Computerized Classification of Surface Spikes in Three-Dimensional Electron Microscopic Reconstructions of Viruses

Younes Benkarroum

The Graduate Center, City University of New York

Recommended Citation

How does access to this work benefit you? Let us know!
Follow this and additional works at: https://academicworks.cuny.edu/gc_etds
Part of the Biology Commons, and the Numerical Analysis and Scientific Computing Commons

This Dissertation is brought to you by CUNY Academic Works. It has been accepted for inclusion in All Dissertations, Theses, and Capstone Projects by an authorized administrator of CUNY Academic Works. For more information, please contact deposit@gc.cuny.edu.
COMPUTERIZED CLASSIFICATION OF SURFACE SPIKES IN
THREE-DIMENSIONAL ELECTRON-MICROSCOPIC RECONSTRUCTIONS
OF VIRUSES

by

YOUNES BENKARROUM

A dissertation submitted to the Graduate Faculty in Computer Science
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy, The City University of New York

2016
This manuscript has been read and accepted for the
Graduate Faculty in Computer Science in satisfaction of the
dissertation requirements for the degree of Doctor of Philosophy.

Date

Dr. Gabor T. Herman, Chair of Examining Committee

Date

Dr. Robert M. Haralick, Executive Officer

Dr. Robert M. Haralick

Dr. Zhigang Zhu

Dr. Paul Gottlieb

Dr. Jose-Maria Carazo
Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK
Abstract

Computerized classification of surface spikes in three-dimensional electron microscopic reconstructions of viruses

by

Younes Benkarroum

Adviser: Gabor T. Herman

The purpose of this research is to develop computer techniques for improved three-dimensional (3D) reconstruction of viruses from electron microscopic images of them and for the subsequent improved classification of the surface spikes in the resulting reconstruction. The broader impact of such work is the following.

Influenza is an infectious disease caused by rapidly-changing viruses that appear seasonally in the human population. New strains of influenza viruses appear every year, with the potential to cause a serious global pandemic. Two kinds of spikes – hemagglutinin (HA) and neuraminidase (NA) – decorate the surface of the virus particles and these proteins are primarily responsible for the antigenic changes observed in influenza viruses. Identification of the locations of the surface spikes of both kinds in a new strain of influenza virus can be of critical importance for the development of a vaccine that protects against such a virus.

Two major categories of reconstruction techniques are transform methods such as weighted backprojection (WBP) and series expansion methods such as the algebraic reconstruction techniques (ART) and the simultaneous iterative reconstruction technique (SIRT). Series expansion methods aim at estimating the object to be reconstructed by a linear combination of some fixed basis functions and they typically
estimate the coefficients in such an expansion by an iterative algorithm. The choice of the set of basis functions greatly influences the efficacy of the output of a series expansion method. It has been demonstrated that using spherically symmetric basis functions (blobs), instead of the more traditional voxels, results in reconstructions of superior quality. Our own research shows that, with the recommended data-processing steps performed on the projection images prior to reconstruction, ART (with its free parameters appropriately tuned) provides 3D reconstructions of viruses from tomographic tilt series that allow reliable quantification of the surface proteins and that the same is not achieved using WBP or SIRT, which are the methods that have been routinely applied by practicing electron microscopists.

Image segmentation is the process of recognizing different objects in an image. Segmenting an object from a background is not a trivial task, especially when the image is corrupted by noise and/or shading. One concept that has been successfully used to achieve segmentation in such corrupted images is fuzzy connectedness. This technique assigns to each element in an image a grade of membership in an object.

Classifications methods use set of relevant features to identify the objects of each class. To distinguish between HA and NA spikes in this research, discussions with biologists suggest that there may be a single feature that can be used reliably for the classification process. The result of the fuzzy connectedness technique we conducted to segment spikes from the background confirms the correctness of the biologists’ assumption. The single feature we used is the ratio of the width of the spike’s head to the width of its stem in 3D space; the ratio appears to be greater for NA than it is for HA. The proposed classifier is tested on different types of 3D reconstructions derived from simulated data. A statistical hypothesis testing based methodology allowed us to evaluate the relative suitability of reconstruction methods for the given classification task.
To my parents, wife and children.
Acknowledgments

I would like to express my profound gratitude and sincere thanks to my adviser, Dr. Gabor T. Herman, for his continuous support, valuable guidance and permanent encouragement throughout my PhD study and related research. His immense knowledge has been a source of inspiration at all stages of the research. I am so fortunate and honored to have had such a great adviser.

I would also like to extend my sincere appreciation to the rest of my committee members Dr. Robert M. Haralick, Dr. Zhigang Zhu, Dr. Paul Gottlieb and Dr. Jose-Maria Carazo for their advice and insightful comments during my research.


Special thanks to my colleagues Dr. Joanna Klukowska, Dr. Lucas de Melo Oliveira, Dr. Marcelo Zibetti, Dr. Chuan Lin, Ze Ye and Amir Toliyat for their support and cooperation.

Finally, and most importantly, I would like to thank my wonderful wife Hind for the countless sacrifices she made to help me get to this point. I would also like to extend my deepest gratitude to all my family members for their prayers, moral support and encouragement throughout the course of my study and my life in general.
# Contents

**Introduction** ................................................................. 1

1 **Background** ................................................................. 4
   1.1 Virus Structure ......................................................... 5
   1.2 Iterative Reconstruction Methods and Their
      Implementation ......................................................... 9
      1.2.1 Algebraic Reconstruction Techniques (ART) .............. 11
      1.2.2 Basis Functions ................................................. 12
   1.3 3D Grids ................................................................. 14

2 **Desirable Blob Parameters** .............................................. 19
   2.1 Blobs and Their Fourier Transforms .............................. 20
   2.2 Approximation of Constant-Valued Functions
      Using Blobs ............................................................. 23
   2.3 Representation of Images Using Blobs with
      Desirable Parameters ................................................. 30
   2.4 Image Reconstruction Using Blobs with Desirable Parameters 47

