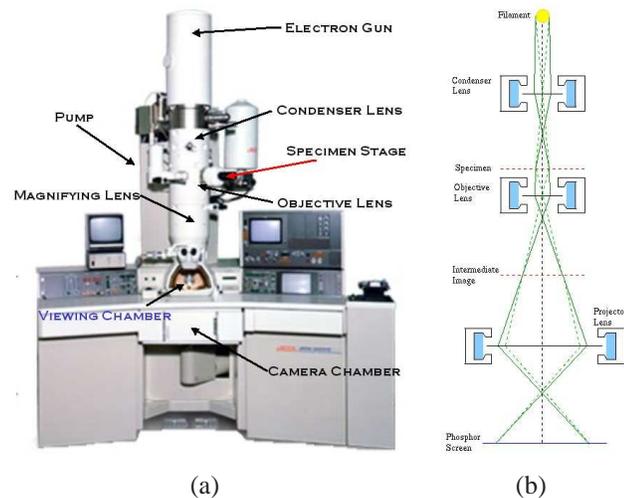


Fig.2.2: (a) Actual configuration of a JEOL[©] 2010 F TEM. (b) Schematic operation of a TEM.

that is inserted horizontally through the airlock.

Theoretically the resolution of a TEM is determined by the wavelength of the electrons in the beam. For a typical energy of 100 keV the wavelength of an electron is 0.004 nm which would result in a resolution of 0.002 nm. However, the lens aberrations are so great, and difficult to correct, that it is necessary to use very small objective apertures to achieve resolutions of 0.1-0.5 nm (1-5 Å).

2.3.1 Specimen Preparation

The electrons in the beam interact with any object along its path, which includes air particles. Consequently, the interior of the microscope has to be kept in vacuum, at least 10^{-5} torr (1 torr $\approx 133.322368421 \frac{\text{N}}{\text{m}^2}$) [77]. An undesirable side effect of preserving vacuum inside the microscope is the need to extract the water from the biological specimens. Unfortunately the extraction of water from a biological specimen leads to the destruction of the three-dimensional structure of the specimen.

Moreover, the biological specimens are extremely sensitive to electron radiation. The radiation from the electron beam causes the formation of free radicals and ions in the specimen. Damage to the secondary structure occurs at an electron dose of less than 1 electron per square angstrom ($e/\text{Å}^2$), and further exposure causes the reorganization of the tertiary structure involving the loss of specific groups and alteration of structural composition. In the end, the specimen loses between 10% and 30% of its mass during one exposure of 10-

$100 e/\text{\AA}^2$ [77, 100]. Several techniques have been developed to prevent excessive damage to the specimens and to compensate for the extraction of water from the specimen. Among these techniques, we mention two that have been successful in preserving the specimens during exposure: *negative staining* and *cryomicroscopy*.

Negative staining is the most successful technique for medium resolution. In this technique the specimen is coated with a thin layer of heavy metal salt. The heavy metal salt covers the specimen forming a cast of the protein's exposed surface. This technique helps to preserve the surface shape, but unfortunately it also prevents the information from the specimen's interior from appearing in the final image. The preparation of specimens by this technique is relatively fast and easy, but it limits the resolution to the size of the salt grains in use. Even though the technique avoids to a certain extent the effect of shrinkage and deformation due to the electron-specimen interaction, it is impossible to completely avoid these side effects and the specimen suffers deformations and dehydration that appear as artifacts in the final image.

Cryomicroscopy is an important development that overcomes the limitations of negative staining [26, 27]. In this technique the specimen in solution is rapidly frozen to achieve an amorphous state (most of the current cryomicroscopy work is based on vitreous ice [123]) and, therefore, to prevent the water from forming crystals and disrupting the specimen structure. It is in this frozen state that the specimen is introduced into the TEM. While the technique maintains the hydrated state of the specimen and, more importantly, preserves

the internal characteristics and structure of the specimen; the images produced with it have lower contrast than those produced with negative staining due to the small difference in density between water and protein. Finally, there are important issues that limit the use of this technique such as the need for specialized equipment, the higher level of complexity of the technique as compared to negative staining, the high degree of precision required in the preparation of specimens, and the need for technicians with refined skills.

The technical limitations of the TEM together with restrictions imposed by the preparation of the specimens, the radiation damage to the specimens, and the reduced contrast result in a practical resolution in the images of around 10 Å [31, 77].

2.3.2 Image Formation in the TEM

When the beam arrives at the sample stage most of the electrons have traversed the sample without interacting with it at all (*unscattered electrons*). The electrons interacting with the specimen, and responsible for the image formation, suffer basically two types of scattering: *elastic*, the most frequent, and *inelastic*. In the elastic scattering the electrons do not lose their energy and only suffer a change in phase by the Coulomb force at the nuclei of the specimen atoms. This type of interaction produces *phase contrast* information. The *amplitude contrast* is produced by the loss of energy of those electrons in the beam which interact inelastically with those in the specimen [100]. An important side effect of inelastic scattering is that it deteriorates the image in the low frequencies (a fact that is normally

tolerated), which prevents a quantitative interpretation of the images [109, 110].

The model of a phase contrast image considers the thickness of the specimen preparation to be a stack of several layers of material. Thus, the incident electrons pass through the layers suffering elastic and inelastic interactions in every layer. The accumulated effect of electrons interacting with the layers of preparation can be considered a parallel-beam projection of the specimen only when thin samples and high-energy electrons are utilized (this results in the inelastic scattered electrons preserving their trajectory) [31, 77]. As a consequence, in order to produce high resolution images it is necessary to use high-energy electrons together with thin specimens, thus producing mainly phase contrast information.

A simple model for the formation of an image using a TEM is that the recorded image in the camera chamber is approximately the result of the ideal projection (line integrals) of the specimen convolved with the point spread function (*psf*) of the microscope. We can express this process by

$$y \simeq h * (P\upsilon + \eta_p) + \eta_l, \quad (2.1)$$

where y represents the measured image, $P\upsilon$ represents the ideal projection image of the density function υ , and h represents the *psf* of the microscope. The final image is contaminated with noise at the specimen stage, represented by η_p , and at the lens-system stage, represented by η_l [8, 53, 125]. In electron microscopy it is common to refer to the Fourier transform of h as the *contrast transfer function* or *CTF*. An important characteristic of the

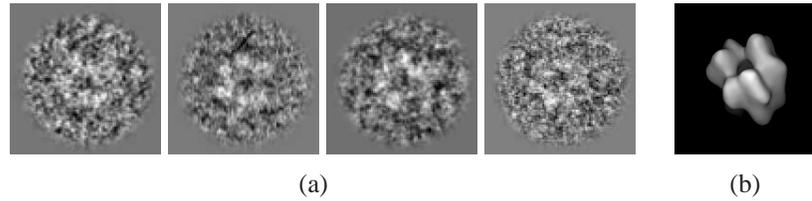


Fig.2.3: (a) A series of TEM images (from [8]) obtained for the study of the protein complex DnaB·DnaC which is shown in (b). The images demonstrate the low signal-to-noise ratio present in a micrograph.

psf is that its central peak is negative, which results in the zeroth-order approximation of the projection $P\psi$ becoming the negative of the y in (2.1). (Mathematical concepts used in this paragraph, such as convolution and Fourier transform, are discussed in the next section on Mathematical Background).

The damage caused to the specimen by electron radiation is one of the most important problems in electron microscopy when high resolution is desired. A common solution is the use of low doses of electron radiation: a dose of $0.5-4 e/\text{\AA}^2$ is appropriate for very high resolution studies of unstained crystalline specimens [3] and a dose of $10-20 e/\text{\AA}^2$ is typical for stained or non-crystalline specimens. The use of low radiation doses translates into a low signal-to-noise ratio (SNR) in the images, see Fig. 2.3. In order to minimize damage to the specimen while focusing the microscope, the so-called minimum dose system (MDS) has been developed based on the assumption that neighboring areas of the preparation have similar number of macromolecules with the desired properties. Consequently, low magnification (i.e., low radiation dose) is utilized when searching for a region of the preparation containing a “good” distribution of macromolecules. Once this area of the preparation has

been located, the final image is acquired by focusing in a contiguous region at the desired magnification.

A common approach to overcome the poor SNR is to align and then average the individual images of individual non-crystallized macromolecular assemblies (*single particles*) in an image; thus, taking advantage of the high similarities between macromolecules with the same primary structure (they are identical copies that have congruence) and the fact that a preparation is largely made of a single type of specimen (purified protein). Clearly, the heterogeneities among members of the population almost vanish by the typical averaging of thousands of particles. However, there are many factors that affect the alignment process such as the shrinkage and deformation of the specimen, the presence of impurities or heterogeneities in the preparation, and the wide range of orientations a single particle can have on the support.

2.4 Mathematical Background

We make an important diversion in order to introduce some mathematical concepts that will be required later on in this dissertation. For this, we follow the notation and theorems in [55] and [87]. We make use of the set of real numbers, denoted by \mathbb{R} , as the information provided by an imaging device is typically composed of real numbers. We also make use of the set of integers, denoted by \mathbb{Z} , as we work with discretized objects. Finally, we make use of the set of complex numbers, denoted by \mathbb{C} , to represent measurements of magnitude

and phase.

All through this dissertation we will use spaces of several dimensions; for example, we work with density functions in three dimensions and images in two dimensions. We represent a point in the continuous n -space, where n is the dimension of the space, by a vector \bar{x} , which is n -tuple (x_1, x_2, \dots, x_n) whose components belong to \mathbb{R} ; the set of all such vectors forms the space \mathbb{R}^n . We also represent points in the discrete n -space by a vector \bar{z} whose components belong to \mathbb{Z} ; similarly, the set of all such vectors forms the space \mathbb{Z}^n . We define the inner product between vectors \bar{x} and \bar{y} in \mathbb{R}^n as $\langle \bar{x}, \bar{y} \rangle = \sum_{j=1}^n x_j y_j$ and the magnitude of a vector \bar{x} as $|\bar{x}| = \sqrt{\langle \bar{x}, \bar{x} \rangle}$. We also represent points $\bar{x} \in \mathbb{R}^3$ by spherical coordinates (r, ϕ, θ) , where $r = |\bar{x}|$.

An important subset of \mathbb{R}^n consists of all those vectors which have unit magnitude; it is defined as $S^{n-1} = \{\bar{x} \mid |\bar{x}| = 1 \text{ and } \bar{x} \in \mathbb{R}^n\}$. Elements of S^{n-1} are directions in \mathbb{R}^n ; we will often use the notation \vec{o} for them.

Every line in n -space is of the form $\{\bar{x} + \tau \vec{o} \mid \tau \in \mathbb{R}\}$ for some uniquely defined $\vec{o} \in S^{n-1}$ and $\bar{x} \in \vec{o}^\perp$, where $\vec{o}^\perp = \{\bar{x} \mid \langle \bar{x}, \vec{o} \rangle = 0\}$ is the hyper-plane perpendicular to \vec{o} ; consequently, the space of lines in n -dimensional space is represented by $T^n = \{(\vec{o}, \bar{x}) \mid \vec{o} \in S^{n-1}, \bar{x} \in \vec{o}^\perp\}$.

We will be making essential use of certain properties of functions and of generalized functions also called distributions). A mathematically precise theory of these in the context of image reconstruction from projections is provided in [55]. We do not repeat that material here, but provide a minimal outline which is necessary for understanding what follows.

The functions that we utilize map the sets \mathbb{R}^n , \mathbb{Z}^n and T^n into either \mathbb{R} or \mathbb{C} . In order for the theorems that we state to be valid, the classes of functions need to be restricted to those which have special properties. One such class is the *Schwartz space* [87] $\mathcal{S}(\mathbb{R}^n)$, whose elements are the rapidly decreasing and infinitely differentiable functions from \mathbb{R}^n into \mathbb{C} [55]. Another important class that we need is $\mathcal{S}'(\mathbb{R}^n)$, the space of *tempered distributions* which are continuous linear forms [55] on $\mathcal{S}(\mathbb{R}^n)$; implying, in particular, that a $T \in \mathcal{S}'(\mathbb{R}^n)$ associates with a function $f \in \mathcal{S}(\mathbb{R}^n)$ a complex number, denoted by Tf (for now). A famous example is provided by the *Dirac function* δ , defined by

$$\delta f = f(\bar{0}), \tag{2.2}$$

for any $f \in \mathcal{S}(\mathbb{R}^n)$. ($\bar{0}$ is the n -dimensional vector of all zeros.)

Now consider any function φ over \mathbb{R}^n and define the linear form T_φ by

$$T_\varphi f = \int_{\mathbb{R}^n} \varphi(\bar{x}) f(\bar{x}) d\bar{x}, \tag{2.3}$$

for any $f \in \mathcal{S}(\mathbb{R}^n)$. If the integral on the right exists for all $f \in \mathcal{S}(\mathbb{R}^n)$, then T_φ is defined.

It is a useful fact that whenever $\varphi \in \mathcal{S}(\mathbb{R}^n)$, then T_φ is defined and is in fact in $\mathcal{S}'(\mathbb{R}^n)$. The same is true for φ having other properties such as being square integrable (i.e., $\varphi \in L_2(\mathbb{R}^n)$, using the notation of [87]) or slowly increasing (see [55], pages 54-55). In such a case we say that φ itself is a tempered distribution and replace the notation on the left hand of (2.3)

by that on its right hand side. This notation can be conveniently extended to other tempered distributions, resulting in formulas such as

$$\int_{\mathbb{R}^n} \delta(\bar{x}) f(\bar{x}) d\bar{x} = f(\bar{0}) \quad (2.4)$$

to replace (2.2).

This notation is useful for understanding how various notions for functions are naturally extended to distributions. For example let, for any function f over \mathbb{R}^n and any $\bar{y} \in \mathbb{R}^n$, $f_{\bar{y}}$ denote the function over \mathbb{R}^n defined by $f_{\bar{y}}(\bar{x}) = f(\bar{x} - \bar{y})$. Then for $\phi \in \mathcal{S}'(\mathbb{R}^n)$ and $\bar{y} \in \mathbb{R}^n$, $\phi_{\bar{y}}$ denotes the tempered distribution which is defined by $\phi_{\bar{y}} f = \phi f_{-\bar{y}}$, for any $f \in \mathcal{S}(\mathbb{R}^n)$.

The appropriateness of this is indicated by

$$\begin{aligned} \phi_{\bar{y}} f &= \int_{\mathbb{R}^n} \phi_{\bar{y}}(\bar{x}) f(\bar{x}) d\bar{x} \\ &= \int_{\mathbb{R}^n} \phi(\bar{x} - \bar{y}) f(\bar{x}) d\bar{x} \\ &= \int_{\mathbb{R}^n} \phi(\bar{z}) f(\bar{z} + \bar{y}) d\bar{z} \\ &= \int_{\mathbb{R}^n} \phi(\bar{z}) f_{-\bar{y}}(\bar{z}) d\bar{z} = \phi f_{-\bar{y}}. \end{aligned} \quad (2.5)$$

A special consequence of this is that

$$\delta_{\bar{y}} f = \delta f_{-\bar{y}} = f_{-\bar{y}}(\bar{0}) = f(\bar{y}). \quad (2.6)$$

It is also clear that if $\varphi \in \mathcal{S}(\mathbb{R}^n)$ or $\varphi \in \mathcal{S}'(\mathbb{R}^n)$, then

$$(\varphi_{\bar{y}})_{\bar{z}} = \varphi_{\bar{y}+\bar{z}}, \quad (2.7)$$

for any \bar{y} and \bar{z} in \mathbb{R}^n .

In what follows we will be defining operators and state theorems about “functions” (including tempered distributions). We will not be precise as to the exact classes for which the operators are defined or in which the theorems are valid; such precision can be obtained from [55] (and to a lesser extent from [87]). Since our aim is an approximate reconstruction of a bounded object from noisy samples of its projections, one can always embed the material presented into an appropriate rigorous mathematical theory.

A most important linear operator is the Fourier transform which maps a function g over \mathbb{R}^n into a function \hat{g} over \mathbb{R}^n defined by

$$\hat{g}(\bar{\xi}) = (2\pi)^{-n/2} \int_{\mathbb{R}^n} g(\bar{x}) e^{-i\langle \bar{x}, \bar{\xi} \rangle} d\bar{x}, \quad (2.8)$$

for all $\bar{\xi} \in \mathbb{R}^n$. If g is in $\mathcal{S}(\mathbb{R}^n)$, then so is \hat{g} (Theorem 2.28 of [55].) Using this fact, the definition of the Fourier transform can be extended to a tempered distribution T as follows:

\hat{T} is the tempered distribution such that for any $f \in \mathcal{S}(\mathbb{R}^n)$, $\hat{T}f = T\hat{f}$ (see (2.64) in [55]).

It is easy to show that this definition is consistent with (2.3) in the sense that, for any

$$\varphi \in \mathcal{S}(\mathbb{R}^n), \quad \widehat{T\varphi} = T\widehat{\varphi}.$$

For each $\bar{y} \in \mathbb{R}^n$, the sinusoidal function $S^{\bar{y}}$ is defined by

$$S^{\bar{y}}(\bar{x}) = e^{-i\langle \bar{x}, \bar{y} \rangle}, \quad (2.9)$$

for all $\bar{x} \in \mathbb{R}^n$. Note that $S^{\bar{0}}$ is the function whose value is always 1. Then, using (2.3), (2.4) and (2.8), we can easily derive that

$$\widehat{\delta} = (2\pi)^{-n/2} S^{\bar{0}}, \quad (2.10)$$

i.e., a constant valued function.

We also define the inverse Fourier transform \check{g} of a function g over \mathbb{R}^n by, for all $\bar{x} \in \mathbb{R}^n$,

$$\check{g}(\bar{x}) = (2\pi)^{-n/2} \int_{\mathbb{R}^n} g(\bar{\xi}) e^{i\langle \bar{x}, \bar{\xi} \rangle} d\bar{\xi} \quad (2.11)$$

and \check{T} of a tempered distribution T by $\check{T}f = T\check{f}$. These are indeed inverses of the Fourier transform in the usual sense (e.g., T is the inverse Fourier transform of \hat{T}).

Similarly to functions defined over \mathbb{R}^n , the Fourier and inverse Fourier transforms for a function defined on T^n , are given by

$$\hat{g}(\bar{\sigma}, \bar{\xi}) = (2\pi)^{\frac{1-n}{2}} \int_{\bar{\sigma}^\perp} g(\bar{\sigma}, \bar{x}) e^{-i\langle \bar{x}, \bar{\xi} \rangle} d\bar{x} \quad (2.12)$$

and

$$g(\vec{\sigma}, \bar{x}) = (2\pi)^{\frac{1-n}{2}} \int_{\vec{\sigma}^\perp} \hat{g}(\vec{\sigma}, \bar{\xi}) e^{i\langle \bar{x}, \bar{\xi} \rangle} d\bar{\xi} \quad (2.13)$$

where $\vec{\sigma} \in S^{n-1}$ and $\bar{\xi}, \bar{x} \in \vec{\sigma}^\perp$.

The Fourier transform has important properties that we will use in this dissertation. The shifting property of the Fourier transform is that, for all $\bar{y} \in \mathbb{R}^n$,

$$\widehat{f_{\bar{y}}} = S^{\bar{y}} \times \hat{f}. \quad (2.14)$$

In case \hat{f} is a function, the right hand side of (2.14) denotes just a pointwise multiplication of two functions. It is also meaningful in case \hat{f} is a distribution, as is defined in (A.1) of the Appendix A.

The scaling property is that, if $\lambda f(\bar{x}) = f(\lambda \bar{x})$ and $\lambda \in \mathbb{R}$, then

$$\widehat{\lambda f}(\bar{\xi}) = \lambda^{n-1} \hat{f}(\bar{\xi}). \quad (2.15)$$

An important operation on functions is convolution; it is defined between two functions

f and g over \mathbb{R}^n by

$$[f * g](\bar{x}) = \int_{\mathbb{R}^n} f(\bar{x} - \bar{y}) g(\bar{y}) d\bar{y}, \quad (2.16)$$

or between two functions g and h over T^n by

$$[g * h](\vec{o}, \bar{x}) = \int_{\vec{o}^\perp} g(\bar{x} - \bar{y}) h(\bar{y}) d\bar{y}. \quad (2.17)$$

The convolution of a distribution φ with a function g is a function whose value for any \bar{x} is given by

$$[\varphi * g](\bar{x}) = [\varphi_{-\bar{x}}]_{-1} g, \quad (2.18)$$

see (2.28) of [55]. That this is a natural extension of (2.16), is easily seen by using the interpretation provided by (2.3). In particular, (2.18) combined with (2.6) and (2.7), yields that

$$\delta_{\bar{y}} * g = g_{\bar{y}}. \quad (2.19)$$

In words, this can be described by saying that convolving a function with $\delta_{\bar{y}}$ is the same as “shifting” it to \bar{y} .

This notion can be extended as follows. Let G be a set of points in \mathbb{R}^n and define the

distribution III_G by

$$\text{III}_G f = \sum_{\bar{y} \in G} f(\bar{y}). \quad (2.20)$$

An alternative way of describing the same idea is $\text{III}_G = \sum_{\bar{y} \in G} \delta_{\bar{y}}$, and so III_G is referred to as a *train of pulses* on the grid G . Then, for any function g ,

$$\text{III}_G * g = \sum_{\bar{y} \in G} g_{\bar{y}}. \quad (2.21)$$

In words, convolving a function with III_G results in the sum of the copies of g shifted to the points G .

An important relationship between the convolution of two functions and their Fourier transforms is given by the following theorem.

Theorem 2.1 (Convolution) *Let f and g be two functions over \mathbb{R}^n , then*

$$\widehat{f * g} = (2\pi)^{n/2} \hat{f} \times \hat{g} \quad \text{and} \quad \widehat{f \times g} = (2\pi)^{-n/2} \hat{f} * \hat{g}. \quad (2.22)$$

If f and g are two functions over T^n , then

$$\widehat{f * g} = (2\pi)^{(n-1)/2} \hat{f} \times \hat{g} \quad \text{and} \quad \widehat{f \times g} = (2\pi)^{(1-n)/2} \hat{f} * \hat{g}. \quad (2.23)$$

We consider a projection (a micrograph) to be a collection of line integrals, see Fig. 2.4. For a function \mathfrak{v} over \mathbb{R}^n (in our application this is the density function describing a macromolecule) and for any $\vec{\sigma} \in S^{n-1}$ and $\bar{x} \in \vec{\sigma}^\perp$, we define

$$[\mathcal{P}\mathfrak{v}](\vec{\sigma}, \bar{x}) = \int_{\mathbb{R}^1} \mathfrak{v}(\bar{x} + \tau\vec{\sigma}) d\tau. \quad (2.24)$$

The operator \mathcal{P} is called the *ray transform*, since $\mathfrak{v}(\bar{x} + \tau\vec{\sigma})$ provides, as τ varies, all the densities along a line. The function $\mathcal{P}\mathfrak{v}$ is defined over the space T^n . If $\mathfrak{v} \in \mathcal{S}(\mathbb{R}^n)$, then $[\mathcal{P}\mathfrak{v}] \in \mathcal{S}(T^n)$ [87]. The ray transform of \mathfrak{v} is related to its Fourier transform by the following theorem.

Theorem 2.2 (Central Slice) *Let \mathfrak{v} be a function over \mathbb{R}^n . Then for $\vec{\sigma} \in S^{n-1}$ and $\bar{\xi} \in \vec{\sigma}^\perp$,*

$$[\widehat{\mathcal{P}\mathfrak{v}}](\vec{\sigma}, \bar{\xi}) = \sqrt{2\pi} \hat{\mathfrak{v}}(\bar{\xi}). \quad (2.25)$$

For a fixed $\vec{\sigma}$, the values of $\bar{\xi}$ all lie in the hyper-plane $\vec{\sigma}^\perp$ perpendicular to $\vec{\sigma}$ which crosses the origin (a central hyper-plane of the Fourier transform of the density function \mathfrak{v}). As a consequence, by taking the ray transform for every orientation, it is possible to recover the Fourier transform of the density function \mathfrak{v} and then to recover the density function itself by means of the inverse Fourier transform.

The *backprojection* operator \mathcal{B} , which maps a function g over T^n into a function from

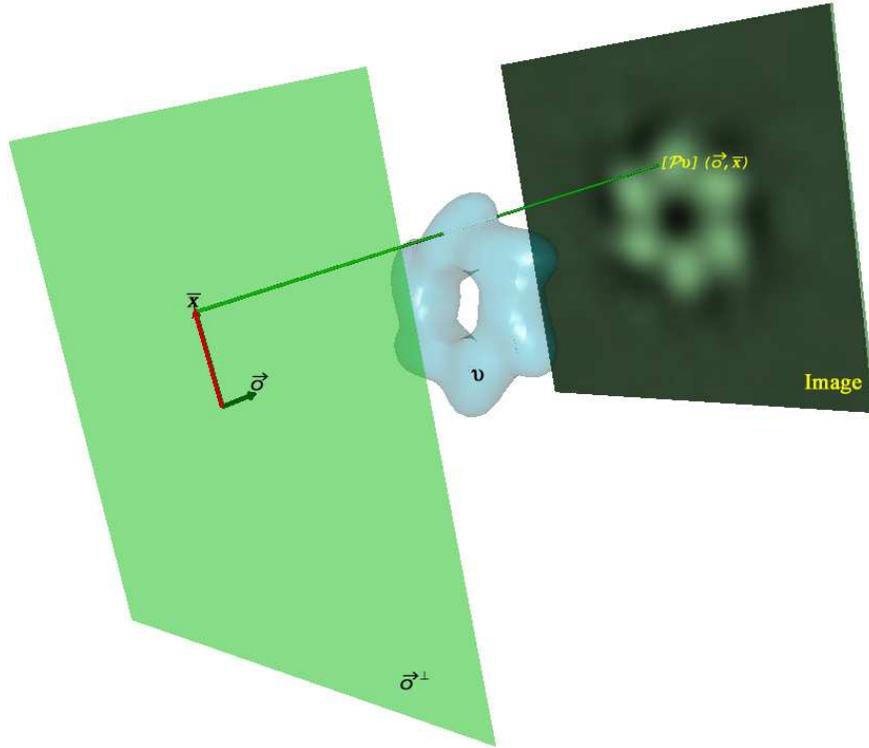


Fig.2.4: Under specific conditions, a micrograph can be considered a collection of line integrals $\mathcal{P}\mathfrak{v}$ crossing the specimen, denoted by \mathfrak{v} . Every line integral is determined by an orientation vector \vec{o} and a position vector $\bar{x} \in \vec{o}^\perp$.

\mathbb{R}^n into \mathbb{R} , is defined by

$$[\mathcal{B}g](\bar{x}) = \int_{S^{n-1}} g(\vec{o}, \bar{x} - \langle \vec{o}, \bar{x} \rangle \vec{o}) d\vec{o}. \quad (2.26)$$

For $g = [\mathcal{P}\mathfrak{v}]$ the backprojection of g is the average of the line integrals of \mathfrak{v} taken over all lines crossing the point \bar{x} . There is a relationship between the backprojection operator and the ray transform given by the following theorem.

Theorem 2.3 (Backprojection) *Let f be defined over \mathbb{R}^n and g over T^n , then*

$$[\mathcal{B}g] * f = [\mathcal{B}(g * [\mathcal{P}f])]. \quad (2.27)$$

2.5 Data Collection in TEM

In order to achieve a three-dimensional reconstruction from electron microscopic information it is necessary to collect several projections of the specimen from different directions. The techniques to collect micrographs in a TEM can be separated into those requiring tilts and those with ‘zero-tilt’ [123].

For techniques that require tilting the specimen, the different views of the specimen can be obtained by tilting it inside the TEM with the assistance of the goniometer, a piece of the specimen holder. Despite the fact that commercial goniometers can achieve tilting angles up to $\pm 80^\circ$, the thickness of the tilted specimen and the supporting grid bring an important limitation to the practical tilting angles that are useful in TEM, see Section 2.3.2 above. The classical tomography approach consists of acquiring many pictures of the same particle tilted into different orientations [57, 91]. However, a restriction on this collection scheme is the accumulation of radiation damage in the specimen. Another collection scheme in use is the *single axis tilting* (very useful when it is possible to crystallize a specimen or when the symmetrical structure of it is previously known). In this technique the specimen is tilted in small increments around only one axis and images are recorded at every step.

Another tilting-based method to collect micrographs that avoids, as much as possible, accumulative damage to the specimen and compensates somewhat for the limited tilt angle is the *random conical tilting*. Biological macromolecules interact with the specimen support at the time of preparation; such an interaction frequently results in the individual macromolecules adopting a few well-determined orientations on the specimen support. Random conical tilting is applicable to those macromolecules that exhibit an orientation which is specifically perpendicular to the plane of the support and is random parallel to that plane [98]. In this scheme many projections are determined by the tilting of the specimen (typically $45 - 60^\circ$) around an axis of the goniometer and by the different rotations of the individual macromolecules in the specimen around the support's normal axis. Significantly, only one exposure of the tilted specimen preparation is necessary to collect many different orientations. To obtain the information of the angle distribution parallel to the support's plane, a second exposure of the same area is taken without tilting the goniometer. The information from the tilted support is the one used for the 3D reconstruction process; it is the first to be acquired in order to minimize the damage to the specimen. After digitization, the individual-particle images must be isolated together with their respective orientation information. Interactive particle selection is still a common way to select the individual images and correlation techniques are the preferred methods to align the individual macromolecule images [31]. In practice the individual macromolecule images have to be aligned by in-plane translations and then by rotations; such alignments are not perfect, and they are

reflected in the final reconstruction's lower resolution.

Zero-tilt techniques also take advantage of the random orientations that particles assume inside the specimen preparation. The two main zero-tilt techniques are *projection matching* and *angular reconstitution*. In the projection matching method the projection orientations of the individual macromolecules are found by correlating the images (previously isolated) to projections of a reference density function; when the individual macromolecules adopt only a few well-determined orientations on the specimen, a suitable method to obtain the reference reconstruction is by the random conical tilting scheme. The angles and in-plane translations are refined by iteratively applying the projection matching method. Angular reconstitution is a more general method in the sense that it does not require a previous reference reconstruction. This method finds a common line in two projections of the 3D density function [112, 122, 123]. For this purpose, projections of all possible lines in the 2D projections, through the origin of the 3D object, are first calculated. These line projections can be displayed as a 2D image known as a *sinogram*. Every pair of line projections is cross-correlated to obtain the orientation of the common tilt axis between the two projections. After obtaining a first approximation to the orientations of the individual-macromolecule images, a reference reconstruction of the density function is obtained. The information of the individual image orientations can be refined by correlating the sinograms of images produced by projecting the reference density function. This procedure is iteratively refined until differences between the TEM-acquired images and the reference projection images

are negligible from a practical point of view.

2.6 Three-Dimensional Reconstruction from TEM Images

Under the conditions given in the previous sections, images produced by a TEM can be considered to be parallel projections of the specimens; an individual image can be considered to be a collection of line integrals with a given orientation. Although there is a direct inversion formula for (2.24), it is rarely used in practice [87]; practical approaches make use of properties of the ray transform that relate it to other transforms. We briefly introduce the principles of such approaches. Prior to introducing these principles, it is worth mentioning that an important consequence of the limited range of angles that the individual macromolecules adopt, or of the admissible tilting angle of the goniometer, is the failure to collect projection information in a region of Fourier space. For example, in the case of single axis tilting, the missing-information region is a wedge, and it is a cone for the random conical tilting. This lack of information is important because it is reflected in the final reconstruction, e.g., an elongation along the principal axis of the cone in the case of the random conical tilting.

One approach to reconstruction takes advantage of Theorem 2.2. When reconstructing by making use of this idea, it is necessary to take care of the missing data inherent to the imaging modality used for the data collection (e.g., projections are not available for certain ranges of $\vec{\sigma}$, see Section 2.5). Also, prior to taking the inverse Fourier transform

(commonly implemented assuming sampling on a Cartesian grid), interpolation is required from the sampling of Fourier space that is implied by the Central Slice Theorem.

Other practical methods to invert the ray transform include *Filtered Backprojection*, one of the most widely-used methods of reconstruction. The Filtered Backprojection method is based on Theorem 2.3 and the idea is to choose the function $\mathcal{B}g$ in (2.27) to be as close as possible to the delta function δ defined by (2.2); thus, by (2.19), the left hand side of (2.27) will result in a close approximation to the function f . An efficient way to perform the convolution on the right-hand side of (2.27) is by performing the operation in Fourier space (by making use of Theorem 2.1). Thus, we can take advantage of efficient discrete implementations of the Fourier transform, e.g., the *Fast Fourier Transform* [16, 17, 23, 99], to recover the density function. For this approach to work, it is necessary to find a g such that $\mathcal{B}g \approx \delta$. Many such functions have been proposed and studied in [49, 62, 86, 87].

The mathematical foundation of methods such as the one presented above assume the acquisition of projections for a continuum of orientations determined by the direction vector \vec{o} in (2.24), something that is impossible in practice. Yet, under certain conditions it is possible to recover an approximation to the density function \mathfrak{v} from a limited number of projections [49, 62, 86, 87], but a presentation of such an analysis is beyond the scope of this dissertation.

Filtered Backprojection belongs to the category of *transform methods* because it takes advantage of the ray transform and its relationship to other transforms, such as the Fourier

transform. Furthermore, these methods treat the reconstruction problem as a continuous one until the end, when an inversion formula is discretized in order to evaluate it on a computer. There are other methods that treat the reconstruction problem as a discrete problem from the beginning. These methods assume that any density function can be represented by a series expansion into basis functions; thus, these methods are called *series expansion methods*. An important example of such methods are the algebraic reconstruction techniques (*ART*) [40].

In the series expansion methods the density function is approximated by a linear combination of basis functions as

$$\mathbf{v}(\bar{x}) \approx \sum_{j=1}^J c_j b_j(\bar{x}), \quad (2.28)$$

where \bar{x} is a point in the space (in this work we work with points in \mathbb{R}^3 space), \mathbf{v} is the density function to be reconstructed, $\{b_j\}$ is the set of basis functions each of which is weighted by a coefficient c_j . While the basis functions are selected *a priori*, the coefficients have to be determined by the reconstruction method. As an example for (2.28) we refer to Fig. 2.5 in which a density function is represented by the donut-like object that has to be approximated by basis functions that have a unit value inside a cubic volume and zero outside (known as cubic voxels, in Subsection 4.3.1 we formally define them). In Fig. 2.5, the set of coefficients is represented by the small dots at the center of the cubic voxels. Such an approximation of a density function by cubic voxels is the most common manifestation

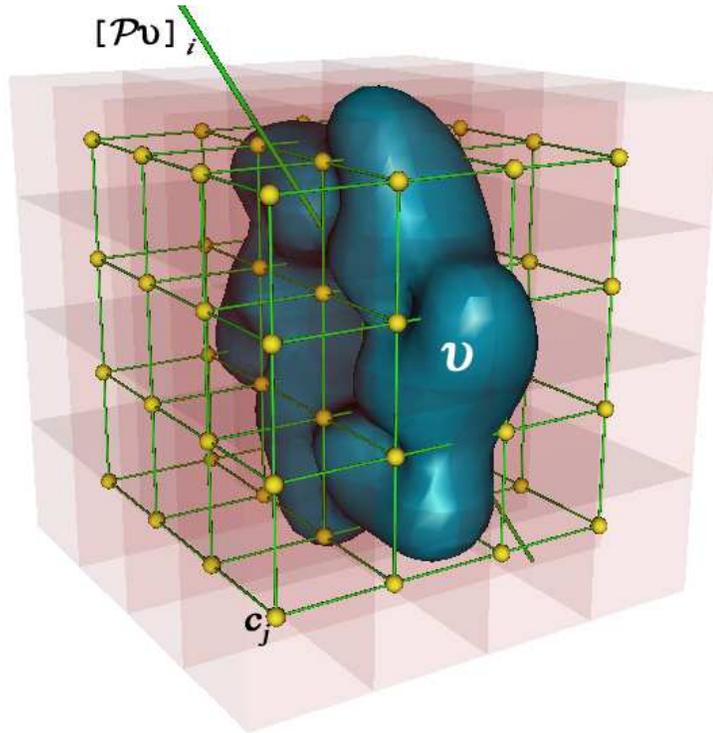


Fig.2.5: In the *series expansion methods* a density function v is approximated by the linear combination of basis functions. A common choice for basis functions are functions that have a unit value inside a cube and zero outside. To approximate a density function, each instance of the basis functions is weighted by coefficients c_j . Under this model, a line integral $[\mathcal{P}v]_i$ will intersect only a few basis functions.

of (2.28). Recently, functions that are radially symmetric and limited in space have been used as basis functions in (2.28) instead of cubic voxels. In Section 3.1 we discuss in detail a particular family of such functions, which will be referred to as *blobs* [69, 70, 73, 80]. In fact, only blobs are used in this dissertation for the basis functions of (2.28).

It follows from (2.24) and (2.28) that, for any $(\vec{o}, \bar{x}) \in T^n$,

$$[\mathcal{P}\mathbf{v}] (\vec{o}, \bar{x}) \approx \sum_{j=1}^J c_j [\mathcal{P}_v b_j] (\vec{o}, \bar{x}). \quad (2.29)$$

Suppose that our measurements are made for I lines, characterized by (\vec{o}_i, \bar{x}_i) , for $1 \leq i \leq I$.

Then we get

$$y_i \approx \sum_{j=1}^J \ell_{i,j} c_j \quad (2.30)$$

where y_i is the i th measurement and $\ell_{i,j} = [\mathcal{P}_v b_j] (\vec{o}_i, \bar{x}_i)$, a value which can usually be analytically determined (in case of cubic voxels, it is the length of the segment of the i th line inside the j th basis function). Equation (2.30) is approximately a system of linear equations $\bar{y} = \mathbf{L}\bar{c}$ that can be solved either by some sort of direct inversion or by an iterative method. In practice the system $\bar{y} = \mathbf{L}\bar{c}$ is often overdetermined (to compensate for the noise in the measurements, we typically take many more measurements than there are unknowns c_j).

A particular iterative method to solve the system $\bar{y} = \mathbf{L}\bar{c}$ is a generalization of Kaczmarz's method [61] for solving both over- and underdetermined linear systems of equations. For this generalization, we consider that the matrix \mathbf{L} can be divided in N blocks (n denotes the number of a block, $1 \leq n \leq N$) and there are M rows in each block and so

$I = M \times N$. The k th iterative step is

$$\bar{c}^{(k+1)} = \bar{c}^{(k)} + \lambda^{(k)} \sum_{i=(n-1)M+1}^{nM} \frac{y_i - \langle \bar{\ell}_i, \bar{c}^{(k)} \rangle}{\sum_{j=1}^J \ell_{i,j}^2} \bar{\ell}_i, \text{ for } n = [k \pmod{N}] + 1, \quad (2.31)$$

where $\bar{\ell}_i$ is a J -dimensional vector whose j th component is $\ell_{i,j}$ and $\lambda^{(k)}$ is a relaxation parameter which determines how much the solution is updated in the k th iteration. In practice, the blocks are chosen to correspond to a single projection (an electron micrograph). Thus $\bar{\ell}_i$ has the same value for all i 's in a single block. (N is the number of electron micrographs and M is the number of sample points in an electron micrograph.) This allows us a very efficient *simultaneous* calculation of $\langle \bar{\ell}_i, \bar{c}^{(k)} \rangle$, for $(n-1)M+1 \leq i \leq nM$, by using the ‘‘footprints’’ of the blobs [80], and makes the iterative algorithm specified in (2.31) quite fast for practical applications. In fact, it was reported in [73] that for reconstruction from data collected by the random conical tilt geometry, the algorithm based on (2.31) required 3.34 min, while a standard transform method needed 24.16 min to reconstruct from the same data on the same computer.

The ordering of the projections which results in the blocks used in the formula above is important for achieving an efficient performance [54]. The main idea is that consecutive blocks should introduce independent changes. This is achieved by ordering the projections so that in the resulting sequence of $\bar{\sigma}_n$ (the common $\bar{\sigma}_i$ associated with the n th block), each vector tends to be orthogonal to the ones immediately preceding it.

The relaxation parameters $\lambda^{(k)}$ are also of great practical importance, see [49, 50, 62, 76, 78, 87], and the optimal values depend on the quality of the projections, the data collection geometry and the information expected from the reconstruction. It is important to point out that generally the values of $\lambda^{(k)}$ would be different for distinct reconstruction tasks.

It turns out that, in our application, solving the linear system by (2.31) is faster than inverting the matrix \mathbf{L} because a line intersects only a very few basis functions (see Fig. 2.5), resulting in a matrix \mathbf{L} which is *sparse* (most of its entries are zero valued) whereas a generalized inverse \mathbf{L}^+ [9] (typically \mathbf{L} is not invertible) has mostly non-zero entries. The sparsity of \mathbf{L} implies that (2.31) can be efficiently implemented (specifically by using the already mentioned idea of footprints [80]).

2.7 Final Representation

After reconstruction, the approximation to the density function \mathfrak{v} is generally corrupted by noise, resulting from the noise in the data collection, see (2.1), and by errors inherent in the reconstruction algorithm. As a result, \mathfrak{v} is typically an imprecise approximation to the macromolecular complex. Consequently, this approximation has to be further processed to obtain an accurate representation of the macromolecule. Despite the need for further processing, TEM technology has been commonly used to obtain reconstructions of macromolecular structures because of the meaningful information this technology produces under the conditions described in previous sections [24, 31, 77].

Chapter 3

VISUALIZATION USING KAISER-BESSEL FUNCTIONS

3.1 Reconstruction Algorithm

In Chapter 2 we introduced some methods for recovering a density function υ from its projections. All those methods produce “good” reconstructions under the appropriate conditions. In this dissertation we are interested in the algebraic reconstruction techniques because these have proved to be efficacious methods for a number of reconstruction tasks [49, 50, 65, 78] and in particular the authors of [73, 76] obtained results suggesting that the combined use of ART and a set of smooth basis functions produce better reconstructions than (or at least as good reconstructions as) those produced by other algorithms.

As we mentioned in Section 2.6, ART attempts to recover a density function υ by a linear combination of known basis functions, see (2.28). The choice of the set of basis functions $\{b_j\}$ greatly influences the result of the reconstruction algorithm [69, 70, 80]. When the basis functions are cubic voxels, as in Fig. 2.5, the resulting approximation to υ is a piecewise constant function that has undesirable artificial sharp edges; consequently, it

appears to be more appropriate to use functions with a smooth transition from one to zero.

In the field of reconstruction from projections Lewitt [69, 70] and Matej [80] have proposed the use of basis functions, called *blobs*, with spherical symmetry and a smooth transition from one to zero (they considered that using blobs as the basis function would reflect more accurately the smoothness of natural objects). Blobs are a generalization of a well-known class of window functions in digital signal processing called *Kaiser-Bessel* [69]. The general form of a single blob is:

$$b(m, \alpha, a; r) = \begin{cases} \frac{I_m\left(\alpha\sqrt{1 - \left(\frac{r}{a}\right)^2}\right)}{I_m(\alpha)} \left(\sqrt{1 - \left(\frac{r}{a}\right)^2}\right)^m, & \text{if } 0 \leq r \leq a, \\ 0, & \text{otherwise,} \end{cases} \quad (3.1)$$

where r is the radial distance from the blob center, I_m denotes the modified Bessel function of order m , a is the radius of the blob and α is a parameter controlling the blob shape. The three parameters m (a non-negative integer), a and α (non-negative real numbers) control the smoothness and shape of a blob and influence the results yielded by the reconstruction algorithm; therefore, the appropriate selection of them is highly important. Hereafter, we set m equal to 2, which makes the blobs to be twice differentiable and its first derivative continuous.

The selection of blobs as basis functions for (2.28) is based on several desirable properties they possess. First of all, they have a smooth transition from one to zero in a bounded

region of space, whereas Gaussian functions decay to zero at infinity. Second, the projection of an n -dimensional blob results in an $(n - 1)$ -dimensional blob, a fact that we can make good use of in the implementation of (2.31). Third, there is a closed analytical formula for the three-dimensional Fourier transform of any blob defined by (3.1) (in case $m = 2$, it is the spherically symmetric function determined by

$$\hat{b}(2, \alpha, a; R) = \frac{a^3 \alpha^2}{I_2(\alpha)} \begin{cases} \frac{I_{\frac{7}{2}}(\sqrt{\alpha^2 - (aR)^2})}{(\sqrt{\alpha^2 - (aR)^2})^{\frac{7}{2}}}, & \text{if } aR \leq \alpha, \\ \frac{J_{\frac{7}{2}}(\sqrt{(aR)^2 - \alpha^2})}{(\sqrt{(aR)^2 - \alpha^2})^{\frac{7}{2}}}, & \text{if } aR \geq \alpha, \end{cases} \quad (3.2)$$

where J is the Bessel function). Finally, blobs are practically bandlimited [69]. We illustrate this in Fig. 3.1, which plots the value $\log\left(\frac{\hat{b}(2,13.36,2.40;R)}{\hat{b}(2,13.36,2.40;0)}\right)$ as a function of the frequency R . It is clear from this figure that $\hat{b}(2,13.36,2.40;R)$ is less than a ten-thousandth

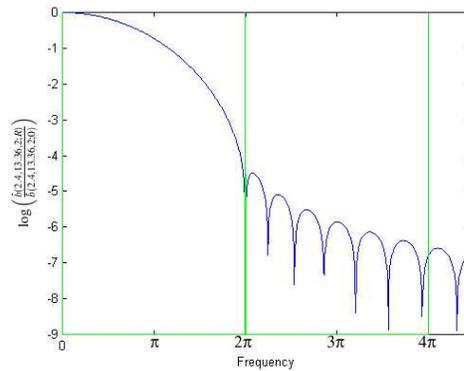


Fig.3.1: The Fourier transform of a blob. We plot $\log\left(\frac{\hat{b}(2,13.36,2.40;R)}{\hat{b}(2,13.36,2.40;0)}\right)$ as a function of the frequency R .

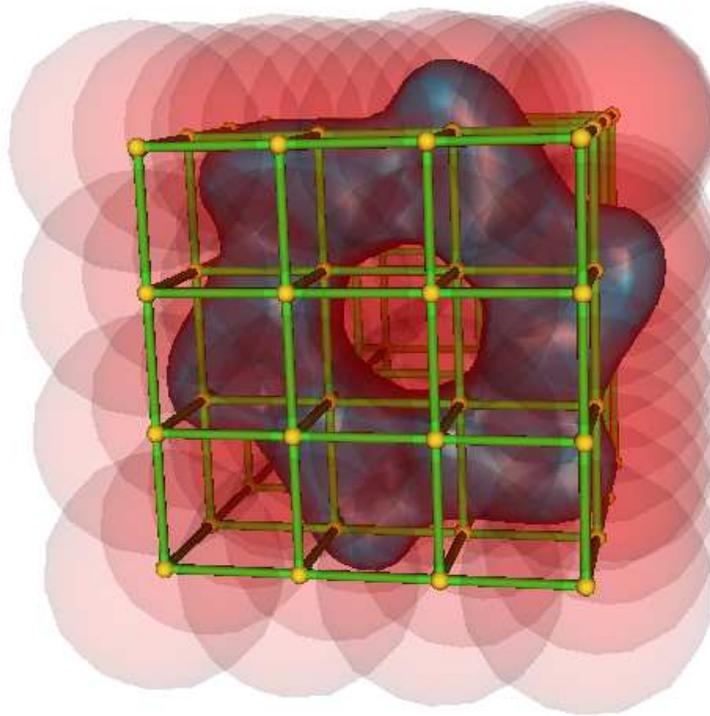


Fig.3.2: Scheme for the approximation of a density function, represented in blue, by the linear combination of weighted blobs (the center of each blob is represented in the image by a small yellow sphere). The support of a blob is a sphere and we indicate the support of each blob in transparent red.

of its peak value if $R \geq 2\pi$ and it is less than a millionth of its peak value if $R \geq 4\pi$.

When blobs are used as basis functions in (2.28), the individual blob functions b_j are shifted versions of the blob b defined in (3.1), see Fig. 3.2. We refer to the set of points $\{\bar{p}_j\}$ to which the centers of the blobs are shifted in such a blob representation as a *grid*, see the yellow spheres in Fig. 3.2.

Incidentally, the linear combination of blob-like basis functions has also been proposed as a way for modeling three-dimensional objects in the computer graphics field [14, 29, 84].

Some examples of basis functions with smooth transition from their maximum to zero are the Gaussian function, used in [14], and multiscale wavelets, used in [84, 85].

3.2 Blob Parameters and Grids

The choice of the spatial arrangement of the set of grid points $\{\overline{p}_j\}$ is important for the quality of the final reconstruction. Three grids are of particular interest:

a. The simple cubic grid (*sc*) is defined by

$$G_\Delta = \{\Delta\overline{k} \mid \overline{k} \in \mathbb{Z}^3\}, \quad (3.3)$$

where Δ is a positive real number (the sampling distance).

b. The body-centered cubic grid (*bcc*) is defined by

$$B_\Delta = \{\Delta\overline{k} \mid \overline{k} \in \mathbb{Z}^3 \text{ and } k_1 \equiv k_2 \equiv k_3 \pmod{2}\}, \quad (3.4)$$

where Δ is a positive real number.

c. The face-centered cubic grid (*fcc*) is defined by

$$F_\Delta = \{\Delta\overline{k} \mid \overline{k} \in \mathbb{Z}^3 \text{ and } k_1 + k_2 + k_3 \equiv 0 \pmod{2}\}, \quad (3.5)$$

where Δ is a positive real number.

In order to visualize these grids, we can use a small portion of them and take advantage of their periodic repetition, see Fig. 3.3.

The sampling of a function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ can be carried out over any of the grids presented above. For this purpose let III_{G_Δ} , III_{B_Δ} , and III_{F_Δ} denote the trains of pulses on

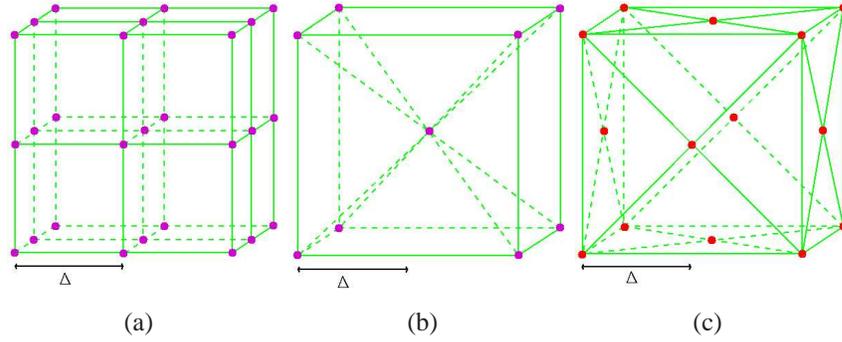


Fig.3.3: Points in the simple cubic (a), body-centered cubic (b), and face-centered cubic (c) grids in a $2 \times 2 \times 2$ portion of space (assuming $\Delta = 1$). The rest of the points can be obtained by filling in space by the most natural repetition of the indicated $2 \times 2 \times 2$ portion.

the grids G_Δ , B_Δ and F_Δ , respectively see (2.20). Multiplying f by the appropriate one of these distributions defines sampling on the corresponding grid (see Appendix A for a mathematically more precise treatment). It is easy to verify (see (A.5), (A.8) and (A.11) in the Appendix) that

$$\widehat{\text{III}}_{G_\Delta} = \left(\frac{\sqrt{2\pi}}{\Delta} \right)^3 \text{III}_{G_{\frac{2\pi}{\Delta}}} \quad (3.6)$$

and

$$\widehat{\text{III}}_{B_\Delta} = \frac{1}{\sqrt{2}} \left(\frac{\sqrt{\pi}}{\Delta} \right)^3 \text{III}_{F_{\frac{\pi}{\Delta}}}. \quad (3.7)$$

The interest in the grids defined above is that the simple cubic grid is the most used and is the easiest to implement in current computers. It has been shown in [81, 97] that

the *bcc* grid is the most “efficient” sampling in three-dimensional Euclidean space when a function is bandlimited with a spectrum that is radially symmetric. To illuminate this statement, we consider a distribution \mathfrak{v} whose Fourier transform $\hat{\mathfrak{v}}$ has the property that $\hat{\mathfrak{v}}(R, \Phi, \Theta) = 0$, if $R \geq \pi$ (i.e., the bandwidth of \mathfrak{v} is 2π). Sampling \mathfrak{v} at points of G_1 (which is the same as multiplying by III_{G_1}) results, see Theorem 2.1, in a Fourier transform which is the convolution of $\hat{\mathfrak{v}}$ with $\text{III}_{G_{2\pi}}$; see (3.6). Due to (2.21) and the assumed property of $\hat{\mathfrak{v}}$, this Fourier transform coincides with the Fourier transform of \mathfrak{v} within a sphere of radius π centered at the origin (see Fig. 3.3(a)), and so \mathfrak{v} can be unambiguously recovered from the samples. However, the same cannot be guaranteed if sampling is done at points of G_Δ with $\Delta > 1$, because of the resulting overlap of the repeated spheres of radius π in the Fourier transform of the sampled function (*aliasing*). On the other hand, sampling \mathfrak{v} at points of B_Δ results in the spheres repeating with their centers at the grid points $F_{\frac{\pi}{\Delta}}$; see (3.7). As can be seen from Fig. 3.3(c), to avoid aliasing we may set Δ to $\frac{1}{\sqrt{2}}$ (or anything less). The number of points needed to cover a fixed portion of space is lower when the points come from $B_{\frac{1}{\sqrt{2}}}$ than when they come from G_1 . This is the sense in which the *bcc* grid is more efficient than the *sc* grid. The *fcc* grid is more efficient than the simple cubic grid but is less efficient than the *bcc* grid. Interestingly the *bcc* and the *fcc* grids are reciprocals of each other in real space and Fourier space, as expressed in (3.7).

For reconstruction purposes, Matej and Lewitt [79] demonstrated that whenever a linear combination of blobs is employed to obtain an approximation of \mathfrak{v} , the grid used should be

different from the simple cubic grid, and that the *bcc* grid is the most desirable. Therefore, we will consider the *bcc* grid for the set $\{\overline{p_j}\}$ where the centers of the blobs $\{b_j\}$ should be located; i.e., the set is obtained by the intersection of some finite convex region of space with the B_Δ of (3.4).

Having decided that we use $m = 2$ and the *bcc* grid, there are three parameters to be chosen: α , a , and Δ . Clearly, to be able to approximate arbitrary distributions using (2.28), the value of Δ should be small. However, in a fixed volume of space, the number of grid points (and consequently the cost of our reconstruction algorithm) is proportional to $\frac{1}{\Delta^3}$ and so practical considerations do not allow us to choose Δ very small. The cost of reconstruction (in our implementation using footprints [70, 80]) is also proportional to a^2 ; we soon report on the influence of the size of a on reconstruction quality. In our implementation, computational cost does not depend on α , and so this parameter may be chosen purely based on the quality of the resulting reconstructions.

One reasonable criterion for choosing our parameters is that a linear combination of blobs with $c_j = 1$, for $1 \leq j \leq J$, should be an approximation of a constant valued function. In such a case, the right hand side of (2.28) is a convolution of the blob b in (3.1) with a truncated version of III_{B_Δ} , and so by Theorem 2.1 its Fourier transform is approximately $\widehat{b} \times \frac{1}{\sqrt{2}} \left(\frac{\sqrt{\pi}}{\Delta}\right)^3 \text{III}_{F_{\frac{\pi}{\Delta}}}$ (see Appendix A for a discussion of the multiplication of a function by a distribution). For this to best approximate the Fourier transform of a constant-valued function (which is an impulse at the origin) it is useful to select b in such a way that

$\hat{b}(2, \alpha, a; R)$ is zero-valued at the locations of $F_{\frac{\pi}{\Delta}}$ which have the smallest positive distance from the origin; i.e., at the frequency $R = \frac{\sqrt{2}\pi}{\Delta}$. Since $I_{\frac{7}{2}}$ is not zero-valued and the smallest positive x for which $J_{\frac{7}{2}}(x) = 0$ is $x = 6.987932$, it follows from (3.2) and the discussion in this paragraph that, for any a and Δ , a reasonable choice is

$$\alpha = \sqrt{2\pi^2 \left(\frac{a}{\Delta}\right)^2 - 6.987932^2}. \quad (3.8)$$

Following this approach we reduce the number of unknowns from three to just two, i.e., a and Δ . Note that α depends only on the ratio (and not on the actual values) of these two variables.

In Fig. 3.4 we report on an experiment based on those proposed in [80]. The plots represent the level sets (with indicated values) of the root mean square (*rms*) error between an appropriate constant and the right hand side of (2.28) (with $c_j = 1$ for $j = 1 \dots J$) sampled at an appropriate subset of G_1 for various choices of $\frac{a}{\Delta}$ and α (we set, for this experiment, $\Delta = \frac{1}{\sqrt{2}}$).

In Fig. 3.4 we indicate by a continuous heavy line the locus of points for which (3.8) is satisfied (a hyperbola). As can be seen from Fig. 3.4, for any fixed value of $\frac{a}{\Delta}$ the α determined by (3.8) provides a low rms error. If we restrict ourselves to points on the hyperbola indicated in Fig. 3.4 by the continuous heavy line, the rms error for a fixed Δ can be decreased by using a higher value a , but this is at the expense of increased computational cost. Also, this study ignores resolution, which is another important criterion. There is

indeed provide us with relatively low rms errors.

3.3 Implicit Surfaces and Visualization

In computer graphics implicit surfaces (also called isosurfaces or isointensity surfaces) have been used to represent objects of different topologies and geometries. An implicit surface S is mathematically defined as

$$S = \{(r, \phi, \theta) | v(r, \phi, \theta) = t\}. \quad (3.9)$$

In many areas of science, (3.9) is used to visualize a reconstructed distribution described by (2.28). The assumption is that there is a fixed threshold t such that the object of interest consists of exactly those points at which the value of the distribution v is greater than the threshold. If the total volume of the object of interest is known (as is the case in some applications, such as electron microscopy), then t is uniquely determined by the criterion that S should enclose exactly the known volume. For computerized visualization of the object of interest it is then sufficient to display its surface S , as defined by (3.9).

A suitable method for visualizing the surface in (3.9) is *raycasting*. In one of its forms this technique consists of casting a finite number of rays perpendicular to a plane, typically the computer screen, towards S ; this form of raycasting produces an orthogonal projection onto the plane. In order to produce a foreshortening effect in the final image (the farther the objects, the smaller they appear in it) it is possible to use a perspective projection in which

all the rays cast from the plane intersect in a point called the *center of projection* [29, 126]. Because we work with objects in which foreshortening is not important, we present images with orthogonal projection only.