3 **Influenza Virus Reconstruction** ....................................... 52
   3.1 Data Collection ....................................................... 53
   3.2 Data Processing ...................................................... 54
   3.3 Virus Reconstruction ................................................ 63
3.4 Results and Discussion .................................................. 66

4 Virus Spikes Classification .............................................. 68
   4.1 Identification of the Region of Interest .......................... 68
   4.2 Segmentation .......................................................... 70
      4.2.1 Description of the Approach ................................. 70
      4.2.2 Theory and Algorithm ....................................... 71
      4.2.3 Experiments .................................................. 73
   4.3 Feature Extraction and Classification ............................ 77
      4.3.1 Reorientation of Spike Structures ........................... 77
      4.3.2 Feature Extraction .......................................... 80
      4.3.3 Classification ............................................... 84

5 Evaluation .............................................................. 88
   5.1 Evaluation Methodology .......................................... 88
      5.1.1 Generation of Phantoms and Their Projection Data ...... 89
      5.1.2 Reconstructions by the Algorithms to be Compared ..... 91
      5.1.3 Assignment of an FOM to Each Reconstruction .......... 92
      5.1.4 Calculation of Statistical Significance ................. 93
   5.2 Comparisons of Algorithm Efficacy for Spike
        Classification ................................................... 96

6 Discussion .............................................................. 98
   6.1 Printed Publications Based on Work for This
       Dissertation ....................................................... 98
   6.2 Workshop Presentations Based on Work for This
       Dissertation ....................................................... 99
   6.3 Summary of the Contributions of this
       Dissertation ....................................................... 99
List of Tables

2.1 First nine roots (zero crossings) of $J_{7/2}$ with $j_r$ denoting the $r$th root. 23
2.2 Desirable blob parameters for various pairs of roots of $J_{7/2}$. 26
2.3 Values $q_3/q_0$, $q_4/q_0$ and $q_5/q_0$ of $\hat{w}_{2,a,\alpha}(R)/\hat{w}_{2,a,\alpha}(0)$ at distances $R_3 = \sqrt{6}/2\beta$, $R_4 = \sqrt{10}/2\beta$ and $R_5 = 2/\beta$, respectively, from the origin. The $a$ and $\alpha$ are as in Table 2.2 with $\beta = 1/\sqrt{2}$. 27
2.4 Values of the error $E_{\beta,\theta,r_1,r_2}$ for the $(r_1,r_2)$ pairs $(1 \leq r_1 < r_2 \leq 9)$ with $\beta = 1/\sqrt{2}$ and $\theta = 4$. 44
2.5 The pairs $(r_1,r_2)$ that minimize the error $E_{\beta,\theta,r_1,r_2}$ for various values of $\theta$ with $\beta = 1/\sqrt{2}$, together with $E_{\beta,\theta,STD}$ (which is greater than the optimal $E_{\beta,\theta,r_1,r_2}$ in all cases). 44
5.1 Example of an array created to calculate the classification purity of a reconstruction. 93
5.2 Average values of the classification purity over 30 data sets. 94
5.3 P-values for pairwise comparisons of reconstruction algorithms. 97
List of Figures

1.1 Schematic of influenza virus. ........................................... 6

1.2 Distributions and shape-based differentiation of HA and NA spikes.
(a) HA cluster (left); single NA (marked) in a cluster of HAs (center); and cluster of mainly NA spikes (right) (scale bar, 50 nm). (b) HA and (c) NA stems (scale bar, 5 nm). (d) Patches of glycoprotein spikes depicted in tangential sections; triangular HA spikes (e.g., white arrowhead) are distinguishable from square NA spikes (e.g., white arrowhead with black border). Images are reproduced from [29]. ............................................................... 7

1.3 An aligned micrograph of influenza B/Lee/40 virions. ............... 8

1.4 Grid points of (a) the simple cubic grid $G_{\sigma}$, (b) the body-centered cubic grid $B_{\beta}$ and (c) the face-centered cubic grid $F_{\phi}$ in a $2\sigma \times 2\sigma \times 2\sigma$, $2\beta \times 2\beta \times 2\beta$, and $2\phi \times 2\phi \times 2\phi$ portion of 3D space, respectively. ....................................................... 16

2.1 Values of $w_{m,a,\alpha}(\vec{x})$ as a function of the norm $r$ of $\vec{x}$ for the (standard) parameters $m = 2$, $a = 2$ and $\alpha = 10.4$ (green) and for the (recommended) parameters $m = 2$, $a = 2.453144$ and $\alpha = 13.738507$ (red). ............................................................. 21

2.2 Plot of the Bessel function $J_{7/2}$. ..................................... 22
2.3 Central slices of (a) the ball phantom and its blob representations using (b) \( r_2 = 2 \), (c) \( r_2 = 3 \), (d) \( r_2 = 4 \), (e) \( r_2 = 5 \), (f) \( r_2 = 6 \), (g) \( r_2 = 7 \), (h) \( r_2 = 8 \) and (i) \( r_2 = 9 \). The blob parameters \( a \) and \( \alpha \) are as in Table 2.2 with \( \beta = 1/\sqrt{2} \). The red circles indicate the location of the ball’s circumference. Display thresholds are 0.999 and 1.001.

2.4 Plot of the ball phantom (blue) and the blob representations using \( r_2 = 4 \) (red) and \( r_2 = 9 \) (black) along the line \((r,0,0)\). The blob parameters \( a \) and \( \alpha \) are as in Table 2.2 with \( \beta = 1/\sqrt{2} \).

2.5 Plot of the ball phantom (blue) and the blob representations using the recommended parameters \( a = 2.453144 \) and \( \alpha = 13.738507 \) (red) and the standard parameters \( a = 2 \) and \( \alpha = 10.4 \) (green) along the line \((r,0,0)\).