For every ray, we need to find the point \bar{q} in S nearest to the plane and compute its distance and the normal to S at \bar{q} (these are used to assign an intensity value on the computer screen [29, 126]). In practice, finding the points \bar{q} is computationally expensive. In general there is no prior estimate of how far \bar{q} is from the plane. Based on an approach in [14], we designed the following methodology. We first do a preprocessing of the set of grid points $\{\bar{p}_j\}$ at the end of which, for every point on the plane from which we need to cast a ray, we have the list of those grid points (arranged in order of increasing distance from the plane) whose associated coefficients can possibly influence the value of the distribution υ anywhere along the ray. (These grid points all lie within a cylinder of radius a whose central axis is the ray in question.) This preprocessing is easily done by identifying on the plane the shadows of the blobs centered at the grid points, one-by-one in an appropriate order. In locating \bar{q} for a particular ray, we make use of the associated list of grid points. For all grid points in the list (recall that these are arranged in order of increasing distance from the plane), we evaluate υ at the projection of the grid point onto the ray (for this we need the blob coefficients for only a few grid points, all of which are at a similar position in the list), until we find (if ever) two consecutive projections \bar{q}_a and \bar{q}_b such that the value of υ is below the threshold at \bar{q}_a and is above it at \bar{q}_b . Then \bar{q} is located by a binary search

between \bar{q}_a and \bar{q}_b (for this we need the coefficients of only those blobs which were used for calculating \mathfrak{v} at \bar{q}_a and at \bar{q}_b).

Assuming that the approximation in (2.28) is exact, we know that \mathfrak{v} is a continuously differentiable function and that the gradient of \mathfrak{v} , at any point, is given by

$$\nabla \mathfrak{v}(r, \phi, \theta) = \sum_{j=1}^J c_j \nabla b_j(r, \phi, \theta). \quad (3.10)$$

The set $\{c_j\}$ is produced by the reconstruction algorithm and we have closed formulas to compute ∇b_j [69]. The representation obtained by (3.9) and raycasting is therefore an accurate representation of the object of interest, limited only by the reconstruction and thresholding processes.

3.4 Selection of Blobs for Visualization

The principles described in Sections 2.6, 3.1 and 3.2 have been applied in the field of Structural Analysis of Proteins by Transmission Electron Microscopy [8, 73, 76, 77, 107]. In particular, the authors of [8] obtained a set of coefficients $\{c_j\}$ by applying ART to projection images originating from a protein (the macromolecular complex DnaB·DnaC) imaged by a transmission electron microscope, using parameters α , a and Δ satisfying (3.8). When we used the resulting set $\{c_j\}$ to produce a visual representation of the surface of the macromolecular complex DnaB·DnaC by the raycasting method explained above, we found, to our surprise, that the surface representation had artifacts that were not observable

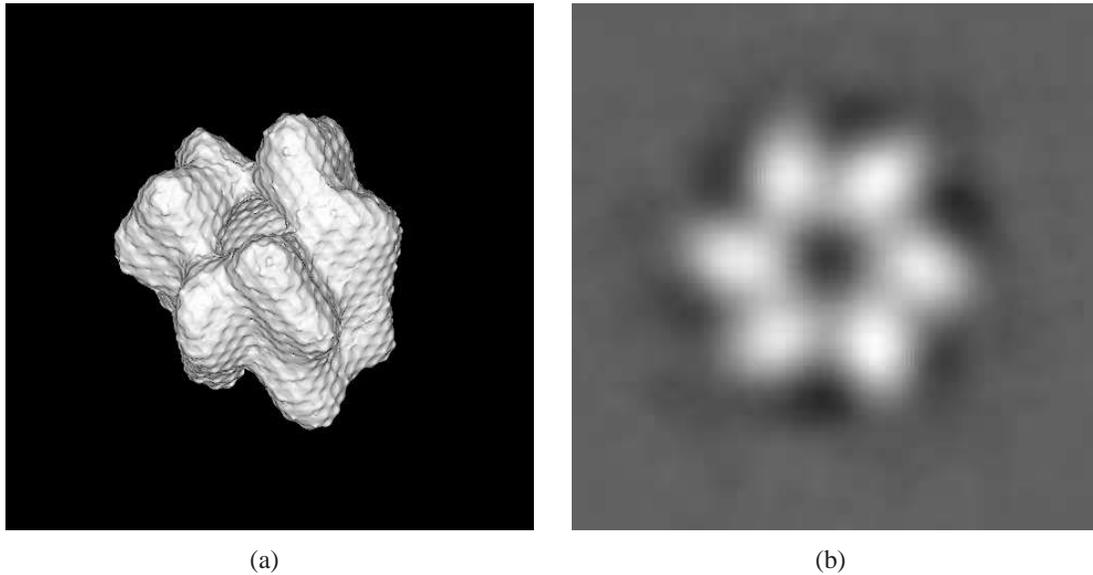


Fig.3.5: (a) Surface representation of the macromolecular complex DnaB·DnaC with blob parameters $a = 1.25$, $\alpha = 3.60$ with grid separation $\Delta = \frac{1}{\sqrt{2}}$ [8] (the distance between pixels is equal to $\frac{1}{4}$). (b) A central slice from the reconstruction by ART of the macromolecular complex DnaB·DnaC with the same parameters as for (a) with $\Delta = 1.00$.

in the slice-by-slice presentation of the reconstructed distribution, see Fig. 3.5 (In order to fully appreciate the three-dimensional structure of the macromolecule, the distance between pixels in Fig. 3.5(a) is equal to $\frac{1}{4}$.) Clearly, the parameters which were considered sufficiently good for reconstruction are not particularly good for visualization.

Similarly to the approach of approximating functions by linear combinations of blobs, the representation of surfaces using linear combinations of blobs requires finding a “good” set of parameters for the blobs and for the grid of blob centers. In Fig. 3.6, we illustrate the enormous influence of the values of a and α have on the appearance of the resulting surface.

In order to study further the effect of blob parameters for visualization, we used an approach similar to that in Section 3.2, but now we aimed at minimizing the error between a surface and its approximation, as measured by the difference between the surface normals [114]. For this test we selected a distribution \mathfrak{v}_s with a constant value 1 inside a sphere

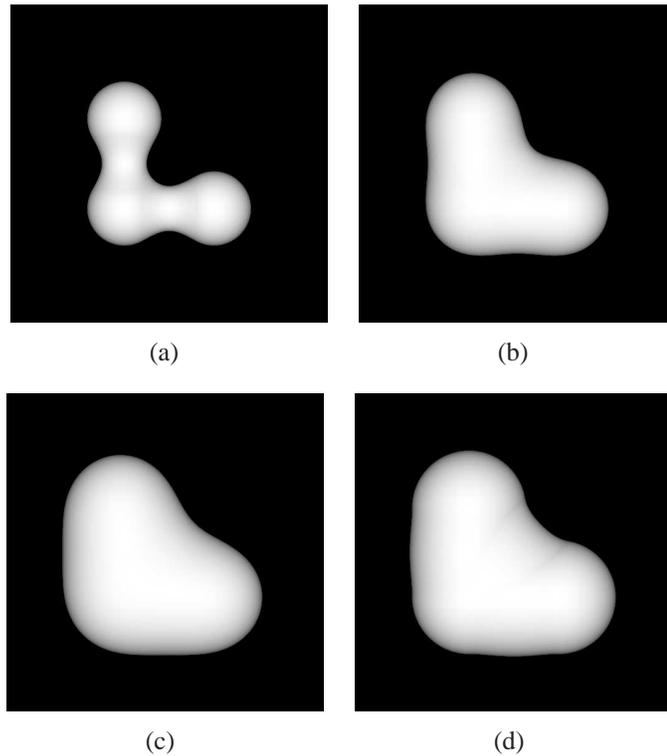


Fig.3.6: In all cases the distribution \mathfrak{v} is defined by (2.28) using the grid $B_{\frac{1}{\sqrt{2}}}$ with blob coefficient 1 at the points $(0,0,0)$, $(\sqrt{2},0,0)$ and $(0,\sqrt{2},0)$ and blob coefficient 0 at all other points. The displayed surfaces are defined by (3.9) with $t = 0.5$. The values of a and α are (a) 1.25 and 3.60 (same as for Fig. 3.5(a)), (b) 2.40 and 13.36, (c) 3.20 and 18.85 and (d) 1.65 and 0.00. The first three pairs of (a, α) satisfy (3.8) and the fourth is computed with the value of x for the second zero crossing of $J_{\frac{7}{2}}(x)$. (Throughout this dissertation we report on a and α to an accuracy of 0.01, however the values actually used by our programs were always calculated to the accuracy of our computations.)

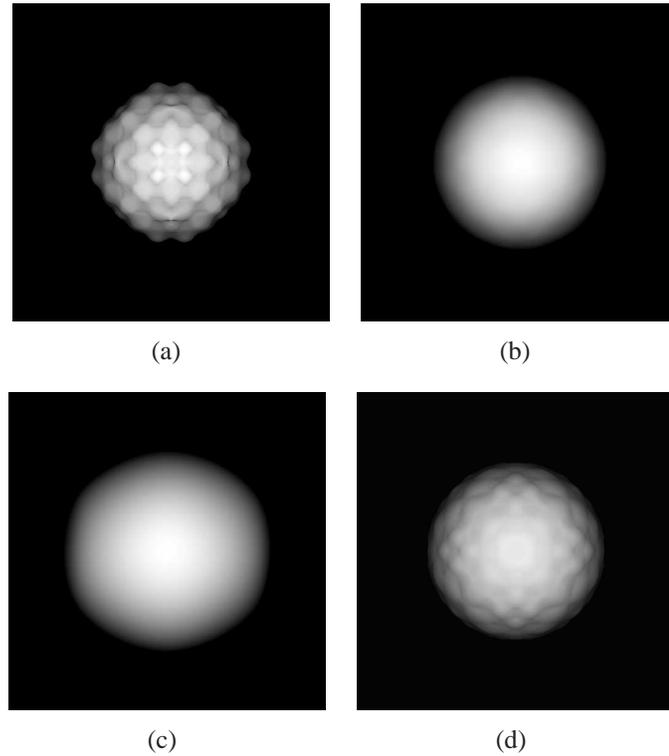


Fig.3.7: Visualizations of the implicit surface ($t = 0.5$) of reconstructions of a sphere. For the choices of the parameters in cases (a), (b), (c) and (d), see the caption of Fig. 3.6.

and 0 outside. We then simulated the random conical tilt scheme of data collection, see Section 2.5, to generate the projections from \mathcal{V}_s . The projections were utilized to create reconstructions using ART with different values for $\frac{a}{\Delta}$ and α (keeping again $\Delta = \frac{1}{\sqrt{2}}$). For each set $\{c_j\}$ produced by the reconstruction algorithm, raycasting was used to create a visualization of the implicit surface of the reconstructed sphere at threshold 0.5. (Four of the resulting surface displays, for the same choices of parameters as were made for Fig. 3.6 are shown in Fig. 3.7.)

For every ray we computed the angle between the normal to the true spherical surface

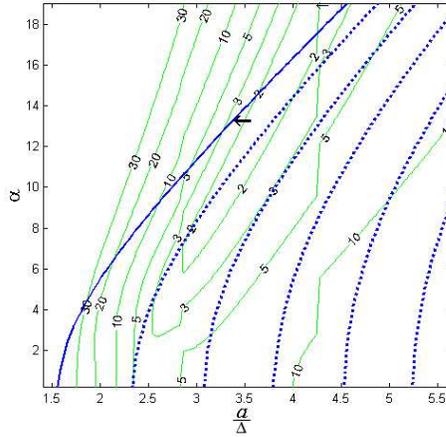


Fig.3.8: The rms error between analytic normals to a sphere and normals to the implicit surface of its reconstruction at each display point for which the ray casted crosses both surfaces (see Fig. 3.7). The continuous heavy curve indicates the locus of the points (a hyperbola) which satisfy (3.8). The broken heavy curves indicate the hyperbolas which correspond to the other zero-crossing of the Fourier transform of b .

and the normal to the implicit surface in the reconstructed distribution. We define the *rms error* to be the norm of the vector whose components are these angles (for all display points for which the casted ray crosses both surfaces). The results are displayed in Fig. 3.8 in a manner similar to the display in Fig. 3.4.

Ignoring for now the exact choice of α , what we see in Fig. 3.8 is that, generally speaking, $\frac{a}{\Delta}$ should be neither too small (this results in bumps as in Fig. 3.7(a)) nor too large (this results in a blurred-out sphere as in Fig. 3.7(c)). To allow us to make a definite choice, we propose the following criterion: a should be chosen as small as possible consistent with achieving the following. If two blobs at nearest grid points in the grid B_{Δ} are given coefficients 1 with all other blobs given coefficients 0, then the implicit surface thresholded

at $t = 0.5$ should enclose a convex set. Such implicit surfaces (for the values of a also used in Figs.3.6, and 3.7 and with α determined by (3.8), assuming that $\Delta = \frac{1}{\sqrt{2}}$) are shown in Fig. 3.9. Fixing Δ to be $\frac{1}{\sqrt{2}}$ and using an α which is determined by (3.8), we find that the smallest a which satisfies our new criterion is 2.40 (this corresponds to (b) in Figs.3.6 and 3.7). The corresponding α is 13.36 and the location of this $(\frac{a}{\Delta}, \alpha)$ pair is indicated by arrows in Figs.3.4 and 3.8.

We see that, as long as we insist that (3.8) be satisfied, the value of a determined by the criterion stated in the previous paragraph will be a fixed multiple of Δ . (This can be easily derived by looking at (3.1) and expressing both a and r in it as multiples of Δ and using the α specified with (3.8).) So the criterion, together with (3.8), uniquely determines $\frac{a}{\Delta}$ (it is 3.39) and α (it is 13.36). We still have the freedom of choosing Δ ; its choice to a large extent depends on the nature of our measurements.

We could argue that the criterion just described could be combined with (3.8), but using the hyperbola associated with the next zero-crossing of the Fourier transform of (3.1), to obtain a lower rms error. For example, the hyperbola associated with the second zero-crossing of b in Fig. 3.8 crosses lower rms errors. Disappointingly, using the same criterion for this hyperbola $\left(\alpha = \sqrt{2\pi^2 \left(\frac{a}{\Delta}\right)^2 - 10.41711^2} \right)$ does not provide a value for $\frac{a}{\Delta}$ such that the surface of two closest-neighboring blobs transits from concave to convex. There is no transition since the a provided by with $\alpha = 0$ already produces a convex surface, shown in Fig. 3.9(d). It can be seen in Fig. 3.7(d) that the implicit surface produced with such

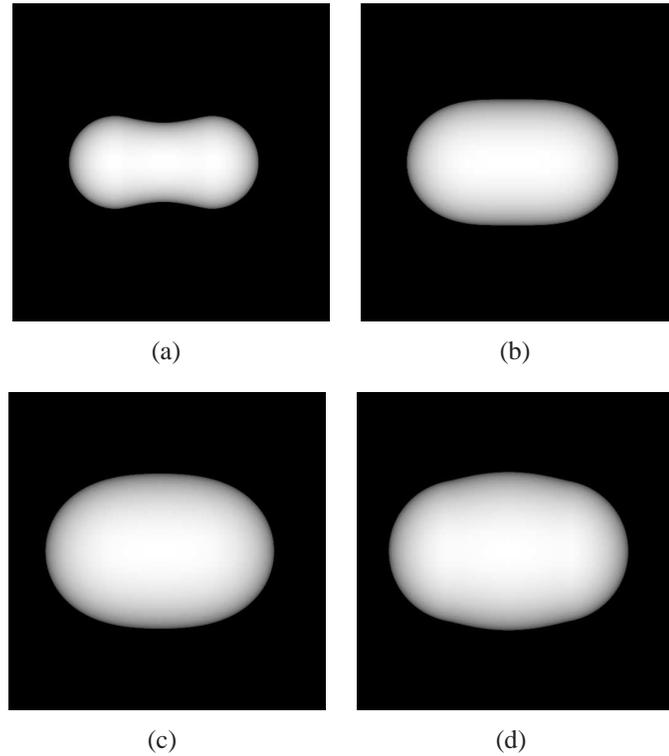


Fig.3.9: Representations of the implicit surface at level $t = 0.5$ for the combination of two blobs whose centers are immediate neighbors in the bcc grid $B_{\frac{1}{\sqrt{2}}}$ and whose coefficients are 1. The parameters of the blobs match those in Figs.3.6 and 3.7.

parameters is a bad approximation of a sphere. Not surprisingly, for these values, the rms error presented in Fig. 3.8 is outside the region of “good” values.

We illustrate the performance of these choices of the parameters used in conjunction with the electron microscopic data from the macromolecular complex DnaB·DnaC. The resulting implicit surface is shown in Fig. 3.10(b); it is superior to the previously shown surface of Fig. 3.5(a), which is reproduced here as Fig. 3.10(a). It is also worth noting that it is just about impossible to anticipate the radical differences between the surface

displays in Fig. 3.10 from the traditional slice-by-slice presentation of the reconstructed distributions, see Fig. 3.11.

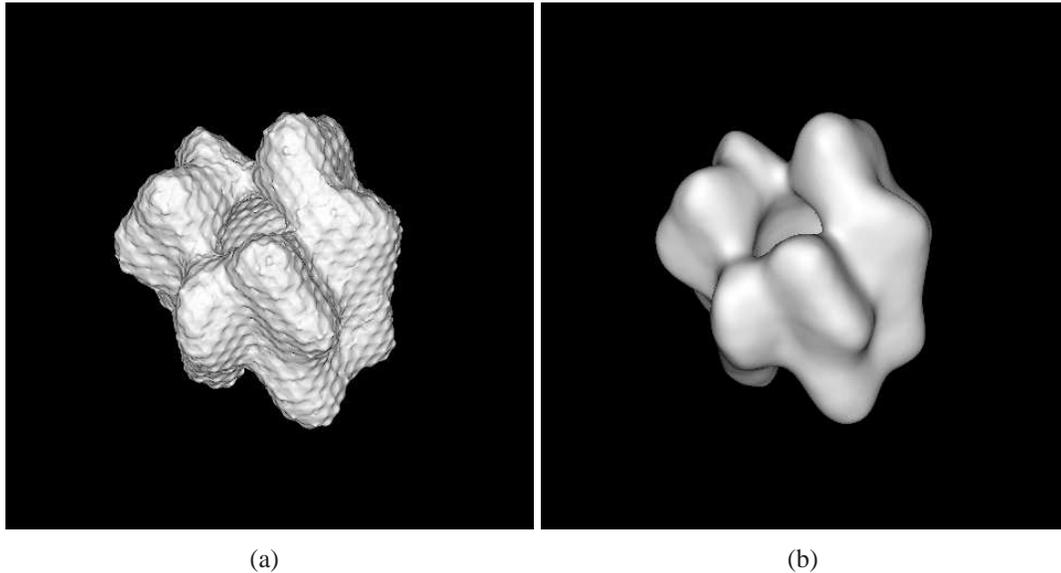


Fig.3.10: Representations of the implicit surface for the macromolecular complex DnaB·-DnaC. The reconstructions using ART and visual representations of the DnaB·DnaC were created with (a) parameters $\Delta = \frac{1}{\sqrt{2}}$, $a = 1.25$ and $\alpha = 3.60$ as used in [8], (b) parameters $\Delta = \frac{1}{\sqrt{2}}$, $a = 2.40$ and $\alpha = 13.36$.

3.5 Comparison with Explicit Surface Visualization

While the implementation of raycasting to visualize implicit surfaces is straightforward, such visualization is usually computationally demanding and slow because of the constant search for the intersecting points \bar{q} . Polygon-projection methods are an alternative because of their fast performance. These methods explicitly approximate a surface by a collection of polygons [29, 35, 126]. Here we compare the performance of the raycasting method

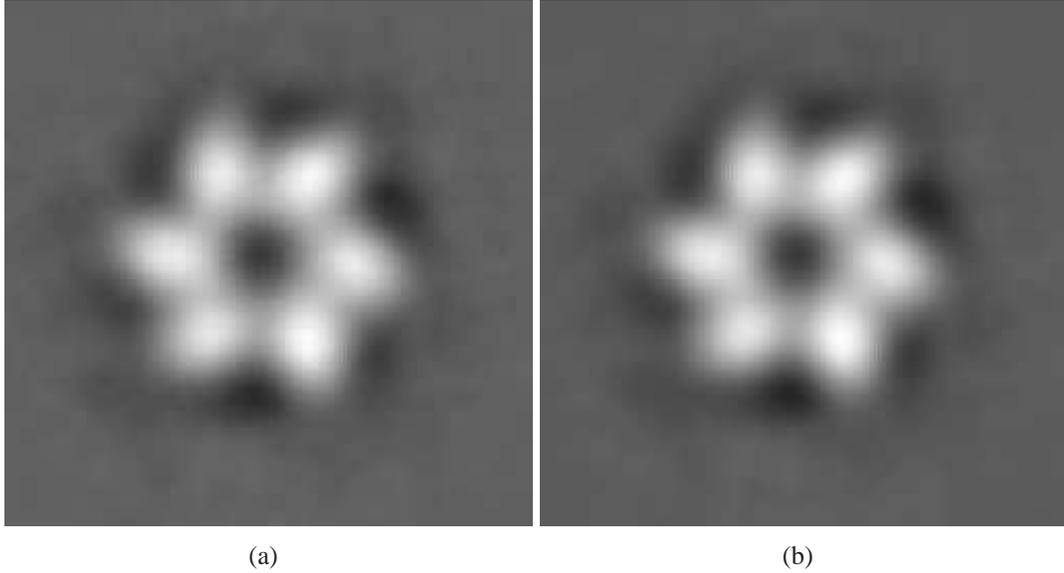


Fig.3.11: Central section of the reconstructions of the macromolecular complex DnaB-DnaC by ART with (a) parameters $\Delta = \frac{1}{\sqrt{2}}$, $a = 1.25$ and $\alpha = 3.60$, (b) parameters $\Delta = \frac{1}{\sqrt{2}}$, $a = 2.40$ and $\alpha = 13.36$. Both images are central slices of the discretized density function v using the *sc* grid with $\Delta = 1.00$.

described in Section 3.3 with that of the polygon-projection method implemented in the visualization software OpenDXTM [1].

In order to apply OpenDXTM to our reconstructions, we first need to evaluate the values at points of some grid (for our illustrations we used the simple cubic grid). For reasons already explained in Section 3.2, if $B_{\frac{1}{\sqrt{2}}}$ was used for reconstruction, it is appropriate to estimate the reconstructed values, using (2.28), at points of G_1 . Just like the raycasting method, the polygon projection method of OpenDXTM requires the specification of a threshold, based on which it automatically calculates the polygons which form the explicit surface to be displayed.

We applied this approach to the reconstructions of the macromolecular complex DnaB-DnaC whose central slices are reported in Fig. 3.11. (By the way, these slices are displays of the estimated values at points of G_1 .) The calculation of the values at points of G_1 took 10 s for the blobs with $a = 1.25$ and $\alpha = 3.60$ and 56 s for the blobs with $a = 2.40$ and $\alpha = 13.36$. (All times are for a Pentium 4TM based computer, 2 GHz, 512 Mbytes of RAM, under Linux©.) In both cases, the computation of the explicit surface took 2 sec. The display of the surface, once computed, is essentially instantaneous. These times are much better than what is needed for raycasting at the same resolution (480×480 pixels): 13 min 19 s for the blobs with $a = 1.25$ and $\alpha = 3.60$ and 1 h 35 min 50 s for the blobs with $a = 2.40$ and $\alpha = 13.36$. However, the reconstruction times (the calculation of the $\{c_j\}$ by ART) are 32 h 15 min 36 s and 65 h 22 min 48 s, respectively, in these two cases and the time needed for specimen preparation and the collection of electron microscopic data is measured in weeks. Hence the quality of the visualizations may very well be considered more important than the computer time needed to produce them.

The results produced by the polygon-projection method in OpenDXTM are presented in Fig. 3.12. All parameters (for the blobs, the grid, the threshold, the assumed orientation of the surface, etc.) were selected to be the same as those in the corresponding displays in Fig. 3.10. With the computer graphic display methodology embedded in OpenDXTM, the explicit surface displays of Fig. 3.12 appear to be smoother than the implicit surface displays of Fig. 3.10, which is an advantage in case (a), but seems to be a disadvantage in

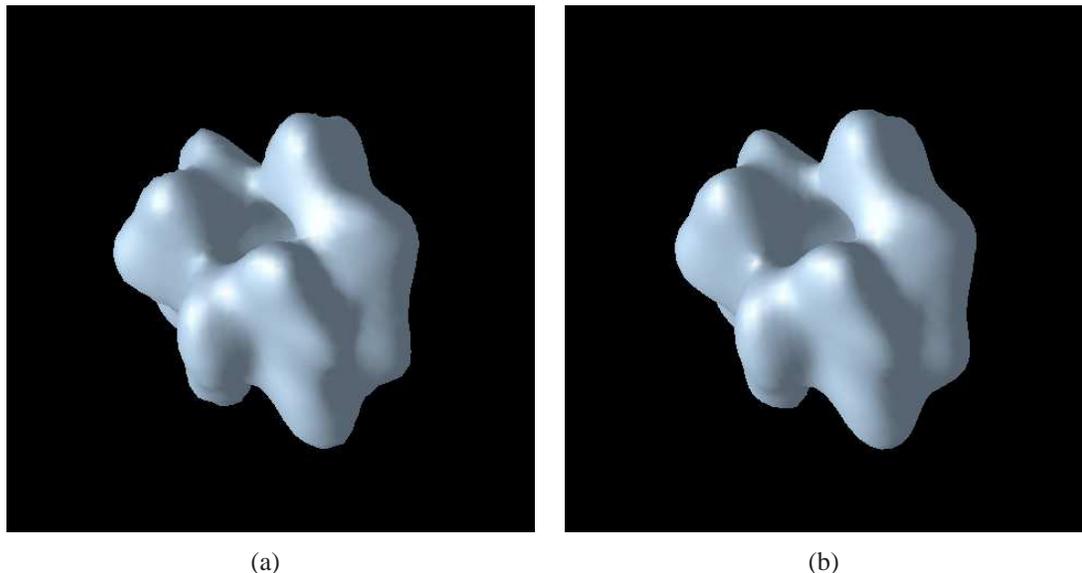


Fig.3.12: Representations of the explicit surfaces obtained from the reconstructions of the macromolecular complex DnaB-DnaC using OpenDX with a $128 \times 128 \times 128$ voxelized distribution over the sc grid ($\Delta = 1.00$). Parameters used for reconstruction were: (a) $\Delta = \frac{1}{\sqrt{2}}$, $a = 1.25$ and $\alpha = 3.60$ and (b) $\Delta = \frac{1}{\sqrt{2}}$, $a = 2.40$ and $\alpha = 13.36$. It is clear that for this example the general visualization software hides some important details; for a comparison with the corresponding output of the raycasting method, see Fig. 3.10.

case (b), which is the case that uses the blob parameters recommended by us in Section 3.4.

In order to further increase the rendering resolution, it was necessary to sample the density function at the points of $G_{\frac{1}{4}}$. By using this resolution, the artifacts in Fig. 3.10(a) are also visible in the OpenDX rendering using reconstruction parameters $a = 2.40$ and $\alpha = 13.36$, see Fig. 3.13. However, the increase of the sampling implies an increment in time (60 seconds) and memory (800 Mbytes) to generate the explicit surface using this software, but the time to render an image is still essentially instantaneous.

It is difficult to make a visual comparison between the results obtained by our raycasting

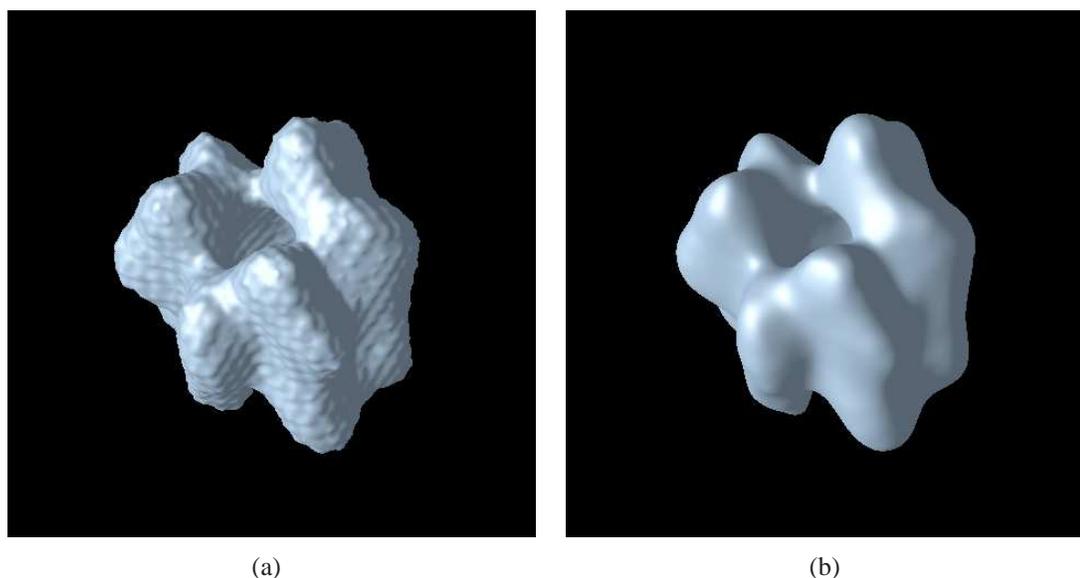


Fig.3.13: Representation of the explicit surface obtained from the reconstructions of the macromolecular complex DnaB·DnaC using OpenDX with a $400 \times 400 \times 400$ voxelized distribution over the *sc* grid ($\Delta = 0.25$). (a) Parameters used for reconstruction were $\Delta = \frac{1}{\sqrt{2}}$, $a = 1.25$ and $\alpha = 3.60$. (b) Parameters used for reconstruction were $\Delta = \frac{1}{\sqrt{2}}$, $a = 2.40$ and $\alpha = 13.36$. For this representation it is necessary to increase four times the memory allocated by OpenDX.

method shown in Fig. 3.10 and those produced by OpenDX with a finely discretized density function shown in Fig. 3.13, mainly due to the difference in lighting models. However, we can see that the result in Fig. 3.13(a) is a smoother representation of the same surface that is shown in Fig. 3.10(a); we speculate that such smoothing is due to lack of information in the discretized density function used as input to OpenDXTM. Similarly, it seems that the surface in Fig. 3.10(b) is a sharper version of that in Fig. 3.13(b). However, for a fair evaluative comparison it would be necessary to make the lighting models the same and to use a mathematically described object whose features are known.