2.6 Plot of the first term of the right hand side of (2.21) (blue with values on the left vertical axis) along with plots of the second term for the two representations using \( r_2 = 4 \) and \( r_2 = 9 \) and the representation using standard parameters (red, black and green, respectively, with values on the right vertical axis). The three graphs plot the values of these terms for the points \((R/\sqrt{2}, R/\sqrt{2},0)\). The blob parameters \( a \) and \( \alpha \) for \( r_2 = 4 \) and \( r_2 = 9 \) are as in Table 2.2 with \( \beta = 1/\sqrt{2} \); the standard parameters are \( a = 2 \) and \( \alpha = 10.4 \).

2.7 Plots of the second term of the right hand side of (2.21) (a) and its base 10 logarithm (b) for \( r_2 = 4 \) (red), for \( r_2 = 9 \) (black) and for the standard parameters (green). The blob parameters \( a \) and \( \alpha \) for \( r_2 = 4 \) and \( r_2 = 9 \) are as in Table 2.2 with \( \beta = 1/\sqrt{2} \); the standard parameters are \( a = 2 \) and \( \alpha = 10.4 \).
2.8 A central slice through three ball representations: (a) using our recommended parameters \( a = 2.453144 \) and \( \alpha = 13.738507 \), (b) using the standard parameters \( a = 2 \) and \( \alpha = 10.4 \) (these are obtained from (2.14) by rounding to one place after the decimal point), (c) using an improved version of the standard parameters \( a = 2 \) and \( \alpha = 10.444256 \) (obtained by rounding to six places after the decimal point). Display thresholds are 0.999 and 1.001.

2.9 Central slices of (a) the FORBILD abdomen phantom and (b) its blob representation using our recommended parameters \( a = 2.453144 \) and \( \alpha = 13.738507 \); display thresholds are 0.0 cm\(^{-1}\) and 1.5 cm\(^{-1}\).

2.10 Central slices of (a) the FORBILD thorax phantom and (b) its blob representation using our recommended parameters; display thresholds are 0.18 cm\(^{-1}\) and 0.3175 cm\(^{-1}\).

2.11 Central slices \((x_3 = 0)\) through the ball reconstructions and the plots of their values, as \(x_2\) varies from -90 to +90, along the line for which \(x_1 = 10\) and \(x_3 = 0\): (a,c) using our recommended parameters \( a = 2.453144 \) and \( \alpha = 13.738507 \); and (b,d) using the standard parameters \( a = 2 \) and \( \alpha = 10.4 \). Display thresholds in (a) and (b) are 0.99 (black) to 1.01 (white).

2.12 Central slices of the reconstructions of the FORBILD abdomen phantom: (a) using voxels and (b) using blobs with our recommended parameters \( a = 2.453144 \) and \( \alpha = 13.738507 \). Display thresholds are 0.0 - 1.5 cm\(^{-1}\).

2.13 Central slices of the reconstructions of the FORBILD thorax phantom: (a) using voxels and (b) using blobs with our recommended parameters. Display thresholds are 0.18 - 0.3175 cm\(^{-1}\).
3.1 Three projection images of the specimen at angles (from left to right): $-60.32^\circ$, $-0.10^\circ$ and $+59.66^\circ$. .......................... 54
3.2 Line integrals in a projection image of a tomographic tilt series. ........ 55
3.3 Manually determining the disk center in a 2D micrograph. ............... 59
3.4 Projection of $(x_1, x_3)$ after rotation by tilt angle $\alpha_u$. ............. 60
3.5 Plot of the values of $d(x_1, x_3)$ indicating the least squares solution. 61
3.6 Three projections of the isolated virus after processing. .................. 63
3.7 Three different slices ($k_3 = 89, 90$ and $91$) from virus reconstructions: (a), (b) and (c) are from the ART reconstruction; (d), (e) and (f) are corresponding slices from the SIRT reconstruction and (g), (h) and (i) are corresponding slices from the filtered WBP reconstruction; all from the same projection data .......................... 65
3.8 (Left) Slice ($k_3 = 90$) of the reconstructed influenza B/Lee/40. Several HA (yellow) and NA (red) are shown. RNPs are outlined in blue. (Right) Enlarged view of slice with atomic models of HA and NA docked in the image. Bar is 25 nm. ............................... 67
4.1 Gray-value image of the slice displayed on the left of figure 3.8: (a) superimposing the inner and outer circles on it and (b) setting the values of all pixels outside the annulus between the circles to zero. 69
4.2 Interactive seed-selection application used in the segmentation of spikes in the region of interest defined in Section 4.1. ....................... 74
4.3 Surface-smoothed display of the structure system of virus spikes segmented within the region of interest by the MOFS algorithm using the seeds whose selection is illustrated in figure 4.2. ......... 76
4.4 Reorientation of spike $s$ (here $s = 23$). ................................. 78
4.5 Feature extraction steps for the $23^{rd}$ spike structure. .................... 82
4.6 Feature extraction for (a) the 11th, (b) the 12th, and (c) the 16th spike structures. ................................................................. 83
4.7 (a) Fisher’s linear discriminant $J$, see (4.1), plotted against the threshold candidates. (b) $R_{\text{head/stem}}$ numerical values in the increasing order. The threshold $\tau$ that maximizes the Fisher’s linear discriminant is marked by the red line in (b). ........................................ 85
4.8 Reoriented vertical planar sections of all spike structures along with their feature values $R_{\text{head/stem}}$. ................................................................. 86
4.9 Virus spikes classification. Spikes that belong to the classes $c_1$ and $c_2$ are colored with yellow and red, respectively. ................................. 87
5.1 Ellipsoids simulating (a) HA spike’s head with semi-axes $u = 3.5\sigma$, $v = 3.5\sigma$, and $w = 6\sigma$ and (b) NA spike’s head with semi-axes $u = 6\sigma$, $v = 6\sigma$, and $w = 3\sigma$. ......................................................... 90
5.2 Spike distribution: (a) Lateral views of 3 slices where spikes are equidistantly placed and (b) 3D locations of random spikes in an illustrative example. The locations of HA and NA spikes are colored with yellow and red, respectively. ................................. 90
5.3 Central slices of (a) a phantom and its 3D reconstructions: (b) ART using blobs with desirable parameters, (c) ART using blobs with standard parameters, (d) SIRT using blobs with desirable parameters, (e) ART using voxels, and (f) WBP using the default parameters. 92
Introduction

Influenza is a rapidly changing virus that manifests itself seasonally in the human population. Every few years a new strain of the influenza virus appears and causes a serious global pandemic. Knowledge of the structure and density of the virus surface proteins is of critical importance in a vaccine candidate [9, 50, 51]. Each season, the vaccine must be re-engineered to match the current influenza strains with rapid production capability.