As stated at the end of Section 3.3, the accuracy of the visualization using raycasting is limited only by the quality of the reconstruction (the $\{c_j\}$) and the accuracy of the threshold. The polygon-projection methods bring an additional source of inaccuracy into the process: the approximation of the implicit surface by a collection of polygons. Thus, if the reconstruction parameters and the threshold have been well chosen, we can expect that raycasting will be the more reliable visualization tool. Whether or not this is worth the very considerable additional computing time (and the consequent loss of capability of real-time interactive examination of the resulting surfaces) is highly dependent on the application; it is impossible to know ahead of time whether or not the more reliable visualization will lead to increased scientific knowledge.

Chapter 4

FAST AND ACCURATE VISUALIZATION METHODOLOGIES

At the end of the last chapter we presented a short discussion on how the polygon-projection methods bring an additional source of inaccuracy into the computer representation (by approximating an implicit surface by a collection of polygons), and how the results obtained by the raycasting-blobs methods are limited only by the quality of the reconstruction and the accuracy of the threshold. However, the polygon-projection methods are, in general, fast enough to allow real time manipulation of the three-dimensional representations, whereas raycasting-blobs methods are too slow for that. It is desirable to find a method that produces accurate computer representations, such as that presented in Fig. 3.10(b), at affordable speeds.

In this chapter we present two methodologies to create fast and accurate computer representations of a macromolecular complex. The first improves the speed of the method presented in Section 3.3 without any loss of quality in the output. The second takes advantage of a better sampling of the density function ν to create higher-quality polygonal-projection

images.

4.1 Method to Speed Up the Raycasting-Blobs Technique

In this section we consider a method to accelerate the raycasting-blobs method by taking advantage of the properties of the representation in (2.28) when the b_j s are blobs based on (3.1). Recall from Section 3.3 that we are interested in visualizing the implicit surface S , defined by (3.9), of the density distribution function υ at a given threshold t (chosen so the volume enclosed by S equals the known volume of the macromolecular complex). We presented in Section 3.3 such a visualization method. That implementation searches, for every pixel of the image displaying the surface, for two points \overline{q}_a and \overline{q}_b which are end points of a line segment containing the point \overline{q} such that $\upsilon(\overline{q}) = t$. This search is computationally expensive as there is no estimate of how far the points \overline{q}_a and \overline{q}_b are from the projection plane. To reduce this cost, we introduce below a preprocessing method that finds an estimate of the point \overline{q} , for every casted ray in the image.

In the visualization method presented in Section 3.3, we first process the set $\{\overline{p}_j\}$ in such a way that, for every casted ray, we obtain the list of grid points (sorted in order of increasing distance from the plane) for which the blobs centered at them can contribute to the value of υ anywhere along the ray. Then, for every casted ray, the method evaluates υ at the projections of these grid points onto the ray, until we find two consecutive points \overline{q}_a and \overline{q}_b such that $\upsilon(\overline{q}_a)$ is below the threshold and $\upsilon(\overline{q}_b)$ is above it. The search for the

points $\overline{q_a}$ and $\overline{q_b}$ in every ray is time consuming as it is necessary to evaluate υ at many points.

Our new raycasting-blobs method first calculates, for every grid point $\overline{p_j}$, $v_j = \upsilon(\overline{p_j})$ and then proceeds to utilize both the set $\{c_j\}$ and a list of all the j , such that $v_j \geq t$. It also makes use of a *z-buffer*, a popular tool in computer graphics [29, 126] for computing the distances between the projection plane and the points nearest to it.

The version of the *z-buffer* algorithm that we designed for our purposes operates as follows. A value is assigned to every ray of the raycasting algorithm of Section 3.3. Initially this value is “infinity” (in practice, a very large number). Then we loop through all j such that $v_j \geq t$. For each corresponding blob, we calculate the distance d_j of $\overline{p_j}$ from the plane and, for all rays which intersect the blob, we replace the currently assigned value by d_j if, and only if, d_j is smaller than the currently assigned value. Upon completion of this process, the value assigned to any ray will be an approximation to the distance of the distance of $\overline{q_b}$ for that ray in the raycasting algorithm.

The new method utilizes only those rays whose corresponding value in the *z-buffer* is different from infinity. For every ray of this set, we assume that the value stored in the *z-buffer* provides the point $\overline{q_c}$ that in most of the cases is near to point $\overline{q_b}$. We check whether the condition $\upsilon(\overline{q_c}) \geq t$ is true. When the condition is not true, we search for $\overline{q_a}$ and $\overline{q_b}$ as we did in the original implementation but starting from the point $\overline{q_c}$ (which is a projection of a grid point in the list associated towards the casted ray). In the alternate case, we search

for the points \bar{q}_a and \bar{q}_b in the direction towards the plane, again starting from \bar{q}_c . Once the points \bar{q}_a and \bar{q}_b are found, the point \bar{q} is located as in the original implementation in Section 3.3.

4.2 Experiments for the Faster Raycasting-Blobs Technique

For our experiments we selected two macromolecular complexes: the DnaB·DnaC and the bacteriorhodopsin (*bR*), see Appendix B for a discussion of the bR. The reasons for selecting these two macromolecular complexes are that we already presented in the previous chapter results with the complex DnaB·DnaC and that there is an atomic model for the bR which permits us to know the truth regarding this complex (see Fig. B.2).

First, we obtained micrographs for both complexes, see Fig. 4.1. In the case of the complex DnaB·DnaC we used those supplied for the experiments in Chapter 3, see Fig. 4.1(a). For the bR we obtained an atomic description from the Protein Data Bank (*PDB*) [10]. We used this description file to simulate the process in (2.1) with programs that permit us to simulate projections with noise from PDB models [30, 32, 75]. We decided to utilize the conical tilt scheme to create the projections. For the selection of uncertainty for the angles we used ± 5 degrees for rotation and ± 1 degree for tilt, these values are suggested in [73]. For the level of noise in the pixels, we selected the value of 0.5 SNR (see Fig. 4.1(b)), a value within the range presented in [31] for uncorrelated additive noise with zero mean

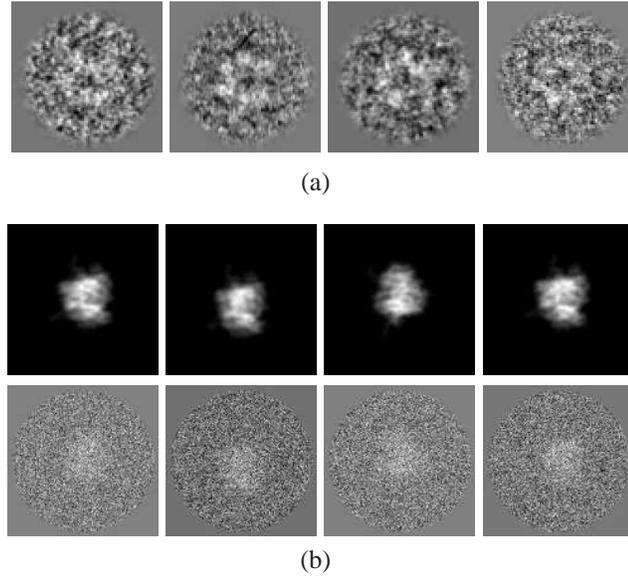


Fig.4.1: (a) Four micrographs of specimens of the DnaB-DnaC [8]. (b) Projections (*above*) without noise and (*below*) with noise from the macromolecular complex bacteriorhodopsin created with the assistance of atomic descriptions [10] and simulation programs [30, 32, 75].

and defined [31] as

$$SNR = \frac{\text{variance of signal}}{\text{variance of noise}} = \frac{\int |\hat{o}(\bar{\xi})|^2 \hat{h}^2(\bar{\xi}) d\bar{\xi}}{\int |\hat{n}(\bar{\xi})|^2 d\bar{\xi}}, \quad (4.1)$$

where o represents the signal, h the psf , see (2.1), and n is the noise. In the equation above, the terms $\hat{o}(\bar{0})$, $\hat{h}(\bar{0})$ and $\hat{n}(\bar{0})$ are excluded.

We produced reconstructions of the macromolecular complexes by ART with blobs as described in Section 2.6. In conformity with our discussion in Section 3.3, we made a selection of blobs based on the criteria discussed in that section: $a = 2.40$, $\alpha = 13.36$

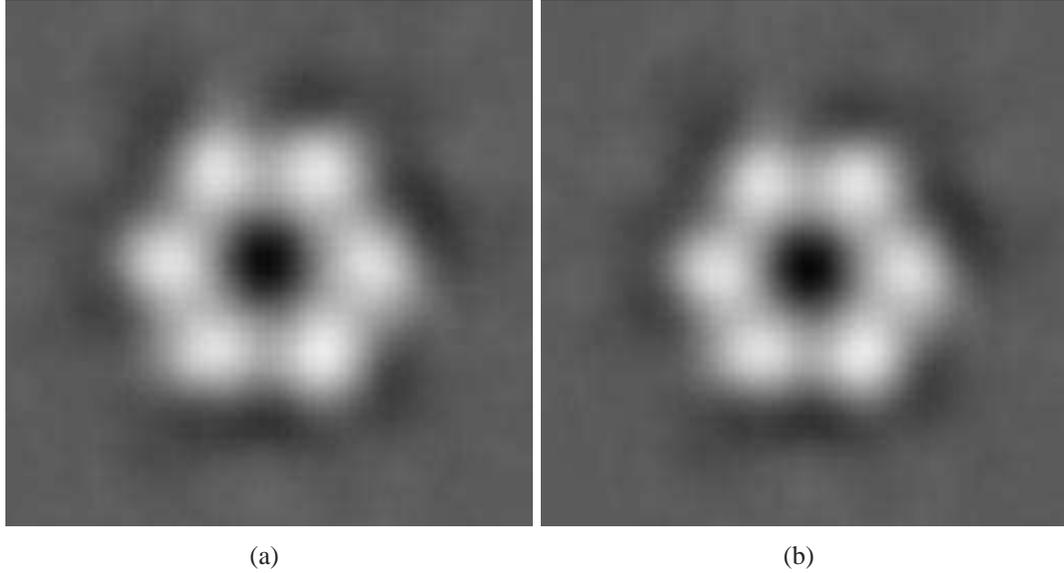


Fig.4.2: Central slices corresponding to the DnaB·DnaC reconstructed with ART with blob parameters $a = 2.40$, $\alpha = 13.36$ and $\Delta = \frac{1}{\sqrt{2}}$: (a) the set $\{c_j\}$, (b) the discretized density function v at the locations of $\{\bar{p}_j\}$.

and $\Delta = \frac{1}{\sqrt{2}}$. In Fig. 4.2(a) we present a slice of the set $\{c_j\}$ for the macromolecular complex DnaB·DnaC. A slice of the *bcc* grid can be visualized as a square array where only even or odd positions are assigned with a coefficient value depending on the slice (see the definition of the *bcc* grid in (3.4) and Fig. 3.3(b)). Therefore, for display purposes, we assigned values to points in the simple cubic grid (see Fig. 3.3(a)) in the following way. First, the values assigned to points in the *bcc* grid are assigned to the same positions on the simple cubic grid. The other values in the simple cubic grid are computed as follows, refer to Fig. 3.3(b). First, the values for the positions in the middle of the cube faces are obtained by averaging the two neighbors with distance Δ and the four neighbors with distance $\sqrt{2}\Delta$.

Afterwards, the values for the positions on the middle of the edges of the cube are obtained by averaging the six neighbors with distance Δ . We illustrate the similarity between the set $\{c_j\}$ and the discretized density function υ in Fig. 4.2(b) where we present a slice of the density function υ , calculated at the same positions as used in Fig. 4.2(a).

After reconstruction, we created the set $\{v_j\}$. This process took 34 s for the set $\{c_j\}$ corresponding to the DnaB·DnaC, with 1,600,065 values, and 37 s for that of the bR, with 1,482,624 values. The result of this process for the DnaB·DnaC is illustrated in Fig. 4.2(b).

After creating the set $\{v_j\}$, we used our modified raycasting method to obtain the computer representation of both the macromolecular complexes DnaB·DnaC and bR. We present these computer representations in Fig. 4.3(a) and (b). These images can be compared with those created with the raycasting method introduced in Section 3.3 in Fig. 4.3(c) and (d), respectively. It can be seen that the images are exactly the same. However, the times to generate the images in Fig. 4.3(c) and (d) are around 1.5 hours (using an Pentium 4TM with under Linux©computer at 2 GHz), while the times to generate those images in Fig. 4.3(a) and (b) are 190 s and 225 s for the DnaB·DnaC and bR, respectively; thus reducing the computing time between 24 and 28 times.

In spite of the use of both the z -buffer and the set $\{v_j\}$ our raycasting-blobs method still does not allow for real-time interaction. Therefore, it is desirable to have a method that preserves the detail of Fig. 4.3(a) in conjunction with real-time interaction. In the next section we introduce a method that takes advantage of the polygon-projection methods as

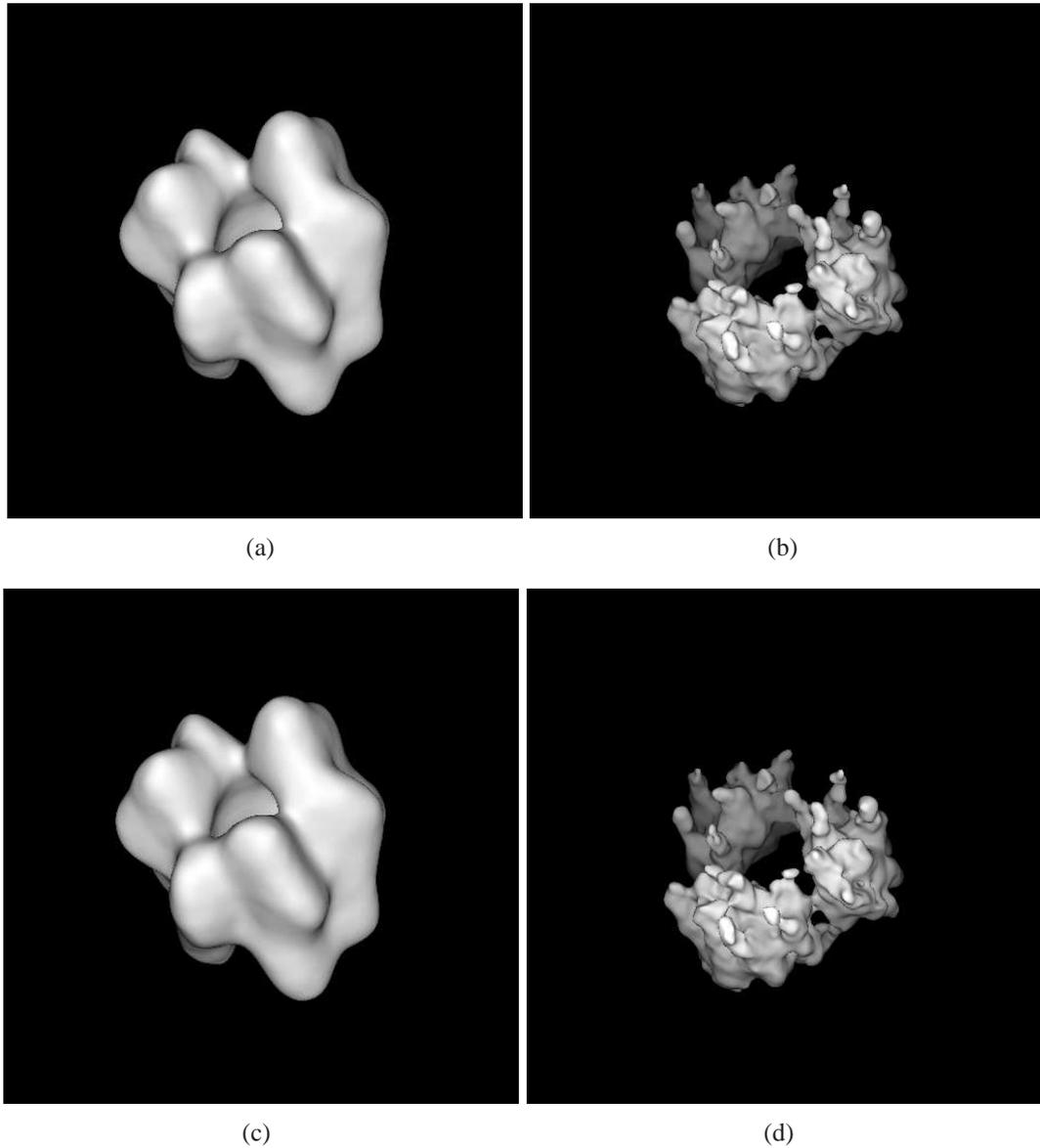


Fig.4.3: Three-dimensional representations of the macromolecular complexes (a) DnaB·DnaC and (b) bacteriorhodopsin produced by the modified raycasting technique described in Section 4.1. They are identical to those produced by the original implementation of the raycasting-blobs method show in (c) DnaB·DnaC and (d) bacteriorhodopsin. While the real shape of the DnaB·DnaC is unknown, we can compare (b) and (d) with the “real” shape of the protein in Fig. B.2 (at an arbitrary threshold that allows the visualization of the seven α -helices). Both images (b) and (d) represent the surface that encloses the same volume as that represented in Fig. B.2.

well as of the non-cubic grids.

4.3 Representation of Implicit Surfaces from Voxelized Sets

In this section we present an alternative way to display an accurate approximation to the implicit surface of a density function υ . Since this method allows the display of the approximate surface in a fraction of a second, it also provides us with a real-time capability to examine the surfaces. Yet, as we will see, there is only some deterioration in image quality as compared to Fig. 4.3(a) and (b) (and similarity in image quality as compared to Fig. 3.13).

4.3.1 Voxels

We need to introduce the concept of *voxels*, short for volume elements, that are commonly used to describe images in 3D. We define the *Voronoi neighborhood* of any element \bar{g} (in a grid Γ which is a set of points in \mathbb{R}^3), by

$$N_{\Gamma}(\bar{g}) = \{\bar{x} \in \mathbb{R}^3 \mid \text{for all } \bar{h} \in \Gamma, |\bar{x} - \bar{g}| \leq |\bar{x} - \bar{h}|\}, \quad (4.2)$$

i.e., the Voronoi neighborhood of \bar{g} consists of all points that are not nearer to any other point of Γ than they are to \bar{g} . The Voronoi neighborhoods associated with a grid in three-dimensional space are commonly known as voxels. The voxels of the grid G_{Δ} of (3.3) are cubes with volume equal to Δ_{sc}^3 , see Fig. 4.4(a). The voxels of B_{Δ} of (3.4) are truncated

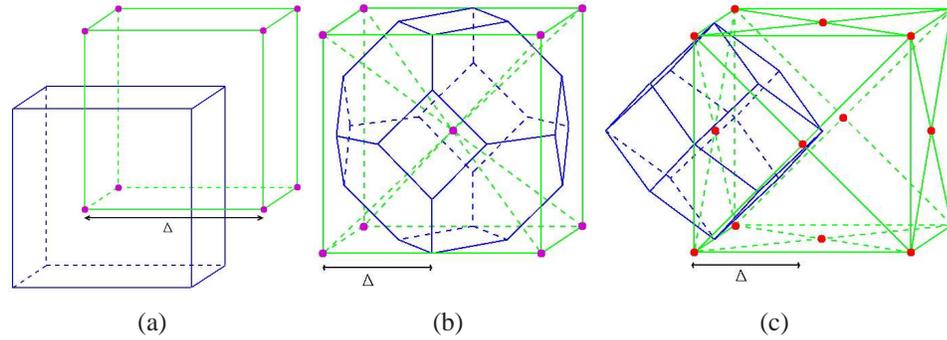


Fig.4.4: We present the voxels associated with (a) the simple cubic grid (which are cubes whose centers are points of the grid), (b) the body-centered cubic (which are truncated octahedra), and (c) the face-centered cubic grid (which are rhombic dodecahedra).

octahedra (polyhedra with six square faces and eight hexagonal faces) with volume equal to $4\Delta_{bcc}^3$, see Fig. 4.4(b). Finally, the voxels of F_Δ of (3.5) are rhombic dodecahedra (polyhedra with 12 identical rhombic faces) with volume equal to $2\Delta_{fcc}^3$, see Fig. 4.4(c). In all cases, a voxel can be referred to by the grid point with which it is associated; thus we talk about a cubic voxel \bar{c} , where $\bar{c} \in G_\Delta$.

In Section 3.2 it was mentioned that the *bcc* grid is the best grid to sample a bandlimited function in \mathbb{R}^3 . However, the *fcc* grid is easier to handle than the *bcc* grid for the purposes of image processing and graphics. (For example, the voxels in the *fcc* grid have only one kind of face, whereas the *bcc* voxels have two kinds of faces.) Much of what is reported in the rest of this chapter is based on the *fcc* grid.

Any grid in \mathbb{R}^3 provides us with as a tessellation of it into voxels. If some voxels are assigned the value 1 and the rest the value 0, then we obtain a *3D binary image*. Boundary elements (*bels*) are common faces of 1-voxels and 0-voxels. We call a component (i.e., a

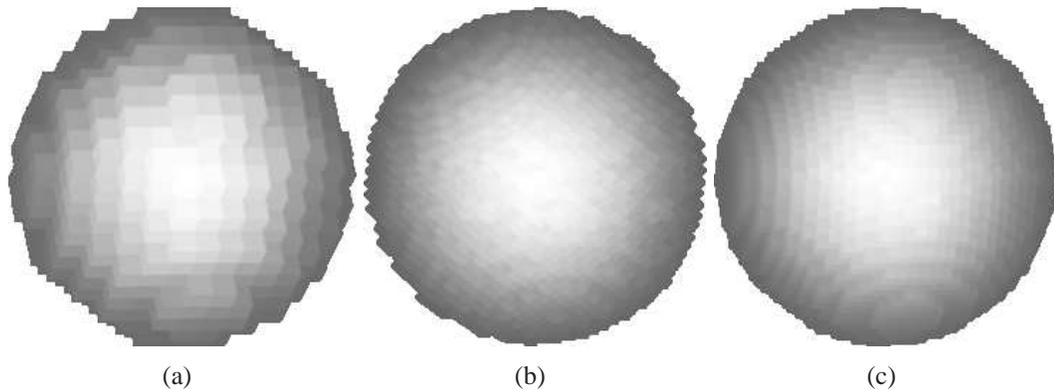


Fig.4.5: Computer graphic display of a sphere using different grids. (a) is the display based on a simple cubic grid with voxels of the same volume as the display based on the *fcc* grid used for (b). The image (c) corresponds to a display based on a simple cubic grid with voxels of volume equal to one eighth of the voxel volume in the other two images (reproduced from [51]).

maximal “connected” subset) of the set of bels, a *boundary* (this definition depends on how connectivity of bels is defined; more is said about this below).

Several methods have been developed to track a boundary in a 3D binary image [5, 39, 95], most of them work with cubic voxels. However, boundaries utilizing cubic voxels have some undesirable properties. One of them is that the 90° angle between the faces of a cubic voxel results in a “blocky” appearance when displayed on a computer screen; see Fig. 4.5(a). A way to overcome such problems is by utilizing a different type of voxel.

This is the approach we take, utilizing rhombic dodecahedral rather than cubic voxels (see Fig. 4.5(b)). It turns out that a minor modification of an algorithm for boundary tracking for cubic voxels [5] results in an algorithm capable of boundary tracking for both cubic and rhombic dodecahedral voxels [36].

Since F_Δ is a subset of G_Δ (see Fig. 3.3), every binary image over F_Δ gives rise to a *derived* binary image over G_Δ , simply by assigning 0 to those elements of G_Δ which are not in F_Δ . This is the observation which allows us to take advantage of an existing algorithm for boundary tracking for cubic voxels to produce our boundary-tracking algorithm for rhombic dodecahedral voxels.

4.3.2 Adjacencies for Cubic and Dodecahedral Voxels

For any element \bar{g} in a grid Γ there is a set of “neighbors.” Such neighbors are generally determined by a symmetric binary relationship (or *adjacency*) on Γ [51]. We define two adjacencies (symmetric binary relationships) ω and ϖ on G_Δ (and hence on F_Δ) by:

$$(\bar{c}, \bar{d}) \in \omega \Leftrightarrow |c - d| = \Delta \tag{4.3}$$

and

$$(\bar{c}, \bar{d}) \in \varpi \Leftrightarrow \Delta \leq |\bar{c} - \bar{d}| \leq \sqrt{2}\Delta \tag{4.4}$$

The grid points in G_Δ can be both ω -adjacent and ϖ -adjacent. When two grid points in G_Δ are ω -adjacent then their associated cubic voxels share a face, when two points in G_Δ are ϖ -adjacent then the associated voxels can share either a face or an edge. In fact, just as we can identify a cubic voxel with a $\bar{c} \in G_\Delta$, we can also identify the six faces of this voxel

by the six possible ordered pairs $(\bar{c}, \bar{d}) \in \omega$ and the twelve edges of the voxel by the twelve possible ordered pairs $(\bar{c}, \bar{d}) \in \bar{\omega} - \omega$.

Two grid points of F_Δ cannot possibly be ω -adjacent; see (3.5) and (4.3). They may be $\bar{\omega}$ -adjacent; in which case the associated rhombic dodecahedral voxels share a face, see Fig. 4.4(c). The twelve faces of the rhombic dodecahedral voxel $\bar{c} \in F_\Delta$ can be identified by the twelve possible ordered pairs $(\bar{c}, \bar{d}) \in \bar{\omega} = \bar{\omega} - \omega$ (since, for F_Δ , ω is empty). Thus, the center of an edge of a cubic voxel \bar{c} which is in F_Δ (i.e., a position in the G_Δ matching

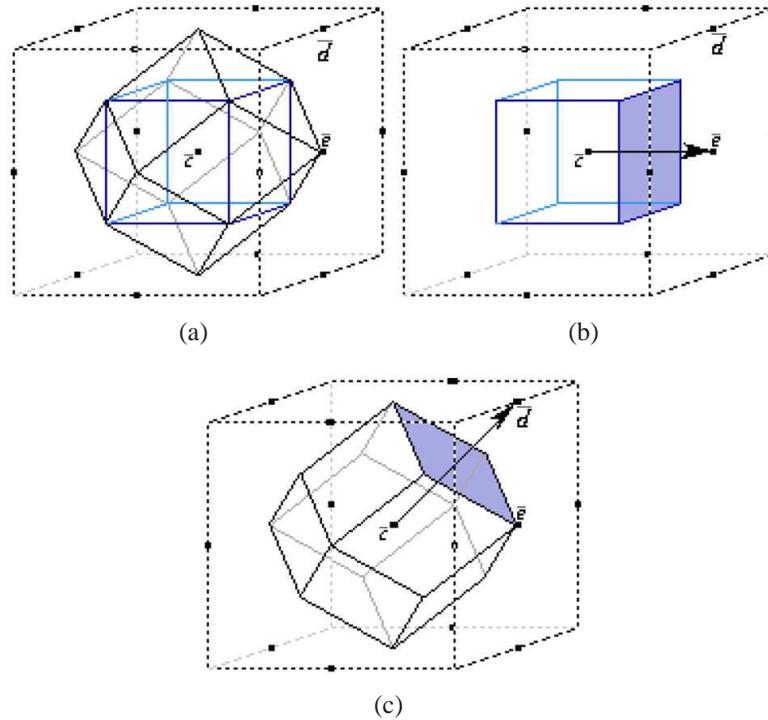


Fig.4.6: The cubic and rhombic dodecahedral voxels, identified by a point \bar{c} in F_Δ . (a) The edge (\bar{c}, \bar{d}) of the cubic voxel is a diagonal of the face (\bar{c}, \bar{d}) of the rhombic dodecahedral voxel. (b) The face (\bar{c}, \bar{e}) of a cubic voxel is produced when the two voxels \bar{c} and \bar{e} in G_Δ are ω -adjacent. (c) The face (\bar{c}, \bar{d}) of a rhombic dodecahedral voxel is produced when the two voxels \bar{c} and \bar{d} in G_Δ are $(\bar{\omega} - \omega)$ -adjacent.

a valid position in the F_Δ) corresponds to the center of a face of the rhombic dodecahedral voxel \bar{c} ; see Fig. 4.6.

4.3.3 Definition of Boundaries

We first define connectivity of bels (and, consequently, the notion of a boundary) for cubic voxels. The notion of a boundary for rhombic dodecahedral voxels will be derived from that for cubic voxels.

On each face (\bar{c}, \bar{d}) of a voxel \bar{c} , we superimpose two directions, as indicated in Fig. 4.7 (these directions depend only on the orientation $\bar{d} - \bar{c}$ of the face). The directions may be represented by two vectors \bar{a}_1 and \bar{a}_2 in G_Δ , which have the properties that they are both ω -adjacent to $\bar{0}$ and are perpendicular to $\bar{d} - \bar{c}$ and to each other. If (\bar{c}, \bar{d}) is a bel in a 3D binary image on G_Δ (which implies that \bar{c} is a 1-voxel, \bar{d} is a 0-voxel and $(\bar{c}, \bar{d}) \in \omega$), then

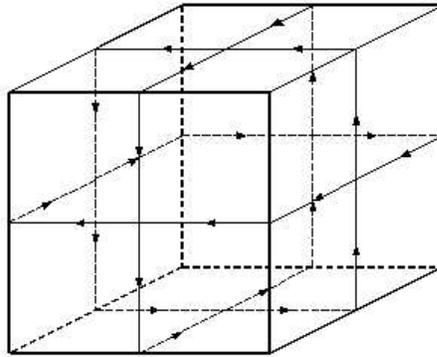


Fig.4.7: Superimposition of directions on voxel faces.

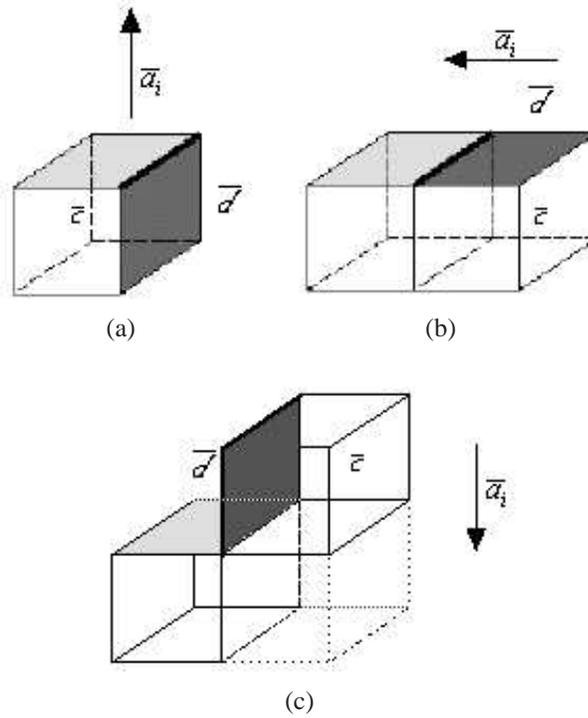


Fig.4.8: Definition of bel-adjacencies for cubic voxels. (The light faces are bel-adjacent from the dark faces.)

\bar{a}_i gives rise to a bel (\bar{c}_i, \bar{d}_i) *bel-adjacent* from it, defined as follows (consult Fig. 4.8). If $\bar{d} + \bar{a}_i$ is 1-voxel (case (c) in Fig. 4.8), then $\bar{c}_i = \bar{d} + \bar{a}_i$ and $\bar{d}_i = \bar{d}$. If $\bar{d} + \bar{a}_i$ is a 0-voxel and $\bar{c} + \bar{a}_i$ is a 1-voxel (case (b) in Fig. 4.8), then $\bar{c}_i = \bar{c} + \bar{a}_i$ and $\bar{d}_i = \bar{d} + \bar{a}_i$. If both $\bar{d} + \bar{a}_i$ and $\bar{c} + \bar{a}_i$ are 0-voxels (case (a) in Fig. 4.8), then $\bar{c}_i = \bar{c}$ and $\bar{d}_i = \bar{c} + \bar{a}_i$.

In any 3D binary image over the grid G_Δ , we say that a bel \bar{b}' is *connected* from a bel \bar{b} , if there exists a sequence $\langle \bar{b}^{(0)}, \bar{b}^{(1)}, \dots, \bar{b}^{(K)} \rangle$ of bels, such that $\bar{b}^{(0)} = \bar{b}$, $\bar{b}^{(K)} = \bar{b}'$ and, for $1 \leq k \leq K$, $\bar{b}^{(k)}$ is bel-adjacent from $\bar{b}^{(k-1)}$. It has been shown in [56] that connectivity of bels is an equivalence relation and hence partitions the set of all bels into components;

each of these components is defined to be a *boundary*. Hence, given any bel, the unique boundary containing it is the set of all bels connected from it.

Consider now a binary image over F_Δ . In the derived binary image over G_Δ (refer to Fig. 4.8):

1. case (b) cannot arise;
2. in case (c), the rhombic dodecahedral face $(\bar{c}, \bar{d} + \bar{a}_i)$ is not a bel, since both \bar{c} and $\bar{d} + \bar{a}_i$ are 1-voxels;
3. in case (a), $(\bar{c}, \bar{d} + \bar{a}_i)$ is a bel of the binary image over F_Δ .

This leads to the following definition of boundaries in binary images over F_Δ . Every boundary L_G in the derived binary image over G_Δ gives rise to a boundary L_F in the binary image over F_Δ : L_F is defined as the collection of all bels $(\bar{c}, \bar{d} + \bar{a}_i)$ for which (\bar{c}, \bar{d}) and $(\bar{c}, \bar{c} + \bar{a}_i)$ are in L_G . It can be shown that the boundaries defined in this way are also components of bels under an appropriate definition of bel-adjacency for the *fcc* grid [51].

4.3.4 A Boundary Tracking Algorithm

A boundary-tracking algorithm for 3D binary images on the cubic grid was proposed in [5]. Given an arbitrary bel (\bar{c}_0, \bar{d}_0) of the binary image, this algorithm outputs the list L of all bels in the boundary that contains (\bar{c}_0, \bar{d}_0) . It does this by iteratively determining the bels which are bel-adjacent from those already known to be in L , using the \bar{a}_i in Fig. 4.8,

and the definition of Fig. 4.7. The mathematical properties of this algorithm are analyzed in [56] and its computational behavior is discussed in [33]. Essential in the algorithm is the use of an auxiliary set M of the once visited bels, which greatly simplifies the prevention of retracking already tracked parts of the boundary. Due to the relationship discussed in the last section, the algorithm can be modified so that it is capable of dealing with both the cubic grid and the rhombic dodecahedral grid (see Algorithm 4.1). The output of the algorithm can also be characterized (both in the G_Δ and in the F_Δ grids) as the set of bels between elements of that connected (in some appropriate sense) component of the 1-voxels which contains \bar{c}_0 and elements of that connected (in some appropriate sense) component of the 0-voxels which contains \bar{d}_0 [51].

4.4 Experiments

We illustrate the output of our algorithm on data from electron microscopy from the DnaB·DnaC and the bacteriorhodopsin. Both the DnaB·DnaC and the bR were reconstructed using ART with the set $\{\bar{p}_j\}$ arranged on $B_{\frac{1}{\sqrt{2}}}$ and blob parameters $a = 2.40$ and $\alpha = 13.36$. In the first instance we evaluated the density function using (2.28) at points of $F_{1.68}$. However, for display purposes it seemed better to increase the sampling resolution, see Fig. 4.9. Therefore, to create images more similar to those in Fig. 3.10 we discretized the density function at points of $G_{0.25}$ and $F_{0.42}$. This way we made the area of the faces in the two data sets the same; thus, providing a reasonable way of comparing the outputs of

Algorithm 4.1 Algorithm to track boundaries in both the cubic and the rhombic dodecahedral grid.

1. Insert two copies of (\bar{c}_0, \bar{d}_0) into M .
2. If using Cubic Grid, then output (\bar{c}_0, \bar{d}_0) into L .
3. Queue (\bar{c}_0, \bar{d}_0) in X .
4. While X is not empty do:
 - (a) Remove a (\bar{c}, \bar{d}) from X .
 - (b) Find the bels (\bar{c}_1, \bar{d}_1) and (\bar{c}_2, \bar{d}_2) which are bel-adjacent from (\bar{c}, \bar{d}) .
 - (c) For $i=1,2$ do:

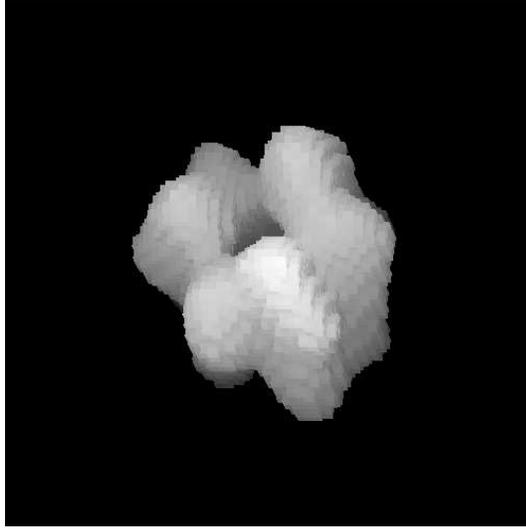
Boundary if in $F_\Delta \rightarrow$	If using Rhombic Dodecahedral Grid and $\bar{c}_i = \bar{c}$, then output $(\bar{c}, \bar{d} + \bar{a}_i)$ into L .
---	--

If (\bar{c}_i, \bar{d}_i) is in M :
 Then remove (\bar{c}_i, \bar{d}_i) from M .
 Else:

Boundary if in $G_\Delta \rightarrow$	If using Cubic Grid, then output (\bar{c}_i, \bar{d}_i) into L .
---	--

Queue (\bar{c}_i, \bar{d}_i) in X .
 Insert (\bar{c}_i, \bar{d}_i) into M .
 End else.
 End if.
 End for.
 End while.

the algorithm on the two different kinds of grid. In order to create the binary data sets we use the same threshold used for those images in Fig. 4.3 assigning 1 to those grid points which correspond to the macromolecular complexes and 0 to all other grid points. Then



(a)

Fig.4.9: Three-dimensional display of the DnaB·DnaC approximated as a collection of rhombic faces from rhombic dodecahedral voxels at points of $F_{1.68}$.

we applied our algorithm to detect the boundary of the digital approximations (in the two spaces) of the macromolecular complexes. The resulting surfaces are displayed in Fig. 4.10 and Fig. 4.11.

The first thing to note about these figures is that (even though the same set of TEM slices, the same threshold, and the same area for the faces were used to produce the surfaces) the number of faces in the digital approximation on the cubic grid is approximately a third more than the number of faces in the digital approximation based on the rhombic dodecahedral grid. This is because the larger number of orientations of the rhombic dodecahedral faces allows us to fit the underlying biological surface more tightly. This also has a consequence on computational costs: the time required by the algorithm to detect

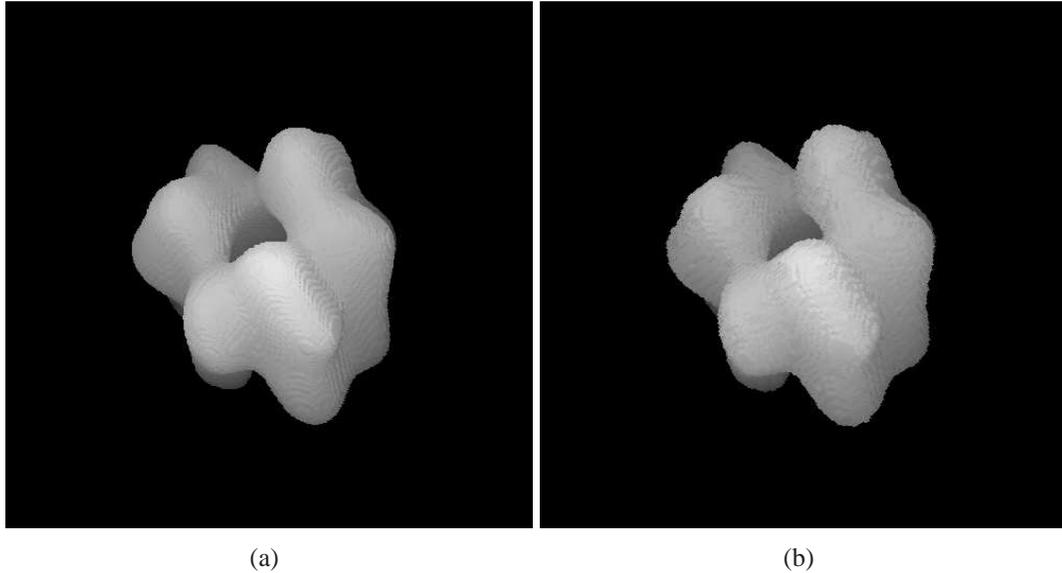


Fig.4.10: Three-dimensional display of the DnaB·DnaC (reconstructed by ART using the same parameters as in Fig. 3.10(b)) approximated as (a) a collection of 82,310 faces of cubic voxels at points of $G_{0.25}$ and (b) a collection of 63,769 faces of rhombic dodecahedral voxels at points of $F_{0.42}$. These images were created with the same threshold value used in Fig. 4.3(a).

the surfaces displayed in Figures 4.10(b) and 4.11(b) is less than the time required by the same algorithm to detect the surfaces displayed in Figures 4.10(a) and 4.11(a). The time required to produce the computer graphic display of the already detected surface was approximately one and a half times longer when using the cubic grid than when using the rhombic dodecahedral grid.

In displaying the surfaces we have chosen a rendering methodology which does not hide the digital nature of the surfaces in question. We have done this so that the reader can observe the appearance of the digital surfaces themselves (from the same data one can produce much smoother appearing surfaces). A great advantage of this approach is that

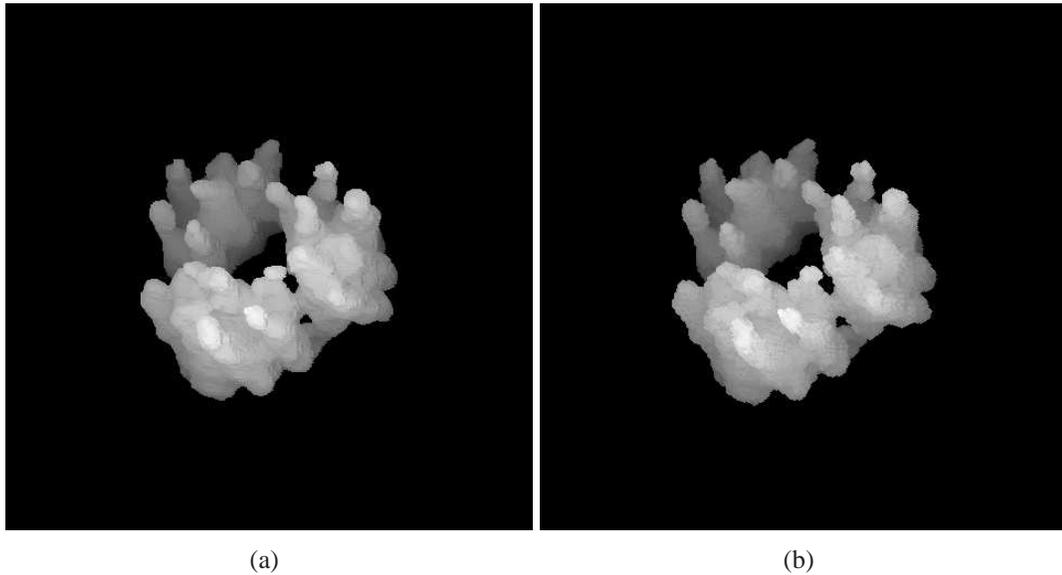


Fig.4.11: Three-dimensional display of the bR (reconstructed by ART using the same parameters as in Fig. 3.10(b)) approximated as (a) a collection of 82,310 faces of cubic voxels at points of $G_{0.25}$ and (b) a collection of 63,769 faces of rhombic dodecahedral voxels at points of $F_{0.42}$. These images were created with the same threshold value used in Fig. 4.3(b).

once the voxelized density functions have been created, the most expensive part of the operation in our experience (for these sets the maximum time was 120 s), the boundary tracking algorithm and the rendering operate rapidly: the boundary tracking took at most 1.3 s using the same computer as reported in the previous section and the time to render the result is a fraction of a second; thus, allowing us to rotate the boundary in real time.

The time necessary to produce a high quality image by the raycasting-blobs algorithm described in Section 3.3 is considerable (except for very simple objects) as reported in Section 3.5. We have introduced a preprocessing stage to our original implementation of the raycasting-blobs method that reduces the image-rendering time by reducing the space

of search in two ways: a) by selecting a small number of coefficients (which contribute to the formation of the object of interest) and b) by providing a good estimate of where the surface is located for every ray. This improvement has resulted in more practical timings while preserving the quality of the generated images. Yet, the new implementation does not allow for real time interaction.

As part of our research we have developed a method that approximates a surface by using identical rhombi (which are the faces of the rhombic dodecahedral voxels) instead of squares (the faces of the common cubic voxel). This algorithm requires discretizing the density function at fewer grid points than those necessary for the simple cubic grid for the same image quality. The method also takes advantage of algorithms developed previously for the simple cubic grid. Finally, our algorithm allows for real time interaction and produces images of similar quality to those produced with OpenDXTM but requires less resources and time than OpenDX to create the approximation to a surface.

The two algorithms presented in this chapter provide two tools that produce three-dimensional displays in practical times. In experiments, we have seen that the raycasting-blobs method offers slightly better visual results. However, we do not, at this stage, recommend either of them as the method of choice for 3D display. That would require further studies measuring quality, accuracy and speed of these methods in comparison with other state of the art methods, such as shell rendering [41, 117].

Chapter 5

EXTRACTION OF SUBUNITS FROM DENSITY FUNCTIONS

As mentioned in Section 2.1, biological macromolecules very frequently present some kind of symmetry and the localization of all the subunits forming a polymer is of great importance for the understanding of the conformational changes a macromolecule can experience. We have already presented, in the same section, methods used to minimize damage to biological macromolecules by considering the presence of several copies of the same macromolecule in a preparation. The symmetry of macromolecules is also made use of during the process of reconstruction, see [74]. However, on many occasions the density functions (produced by reconstruction algorithms) of macromolecules show repetitive subunits (presumably forming a single polypeptide chain) that are not perfectly distinguishable, see Fig. 3.11. In this chapter we propose that the existence of such repetitive subunits can be utilized to improve the presentation of the macromolecule after the reconstruction process.

5.1 Identification of Classes of Subunits and Their Alignment

The basic idea of our approach comes from previous work to extract characters in a digitized document written in a language whose characters are not known *a priori* [44]. In such an approach the individual digitized characters in the document are first identified by thresholding. In order to improve the quality of the characters, an alphabet of the characters in the document is created by grouping the characters into classes. A model for every class is created by averaging all the characters composing a single class. Finally, every character in a class is replaced in the document by its model.

We propose a similar approach in which we identify subunits of unknown shape that will be grouped into different classes. The number of classes into which the subunits are grouped should match the number of different basic subunits that form the macromolecule. The subunits grouped into a particular class can be used to obtain a model of the ideal subunit of that class by averaging its elements. Thereafter, the model can be utilized to replace the subunits of the corresponding class. We conjecture that this averaging process increases the quality of the representation of the subunits and, consequently, of the macromolecule.

An important difference between our approach and that presented in [44] is that in our application the possible subunits that belong to a class differ in their location and their orientation whereas the characters in a document differ only in their location; this introduces another level of complexity to our problem. On the other hand, in our problem we have access to accurate information regarding the mass of the macromolecular complex (by means

of biochemical studies) and to the possible number of subunits (by means of techniques such as crystallography).

5.1.1 Segmentation

It is known that segmenting a volume by thresholding is often not the best approach, because the structural components in a volume can overlap in their density values and therefore they are not uniquely defined by ranges of such values [71, 92, 93, 115, 119]. There are alternative approaches to segmenting a volume, namely physics-based methods, artificial intelligence, statistical-probabilistic methods, region growing methods, and methods based on fuzzy set theory [45, 92]. In this work we will use a technique based on fuzzy set theory [7, 12, 102, 128]. The reason for this selection is that there is a wealth of successful experience using these techniques acquired in the medical imaging community [15, 22, 34, 60, 90, 93, 105, 118, 116, 119, 120] and in our group [20, 21, 52]. In particular, in [20] we have used this approach on *fcc* grids to produce good segmentations from noisy reconstructed distributions produced by PET.

5.1.2 Fuzzy Segmentation

Rosenfeld [102] suggested the application of fuzzy concepts to the field of image processing. However, the introduction of fuzzy connectedness and of fuzzy affinity were first introduced by Udupa and Samarasekera [119]. Our approach (introduced in [52]) is based on the theory presented in [119], but is generalized to arbitrary digital spaces [51]. In this

work a *digital space* will be defined as a set of voxels, some pairs of which are identified as being *adjacent*. Segmentation approaches using concepts of fuzzy set theory (for the sake of simplicity we will refer to them as *fuzzy segmentation*) assign to each voxel and each object, a grade of membership of that voxel in that object, and this grade can be between 0 (nonmembership) and 1 (full membership) [12, 93, 119, 128].

In the fuzzy connectedness context we define a *chain* as a sequence of voxels and the *links* of the chain as the pairs of consecutive voxels in the sequence. In our approach, the *strength* of a link is set to zero for not adjacent voxels and it is automatically determined according to statistical properties of pairs of adjacent voxels in a set of seeds (voxels selected by the user as being inside the object of interest). We set the strength of a chain to be the strength of its weakest link. The *fuzzy connectedness* between any pair of voxels is the strength of the strongest chain between them [103, 119]. In our implementation of multiseeded segmentation we generalize this approach by allowing each object to have its own definition of strength for the links and its own set of seed voxels. Roughly, each object is defined as the set of voxels that are connected in a stronger way to one of the seeds of that object than to any of the seeds of the other objects. A more precise definition (from [52]) is given below.

The computationally most expensive task in determining the objects based on the seeds is the calculation of the multiple fuzzy connectedness of all the voxels to the seed voxels by means of finding the strongest chain between a voxel and one or more seed voxels. For this

task a greedy algorithm, presented in [52], is used to efficiently create the segmentation of a discretized density function υ . It is worth noticing that in the literature dynamic programming approaches have been successfully used [89, 103, 104, 106, 118, 119] , but in [21] it was shown that greedy algorithms have a superior performance. Now we make a brief formal introduction of the fuzzy segmentation algorithm and of some concepts necessary to understand it.

While the general theory allows us to deal with an arbitrary number of objects, in our application there are only two objects: the macromolecule and its background. Hence in the discussion that follows, we replace the arbitrary positive integer M of [52] by the fixed integer 2.

A *2-semisegmentation* of a set V of voxels is a function σ which maps each $\bar{c} \in V$ into an 3-dimensional vector $\sigma^{\bar{c}} = (\sigma_0^{\bar{c}}, \sigma_1^{\bar{c}}, \sigma_2^{\bar{c}})$, such that $\sigma_0^{\bar{c}} \in [0, 1]$ (i.e., it is nonnegative but not greater than 1) and for at least one m , in the range $1 \leq m \leq 2$, $\sigma_m^{\bar{c}} = \sigma_0^{\bar{c}}$ and for the other m it is either 0 or $\sigma_0^{\bar{c}}$. We say that σ is a 2-segmentation if, for every voxel \bar{c} , $\sigma_0^{\bar{c}}$ is positive.

A *fuzzy voxel affinity* on V is a function $\psi : V^2 \rightarrow [0, 1]$. We call $\psi(\bar{c}, \bar{d})$ the ψ -*strength* of (\bar{c}, \bar{d}) . In this work we make fuzzy voxel affinity a “local” concept by insisting that $\psi(\bar{c}, \bar{d}) = 0$ if \bar{c} and \bar{d} are not adjacent.

There are no restrictions on the definition of fuzzy voxel affinity for adjacent voxels and, therefore, the function can be defined based on the problem at hand. In the literature there are examples of different definitions of ψ , see; e.g., [106, 119, 120]. Nevertheless, it

is common to use an affinity function ψ for adjacent grid points \bar{c} and \bar{d} of the general form [21, 51, 52, 119]:

$$\psi(\bar{c}, \bar{d}) = g_1(\nu(\bar{c}) + \nu(\bar{d})) + g_2(|\nu(\bar{c}) - \nu(\bar{d})|). \quad (5.1)$$

Typical choices for the g_i ($i \in \{1, 2\}$) are

$$g_i(x) = \frac{1}{2} e^{-\frac{(x-m_i)^2}{2\zeta_i^2}}, \quad (5.2)$$

where m_1 and ζ_1 are assumed to be the mean and standard deviation of the sum $\nu(\bar{c}) + \nu(\bar{d})$ over all adjacent grid points pairs (\bar{c}, \bar{d}) in the set of seeds and m_2 and ζ_2 are, in turn, assumed to be the mean and standard deviation of $|\nu(\bar{c}) - \nu(\bar{d})|$ over all seed voxel pairs (\bar{c}, \bar{d}) . However, the definition of g_i is not restricted to a Gaussian function, other functions also lead to legitimate affinity functions and might be computationally less expensive. One example is [120]

$$g_i(x) = \begin{cases} \frac{1}{2} - \frac{|x - m_i|}{6\zeta_i}, & \text{if } |x - m_i| \leq 3\zeta_i, \\ 0, & \text{otherwise.} \end{cases} \quad (5.3)$$

Thus, we can select a different function, such as a linear function, that permits simpler and faster computations. Nevertheless, it is our experience that the form in (5.2) produces “good” results at affordable speeds.

We define a *chain* in $U(\subseteq V)$ from $\bar{c}^{(0)}$ to $\bar{c}^{(K)}$ to be a sequence $\langle \bar{c}^{(0)}, \dots, \bar{c}^{(K)} \rangle$ of voxels in U . Such a chain has a ψ -strength given by the ψ -strength of its weakest link $(\bar{c}^{(k-1)}, \bar{c}^{(k)})$, $1 \leq k \leq K$. (In case $K = 0$, the ψ -strength is defined to be 1.) We say that U is ψ -connected if for every pair of distinct voxels in U there is a chain in U of positive ψ -strength from $\bar{c}^{(k-1)}$ to $\bar{c}^{(k)}$.

If there are multiple objects to be segmented, it is reasonable that each should have its own fuzzy voxel affinity, which leads to the following. A *2-fuzzy graph* is a pair (V, Ψ) , where V is a nonempty finite set of voxels and $\Psi = (\psi_1, \psi_2)$ and ψ_m (for $1 \leq m \leq 2$) is a fuzzy voxel affinity such that V is $(\min_{1 \leq m \leq 2} \psi_m)$ -connected. (This is defined by $(\min_{1 \leq m \leq 2} \psi_m)(\bar{c}, \bar{d}) = \min_{1 \leq m \leq 2} \psi_m(\bar{c}, \bar{d})$.) For a 2-semisegmentation σ of V and for $1 \leq m \leq 2$, the chain $\langle \bar{c}^{(0)}, \dots, \bar{c}^{(K)} \rangle$ is said to be a σm -chain if $\sigma_m^{\bar{c}^{(k)}} > 0$, for $0 \leq k \leq K$. Further, for $U \subseteq V$, $W \subseteq V$ and $\bar{c} \in V$, we use $\mu_{\sigma, m, U, W}(\bar{c})$ to denote the maximal ψ -strength of a σm -chain in U from a voxel in W to \bar{c} . (This is equal to 0 if there is no such a chain.)

The feasibility of creating a unique segmentation by using the concepts introduced above is given by the following theorem from [52].

Theorem 5.1 *If (V, Ψ) is a 2-fuzzy graph and, for $1 \leq m \leq 2$, O_m is a subset (of seed voxels) of V such that at least one of these subsets is nonempty, then there exists a unique 2-semisegmentation (which is, in fact, an 2-segmentation) σ of V with the following property.*

For every $\bar{c} \in V$, if for $1 \leq n \leq 2$

$$s_n^{\bar{c}} = \begin{cases} 1, & \text{if } \bar{c} \in O_n, \\ \max_{\bar{d} \in V} (\min(\mu_{\sigma, n, V, O_n}(\bar{d}), \psi_n(\bar{d}, \bar{c}))), & \text{otherwise,} \end{cases} \quad (5.4)$$

then for $1 \leq m \leq 2$

$$\sigma_m^{\bar{c}} = \begin{cases} s_m^{\bar{c}}, & \text{if } s_m^{\bar{c}} \geq s_n^{\bar{c}} \text{ for } 1 \leq n \leq 2, \\ 0, & \text{otherwise.} \end{cases} \quad (5.5)$$

In [52] a greedy algorithm is provided that receives as input two sets of voxels (O_m , for $1 \leq m \leq 2$) and updates, during its execution, the current 2-semisegmentation σ ; producing, at the end, a 2-semisegmentation that satisfies the property of the Theorem 5.1. Below we reproduce such an algorithm for sake of completeness, for a detailed explanation of the performance of the algorithm we refer to [20, 52].

An intuitive picture of the algorithm is the following. There are two competing armies (one corresponding to each object). Initially they both have full strength and they occupy their respective seed voxels. Both armies try to increase their respective territories, but the moving from a voxel to another one reduces the strength of the soldiers to be the minimum of their strength on the previous voxel and the affinity (for that army or object) between the voxels. At any given time, a voxel will be occupied by the soldiers of the armies which were not weaker than any other soldiers who reached that voxel by that time. Eventually a steady

Algorithm 5.1 Multiseeded segmentation algorithm.

```

1. for  $\bar{c} \in V$ 
2.   do for  $m \leftarrow 0$  to 2
3.     do  $\sigma_m^{\bar{c}} \leftarrow 0$ 
4.    $H \leftarrow \emptyset$ 
5.   for  $m \leftarrow 1$  to 2
6.     do for  $\bar{c} \in O_m$ 
7.       do if  $\sigma_0^{\bar{c}} = 0$  then  $H \leftarrow H \cup \{\bar{c}\}$ 
8.          $\sigma_0^{\bar{c}} \leftarrow \sigma_m^{\bar{c}} \leftarrow 1$ 
9.    $l \leftarrow 1$ 
10.  while  $l > 0$ 
11.    for  $m \leftarrow 1$  to 2
12.      do while  $O_m \neq \emptyset$ 
13.        do remove a voxel  $\bar{d}$  from  $O_m$ 
14.           $C \leftarrow \{\bar{c} \in V \mid \sigma_m^{\bar{c}} < \min(l, \psi_m(\bar{d}, \bar{c}))\}$ 
15.          while  $C \neq \emptyset$ 
16.            do remove a voxel  $\bar{c}$  from  $C$ 
17.               $t \leftarrow \min(l, \psi_m(\bar{d}, \bar{c}))$ 
18.              if  $l = t$  and  $\sigma_m^{\bar{c}} < l$  then  $O_m \leftarrow O_m \cup \{\bar{c}\}$ 
19.              if  $\sigma_0^{\bar{c}} < t$  then
20.                if  $\sigma_0^{\bar{c}} = 0$  then  $H \leftarrow H \cup \{\bar{c}\}$ 
21.                for  $n \leftarrow 1$  to 2
22.                  do  $\sigma_n^{\bar{c}} \leftarrow 0$ 
23.                  if  $\sigma_0^{\bar{c}} \leq t$  then  $\sigma_0^{\bar{c}} \leftarrow \sigma_m^{\bar{c}} \leftarrow t$ 
24.            while Maximum-Key( $H$ ) =  $l$ 
25.              Remove-Max( $H$ )
26.             $l \leftarrow$ Maximum-Key( $H$ )
27.          for  $m \leftarrow 1$  to 2
28.             $O_m \leftarrow \{\bar{c} \in H \mid \sigma_m^{\bar{c}} = l\}$ 

```

state is reached; this steady state satisfies the property of the Theorem 5.1. The sequential algorithm simulates this intuitively described parallel behavior of the two armies.