In this computer science dissertation, we will be working with digital representations of the underlying biological objects, as discussed, for example, in Subsection 1.1.4 of [56]. This means that the biological object is virtually partitioned into small abutting cuboidal volume elements, referred to as voxels. The 2D analog of a voxel is a pixel (abbreviation for picture element). A 3D scene of an object is a 3D rectangular array of voxels together with a value assigned to each voxel in the array. The digital representations mentioned at the beginning of this paragraph will be 3D scenes in the just-defined sense. We will adopt the terminology of Subsection 1.1.4 of [56] in a similar fashion in all our discussions related to digital imaging.

Electron microscopy (EM) is an important method for determining the three-dimensional (3D) structure of biological specimens; it allows the 3D reconstruction of an object (a specimen) by gathering the two-dimensional (2D) information present in projection images taken of the specimen at different orientations with an electron microscope [22, 23]. The reconstructions that are produced are 3D scenes
in the sense of the previous paragraph, with the value assigned to a voxel related to the average value of the physical parameter called Coulomb potential of the matter occupying that voxel. Projections are obtained using beams of electrons that are accelerated toward the specimen using a positive electrical potential. Scattering occurs inside the irradiated sample, affecting the electron beam; these interactions and effects are detected and spatially mapped into an image, the values in which are related to the line integrals of the Coulomb potential values in the biological object to be reconstructed. Since the biological tissues react very sensitively to electron beams, data are collected using low electron current resulting in a poor signal-to-noise ratio (SNR).

Influenza is pleomorphic (i.e., the shape and size is subject to environmental conditions). Thus, 3D reconstruction techniques that assume the availability of multiple identical copies of the object to be reconstructed (such as single particle reconstruction [43] or tomogram averaging [8]) cannot be employed to achieve our aim. Electron tomography (ET) [22] is a suitable approach, since it creates its reconstructions from multiple projections of just one copy of the object to be reconstructed. However, there are undesirable consequences of this method of data collection: namely, the limited angular range, the small number and the low SNR of the projections. These interfere with our ability to produce high quality reconstructions of the viruses, which makes it difficult to reliably analyze the structure of the virus surface proteins. Therefore, our first task is to develop a reconstruction procedure that can produce high quality reconstructions of viruses from ET data. Our second task is to develop a classification procedure that can be applied to such reconstructions to provide a reliable classification of the surface proteins.

The two major categories of reconstruction techniques in ET are transform methods such as weighted backprojection (WBP) and series expansion methods such as the algebraic reconstruction techniques (ART); see, for example, [31].
former methods have been widely used because of their fast speed and simplicity of implementation, while the latter methods have a significant capability to provide greater detail with incomplete and/or noisy data [45]. Series expansion methods aim at estimating the object to be reconstructed by a linear combination of some fixed basis functions and they typically estimate the coefficients in such an expansion by an iterative algorithm. The choice of the set of basis functions greatly influences the result of a series expansion method. It has been demonstrated repeatedly that using spherically symmetric basis functions (blobs), instead of the more traditional voxel-based basis functions, results in reconstructions of superior quality, provided that the free parameters that occur in the definition of the family of blobs are appropriately tuned. This general statement is indeed borne out below by the application of ART using blobs to the influenza virus.

The chapters of this dissertation are organized as follows. Chapter 1 describes the structure of the virus and reveals the method we used for the 3D reconstruction. In Chapter 2, a technique for the determination of good parameters for image representation using blob basis functions is presented and demonstrated. Chapter 3 exhibits the data-processing steps performed on the projection images prior to the virus reconstruction. Chapter 4 lists the steps we considered to build the protein classifier and Chapter 5 describes the evaluation methodology for the efficacy of the protein classifier. A discussion of what has been done, including its significance, is provided in Chapter 6.
Chapter 1

Background

Influenza causes acute respiratory disease in humans and animals. Due to the antigenic diversity that is seen in influenza viruses, new vaccines must be reformulated on an annual schedule. The considerable variation in year-to-year efficiency of vaccine production can lead to significant vaccine shortages such as occurred during the H1N1 pandemic in 2009 [21]. The surface glycoprotein antigens are significant immunogens that contribute to the development of an anti-influenza response [6]. Understanding basic virus structure and properties of these viruses would aid in determining the best candidate for maximal antigen yield during vaccine production. Furthermore, the quantities and relative amounts of the types of the virus surface proteins will affect efficacy of the vaccine in producing an immunogenic response.

In this chapter we describe the structure of the virus and discuss background information necessary for iterative reconstruction methods using two types of basis functions: the traditional rectangular voxels and the spherically symmetric blobs; then we discuss the importance of the choice of the spatial arrangement of the set of points where those basis functions are placed in order to obtain image reconstructions of superior quality.
1.1 Virus Structure

Influenza viruses are members of the orthomyxoviridae virus family; they are divided into three types, A, B and C, which are determined by their internal proteins and are antigenically distinct (i.e., they stimulate the production of different antibodies). Types A and B cause annual epidemics of respiratory disease and novel type A influenza virus may cause pandemics such as in 1918.