In our approach we use the *fcc* grid for the algorithm just presented assuming that the two classes in the reconstructed density function υ represent macromolecule and background, respectively. Consequently, we need to select seeds for just these two classes. In order to use our approach to identify the subunits of a macromolecule, we assume that we

can identify a number of points such that each subunit contains exactly one of these identified points. We base this on visual observation of Fig. 3.11, where a point in each subunit can be selected in spite of the components not being sharply defined. We have assumed that the higher values in the reconstructed density function certainly belong to the voxels in the macromolecule while those with the lowest values belong to the background. Under this assumption, we have created a program to automatically select these points. To select the seeds for the macromolecule class, the program applies a low-pass filter. After this operation the minimum, maximum and mean values of the smoothed density function are computed. In order to find the seeds for the macromolecule object the smoothed density function is thresholded to a value between the mean value and the maximum (the threshold value can be adjusted by the user). The result is then eroded in order to separate it into subunits. The erosion is performed iteratively until a single voxel is left in every subunit (in case more than one voxel is left in an iteration before eroding all the voxels in a subunit an arbitrary voxel is chosen). The seeds for the background class are selected in a similar fashion as for the macromolecular class, but with some appropriate adjustments: a) the threshold value is chosen between the mean and the minimum and b) an inverted threshold is performed. We have also implemented a program that allows the user the interactive selection of seed voxels and the modification of a set of seeds already created.

The algorithm described above works on a general set V of voxels. In our application we set V to be a subset of the *fcc* grid because of the suitability of this grid for image processing

(e.g., $\bar{\omega}$ -neighbors are at an equal distance and fewer voxels are needed to obtain the same resolution as compared to the simple cubic grid).

For every seed voxel \bar{o} selected by the method described above, there are twelve $\bar{\omega}$ -adjacent voxels \bar{d} , each of which provides a pair (\bar{o}, \bar{d}) . Because of the adjacency between voxels in F_Δ , some pairs of the twelve neighbors of \bar{o} are also adjacent to each other. This provides 24 more pairs (\bar{c}, \bar{d}) . Thus, each point \bar{o} provides 36 pairs (\bar{c}, \bar{d}) . All these points are used in the calculations of the m_i and ζ_i in (5.2) for $1 \leq i \leq 2$.

We represent the true number of subunits in the macromolecule by N and its true volume by \mathcal{V} . (We assume that both of these values are available to us by some alternative technique.) After creating the 2-segmentation σ we obtain the sets $\mathcal{K}_t = \{\bar{c} \mid \sigma_1^{\bar{c}} > t\}$. Let N_t denote the number of $\bar{\omega}$ -components (maximal $\bar{\omega}$ -connected subsets) in \mathcal{K}_t and \mathcal{V}_t be the sum of the volumes of the voxels in \mathcal{K}_t . We select the threshold t based on two criteria that $N_t = N$ and that $|\mathcal{V} - \mathcal{V}_t|$ is minimal.

5.1.3 Registration Methods

Let the $\bar{\omega}$ -components of \mathcal{K}_t be denoted by κ_i , $1 \leq i \leq N$. Before comparing the subunits κ_i with each other, it is necessary to bring them together or align them. For this operation we treat the κ_i s as rigid objects. (If, as expected, the specimen has been carefully prepared to reflect the three-dimensional configuration of a macromolecular complex at the moment of micrograph acquisition, see Section 2.5, then the shapes of the κ_i s reflect the shapes of

the biological subunits.) Therefore, we assume that two subunits belonging to a class differ only by translation and rotation. Based on these facts, we consider rigid motion approaches to align (also referred to as register) the κ_i s [18, 72, 121].

Principal Axes Method

We consider the Principal Axes Transformation (*PAT*) method to align two subunits κ_i and κ_j with $i \neq j$. This is a well-known and commonly used method in the image processing field [2, 4, 25, 28]. The idea behind this method comes from classical mechanics [37]; a rigid body in space only needs six independent generalized coordinates to specify its position. Chasles' Theorem [37] in mechanics states that it is always possible to represent any general displacement of a rigid body by a translation plus a rotation. Furthermore, it is possible to split the problem of rigid body motion into two separate phases, one solely concerned with the translation of the body and the other with its rotation.

The position of a rigid body is completely specified by locating a Cartesian set of coordinates fixed in the rigid body (*primed axes*) relative to the coordinate axes of the external space (*global axes*). Three of the values are needed to specify the coordinates of the origin of the primed axes. Three more values specify the orientation of the primed axes relative to a coordinate system parallel to the global axes, but with the same origin as the primed axes.

It is common practice to set the body's center of mass as the origin of the primed axes.

We define the center of mass of the subunit κ_i as

$$\bar{\eta}_{\kappa_i} = \frac{1}{|\kappa_i|} \sum_{\bar{c} \in \kappa_i} \bar{c}, \quad (5.6)$$

where $|\kappa_i|$ is the number of voxels in κ_i . To define the primed axes of the subunit κ_i we make use of the *matrix of inertia* \mathbf{A}_i associated with the κ_i :

$$\mathbf{A}_i = \begin{bmatrix} \sum_{\bar{c} \in \kappa_i} (c_2^2 + c_3^2) & - \sum_{\bar{c} \in \kappa_i} c_1 c_2 & - \sum_{\bar{c} \in \kappa_i} c_1 c_3 \\ - \sum_{\bar{c} \in \kappa_i} c_2 c_1 & \sum_{\bar{c} \in \kappa_i} (c_1^2 + c_3^2) & - \sum_{\bar{c} \in \kappa_i} c_2 c_3 \\ - \sum_{\bar{c} \in \kappa_i} c_3 c_1 & - \sum_{\bar{c} \in \kappa_i} c_3 c_2 & \sum_{\bar{c} \in \kappa_i} (c_1^2 + c_2^2) \end{bmatrix}. \quad (5.7)$$

The eigenvectors of the matrix \mathbf{A}_i define a set of Cartesian axes, called the *principal axes*. These Cartesian axes are chosen to be the primed axes and they can be represented by \bar{e}_1 , \bar{e}_2 and \bar{e}_3 , the unit-length eigenvectors of \mathbf{A}_i . Because the matrix \mathbf{A}_i is symmetric with all its elements being real numbers the eigenvalues are real. If the eigenvalues are different, then the eigenvectors form a unique set of three mutually orthogonal vectors. However, when two eigenvalues coincide then there is an overabundance of eigenvectors and this method fails to assign a unique set of primed axes to the subunit.

Generally, if two subunits differ only by rotation and translation, then they can be aligned by bringing together their centers of mass and then aligning their principal axes. The speed of this method and the ease of its implementation made it popular. A limitation

of it [108] is that when two objects are not identical (e.g., if there was a deformation in either of them), then PAT provides only an approximate alignment. This is an important issue to consider because noise or artifacts contaminate the reconstructed density function υ , resulting in slight differences in the repeated appearance of a particular subunit.

Iterative Closest Point

Anticipating the limitations of the PAT technique we decided to follow it up by a method to provide finer alignment of the subunits. We selected the Iterative Closest Point (*ICP* or *point matching algorithm*) for this task, a widely used method in the computer graphics and pattern recognition fields [11, 13, 94, 113, 127]. The ICP algorithm aligns a set κ_j to a reference set κ_i by iteratively finding the closest points between κ_j and κ_i and then computing the rigid body transformation to align the closest points to the set κ_i until convergence, or a margin of acceptable error, is met. The algorithm receives as input the two sets of points κ_j and κ_i in \mathbb{R}^3 and the output is a rotation matrix \mathbf{R} , a translation vector \bar{t} and a real value h such that $\mathbf{R}\kappa_j + \bar{t}$ matches κ_i by h . (All these terms and notation are clarified below.) The algorithm is initialized by roughly matching the sets κ_j and κ_i . In our application we initialize the algorithm by aligning κ_j and κ_i by PAT.

Below we present the ICP algorithm just after initialization followed by an explanation of its several steps. The rotation matrix \mathbf{R} and translation vector \bar{t} used in step 1 are the ones provided by PAT.

Algorithm 5.2 Iterative Closest Point algorithm.

1. $\kappa'_j \leftarrow \mathbf{R}\kappa_j + \bar{t}$
 2. **While** $\left| H(\kappa'_j, \kappa_i) - h \right| > \varepsilon$ **do**
 3. $h \leftarrow H(\kappa'_j, \kappa_i)$
 4. $Y \leftarrow C(\kappa'_j, \kappa_i)$
 5. $\mathbf{V} \leftarrow \mathbf{V}_Y$
 6. $\mathbf{M} \leftarrow \mathbf{M}_V$
 7. $\bar{q} \leftarrow \Xi(\mathbf{M})$
 8. $\mathbf{R} \leftarrow \mathbf{R}_{\bar{q}}$
 9. $\bar{t} \leftarrow \bar{\eta}_Y - \mathbf{R}_{\bar{q}}\bar{\mu}_Y$
 10. $\kappa'_j \leftarrow \mathbf{R}_{\bar{q}}\kappa'_j + \bar{t}$
 11. **end-do**
-

It is clear that we need to have a measure of similarity when comparing the alignment of two subunits κ_i and κ_j as well as when deciding whether the two belong to a given class. For this comparison we use the Hausdorff measure. The Hausdorff measure has become a standard distance in image processing [15, 43, 59]. Consequently, we consider the Hausdorff distance to compare the subunits κ'_j and κ_i .

The Hausdorff distance for two sets κ_i and κ'_j is defined by [59]:

$$H(\kappa_i, \kappa'_j) = \max(h(\kappa_i, \kappa'_j), h(\kappa'_j, \kappa_i)), \quad (5.8)$$

where $h(\kappa'_j, \kappa_i)$ is the non-symmetric directed Hausdorff distance from κ'_j to κ_i defined by

$$h(\kappa'_j, \kappa_i) = \max_{c \in \kappa'_j} \min_{d \in \kappa_i} |\bar{c} - \bar{d}|. \quad (5.9)$$

Intuitively, the function $h(\kappa'_j, \kappa_i)$ describes the amount of mismatch between the aligned subunit κ'_j and the subunit κ_i by finding the greatest distance between a grid point in κ'_j and a grid point in κ_i .

In [44] similar characters in a document are considered to be slightly misaligned only by translation, those with a greater misalignment are considered not to be of the same class. In our experiments we found that we can consider two subunits κ_i and κ_j as being in the same class whenever $H(\kappa_i, \kappa'_j) < 2.5$.

Let κ be a collection of points in \mathbb{R}^3 . We use γ_κ to denote a function from \mathbb{R}^3 into κ which has the property that for any point $\bar{c} \in \mathbb{R}^3$,

$$|\bar{c} - \gamma_\kappa(\bar{c})| \leq |\bar{c} - \bar{d}|, \quad \text{for any } \bar{d} \in \kappa. \quad (5.10)$$

For two sets of points κ'_j and κ_i in \mathbb{R}^3 we define

$$C(\kappa'_j, \kappa_i) = \{(\bar{c}, \gamma_\kappa(\bar{c})) \mid \bar{c} \in \kappa'_j\} \quad (5.11)$$

to be the set of closest points between the sets κ'_j and κ_i .

Let Y be a set of elements of $\mathbb{R}^3 \times \mathbb{R}^3$. Then, we create the cross-covariance matrix \mathbf{V}_Y

between elements of Y by

$$\mathbf{V}_Y = \sum_{(\bar{c}, \bar{d}) \in Y} (\bar{c} - \overline{\mu}_Y) (\bar{d} - \overline{\eta}_Y)^T, \quad (5.12)$$

where $\overline{\mu}_Y = \frac{1}{|Y|} \sum_{(\bar{c}, \bar{d}) \in Y} \bar{c}$ and $\overline{\eta}_Y = \frac{1}{|Y|} \sum_{(\bar{c}, \bar{d}) \in Y} \bar{d}$.

Given a 3×3 matrix \mathbf{V} , the symmetric 4×4 matrix $\mathbf{M}_\mathbf{V}$ is defined as

$$\mathbf{M}_\mathbf{V} = \begin{bmatrix} \text{Tr}(\mathbf{V}) & V_{2,3} - V_{3,2} & V_{3,1} - V_{1,3} & V_{1,2} - V_{2,1} \\ V_{2,3} - V_{3,2} & 2V_{1,1} - \text{Tr}(\mathbf{V}) & V_{1,2} + V_{2,1} & V_{1,3} + V_{3,1} \\ V_{3,1} - V_{1,3} & V_{1,2} + V_{2,1} & 2V_{2,2} - \text{Tr}(\mathbf{V}) & V_{2,3} + V_{3,2} \\ V_{1,2} - V_{2,1} & V_{1,3} + V_{3,1} & V_{2,3} + V_{3,2} & 2V_{3,3} - \text{Tr}(\mathbf{V}) \end{bmatrix}. \quad (5.13)$$

Given a 4×4 symmetric matrix \mathbf{M} we define $\Xi(\mathbf{M})$ to be a unit length eigenvector associated with a largest eigenvalue of the matrix \mathbf{M} .

Given a vector $\vec{q} \in \mathcal{S}^3$, we define the 3×3 matrix $\mathbf{R}_{\vec{q}}$ as

$$\mathbf{R}_{\vec{q}} = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2q_3 - q_1q_4) & 2(q_2q_4 + q_1q_3) \\ 2(q_2q_3 + q_1q_4) & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_3q_4 + q_1q_2) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix}. \quad (5.14)$$

It is easy to see that the matrix $\mathbf{R}_{\vec{q}}$ is a orthonormal matrix (i.e., $\mathbf{R}_{\vec{q}}\mathbf{R}_{\vec{q}}^T = \mathbf{I}$) with $\det \mathbf{R}_{\vec{q}} = 1$. An orthonormal matrix is the realization of an orthonormal transformation,

which is either a rigid rotation or a rotoinversion (rotation followed by a flip). (the matrix \mathbf{A} associated with an orthonormal transformation T has a determinant equal to 1 when T is a rigid rotation and equal to -1 when T is a rotoinversion.)

It is easy to verify that the rotation matrix \mathbf{R} that aligns two points \bar{c} and \bar{d} , whose magnitude is the same, is such that it maximizes $\langle \bar{c}, \mathbf{R}\bar{d} \rangle$. This is the idea behind the ICP algorithm, finding the rotation matrix \mathbf{R} that maximizes the sum

$$\sum_{(\bar{c}, \bar{d}) \in Y} \langle \bar{c}, \mathbf{R}\bar{d} \rangle. \quad (5.15)$$

It has been shown in [58] that the rotation matrix \mathbf{R} that maximizes (5.15) is precisely $\mathbf{R}_{\bar{q}}$. Clearly, before computing the rotation matrix, it is necessary to find the corresponding points between the two sets to be aligned. The algorithm uses (5.11) to determine the corresponding points in the two sets, by means of the Euclidean distance. It is clear that the operation in (5.11) is important for the outcome of the algorithm. In fact, one of the weaknesses of the algorithm is its dependency on the initial approximation of the two sets to be aligned.

The ICP algorithm is known to produce acceptable results as long as a good initial estimate is provided. It is also known to be suited for parallel architecture, to be stable, to be robust even for special cases, and it does not require any local feature extraction. The fact that we need a good estimate to start the alignment of the two sets is only one of the problems with the method, another important problem is the intensive computation

involved in the step $Y \leftarrow C(\kappa'_j, \kappa_i)$. However, we assume that the PAT method can provide a good enough first estimate so that the ICP algorithm can perform in an acceptable time. This was indeed found to be the case in the experiments on which we report below.

The algorithm proceeds to replace each κ_i by

$$\kappa_i = \frac{1}{|C_l|} \sum_{\kappa_j \in C_l} T_{ij} \kappa_j, \quad (5.16)$$

where C_l is the l th class of subunits, $T_{i,j}$ is the transformation that aligns the subunit κ_j to the subunit κ_i .

In practice, the above operation requires of a careful implementation because after applying the transformation $T_{i,j}$ to a subunit κ_j we obtain a set whose points are in \mathbb{R}^3 ; therefore, it is necessary to find the near valid points in the grid, a set that we refer to as κ'_j . Also, after alignment, it is necessary to obtain the density values of the transformed subunit κ'_j in the space occupied by the non-transformed subunit κ_j (the inverse transform $T_{i,j}^{-1}$ that can be easily computed). Again, the inverse transformation of κ'_j produces points in \mathbb{R}^3 . It is common to obtain the density values of the subunit κ'_j by interpolation of the discretized values corresponding to κ_j . However, this operation introduces some extra error. In order to avoid this error we obtain the values from the values of the linear combination of blobs.

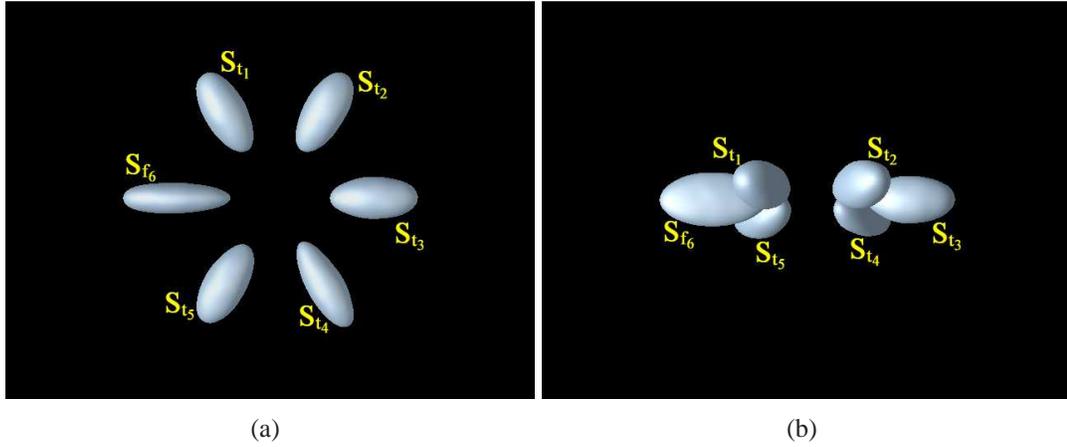


Fig.5.1: Configuration of six subcomponents (ellipsoids) to form a simulated complex macrostructure. The simulated complex macrostructure is formed from two different repeating units: S_f and S_t . We show in (a) a view perpendicular to the plane formed by the six major axes of the subunits and in (b) a view almost parallel to such a plane, respectively.

5.2 Experiments

To test the method described above we decided to use two types of phantoms. The first is a mathematical phantom based on a biological macromolecule. The second uses a model of a macromolecule whose structure is well known.

For the mathematical phantom approach we use phantoms that represent structures formed by putting together six subunits. This type of configuration is based on the torus-like proteins such as many of the DNA helicases, see [63, 107, 124]. In our phantoms, there are two types of repeating units in the shape of ellipsoids that we refer to as S_f and S_t . An ellipsoid is analytically described by three radii that we denote by r_a , r_s and r_c , chosen so that $r_a > r_s > r_c$. The r_a for the ellipsoid S_f is 16% larger than the r_a for S_t

and the other two radii of S_t are decreased in the same proportion as compared to those in S_f in such a way that S_t and S_f end up having the same volume. We construct a phantom by forming a plane containing the major axes of the six ellipsoids. The major axes (with radii r_a) intersect in a common point (the origin of the phantom) with a separation of 60° between consecutive axes. For each of the six axes, the assignment of the subunit S_f and S_t is randomly done. Furthermore, the axis with radius r_s of every ellipsoid can be either parallel or perpendicular to the specimen plane, an orientation that is also randomly assigned. We show an example of such a phantom in Fig. 5.1.

Important issues to decide are the values of the radii r_a , r_s and r_c , and the separation of the ellipsoids from the center of the phantom. Clearly, these will influence the performance of the algorithms. We adjusted the parameters to an appropriate value by running a preliminary experiment such that the current method succeeds in distinguishing the subunits in 50% of the cases.

In the second experiment we used the atomic model of the bacteriorhodopsin. The justification for using this protein is that its structure is well-known and that its atomic structure is available from the protein data bank, see Appendix B.

For both experiments we created projections simulating the random conical tilt scheme for data collection. We used the same uncertainty for the collection angles as we used in Section 4.2. Additionally, we used the same level of noise as used in that section to corrupt the projections. After obtaining the projections, we reconstructed the density distribution

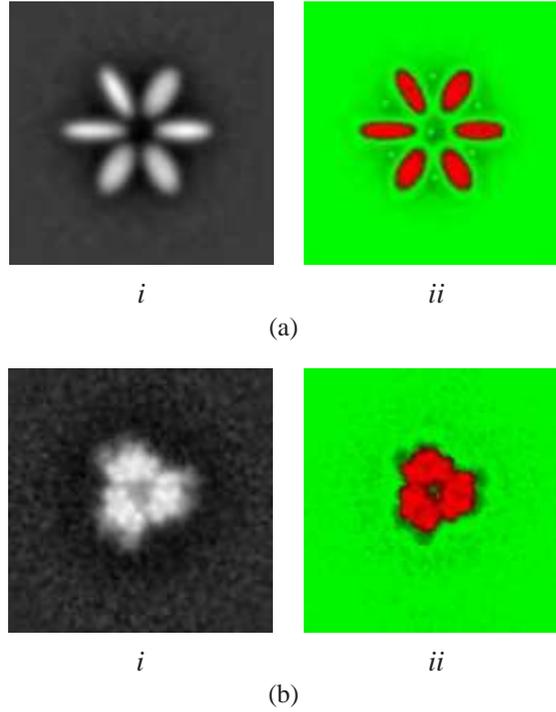


Fig.5.2: Sequences of images showing the central slices for (i) a discretized reconstructed density function and (ii) segmentation using the fuzzy segmentation described in Section 5.1.2 of the reconstructed density function. For the fuzzy segmentation results, the hue indicates the object to which the voxel belongs (i.e., the m such that $\sigma_m^c = \sigma_0^c > 0$), in (ii) the red is for the macromolecule and the green is for the background. The intensity indicates the grade of membership (it is proportional to σ_0^c). We present results for (a) a mathematical phantom and (b) the bacteriorhodopsin.

of the phantoms and the bacteriorhodopsin using ART with the parameters used in Section 3.4. A slice of the *fcc* grid can be visualized as a square array where every other position is assigned with a value depending on the slice (see the definition of the *fcc* grid in (3.5) and Fig. 3.3(c)). Therefore, for display purposes, we assigned values to points in the simple cubic grid (see Fig. 3.3(a)) in the following way. First, the values assigned to points in the *fcc* grid are assigned to the same positions on the simple cubic grid. The other values in

the simple cubic grid are the average of the six neighbors with distance Δ . We show in Fig. 5.2(a) two images showing in (i) a central slice of the reconstructed density function for a mathematical phantom and (ii) a central slice of the segmented density function using the fuzzy segmentation described above. For the display of the reconstructed density function we use a gray scale proportional to the density values, with white representing the highest value and black the lowest. For the fuzzy segmentation results, the hue indicates the object to which the voxel belongs (i.e., the m such that $\sigma_m^{\bar{c}} = \sigma_0^{\bar{c}} > 0$), in (ii) we used red for the macromolecule and green for the background. The intensity indicates the grade of membership (i.e., it is proportional to $\sigma_0^{\bar{c}}$). It can be seen that the fuzzy 2-segmentation is good enough to separate the six subunits in the mathematical phantom and it is not necessary to modify the threshold to preserve the volume, following our criteria presented at the end of Section 5.1.2. However, for the bacteriorhodopsin it is necessary to threshold the $\sigma_1^{\bar{c}}$ in order to achieve the separation into three subunits.

5.3 Results

For the mathematical phantoms we wanted to compare the accuracy of two methods for extracting the subunits from a reconstructed density function: a) simple thresholding and b) the method just described above. For each phantom, we define the inaccuracy to be the number of misclassified voxels of such a phantom in a processed reconstruction and in the

unprocessed phantom. We use the figure of merit (*FOM*) defined as

$$FOM = \frac{\text{No. voxels} - \text{No. misclassified voxels}}{\text{No. voxels}}. \quad (5.17)$$

The level of statistical significance for rejecting the null-hypothesis that both methods produce equally good results in favor of the hypothesis that the one with higher structural accuracy is better is computed as follows. Let β_k and ϵ_k be the inaccuracies of the k th phantom as produced by the two methods, respectively. According to the null-hypothesis, $\beta_k - \epsilon_k$ is a sample of a zero-mean random variable. Then, for a large K

$$\sum_{k=1}^K (\beta_k - \epsilon_k) \quad (5.18)$$

is a sample of a normally distributed zero-mean random variable [82, 83]. We can consider the variance of this variable to be K times that of the zero-mean random variable of which $\beta_k - \epsilon_k$ are samples for $1 \leq k \leq K$. Hence, for large enough K , it is reasonable to assume that the null-hypothesis implies that (5.18) is a sample from normally distributed random variable with mean zero and variance

$$\sum_{k=1}^K (\beta_k - \epsilon_k)^2. \quad (5.19)$$

Therefore, we can use the normal distribution to calculate the significance of the null hy-

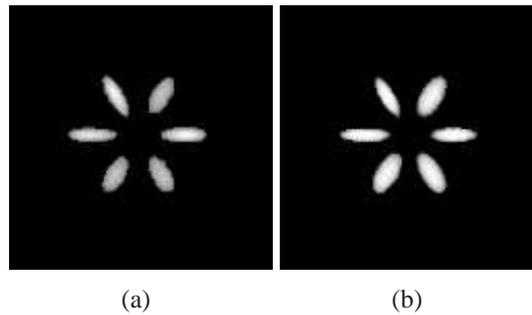


Fig.5.3: (a) Segmentation using simple thresholding based on the volume of the mathematical phantom and (b) result of finding subunits, creating models for them and substituting the models in place of the subunits.

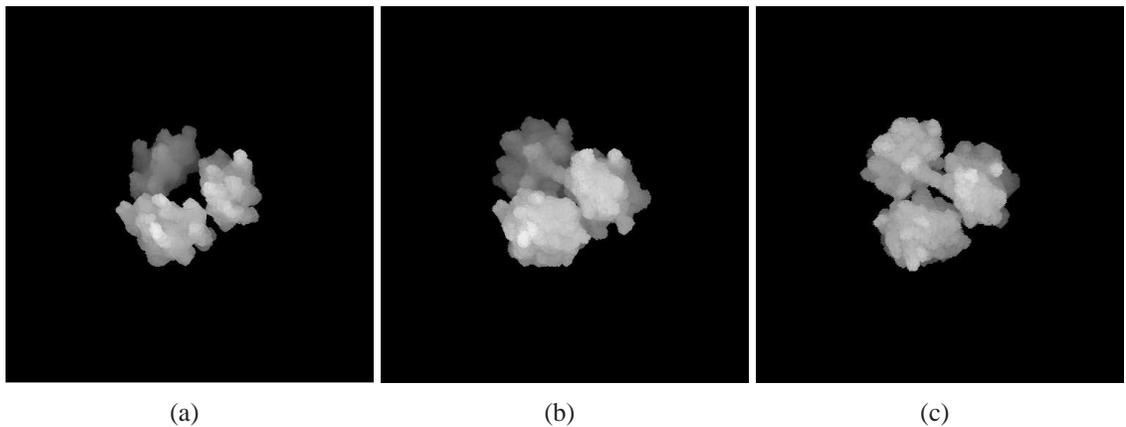
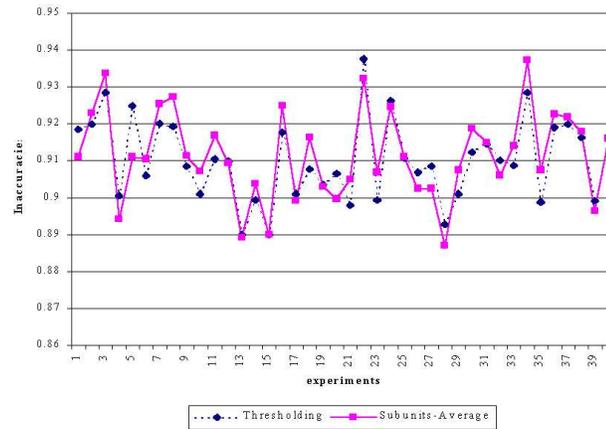


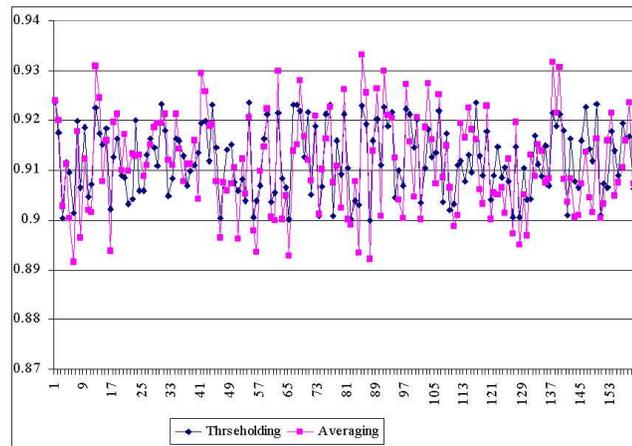
Fig.5.4: We show in (a) the separation of the bacteriorhodopsin's segmentation into three subunits by the subunit-averaging method. We show in (b) and (c) the result of the method with different views.

pothesis [82, 83].

In Fig. 5.3(a) we show the result of thresholding the subunits of the mathematical phantom based on their volume and in Fig. 5.3(b) the result of aligning-averaging them, respectively. We do the same in Fig. 5.4(a) and (b), respectively, for the bacteriorhodopsin, and in Fig. 5.4(c) we show a different orientation of the result in (b).



(a)



(b)

Fig.5.5: Inaccuracies of representations of density functions by (dotted line) thresholding based on the volume of the density function and (continuous line) subunit-model approach. We show in (a) the results after 40 experiments and in (b) after 160 experiments.

In Fig. 5.5 we show the inaccuracies by the simple thresholding method (dotted line) and the subunit-average method (continuous line). We show the results of 40 and 160 experiments in Fig. 5.5(a) and Fig. 5.5(b), respectively. We can see that the behavior of both methods are similar and in fact based on the statistical analysis we can reject the null-hypothesis with a 0.2 and 0.11 confidence, respectively. These results suggest that our method shows little advantage over the thresholding method for this phantom and for this number of experiments. However, we could reject the null hypothesis with more confidence as we carried out more experiments; something that suggests that the aligning-averaging method in fact yields better results than simple thresholding. On the other hand, we hypothesize that the insufficient level of confidence could be the result of, first, the high similarity between the subunits. As we saw, the fuzzy segmentation separated the molecule into the known number of subunits. A second source of difficulty in our algorithm is the alignment process. The alignment process is carried out without any human supervision and it seems necessary to make an evaluation of the reliability of the alignment achieved by PAT in combination with the ICP method for objects that better resemble biological macromolecules. Finally, it would be important to evaluate the similarity of the subunits generated by the fuzzy segmentation as the posterior processes are highly dependent on this first step.

In the case of the bacteriorhodopsin, the algorithm was able to separate segmentation of the bacteriorhodopsin into three subunits, see Fig. 5.4(a). However, the three subunits are below the volume of the “truth” for the bacteriorhodopsin (It can be seen that the α -

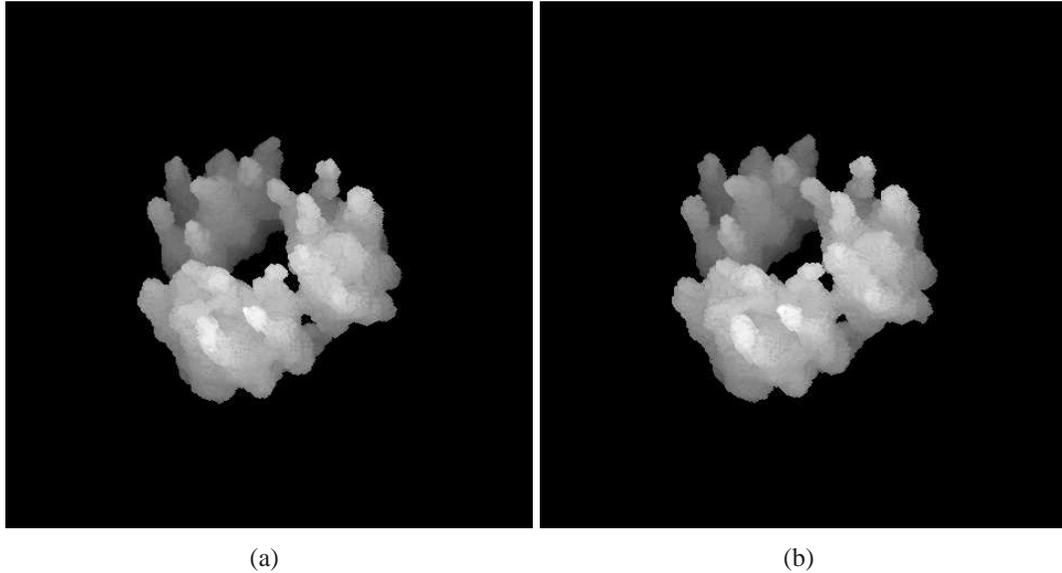


Fig.5.6: (a) Segmentation using simple thresholding based on the volume of the bacteriorhodopsin and (b) result of averaging the three subunits.

helices are greatly deteriorated). We show in Fig. 5.4(b) Fig. 5.4(c) the result of the subunit-averaging algorithm at different orientations; it is clear that the result is incorrect; clearly, the alignment of the subunits is wrong. As we mentioned before, ICP requires of a good estimate to produce acceptable results. From Fig. 5.4(b) and (c) it is presumed that the initial alignment by PAT is not performing well because the initial subunits are basically two ellipsoids, which produces a non-unique set of eigenvectors for the minor axes. In order to see whether averaging in fact improves the extraction of the density function we thresholded the discretized density function with a threshold that preserved the known volume of the protein and then found two rotations along the axis parallel to the α -helices such that it was possible to average the three subunits. We did not find a great

improvement, based on visual inspection, of averaging the three subunits in comparison to non-averaging method. In this case, the resolution of the reconstruction does not allow to separate the individual α -helices and the noise greatly corrupts the reconstruction (this is demonstrated by the difficulty of segmenting the bacteriorhodopsin).

Chapter 6

CONCLUSIONS

In this chapter we present a summary of the motivations and accomplishments of the work presented in the previous pages. After this presentation, we offer a discussion of issues that could be addressed in the future.

6.1 Summary and Contributions

Advances in technology and in research have resulted in the collection of information regarding microscopic objects at high resolution by several technologies (e.g., NMR, crystallography and electron microscopy). In particular, it has become possible to obtain information regarding proteins. The study of the three-dimensional structure of proteins is important, as it is known that several of their functions are determined by shape (e.g., the docking of antibodies to pathogens).

With the development of new computer algorithms, or their improvement, and the advance of computer technologies it is now possible to obtain reconstructions of the three-

dimensional structure of macromolecular objects. Series expansion algorithms for reconstruction can incorporate information from other sources; moreover, they have been shown to be superior to transform methods for specific tasks. Such methods approximate a density function by a linear combination of basis functions. In such a representation, the centers of the basis functions are located on a grid. These algorithms produce as output the coefficients that multiply the basis functions.

The performance of a series expansion method is highly dependent on the selection of the basis functions and on the selection of the grid. Important work has been devoted to study “good” basis functions and grids for these algorithms. In particular, generalized Kaiser-Bessel functions (also known as *blobs*) in conjunction with the body-centered cubic (*bcc*) grid have been used with success in areas such as positron emission tomography and electron microscopy.

Blobs are radially symmetric functions with smooth decay from one to zero. The shape of a blob is determined by three parameters: α which controls the width of the bell-shaped peak, a which determines the support of the function and m which determines the continuity of the function.

The availability of three-dimensional information from proteins has contributed to the development of methods to analyze such information. Among those methods, visualization tools play an important role as much of our knowledge regarding shapes comes through the visual system.

One advantage of series expansion methods is that their output can be used, with little modification, for visualization of the implicit surface enclosing the molecule. Specifically, it is almost straightforward to use the output of the reconstruction method in the so-called *raycasting* algorithm (a method that is also known as the *blobby* scheme). While there had been previous research on the selection of good parameters for reconstructing with blobs on the *bcc* grid, little was known about the behavior of such functions when used for visualization.

The smooth properties of blobs suggest that visualization of density functions reconstructed with blobs would be inherently smooth. However, we found that not all the blobs that are good for reconstruction are good for visualization. In this work, we have proposed a method for selecting blob parameters that takes into account the previous research and adds a new criterion. We propose that for producing smooth surfaces two blobs located at two closest neighbors in the *bcc* grid should produce an implicit surface that is convex, but not too convex. While this criterion indeed results in smooth surfaces it utilizes wider blobs than those indicated by the single criterion of generating “good” reconstructions. A side effect of this method is the greater demand on the computer for generating a reconstruction. Because the constraint that ensures smooth surfaces is more demanding than the one that ensures “good” reconstructions, it seems that selecting the blob parameters for the latter would suffice to generate a “good” reconstruction. However, by selecting wider blobs some features in the reconstruction might be smoothed out. Finding blob parameters

that are a compromise between “good” reconstructions and smooth surface representations at practical times is an issue that requires further research. The loss of features due to selecting wider blobs is also an issue that should be further analyzed with mathematical phantoms appropriately designed to represent features similar to those appearing in real proteins.

In the course of our research we also found that a direct application of raycasting with blobs can result in an unpractical algorithm for visualization, especially considering the computer time in comparison with polygon-projection algorithms (algorithms that approximate the implicit surface by polygons that are projected onto the screen and that use as input a discretized version of the density function, an operation known as voxelization). We have developed a new raycasting-blobs method that preprocesses the coefficients and improves by more than an order of magnitude the times required for producing computer representations without reducing the quality of the renderings. The new raycasting-blobs algorithm produces images of superior quality to those produced by the polygon-projection methods embedded in OpenDXTM (a general purpose open-source program for data manipulation). The time required to generate the approximation to the implicit surface by the same OpenDX polygon-projection algorithm is in the range required by our new raycasting-blobs algorithm for the same rendering quality. However, once the approximation to the implicit surface is generated, rendering by the polygon-projection method, is instantaneous; something that is not possible with our raycasting-blobs method.

Since immediate user interaction is important for the analysis of three-dimensional structures, we developed a polygon-projection method that takes advantage of efficient voxelization of the density function. A common voxelization method is to discretize the density function using a simple cubic grid. It has been shown that there are more efficient grids than the cubic grid, in particular the *bcc* and the *fcc* grids. The *fcc* grid has desirable properties, for example, voxels in the *fcc* grid have twelve neighbors all at the same distance, while the cubic voxels have different types of neighbors at various distances. Furthermore, the *fcc* grid is a subset of a simple cubic grid (with the same sampling distance). This property allows the adaptation of algorithms developed for the simple cubic grid to the *fcc* grid (with minor modifications). An important example of this is the so called-boundary tracking algorithm. This algorithm searches for the voxel faces that enclose a connected set of voxels (polygons that enclose a voxelized-volume). Creating the set of connected voxels that represent the density function is achieved through segmentation. We have developed such an algorithm for connected *fcc*-voxels. We have shown that, for representing the same density function, the *fcc* grid requires fewer voxels than the simple cubic grid for the same rendering quality. This method does not take too much time to produce the approximation to the implicit surface and it allows rendering in real-time. The normals of the surface greatly influence the quality of the final image. As we mentioned in Section 3.3 we have the exact gradient to the density function at any point. Therefore, this technique offers the right platform to incorporate the exact normals in the process of producing better quality

images.

An important characteristic of biological macromolecules is that they are often formed from few subunits that in many occasions show some kind of symmetry. Several methods had been developed in electron microscopy to take advantage of the repetition of the subunits in the process of data acquisition. There had also been methods that take advantage of the subunit repetition in the reconstruction process. Here we have proposed a method that searches for repeated subunits in the reconstructed density functions.

This method is based on previous research suggesting that the quality of digitized documents can be improved by finding characters (subunits) in the image and forming models of the subunits. Later on, the models are substituted in the place of the characters. Clearly, the repeated characters differ by translation only.

In the case of repeated subunits in biological macromolecules the subunits differ in translation and rotation. This introduces an extra level of complexity. We have implemented an algorithm that finds subunits, creates models and substitutes them to improve the quality of a representation. We implemented a three-dimensional version of a fuzzy segmentation algorithm (an algorithm that uses fuzzy set principles) to segment the reconstructed density function into the desired number of subunits. We threshold the result of this segmentation to obtain masks that we use to identify the translation and rotation required to align all the subunits. We implemented two popular alignment methods to obtain the translation and rotation necessary to align the subunits.

For testing our method we designed mathematical phantoms. Using 160 such phantoms we were not able to demonstrate with statistical significance that our approach produces better approximations to the macromolecule than simple thresholding. We hypothesize that this situation could be due, first, to the high similarity between the subunits, second, to the relatively small number of subunits in the molecules of the model (as compared to the number of subunits in a digitized document or the very large number of molecules and images used for averaging when processing data from an electron microscope sample) and, finally, to the limitations of the implemented aligning methods that we used after the segmentation.

6.2 Future Work

There are three important issues that are of immediate interest for future research.

First, further investigation should be carried out regarding the optimization of blobs for reconstruction and visualization in order to make more consistent the selection of parameters for both operations. The criterion for approximating the normals of a convex function is more restrictive than that of approximating a constant function. Thus, it seems sufficient to select the blob parameters based exclusively on the smaller normal rms errors. In this respect, most of the smaller normal rms errors are outside the hyperbolas; consequently, it appears that further research is necessary to select blob parameters that allow create reconstructions in practical times and that result in smooth surfaces.

The second issue for further research concerns the development of a visualization technique that takes advantage of the smoothness of the blobs, producing high quality renderings, and allows real-time interaction. A possible approach is to continue improving the raycasting-blobs method. Another is to approximate the implicit surfaces by polygons but utilizing the analytical normals available from the blob reconstruction.

In the raycasting-blobs method it seems possible to optimize the management of memory in such a way that the several lists that are created make less calls to the system. Furthermore, our implementations have used the support of the blobs to create the z-buffer and the lists of blobs intersecting a ray. Finally, raycasting is an algorithm that is suitable for parallelization. While multi-processors are still uncommon (although more accessible than before), graphic multi-processor cards are becoming cheaper and they could be used to as an alternative for parallelization or some interpolations necessary in the raycasting-blobs method. The operations necessary for every ray casted are independent and they are rather simple: compute the value of the blob and its gradient, add the values for several blobs and search for the position at which the surface is found, for the desired threshold. The smooth transition from one to zero and the continuity in the first derivative allow to create look up tables for different values of the blob and its gradient (something that is currently done). Currently, the final values for a blob and its gradient are computed by interpolation of the values stored in the tables. Furthermore, the addition of all the values is an operation that is constantly performed. However, these operations can be easily performed by the

interpolation, texture mapping and α -buffer features in a graphic card.

The third subject of interest is the development of a method that better identifies and classifies subunits in a reconstructed density function. In this respect, it seems that the algorithm described in Chapter 5 can be improved to better classify the objects into classes by using a statistical analysis system instead of fixed threshold value for the Hausdorff distance. Based on our results, see Fig. 5.4, it seems that alignment methods that we implemented should be further investigated.

Finally, an idea to explore is the modification of both segmentation and subunit-model algorithms to work with the *bcc* grid. By doing this, it would be possible to work with the set of coefficients and create a final display that possesses the quality shown by raycasting-blobs.

Appendix A

Fourier Transforms of Trains of Pulses on Various Grids

The concept of pointwise multiplication between two ordinary functions can be extended to multiplication between a function $f \in \mathcal{S}(\mathbb{R}^n)$ and a distribution $T \in \mathcal{S}'(\mathbb{R}^n)$, see Section 2.4 for this notation, which is defined in [55] as the distribution $f \times T$ such that

$$(f \times T)g = T(f \times g), \quad (\text{A.1})$$

for any function $g \in \mathcal{S}(\mathbb{R}^n)$. For a function ϕ over \mathbb{R}^n , we get (recalling (2.3)) that

$$(f \times T_\phi)g = T_\phi(f \times g) = \int_{\mathbb{R}^n} \phi(\bar{x}) (f(\bar{x})g(\bar{x})) d\bar{x} = \int_{\mathbb{R}^n} (\phi(\bar{x})f(\bar{x}))g(\bar{x}) d\bar{x} = T_{\phi \times f}g. \quad (\text{A.2})$$

Thus multiplying a function f with a distribution T_ϕ determined by a function ϕ gives us a distribution determined by the multiplication of f by ϕ , demonstrating the reasonableness of the claim that (A.1) is a generalization of multiplication between ordinary functions.

For $T = \delta_{\bar{y}}$ we have that

$$(f \times \delta_{\bar{y}}) g = \delta_{\bar{y}}(f \times g) = f(\bar{y}) g(\bar{y}) = (f(\bar{y}) \delta_{\bar{y}}) g, \quad (\text{A.3})$$

which in words says that multiplying a function by a unit impulse at \bar{y} gives us an impulse at \bar{y} whose weight is $f(\bar{y})$. Now consider, III_G , see (2.20), where G is a grid in \mathbb{R}^3 .

Then for any $f \in \mathcal{S}(\mathbb{R}^n)$, we easily derive from (A.1), that

$$f \times \text{III}_G = \sum_{\bar{y} \in G} f(\bar{y}) \delta_{\bar{y}}. \quad (\text{A.4})$$

Thus multiplying f by III_G can be considered to be equivalent to *sampling* f at the points of the set G .

As stated in Section 3.2, we work with functions sampled on the simple cubic (*sc*), the body-centered cubic (*bcc*) and the face-centered cubic (*fcc*) grids. To simplify the notion in their analysis we use, for any positive real number Δ , Δshah as an alternative notation for the distribution III_{G_Δ} . Recalling (3.3), (3.4) and (3.5), and using the notation $\bar{b} = (\Delta, \Delta, \Delta)$, $\bar{e}_1 = (\Delta, \Delta, 0)$, $\bar{e}_2 = (0, \Delta, \Delta)$ and $\bar{e}_3 = (\Delta, 0, \Delta)$, we see that the *trains of pulses* on the grids G_Δ , B_Δ and F_Δ can be expressed by

$$\text{III}_{G_\Delta} = \Delta\text{shah}, \quad (\text{A.5})$$

$$\text{III}_{B_\Delta} = {}^{2\Delta}\text{shah} + {}^{2\Delta}\text{shah}_{\bar{y}} \quad (\text{A.6})$$

and

$$\text{III}_{F_\Delta} = {}^{2\Delta}\text{shah} + {}^{2\Delta}\text{shah}_{\bar{e}_1} + {}^{2\Delta}\text{shah}_{\bar{e}_2} + {}^{2\Delta}\text{shah}_{\bar{e}_3}. \quad (\text{A.7})$$

By what was said in the previous paragraph it follows that sampling a function f on one of these grids is equivalent to multiplying f by the corresponding train of pulses.

From [87], page 4, we have that

$$\widehat{{}^\Delta\text{shah}} = \left(\frac{\sqrt{2\pi}}{\Delta} \right)^3 \frac{2\pi}{\Delta} \text{shah}. \quad (\text{A.8})$$

From this and (2.14)

$$\widehat{{}^\Delta\text{shah}_{\bar{y}}} = S^{\bar{y}} \times \left(\frac{\sqrt{2\pi}}{\Delta} \right)^3 \frac{2\pi}{\Delta} \text{shah}. \quad (\text{A.9})$$

From (A.6), (A.8), (A.9), (A.4), the Fourier transform of the train of pulses arranged in the bcc can be obtained as follows

$$\begin{aligned}
\widehat{\text{III}}_{B_\Delta} &= \widehat{2\Delta\text{shah}} + \widehat{2\Delta\text{shah}_{\bar{b}}} \\
&= \left(\frac{\sqrt{\pi}}{\sqrt{2\Delta}}\right)^3 \frac{\pi}{\Delta} \text{shah} + S^{\bar{b}} \times \left(\frac{\sqrt{\pi}}{\sqrt{2\Delta}}\right)^3 \frac{\pi}{\Delta} \text{shah} \\
&= \left(\frac{\sqrt{\pi}}{\sqrt{2\Delta}}\right)^3 \sum_{\bar{y} \in G_{\frac{\pi}{\Delta}}} \delta_{\bar{y}} + \left(\frac{\sqrt{\pi}}{\sqrt{2\Delta}}\right)^3 \sum_{\bar{y} \in G_{\frac{\pi}{\Delta}}} \delta_{\bar{y}} S^{\bar{b}}(\bar{y}) \\
&= \left(\frac{\sqrt{\pi}}{\sqrt{2\Delta}}\right)^3 \sum_{\bar{y} \in G_{\frac{\pi}{\Delta}}} \delta_{\bar{y}} \left(1 + S^{\bar{b}}(\bar{y})\right) \\
&= \left(\frac{\sqrt{\pi}}{\sqrt{2\Delta}}\right)^3 \sum_{\bar{y} \in G_{\frac{\pi}{\Delta}}} \delta_{\bar{y}} \left(1 + e^{-i\langle \bar{b}, \bar{y} \rangle}\right) \\
&= \left(\frac{\sqrt{\pi}}{\sqrt{2\Delta}}\right)^3 \sum_{\bar{y} \in G_{\frac{\pi}{\Delta}}} \delta_{\bar{y}} (1 + \cos(\Delta(y_1 + y_2 + y_3)) - i \sin(\Delta(y_1 + y_2 + y_3))),
\end{aligned} \tag{A.10}$$

where y_i is of the form $k_i \frac{\pi}{\Delta}$, see (3.3), which results in $\Delta(y_1 + y_2 + y_3) = \pi(k_1 + k_2 + k_3)$ and $\sin(\Delta y_1 + \Delta y_2 + \Delta y_3) = 0$. In a similar way, $1 + \cos(\Delta y_1 + \Delta y_2 + \Delta y_3)$ is equal to zero when $k_1 + k_2 + k_3$ is odd and is equal to two when $k_1 + k_2 + k_3$ is even. This is in fact the definition of the *fcc* grid, see (3.5). Therefore, it follows that (A.10) is equivalent to

$$\widehat{\text{III}}_{B_\Delta} = \frac{1}{\sqrt{2}} \left(\frac{\sqrt{\pi}}{\Delta}\right)^3 \text{III}_{F_{\frac{\pi}{\Delta}}}. \tag{A.11}$$

This indicates that a train of pulses on the *bcc* grid with a sampling distance Δ has a train of pulses on the *fcc* grid as its Fourier transform with sampling distance $\frac{\pi}{\Delta}$ and amplitude $\frac{1}{\sqrt{2}} \left(\frac{\sqrt{\pi}}{\Delta}\right)^3$.

Appendix B

Bacteriorhodopsin

The bacteriorhodopsin (*bR*) is a robust protein from the cell membrane of the *Halobacterium halobium*. The *Halobacterium halobium* is an extreme halophile. Extreme halophiles are prokaryotes that require salt for growth and will not grow at low salt concentrations. Their cell walls, ribosomes and enzymes are stabilized by sodium. The extreme halophiles live in natural environments where the salt concentration is very high, such as the Dead Sea, the Great Salt Lake or evaporating ponds of sea water. The *Halobacterium halobium* is the prevalent species in the Great Salt Lake and adapts to the high-salt environment by the development of a *purple membrane* (patches of light-harvesting pigments in the cell membrane). This is the only known example in nature of non-photosynthetic photophosphorylation [66].

The bR pumps protons out of the cell and provides the cell with energy to survive (in environments with high salinity the oxygen is scarce and the aerobic respiration does not provide enough energy for the cell's needs). This proton pump uses sunlight as its source

of energy.

The bR is an important protein because it serves as a simple model for membrane transporters (biological macromolecules performing the fundamental process, for all forms of life, of ion transportation against an electrochemical potential, up to 10,000-fold difference) and for G-protein-coupled receptors (which include drug targets in humans and which could operate by a similar switch mechanism).

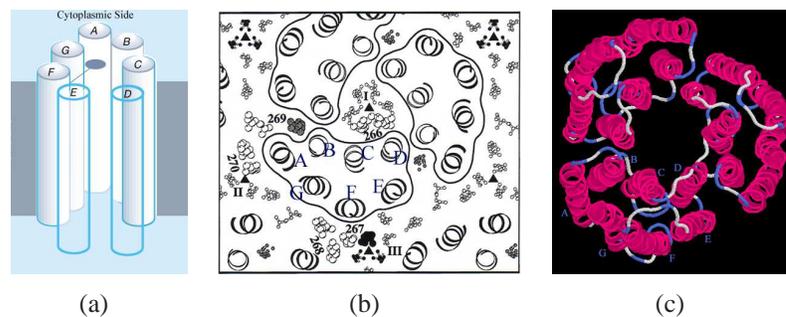


Fig.B.1: (a) Scheme of the seven α -helices forming the bacteriorhodopsin, see text, (reprinted with permission from the author and Nature [66] copyright 2000 Macmillan Publishers Ltd.) (b) View of the arrangement of three Bacteriorhodopsins as they would appear from within the cytoplasm. (c) View of the three bacteriorhodopsins, represented by ribbons, using the model of atoms available in the Protein Data Bank [10].

There are three bacteriorhodopsin in the cell membrane, see Fig. B.1, each of them composed of seven membrane-spanning α -helices, see Fig. B.1(a), perpendicular to the cell's membrane and containing one molecule of a linear pigment called retinal. The seven α -helices are tilted with respect to the plane formed by the membrane and are joined by chains of amino acids on either side of the cell membrane, see Fig. B.1(c).

The bR operates by involving the retinal pigment (which changes its structure in re-

sponse to visible light) in a sequence of five stages in which the protein slightly changes its three-dimensional structure. The primary motions of the proton-pumping action in bR are surprisingly small, involving movements of groups of atoms by 1 Å or less [42, 46, 47, 48, 66, 67].

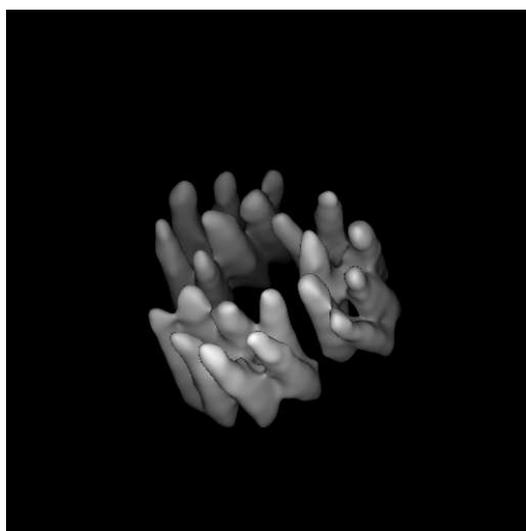


Fig.B.2: Rendering of the bacteriorhodopsin at the desired threshold from perfect projections at 9 Å resolution. This display has the same orientation and volume as that shown in Fig. 4.3(b). This image will serve as the representation of *truth*.

The bR is one of the few proteins that has been crystallized, a fact that has permitted the extensive study of the protein in the last three decades. The structure of the bR has been determined, with high accuracy, at any of its conformational stages to convert light into energy in the bacteria by electron cryomicroscopy of two-dimensional crystals and by X-ray crystallography. The research of the bR has yielded several atomic models of the protein (e.g., 1brd, 2brd), which are available in the Protein Data Bank (*PDB*) [10] at the

World Wide Web address www.rcsb.org/pdb.

For our experiments we created a model at a resolution typical of that achieved by TEM (9 Å), see Fig. B.2. The density function based on the atomic model creates a model whose volume is of importance at almost every value. Consequently, we arbitrarily selected a threshold at which interesting features are visible, in particular at a value at which the seven α -helices are visible.

BIBLIOGRAPHY

- [1] G. D. Abram and L. Treinish, "An extended data flow architecture for data analysis and visualization," in *Proceedings of the IEEE Computer Society Visualization '95*, pp. 263–269, IEEE Computer Society, 1995.
- [2] N. M. Alpert, J. F. Bradshaw, D. Kennedy, and J. A. Correia, "The principal axes transformation – a method for image registration," *Journal of Nuclear Medicine*, vol. 31, pp. 1717–1722, 1990.
- [3] L. A. Amos, R. Henderson, and P. N. T. Unwin, "Three-dimensional structure determination by electron microscopy of two-dimensional crystals," *Progress in Biophysical and Molecular Biology*, vol. 39, pp. 183–231, 1982.
- [4] L. K. Arata, A. P. Dhawan, J. P. Broderick, M. F. Gaskil-Shipley, A. V. Levy, and N. D. Volkow, "Three-dimensional anatomical model-based segmentation of MR brain images through principal axes registration," *IEEE Transactions on Biomedical Engineering*, vol. 42, pp. 1069–1078, 1995.
- [5] E. Artzy, G. Frieder, and G. T. Herman, "The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm," *Computer Graphics and Image Processing*, vol. 15, pp. 1–24, 1981.
- [6] T. S. Baker and J. E. Johnson, "Low resolution meets high: Towards a resolution continuum from cells to atoms," *Current Opinion in Structural Biology*, vol. 6, pp. 585–594, 1996.
- [7] H. Bandemer and S. Gottwald, *Fuzzy Sets, Fuzzy Logic, Fuzzy Methods with Applications*. Chichester: Wiley, 1995.
- [8] M. Bárcena, T. Ruiz, L. E. Donate, S. E. Brown, N. E. Dixon, M. Radermacher, and J. M. Carazo, "The DnaB·DnaC complex: A structure based on dimers assembled around an occluded channel," *European Molecular Biology Organization Journal*, vol. 20, pp. 1462–1468, 2001.

- [9] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*. New York: Wiley, 2nd ed., 2001.
- [10] F. C. Bernstein, T. F. Koetzle, G. J. Williams, E. Meyer, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi, "The Protein Data Bank. A computer-based archival file for macromolecular structures," *Journal of Molecular Biology*, vol. 112, pp. 535–542, 1977.
- [11] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–255, 1992.
- [12] J. C. Bezdek and S. K. Pal, *Fuzzy Models for Pattern Recognition: Background, Significance, and Key Points*, ch. 1, pp. 1–34. New York: IEEE, Inc., 1992.
- [13] G. Blais and M. D. Levine, "Registering multiview range data to create 3D computer objects," *IEEE Transactions of Pattern Analysis and Machine Intelligence*, vol. 17, pp. 820–824, 1995.
- [14] J. F. Blinn, "A generalization of algebraic surface drawing," *ACM Transactions on Graphics*, vol. 1, pp. 235–256, 1982.
- [15] I. Bloch, "Fuzzy techniques in medical image processing: On fuzzy distances and their use in image processing under imprecision," in *Third IEEE EMBS International Summer School on Biomedical Imaging, Ile de Berder, France, June 13–21, 1998*, vol. 1, IEEE, 1998.
- [16] R. N. Bracewell, *The Fourier Transform and its Applications*. New York: McGraw-Hill Book Company, 3rd ed., 1999.
- [17] E. O. Brigham, *The Fast Fourier Transform and Its Applications*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1988.
- [18] L. G. Brown, "A survey of image registration techniques," *ACM Computing Surveys*, vol. 24, pp. 325–376, 1992.
- [19] M. D. Brunch, *NMR Spectroscopy Techniques*. New York: M. Dekker, 1996.
- [20] B. M. Carvalho, E. Garduño, and G. T. Herman, "Multiseeded fuzzy segmentation on the face centered cubic grid," in *Proceedings Second International Conference on Advances in Pattern Recognition ICAPR 2001* (S.-V. Ltd., ed.), ICAPR, Pattern Analysis and Applications Journal, 2002.
- [21] B. M. Carvalho, C. J. Gau, G. T. Herman, and T. Y. Kong, "Algorithms for fuzzy segmentation," *Pattern Analysis and Applications*, vol. 2, pp. 73–81, 1999.