Influenza contains a lipid bilayer envelope surrounding a protein matrix. Inside the matrix is the genome consisting of distinct segments of negative polarity (non-protein coding) RNA that form complex helical structures, termed ribonucleoproteins (RNPs). Types A and B viruses have eight RNPs and type C has seven RNPs. Each RNP segment encodes at least one viral protein. Influenza A viruses are further divided into subtypes based on the amino acid sequences of the spikes projecting from the envelope surface. Influenza virions have variable morphology ranging from spherical to filamentous. Shape variation often affects growth characteristics in cell cultures.

Virus strain identification is based on several factors: initial animal host, geographical origin, strain isolate, and the year of isolation. A high density of two types of glycoprotein spikes (each composed of a protein and a carbohydrate), hemagglutinin (HA) and neuraminidase (NA), are observed projecting from the envelope surface in what appears to be random placement. HA spikes are responsible for viral attachment to the host cell and are the major antigenic determinant (i.e., they are the parts that are recognized by the immune system). NA spikes are responsible for viral exit from the infected host. There are sixteen HA (H1-H16) and nine NA (N1-9) influenza A subtypes (it is the combination of these that leads to designations such as H5N1). X-ray crystallography has revealed the atomic structure of the entire HA [15, 58, 59, 62] and the top segment of the NA [10, 57]. HA are trimers (compounds of three macromolecules) with a cell-receptor-binding domain
that binds the virus to its host cell, and an elongated fusion domain that fuses the viral envelope to the cell envelope [19]. NA are club-shaped tetrameres (proteins with four subunits) with a protein conformation described as anti-parallel $\beta$-sheets arranged in a propeller blade conformation [2]. A schematic of a spherical influenza virus is shown in figure 1.1.

Cryogenic electron microscopic tomography (cryo-EM tomography) [22] has been employed to study the two influenza surface proteins and their distribution on the viral surface. Harris et al. [29] visualized the 3D structure of a type A H3N2 strain X 31 virus using cryo-EM tomography and determined that a typical 120 nm diameter type A influenza virion can contain up to 375 surface spikes, but the actual count could be lower due to bare spots. The shapes of the HA and NA spikes and how they are located relative to the lipid bilayer envelope surrounding the protein matrix are indicated in figure 1.2. In the gray-value images in this figure, the darker values indicate higher Coulomb potentials at the corresponding locations of the reconstructed 3D scenes. Visualization of the spikes is possible due to the fact that the Coulomb potential is higher for protein than for the ice into which the virus is embedded for the purpose of cryogenic electron microscopy. Note the shortness of
Figure 1.2: Distributions and shape-based differentiation of HA and NA spikes. (a) HA cluster (left); single NA (marked) in a cluster of HAs (center); and cluster of mainly NA spikes (right) (scale bar, 50 nm). (b) HA and (c) NA stems (scale bar, 5 nm). (d) Patches of glycoprotein spikes depicted in tangential sections; triangular HA spikes (e.g., white arrowhead) are distinguishable from square NA spikes (e.g., white arrowhead with black border). Images are reproduced from [29].

the stems of the HA spikes relative to those of the NA spikes.

Calder et al. [12] employed cryo-EM tomography to study the structural organization of filamentous influenza A and observed that the interaction between the M1 protein (a matrix protein of the influenza virus) and surrounding envelope determines the morphology of the virion. Giocondi et al. [27] used atomic force microscopy to study the 3D topography of H1N1 influenza and a lateral heterogeneity of the HA and NA spikes was observed for virions at neutral pH and after treatment at pH 5. The distributions of surface glycoproteins on two type A virus particles (A/Udorn/72 and A/Aichi/68 X-31) have recently been determined [61]. Influenza-laboratory-adapted strains are typically ellipsoidal with diameters ranging from approximately 100 to 130 nm. However, the virions also can exist as larger
ellipsoids or filamentous particles that can extend several microns in length and the virus particle morphology often influences growth characteristics [12, 50].

Figure 1.3 shows an aligned projection image of influenza type B/Lee/40; the tilt axis and a red box enclosing the virus we reconstructed for this research are displayed. (The nature of the alignment is explained near the beginning of Chapter 3 below.) Note the variation in size and shape of the virions in the micrograph. The HA and NA surface spikes are visible in the images, but the resolution is not adequate to accurately classify the protein spike type.
1.2 Iterative Reconstruction Methods and Their Implementation

As mentioned earlier, series expansion methods assume that the 3D object \( f \) to be reconstructed can be approximated by a linear combination of a finite set of known and fixed basis functions. More precisely, let \( \{ \vec{x}_n \}_{n=1}^N \) be a set (called a grid) of \( N \) points in 3D space and let \( b \) be a fixed function (called the basic basis function). These together can be used to specify a function \( f^* \), which is constructed as a linear combination of the basis functions, \( b_n \), that are shifted versions of \( b \), as follows:

\[
 f^*(\vec{x}) = \sum_{n=1}^{N} c_n b_n(\vec{x}), \quad (1.1)
\]

where \( \{c_n\}_{n=1}^{N} \) is the set of coefficients of the specification and

\[
b_n(\vec{x}) = b(\vec{x} - \vec{x}_n). \quad (1.2)
\]

For a fixed grid and \( b \), various functions over 3D space (which we often refer to as 3D images) can be approximated by an appropriate choice of the coefficients.

Projections are acquired measurements. Each (of a total number \( M \)) measurement provides an approximation to the integral along a straight line of the unknown spatial distribution of the physical parameter to be reconstructed. Let \( p_i^* \) denote the line integral of \( f^* \) along the straight line of index \( i \) \((1 \leq i \leq M)\). Then

\[
p_i^* = \sum_{n=1}^{N} a_{i,n} c_n, \quad (1.3)
\]

where \( a_{i,n} \) is the line integral, along the straight line \( i \), of the shifted basis function centered at \( \vec{x}_n \).

In order to estimate the coefficients for the 3D object based on the projection
measurements, typically an iterative method is used; it produces a sequence of vectors \( c^{(0)}, c^{(1)}, \ldots \) that is supposed to converge. The \( k \)th iterate determines, according to (1.1), an image \( f^{(k)} \) from the set of coefficients \( \{ c_n^{(k)} \}_{n=1}^{N} \). The algorithm attempts to find a vector of coefficients \( c_n^{(k)} \) such that the line integrals of \( f^{(k)} \) are good approximations of the measured data. Let \( \{ p_i \}_{i=1}^{M} \) be the measured vector that has already been processed to comprise (approximations) of line integrals of the function \( f \) to be reconstructed. Based on the expression (1.3), the algorithm attempts to find a vector \( c \) (having components \( c_n \)) that is an approximate solution to the linear system \( p = Ac \), where \( p \) is the measured data vector and \( A \) is the system matrix of size \( MN \) having elements \( a_{i,n} \). We refer to \( A \) as the projection matrix.

If the size of the projection matrix \( A \) were small, conventional matrix theory methods could be used to invert the system of equations in (1.3). However, in practice the system matrix is often huge; it can have as many as \( 10^{13} \) elements for the fully 3D reconstruction case, which inhibits direct matrix inversion. (The number \( 10^{13} \) is derived as follows. A single projection image is \( 200 \times 200 \) pixels. There are 61 such images. The reconstruction region is \( 200 \times 200 \times 200 \) voxels. For an iterative reconstruction technique using voxels, the size of the system matrix is the product of these numbers, roughly \( 2 \times 10^{13} \).) For that reason iterative methods are used, the coefficient values \( c_n \) are iteratively corrected so that the calculated projections \( p_i^* \) approach the recorded measurements \( p_i \). This iterative correction forms the basis of the algebraic reconstruction algorithms, however the nature and implementation of this correction can vary significantly, and subsequently effect the convergence and quality of the reconstruction. Details of an ART implementation are presented in the following subsection.
1.2.1 Algebraic Reconstruction Techniques (ART)

The algebraic reconstruction technique that we selected to solve the system of linear equations was first proposed by Kaczmarz [38] and introduced to the biomedical field by Gordon et al. [28]. It starts from an initial guess (we used the vector in which all coefficients are zero) for the reconstructed object and then performs an iterative sequence of projections, as in (1.3), and corrective backprojections until the reconstruction converges:

\[
c_n^{(k+1)} = c_n^{(k)} + \lambda^{(k)} \frac{p_{i^{(k)}} - \sum_h a_{i^{(k)}, h} c_h^{(k)}}{\sum_h a_{i^{(k)}, h}^2} a_{i^{(k)}, n},
\]

(1.4)

where \(1 \leq n \leq N\) and \(\lambda^{(k)}\) is a real number, called the relaxation parameter. Mathematical theory allows us the freedom of choosing the relaxation parameter to be between 0 and 2, but practical experience with real projection data indicates that a low value (such as 0.05) is likely to be more efficacious [31, 44, 45]. In (1.4), we denote by \(i^{(k)}\) the \(k\)th index, which is taken cyclically from 1 to \(M\); i.e., \(i^{(k)} = (k \mod M) + 1\).

A distinguishing feature of ART is exactly that it corrects for only one measured line integral in a single iterative step (1.4). As opposed to this, the method SIRT [26] makes corrections simultaneously based on all the measured line integrals in one of its iterative steps. The effect of this difference on algorithm performance is analyzed in detail in Chapters 11 and 12 of [31]. A comparison of ART, SIRT and WBP from the structural biology point of view is reported in [53], with the conclusion that “both ART and SIRT outperform WBP when the free parameters have been properly selected, although ART does so at a fraction of the computational cost (between one and two orders of magnitude) required by SIRT”. This conclusion is further affirmed by the experiment on which we report below. Indeed, all careful comparison studies indicate that ART is more efficacious than either SIRT or WBP; see, for example, [13].
From (1.4), we notice that the image is updated in an additive fashion for each line $i^{(k)}$ in turn using a discrete backprojection, with the scalar that multiplies $a_{i^{(k)},n}$ proportional to the measurement $p_{i^{(k)}}$ for the $i^{(k)}$th line minus the forward projection of the current estimate for that line. The order in which data are accessed during the reconstruction procedure can have a significant effect on the practical performance of the algorithm. Herman and Meyer [34] introduced a computationally efficient data-access ordering for ART. The intuitive principle is that in a subsequence of iterative steps of the type (1.4), the action should be as independent as possible of the previous actions; in other words, the vector whose $n^{th}$ component is $a_{i^{(k)},n}$ should be as orthogonal as possible to the space generated by the recently used corresponding vectors.

For the implementation of ART, we need to specify the initial estimate $c^{(0)}$, the data access ordering function $i^{(k)}$, the relaxation parameter $\lambda^{(k)}$, and finally, we have to decide when to stop the iterative process.

### 1.2.2 Basis Functions

The choice of the set of basis functions greatly influences the result of the reconstruction algorithm [41, 42, 49]. The conventional choice for the basis functions $b_n(\vec{x})$ is the voxel basis functions, which are defined as follows. Given an image $f : \mathbb{R}^3 \rightarrow \mathbb{R}$, we cover its support by an array of voxels that are arranged so that each voxel face that is not on the boundary of the array is shared exactly by two neighboring voxels. Selecting $N$ to be the number of such voxels and using $\vec{x}_n \in \mathbb{R}^3$ to denote the center of the $n$th voxel, the voxel basis functions, for $1 \leq n \leq N$, are defined to be

$$b_n(\vec{x}) = v_\sigma (\vec{x} - \vec{x}_n),$$

(1.5)
with $v_\sigma : \mathbb{R}^3 \to \mathbb{R}$ defined as follows. Let $x_1$, $x_2$ and $x_3$ be the components of $\vec{x}$ and let $d$ denote the number of elements in the set $\{|x_1|, |x_2|, |x_3|\}$ whose value is $\sigma/2$. Then

$$v_\sigma(\vec{x}) = \begin{cases} 
1/2^d, & \text{if } \max\{|x_1|, |x_2|, |x_3|\} \leq \sigma/2, \\
0, & \text{otherwise},
\end{cases}$$

(1.6)

where $\sigma$ is the distance of a voxel’s center to the centers of its neighbors. This strange definition provides reasonable values at the voxel boundaries. The values assigned are 1 in the interiors of voxels, $1/2$ on the faces, $1/4$ on the edges and $1/8$ at the corners. To complete the specification of the representation of $f$ using voxel basis functions, we may define the $c_n$ in (1.1) to have the values $f(\vec{x}_n)$.