- [22] I. Catalaa, R. I. Grossman, D. L. Kolson, J. K. Udupa, L. G. Nyul, L. G. Wei, X. Zhang, M. Polanski, L. J. Mannon, and J. C. McGowan, "Multiple sclerosis: Magnetization transfer histogram analysis of segmented normal-appearing white matter," *Radiology*, vol. 216, pp. 351–355, 2000.
- [23] D. C. Champeney, *Fourier Transforms and Their Physical Applications*. New York: Academic Press, 1973.
- [24] W. Chiu, "What does electron cryomicroscopy provide that X-ray crystallography and NMR spectroscopy cannot?," *Annual Review of Biophysics and Biomolecular Structure*, vol. 22, pp. 233–255, 1993.
- [25] A. P. Dhawan, L. K. Arata, A. V. Levy, and J. Mantil, "Iterative principal axes registration method for analysis of MR-PET brain images," *IEEE Transactions on Biomedical Engineering*, vol. 11, pp. 1079–1087, 1995.
- [26] J. Dubochet, M. Adrian, J.-J. Chang, J.-C. Homo, J. Lepault, A. W. McDowell, and P. Schultz, "Cryo-electron microscopy of vitrified specimens," *Quaternary Review of Biophysics*, vol. 21, pp. 129–228, 1988.
- [27] J. Dubochet, J. Lepault, R. Freeman, J. A. Berriman, and J.-C. Homo, "Electron microscopy of frozen water and aqueous solutions," *Journal of Microscopy (Oxford)*, vol. 128, pp. 219–237, 1982.
- [28] T. L. Faber and E. M. Stokely, "Orientation of 3-D structures in medical images," *IEEE Transactions on Pattern Recognition*, vol. 10, pp. 626–636, 1988.
- [29] J. D. Foley, A. V. Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice*. New York: Addison-Wesley Co., 2nd ed., 1996.
- [30] J. Frank, "SPIDER – a modular software system for electron image processing," *Ultramicroscopy*, vol. 6, pp. 343–358, 1981.
- [31] J. Frank, *Three-Dimensional Electron Microscopy of Macromolecular Assemblies*. New York: Academic Press, 1995.
- [32] J. Frank, M. Radermacher, P. Penczek, J. Zhu, M. Ladjadj, and A. Leith, "SPIDER and WEB: Processing and visualization of images in 3D electron microscopy and related fields," *Journal of Structural Biology*, vol. 116, pp. 190–199, 1996.
- [33] G. Frieder, G. T. Herman, C. Meyer, and J. K. Udupa, "Large software problems for small computers: An example from medical imaging," *IEEE Software*, vol. 2, pp. 37–47, 1985.

- [34] J. C. Fulton, R. I. Grossman, J. K. Udupa, L. J. Mannon, M. Grossman, L. G. Wei, M. Polansky, and D. L. Kolson, "MR lesion load and cognitive function in patients with relapsing-remitting multiple sclerosis," *American Journal of Neuroradiology*, vol. 20, pp. 1951–1955, 1999.
- [35] E. Galin and S. Akkouche, "Incremental polygonization of implicit surfaces," *Graphical Models*, vol. 69, pp. 19–39, 2000.
- [36] E. Garduño, G. T. Herman, and H. Katz, "Boundary tracking in 3D binary images to produce rhombic faces for a dodecahedral model," *IEEE Transactions on Medical Imaging*, vol. 17, pp. 1097–1100, 1998.
- [37] H. Goldstein, C. P. Poole Jr., and J. L. Safko, *Classical Mechanics*. New York: Addison-Wesley Co., 3rd ed., 2001.
- [38] D. S. Goodsell and A. J. Olson, "Structural symmetry and protein function," *Annual Review of Biophysics and Biomolecular Structure*, vol. 29, pp. 105–153, 2000.
- [39] D. Gordon and J. K. Udupa, "Fast surface tracking in three-dimensional binary images," *Computer Vision, Graphics, and Image Processing*, vol. 45, pp. 196–214, 1989.
- [40] R. Gordon, R. Bender, and G. T. Herman, "Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography," *Journal of Theoretical Biology*, vol. 29, pp. 471–481, 1970.
- [41] G. J. Grevera, J. K. Udupa, and D. Odhner, "An order of magnitude faster isosurface rendering in software on a PC than using dedicated, general purpose rendering hardware," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, pp. 335–345, 2000.
- [42] N. Grigorieff, T. A. Ceska, K. H. Downing, J. M. Baldwin, and R. Henderson, "Electron-crystallographic refinement of the structure of bacteriorhodopsin," *Journal of Molecular Biology*, vol. 259, pp. 393–421, 1996.
- [43] A. D. Gross, "Digital geometric methods in image analysis and compression," in *Lecture Notes in Computer Science, Computer Vision ACCV'98*, vol. 1351, ACCV, 1998.
- [44] A. D. Gross and R. Lusinyants, "Fast nonparametric detection of regular structure using digital geometric primitives," in *Proceedings of SPIE, Vision Geometry VI*, vol. 3168, pp. 338–347, 1997.
- [45] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Computer Vision, Graphics, and Image Processing*, vol. 29, pp. 100–132, 1985.

- [46] R. Henderson, J. M. Baldwin, T. A. Ceska, F. Zemlin, E. Beckmann, and H. K. Downing, "Model for the structure of bacteriorhodopsin based on high-resolution electron cryo-microscopy," *Journal of Molecular Biology*, vol. 213, pp. 899–929, 1990.
- [47] R. Henderson, J. M. Baldwin, J. Downing, J. Lepault, and F. Zemlin, "Structure of purple membrane from *Halobacterium halobium*: Recording, measurement and evaluation of electron micrographs at 3.5 Å resolution," *Ultramicroscopy*, vol. 19, pp. 146–178, 1986.
- [48] R. Henderson and P. Unwin, "Three-dimensional model of purple membrane obtained by electron microscopy," *Nature*, vol. 257, pp. 28–32, 1975.
- [49] G. T. Herman, *Image Reconstruction from Projections: The Fundamentals of Computerized Tomography*. New York: Academic Press, 1980.
- [50] G. T. Herman, "Algebraic reconstruction techniques in medical imaging," in *Medical Imaging, Systems Techniques and Applications - Computational Techniques* (C. T. Leondes, ed.), pp. 1–42, Amsterdam: Gordon and Breach Science Publishers, 1997.
- [51] G. T. Herman, *Geometry of Digital Spaces*. Boston: Birkhäuser, 1998.
- [52] G. T. Herman and B. M. Carvalho, "Multiseeded segmentation using fuzzy connectiveness," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 460–474, 2001.
- [53] G. T. Herman, R. Marabini, J. M. Carazo, E. Garduño, R. M. Lewitt, and S. Matej, "Image processing approaches to biological three-dimensional electron microscopy," *International Journal of Imaging Systems & Technology*, vol. 11, pp. 12–29, 2000.
- [54] G. T. Herman and L. B. Meyer, "Algebraic reconstruction techniques can be made computationally efficient," *IEEE Transactions on Medical Imaging*, vol. 12, pp. 600–609, 1993.
- [55] G. T. Herman and H. K. Tuy, "Background from mathematical analysis," in *Basic Methods of Tomography and Inverse Problems* (P. C. Sabatier, ed.), pp. 21–68, Bristol: IOP Publishing, 1987.
- [56] G. T. Herman and D. Webster, "A topological proof of a surface tracking algorithm," *Computer Vision, Graphics, and Image Processing*, vol. 5, pp. 162–177, 1983.

- [57] W. Hoppe, J. Gaßmann, N. Hunsmann, H. J. Schramm, and M. Sturm, “Three-dimensional reconstruction of individual negatively stained yeast fatty acid synthetase molecules from tilt-series in the electron microscope,” *Hoppe-Seylers Zeitschrift für Physiologische Chemie*, vol. 355, pp. 1483–1487, 1974.
- [58] B. K. P. Horn, “Closed form solution of absolute orientation using unit quaternions,” *Journal of the Optical Society of America A-Optics & Image Science*, vol. 4, pp. 629–642, 1987.
- [59] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, “Comparing images using the Hausdorff distance,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 850–863, 1993.
- [60] T. N. Jones and D. N. Metaxas, “Image segmentation based on the integration of pixel affinity and deformable models,” in *Proceedings of the IEEE Computer Vision and Pattern Recognition '98*, pp. 330–337, IEEE Computer Society, 1998.
- [61] S. Kaczmarz, “Angenäherte Auflösung von Systemen Linearer Gleichungen,” *Bulletin de l'Académie Polonaise des Sciences, Lettres, Series A*, vol. 35, pp. 355–357, 1937.
- [62] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*. New York: IEEE Press, 1988.
- [63] G. Karp, *Cell and Molecular Biology: Concepts and Experiments*. New York: John Wiley and Sons, Inc., 1996.
- [64] J. Kendrew, “Myoglobin and the structure of proteins,” *Science*, vol. 139, pp. 1259–1266, 1966.
- [65] P. E. Kinahan, S. Matej, J. S. Karp, G. T. Herman, and R. M. Lewitt, “A comparison of transform and iterative techniques for a volume-imaging PET scanner with a large acceptance angle,” *IEEE Transactions on Nuclear Science*, vol. 42, pp. 2281–2287, 1995.
- [66] W. Kühlbrandt, “Bacteriorhodopsin – the movie,” *Nature*, vol. 406, pp. 569–570, 2000.
- [67] W. Kühlbrandt, D. N. Wang, and Y. Fujiyoshi, “Atomic model of plant light-harvesting complex by electron crystallography,” *Nature*, vol. 367, pp. 6144–6621, 1994.
- [68] K. Kyte, *Structure in Protein Chemistry*. New York: Garland Publishers, 1995.

- [69] R. M. Lewitt, "Multidimensional digital image representations using generalized Kaiser–Bessel window functions," *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, vol. 7, pp. 1834–1846, 1990.
- [70] R. M. Lewitt, "Alternatives to voxels for image representation in iterative reconstruction algorithms," *Physics in Medicine and Biology*, vol. 37, pp. 705–716, 1992.
- [71] G. Lohmann, *Volumetric Image Analysis*. New York: Wiley-Teubner, 1998.
- [72] J. B. A. Maintz and M. A. Viergever, "A survey of medical image registration," *Medical Image Analysis*, vol. 2, pp. 1–37, 1998.
- [73] R. Marabini, G. T. Herman, and J. M. Carazo, "3D reconstruction in electron microscopy using ART with smooth spherically symmetric volume elements (blobs)," *Ultramicroscopy*, vol. 72, pp. 53–65, 1997.
- [74] R. Marabini, G. T. Herman, C. O. Sánchez-Sorzano, and J. M. Carazo, "Volumetric constraints in 3D tomography applied to electron microscopy," in *Proceedings IEEE International Symposium on Biomedical Imaging: Macro to nano* (IEEE, ed.), (Washington D. C.), pp. 641–644, IEEE Engineering in Medicine and Biology Society and IEEE Signal Processing Society, 2002.
- [75] R. Marabini, I. M. Masegosa, M. C. San Martin, S. Marco, J. J. Fernandez, L. G. de la Fraga, C. Vaquerizo, and J. M. Carazo, "Xmipp: An image processing package for electron microscopy," *Journal of Structural Biology*, vol. 116, pp. 237–240, 1996.
- [76] R. Marabini, E. Rietzel, R. Schröder, G. T. Herman, and J. M. Carazo, "Three-dimensional reconstruction from reduced sets of very noisy images acquired following a single-axis tilt schema: Application of a new three-dimensional reconstruction algorithm and objective comparison with weighted backprojection," *Journal of Structural Biology*, vol. 120, pp. 363–371, 1997.
- [77] R. Marabini, M. C. San Martin, and J. M. Carazo, "Electron tomography of biological specimens," in *Contemporary Perspectives in Three-Dimensional Biomedical Imaging* (R. J. L. Coatrieux, ed.), vol. 30 of *Studies in Health Technology and Informatics*, pp. 53–77, IOS Press, 1997.
- [78] S. Matej, G. T. Herman, T. K. Narayan, S. S. Furuie, R. M. Lewitt, and P. E. Kinahan, "Evaluation of task-oriented performance of several fully 3D PET reconstruction algorithms," *Physics in Medicine and Biology*, vol. 39, pp. 355–367, 1994.
- [79] S. Matej and R. M. Lewitt, "Efficient 3D grids for image-reconstruction using spherically-symmetrical volume elements," *IEEE Transactions on Nuclear Science*, vol. 42, pp. 1361–1370, 1995.

- [80] S. Matej and R. M. Lewitt, "Practical considerations for 3-D image reconstruction using spherically symmetric volume elements," *IEEE Transactions on Medical Imaging*, vol. 15, pp. 68–78, 1996.
- [81] H. Miyakawa, "Sampling theorem of stationary stochastic variables in multi-dimensional space," *Journal of the Institute of Electronic and Communication Engineers of Japan*, vol. 42, pp. 421–427, 1959.
- [82] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineering*. New York: John Wiley and Sons, Inc., 1994.
- [83] R. F. Mould, *Introductory Medical Statistics*. Bristol, England: Adam Hilger, 2nd ed., 1989.
- [84] S. Muraki, "Volumetric shape description of range data using "Blobby Model"," *Computer Graphics*, vol. 25, pp. 227–235, 1991.
- [85] S. Muraki, "Multiscale volume representation by a DoG wavelet," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, pp. 109–116, 1995.
- [86] F. Natterer, *The Mathematics of Computerized Tomography*. Chichester, England: John Wiley & Sons, 1986.
- [87] F. Natterer and F. Wübbeling, *Mathematical Methods in Image Reconstruction*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2001.
- [88] E. Nogales, S. G. Wolf, and K. H. Downing, "Structure of the $\alpha\beta$ tubulin dimer by electron crystallography," *Nature*, vol. 391, pp. 199–203, 1998.
- [89] L. G. Nyul and J. K. Udupa, "Fuzzy-connected 3D image segmentation at interactive speeds," in *Proceedings of SPIE: Medical Imaging*, vol. 3979, (San Diego, CA), pp. 212–223, International Society of Optical Engineering, 2000.
- [90] L. G. Nyul and J. K. Udupa, "MR image analysis in multiple sclerosis," *Neuroimaging Clinics of North America*, vol. 10, pp. 799–, 2000.
- [91] H. Oettl, R. Hegerl, and W. Hoppe, "Three-dimensional reconstruction and averaging of 50S ribosomal subunits of *Escherichia coli* from electron micrographs," *Journal of Molecular Biology*, vol. 163, pp. 431–450, 1983.
- [92] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, pp. 1277–1294, 1993.
- [93] S. K. Pal and D. K. Majumder, *Fuzzy Mathematical Approach to Pattern Recognition*. New Delhi: John Wiley and Sons, Inc., 1997.

- [94] X. Pennec and J. Thirion, "Validation of 3-D Registration Methods based on Points and Frames," Tech. Rep. INRIA Research Report 2470, January 1995.
- [95] L. Perroton, "A new 26-connected objects surface-tracking algorithm and its related pram version," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 9, pp. 719–734, 1995.
- [96] M. Perutz, "X-ray analysis of hemoglobin," *Science*, vol. 140, pp. 863–869, 1963.
- [97] D. P. Petersen and D. Middleton, "Sampling and reconstruction of wave-number-limited functions in N -dimensional Euclidean spaces," *Information and Control*, vol. 5, pp. 279–323, 1962.
- [98] R. Radermacher, T. Wagenknecht, A. Verschoor, and J. Frank, "Three-dimensional reconstruction from a single-exposure, random conical tilt series applied to the 50S ribosomal subunit of *Escherichia coli*," *Journal of Microscopy*, vol. 146, pp. 113–136, 1987.
- [99] R. W. Ramirez, *The FFT: Fundamentals and Concepts*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1985.
- [100] L. Reimer, *Transmission Electron Microscopy: Physics of Image Formation and Microanalysis*, vol. 36 of *Springer series in optical sciences*. Berlin: Springer-Verlag, 1984.
- [101] G. Rhodes, *Crystallography Made Crystal Clear: A Guide for Users of Macromolecular Models*. San Diego, CA: Academic Press, 2nd ed., 2000.
- [102] A. Rosenfeld, "Fuzzy digital topology," *Information Control*, vol. 40, pp. 76–87, 1979.
- [103] P. K. Saha and J. K. Udupa, "Fuzzy connected object delineation: Axiomatic path strength definition and the case of multiple seeds," *Computer Vision and Image Understanding*, vol. 83, pp. 275–295, 2001.
- [104] P. K. Saha and J. K. Udupa, "Relative fuzzy connectedness among multiple objects: Theory, algorithms, and applications in image segmentation," *Computer Vision and Image Understanding*, vol. 82, pp. 42–56, 2001.
- [105] P. K. Saha, J. K. Udupa, E. F. Conant, D. P. Chakraborty, and D. Sullivan, "Breast tissue density quantification via digitized mammograms," *IEEE Transactions on Medical Imaging*, vol. 8, pp. 792–803, 2001.

- [106] P. K. Saha, J. K. Udupa, and D. Odhner, "Scale-based fuzzy connected image segmentation: Theory, algorithms, and validation," *Computer Vision and Image Understanding*, vol. 77, pp. 145–174, 2000.
- [107] M. C. San Martin, C. Gruss, and J. M. Carazo, "Six molecules of SV40 large T antigen assemble in a propeller-shaped particle around a channel," *Journal of Structural Biology*, vol. 268, pp. 15–20, 1997.
- [108] T. Schormann and K. Zilles, "Limitations of the principal-axes theory," *IEEE Transactions on Medical Imaging*, vol. 16, pp. 942–947, 1997.
- [109] R. R. Schröder, "Zero-loss energy-filtered imaging of frozen-hydrated proteins: Model calculations and implications for future developments," *Journal of Microscopy*, vol. 166, pp. 389–400, 1992.
- [110] R. R. Schröder, W. Hofmann, and J.-F. Menetret, "Zero-loss energy filtering as improved imaging mode in cryoelectronmicroscopy of frozen hydrated specimens," *Journal of Structural Biology*, vol. 105, pp. 28–34, 1990.
- [111] D. E. Schumm, *Essentials of Biochemistry*. Boston, Little: Brown and Company, Inc., 2nd ed., 1995.
- [112] I. I. Serysheva, E. V. Orlova, W. Chiu, M. B. Sherman, S. L. Hamilton, and M. van Heel, "Electron cryomicroscopy and angular reconstitution used to visualize the skeletal muscle calcium release channel," *Nature: Structural Biology*, vol. 2, pp. 18–24, 1995.
- [113] G. C. Sharp, S. W. Lee, and D. K. Wehe, "ICP registration using invariant features," *IEEE Transactions of Pattern Analysis and Machine Intelligence*, vol. 24, pp. 90–102, 2002.
- [114] U. Tiede, K. H. Hoehne, M. Bomans, A. Pommert, M. Riemer, and G. Wiebecke, "Investigation of medical 3D-rendering algorithms," *IEEE Computer Graphics and Applications*, vol. 10, no. 2, pp. 41–53, 1990.
- [115] J. K. Udupa and G. T. Herman, *3D Imaging in Medicine*. Boca Raton, Florida: CRC Press, Inc., 1991.
- [116] J. K. Udupa, L. G. Nyul, Y. L. Ge, and R. I. Grossman, "Multiprotocol MR image segmentation in multiple sclerosis: Experience with over 1,000 studies," *Academic Radiology*, vol. 8, pp. 1116–1126, 2001.
- [117] J. K. Udupa and D. Odhner, "Shell rendering," *IEEE Computer Graphics and Applications*, vol. 13, pp. 58–67, 1993.

- [118] J. K. Udupa, P. K. Saha, and R. Lotufo, "Fuzzy connected object definition in images with respect to co-objects," in *Image Processing* (K. Hanson, ed.), vol. 3661 of *Proceedings of SPIE*, pp. 236–245, 1999.
- [119] J. K. Udupa and S. Samarasekera, "Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation," *Graphical Models and Image Processing*, vol. 58, pp. 246–261, 1996.
- [120] J. K. Udupa, L. Wei, S. Samarasekera, Y. Miki, A. van Buchem, and R. I. Grossman, "Multiple sclerosis lesion quantification using fuzzy-connectedness principles," *IEEE Transactions on Medical Imaging*, vol. 16, pp. 598–609, 1997.
- [121] P. A. van den Elsen, E.-J. D. Pol, and M. A. Viergever, "Medical image matching - a review with classification," *IEEE Engineering in Medicine and Biology*, vol. 12, pp. 26–39, 1993.
- [122] M. van Heel, "Angular reconstitution: A posteriori assignment of projection direction for 3D reconstruction," *Ultramicroscopy*, vol. 21, pp. 111–124, 1987.
- [123] M. van Heel, B. Gowen, R. Matadeen, E. V. Orlova, R. Finn, T. Pape, D. Cohen, H. Stark, R. Schmidt, M. Schatz, and A. Patwardhan, "Single-particle electron cryomicroscopy: Towards atomic resolution," *Quarterly Reviews of Biophysics*, vol. 33, pp. 307–369, 2000.
- [124] L. G. Wade, *Organic Chemistry*. Englewood Cliffs, New Jersey: Prentice Hall, Inc., 3rd ed., 1995.
- [125] R. H. Wade, "A brief look at imaging and contrast transfer," *Ultramicroscopy*, vol. 46, pp. 145–156, 1992.
- [126] A. H. Watt, *3D Computer Graphics*. Reading, Mass.: Addison-Wesley Pub. Co., 2nd ed., 1993.
- [127] Z. Y. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, vol. 13, pp. 119–152, 1994.
- [128] H. J. Zimmermann, *Fuzzy Set Theory and Its Applications*. London: Kluwer Academic Publishers, 3rd ed., 1996.

INDEX

Symbols

III

body-centered cubic grid, 45, 129
 Fourier transform, 46
convolution, 27
face-centered cubic grid, 45, 129
sampling, 46, 128
simple cubic grid, 27, 45, 128
 Fourier transform, 45
train of pulses, 27
 \AA , *see* angstrom
 \mathbb{C} , 20
 δ , *see* Dirac function
 Δ_{bcc} , *see* sampling distance
 Δ_{fcc} , *see* sampling distance
 Δ_{sc} , *see* sampling distance
Da, *see* Dalton
 $\vec{\sigma}^\perp$, 20
 \mathbb{R} , 19
 \mathbb{R}^n , 20
 $S^{\bar{y}}$, 24
 S^{n-1} , 20
 \mathcal{S} , *see* Schwartz space
 S' , *see* tempered distributions
 T^n , 20
torr, 14
 \mathbb{Z} , 19
 \mathbb{Z}^n , 20

A

adjacency, 77, 92
 ω , 77
 $\bar{\omega}$, 77
symmetric binary, 77

algebraic reconstruction techniques,
 see ART

algorithm

 boundary tracking, 83
 fuzzy segmentation, 97
 ICP, 103
amino acid, 6
angstrom, 12
angular momentum, 101
ART, 35, 37–40
 blobs, *see* blobs

B

backprojection, 29
 operator, 29
 theorem, 30
bacteriorhodopsin, 69, 131–134
basis functions, 35, 40
 blobs, *see* blobs
bcc, *see* body-centered cubic grid
bels, 76
 connectivity, 79–80
biological macromolecules, 5
blobs, 41–43
 definition, 41
 Fourier transform, 42
 raycasting, 51
 reconstruction criterion, 47–50
 visualization criterion, 55–59
body-centered cubic grid
 slice, 71
body-scale, 1
boundary, 76, 80
 face-centered cubic grid, 81

simple cubic grid, 80
 boundary elements, *see* bels
 bR, *see* bacteriorhodopsin

C

center of mass, 101
 Central Slice Theorem, 28
 Chasles' Theorem, 100
 closest points, 104
 contrast transfer function, 18
 convolution, 26–27, 46
 III, 27
 theorem, 27
 cryomicroscopy, 15–16
 CTF, *see* contrast transfer function

D

Dalton, 6
 data collection
 angular reconstitution, 32
 projection matching, 32
 random conical tilting, 31
 single axis tilting, 30
 DnaB·DnaC, 53, 59, 69
 density function, 4, 17, 20, 28
 discretized, 61, 63, 68
 digital space, 92
 Dirac function, 21, 23, 24

F

fcc, *see* face-centered cubic grid
 figure of merit, 112
 FOM, *see* figure of merit
 Fourier transform, 23–28
 III_G, 45
 III_B, 46
 convolution, 46
 Convolution Theorem, 27
 Dirac function, 24
 inverse, 24
 scaling, 26

shifting property, 25
 tempered distributions, 25

Fourier transform

blobs, 42
 fuzzy connectedness, 92
 fuzzy segmentation, 91–99
 M classes, 93
 ψ -connectivity, 95
 ψ -strength, 93, 95
 2-semisegmentation, 93
 affinity, 93, 94
 algorithm, 97
 analogy, 97
 chain, 92, 95
 fuzzy connectedness, 92
 fuzzy graph, 95
 fuzzy voxel affinity, 93
 links, 92
 membership, 92
 seeds, 92, 94, 95
 semisegmentation, 93
 strength, 92, 93, 95
 theorem, 95
 voxel affinity, 93

G

global axes, 100
 grid, 43
 body-centered cubic, 44
 face-centered cubic, 44
 face-centered cubic grid, 91
 simple cubic, 44

H

Halobacterium halobium, 131
 Hausdorff, 103–104

I

ICP, 102–107
M matrix, 105
 algorithm, 103

closest points, 104
 Hausdorff, 103–104
 principle, 106
 rotation matrix **R**, 105
 variance, 105
 image, 20
 binary image, 76
 micrograph, 12
 implicit surface, 50
 inner product, 20
 Iterative Closest Point, *see* ICP

K

Kaiser-Bessel functions, *see* blobs

L

line, 20
 line integral, 28
 lipids, 5

M

matrix of inertia, 101
 micro-scale, 1
 micrograph, 12, 28, 69
 noise, 17, 70
 projection, 17
 signal-to-noise ratio, 18, 70

N

nano-scale, 2
 negative staining, 15
 NMR, *see* Nuclear Magnetic Resonance
 Nuclear Magnetic Resonance, 11
 nucleic acids, 5
 null hypothesis, 112
 random variable, 113

O

OpenDX™, 60, 63, 87

P

PAT, 100–102

angular momentum, 101
 center of mass, 101
 global axes, 100
 matrix of inertia, 101
 primed axes, 100
 principal axes, 101
 PDB, *see* Protein Data Bank
 point spread function, 17, 70
 polygon projection, 59
 polysaccharides, 5
 primed axes, 100
 principal axes, 101
 Principal Axes Technique, *see* PAT
 projection, 17, 28
 protein, 5–9
 heterooligomers, 8
 homooligomers, 8
 primary structure, 6
 quaternary structure, 6
 secondary structure, 6
 size, 6
 structure analysis, 9–12
 symmetry, 8
 groups, 8
 tertiary structure, 6
 Protein Data Bank, 69, 134
 psf, *see* point spread function

R

ray transform, 28
 raycasting, 51–52
 blobs, 51, 66, 67
 increased speed, 67–74
 comparison with OpenDX™, 60–65
 reconstruction, 33–39
 ART, 35, 37–40
 blobs, *see* blobs
 basis functions, 35, 40
 filtered backprojection, 34
 Kaczmarz's method, 37
 series expansion methods, 35

transform methods, 35
 registration, 99
 rigid body, 100
 rms error, 48
 criterion for reconstruction, 49
 criterion for visualization, 55

S

sampling, 45, 46, 128
 sampling distance
 body-centered cubic grid, 44
 face-centered cubic grid, 45
 simple cubic grid, 44
sc, *see* simple cubic grid
 scattering
 elastic, 16
 inelastic, 16
 Schwartz space, 21
 segmentation, 91
 thresholding, 50, 52, 66, 67
 shah, *see* III
 signal-to-noise ratio, 18, 70
 sinogram, 32
 SNR, *see* signal-to-noise ratio
 specimen
 cryomicroscopy,
 see cryomicroscopy
 negative staining, *see* negative stain-
 ing
 preparation, 14–16
 radiation damage, 14, 18
 spherical coordinates, 20

T

TEM, 11–14
 airlock, 13
 contrast transfer function, 18
 cryomicroscopy,
 see cryomicroscopy
 data collection, 19
 image formation, 16–19

amplitude contrast, 16
 phase contrast, 16
 micrograph, 12
 negative staining, *see* negative stain-
 ing
 point spread function, 17, 70
 single particles, 19
 specimen stage, 13
 tempered distributions, 21–23
 tessellation, 75
 train of pulses, 27
 transmission electron microscope,
 see TEM

U

unscattered electrons, 16

V

vector, 20
 unit vector, 20
 volume elements, *see* voxels
 Voronoi neighborhood, 74
 voxels, 74
 body-centered cubic grid, 75
 face-centered cubic grid, 75
 simple cubic grid, 75

X

X-ray crystallography, 10–11

Z

z-buffer, 68–69