It was pointed out by Lewitt [41, 42] that, due to the cubical shape of their supports and the discontinuity in their values at the boundaries of their supports, voxel basis functions do not appear to be appropriate for efficacious representation of biomedical objects. He recommended that one should use instead what we in this research (in common with many others) refer to as blobs, which are spherically symmetric continuously differentiable functions with overlapping supports; they are generalizations of a well-known class of functions used in digital signal processing called Kaiser-Bessel window functions. The appropriateness of this recommendation has been demonstrated in many publications since; an early example is [47].

Matej and Lewitt [48, 49] provided a careful investigation of how the blob basis functions should be chosen when they are used in the context of 3D (three-dimensional) image reconstruction. Since then blobs have been used extensively for 3D image reconstruction in X-ray computed tomography [35], positron emission tomography [11, 17], single photon emission computed tomography [1, 60, 63], and electron microscopy [5, 20, 44, 54].

Note that the voxel basis function is uniquely defined by a single parameter
which is the size of the voxel (the distance of a voxel’s center to the centers of its neighbors). The definition of the blob basis functions, to be given in the next chapter, is more complex since it involves additional parameters that allow the user to control the characteristics of the image representation. Two of these are named $a$ and $\alpha$, where $a$ is the radius of the support of the blob and $\alpha$ controls the blob’s shape. It was demonstrated by Matej and Lewitt [49] that in order to be able to approximate closely a nonzero-constant-valued function by a representation of the form (1.1), with the $b_n$ being blob basis functions, the parameters $a$ and $\alpha$ should satisfy a particular quadratic equation. A pair of $a$ and $\alpha$ that approximately satisfy this equation became known as the standard values and have been used in research papers such as [5, 20] and software packages such as Xmipp\(^1\). In this research, we will push further the analysis of [49] and so obtain pairs of simultaneous quadratic equations satisfaction of which results in better approximations to nonzero-constant-valued functions than what we get using the standard values. We also show that it is important to have very exact satisfaction of the equations, even small perturbations in the parameter values can result in unsatisfactory representations. We also discuss, by making use of an extra degree of freedom, which has been previously ignored, how to select from the options that are considered satisfactory for the representation of nonzero-constant-valued functions the parameters that also blur edges the least.

1.3 3D Grids

The choice of the points $\vec{x}_n$ in (1.2) is pretty well determined by the nature of the voxel basis function and the desire to have a reasonable approximation to an arbitrary image by its digital representation. The nature of the blob basis functions

\(^1\)http://xmipp.cnb.csic.es/twiki/bin/view/Xmipp/Reconstruct_art_v3
allows us much more freedom. Nevertheless, the choice of the spatial arrangement of the set of points at which the blobs are placed is important for the efficacy of the blob basis functions for image representation and reconstruction. Three types of grids are of particular interest [25]: the sc, the bcc and the fcc grids, which we now proceed to define.

- A simple cubic (sc) grid is defined by

\[
G_\sigma = \{ (\sigma k_1, \sigma k_2, \sigma k_3) | k_1, k_2, k_3 \in \mathbb{Z} \}, \tag{1.7}
\]

where $\mathbb{Z}$ is the set of integers and $\sigma$ is a positive real number (the sampling distance). We see that the voxels in a 3D scene may be indexed by triples $k_1, k_2, k_3$. We refer a part of a 3D scene that contains all voxels for which $k_3$ has a single fixed value as a slice.

- A body-centered cubic (bcc) grid is defined by

\[
B_\beta = \{ (\beta k_1, \beta k_2, \beta k_3) | \]
\[
k_1, k_2, k_3 \in \mathbb{Z} \text{ and } k_1 \equiv k_2 \equiv k_3 (\text{mod } 2) \}, \tag{1.8}
\]

where $\beta$ is a positive real number.

- A face-centered cubic (fcc) grid is defined by

\[
F_\phi = \{ (\phi k_1, \phi k_2, \phi k_3) | \]
\[
k_1, k_2, k_3 \in \mathbb{Z} \text{ and } k_1 + k_2 + k_3 \equiv 0 (\text{mod } 2) \}, \tag{1.9}
\]

where $\phi$ is a positive real number.

To visualize the above grids, we can use small portions of them and take advantage of their periodic repetitions. Figure 1.4 displays the three types of grids in a $2\sigma \times 2\sigma \times 2\sigma$, $2\beta \times 2\beta \times 2\beta$, and $2\phi \times 2\phi \times 2\phi$ portion of space.
It is discussed in [48] that the sc grids are not as efficient as the two other types of grids. We now give an interpretation of this statement. For this, we make use of Fourier transforms. Our particular choice for the definition of the Fourier transform \( \hat{f} \) of a function \( f : \mathbb{R}^3 \rightarrow \mathbb{R} \) is that, for all \( \vec{X} \in \mathbb{R}^3 \),

\[
\hat{f}(\vec{X}) = \int_{\mathbb{R}^3} f(\vec{x}) e^{-2\pi i \vec{x} \cdot \vec{X}} d\vec{x},
\]

(1.10)

where \( \vec{x} \cdot \vec{X} \) denotes the inner product of the vectors \( \vec{x} \) and \( \vec{X} \).

Let \( L \) be an arbitrary positive real number. The sampling theorem says that any function \( f \) over \( \mathbb{R}^3 \) that has the property that its Fourier transform is zero for all frequencies larger than \( L \) (that means that \( \hat{f}(\vec{X}) = 0 \), whenever the norm of the vector \( \vec{X} \) is greater than \( L \)) is uniquely determined by its samples at the points of \( G_{1/2L} \), but the same statement is not true for any sc grid \( G_{\sigma} \) with \( \sigma > 1/2L \). Similarly, one can derive (this can be done, for example, by using (1.12) below) that any function over \( \mathbb{R}^3 \) that has the property that its Fourier transform is zero for all frequencies larger than \( L \) is uniquely determined by its samples at the points of \( B_{1/2\sqrt{2}L} \), but the same statement is not true for any bcc grid \( B_\beta \) with \( \beta > 1/2\sqrt{2}L \). One way of expressing the contents of the last two sentences is to say that the grids \( G_{1/2L} \) and \( B_{1/2\sqrt{2}L} \) are
equivalent. More generally, since $L$ is an arbitrary positive real number, we can say that, for any $\sigma > 0$, the grids $G_\sigma$ and $B_{\sigma/\sqrt{2}}$ are equivalent. Now consider an arbitrary image $f : \mathbb{R}^3 \to \mathbb{R}$ and a bounded region of $\mathbb{R}^3$ that contains the support of $f$. As can be seen by studying figure 1.4, the number of grid points of the sc grid $G_\sigma$ that fall within that bounded region of $\mathbb{R}^3$ is approximately $\sqrt{2}$ times the number of grids points of the equivalent bcc grid $B_{\sigma/\sqrt{2}}$ that fall within it. Thus fewer samples are needed by a bcc grid than by the equivalent sc grid. In the same sense, the fcc grids are more efficient than the sc grids, but are not as efficient as the bcc grids [48]. Such considerations led to the suggestion of using the bcc grid as the set of points at which blobs are placed. This suggestion has been repeatedly validated in practice and has been widely adopted; we will follow it in this research as well.

In order to analyze the consequences of this decision on the appropriate choice of blob parameters, we move beyond using just ordinary functions on $\mathbb{R}^3$, but make use of generalized functions (also called distributions) as well. In particular, following the standard $\mathcal{S}$ notation [7], we make the following definition. If $P$ is a set of points in $\mathbb{R}^3$, then the distribution of $P$ is defined as

$$\Pi_P(\vec{x}) = \sum_{\vec{p} \in P} \delta(\vec{x} - \vec{p}),$$  

(1.11)

where $\delta$ is the Dirac delta distribution (i.e., an impulse of unit strength at the origin of $\mathbb{R}^3$). In other words, $\Pi_P$ denotes the distribution that one obtains by placing impulses (of unit-strength) at the points of $P$.

Generalized functions have Fourier transforms that behave in a manner analogous to their behavior for ordinary functions [7]. The Fourier transform of the Dirac delta distribution $\delta$ is the ordinary function $\hat{\delta} : \mathbb{R}^3 \to \mathbb{R}$ such that, for all $\vec{x} \in \mathbb{R}^3$, $\hat{\delta} (\vec{x}) = 1$. The Fourier transform $\hat{\Pi}_{B_\beta}$ of $\Pi_{B_\beta}$ can be shown to be

$$\hat{\Pi}_{B_\beta} = \frac{1}{4\beta^3} \Pi_{F_{1/2\beta}},$$  

(1.12)
see, for example, [25]. In words, this means that the Fourier transform of the distribution of the bcc grid $B_\beta$ is the distribution of the fcc grid $F_{1/2\beta}$ multiplied by the constant $1/4\beta^3$.

Some ordinary functions $f$ whose Fourier transforms $\hat{f}$ do not exist as an ordinary function defined by (1.10) can nevertheless have a Fourier transform that is a generalized function. For example, the Fourier transform of any nonzero-constant-valued ordinary function over $\mathbb{R}^3$ is an impulse (a nonzero multiple of $\delta$) at the origin; we shall be making essential use of this fact below.
Chapter 2

Desirable Blob Parameters

Blobs are superior to voxels for the estimation of the shapes of biological objects and allow for the fact that biological elements usually lack perpendicular edges [44]. They also account for overlap creating smooth transitions, a property useful for reconstruction of influenza virions with the large number of surface spikes. In this chapter we investigated the selection of blob parameters using an extra degree of freedom which has previously been ignored. Using that extra degree of freedom, we produced a family of blob parameters that accurately represent an object that has the same density everywhere. We then investigated how well different members of this family can represent a ball showing that there is a trade-off between the blurring of the edge of the ball and the magnitude of the oscillations inside the ball. In the next section we define blobs and study their Fourier transforms. In Section 2.2 we discuss the effects of attaching blobs to points of a grid and the nature of those blob parameters that allow us to approximate nonzero-constant-valued functions by blobs attached to a grid. In Section 2.3 we illustrate how well a piecewise-constant image can be represented depending on the blob parameters. Illustration of the effects of the blob parameters in image reconstruction from projections is provided in Section 2.4. We note that the material of this chapter appeared in a publication
that is written together with Stuart W. Rowland [4].

### 2.1 Blobs and Their Fourier Transforms

The general form of a basic blob $w_{m,a,\alpha} : \mathbb{R}^3 \rightarrow \mathbb{R}$ as given by Lewitt [41] is

$$b(m,a,\alpha;r) = \begin{cases} \frac{I_m(\alpha\sqrt{1-(\frac{r}{a})^2})}{I_m(\alpha)} \left[ \sqrt{1-(\frac{r}{a})^2} \right]^m, & \text{if } 0 \leq r \leq a, \\ 0, & \text{otherwise,} \end{cases} \tag{2.1}$$

where $r$ denotes the norm $\|\vec{x}\|$ of the vector $\vec{x}$, $I_m$ denotes the modified Bessel function of order $m$, $a > 0$ is the radius of the spherical support of the blob and $\alpha$ is a parameter controlling the blob shape. Figure 2.1 plots the two versions of such a blob as a function of $r$, one for the so-called standard parameters $m = 2$, $a = 2\sigma$ and $\alpha = 10.4$ (in green) and the other for the so-called recommended parameters $m = 2$, $a = 2.453144\sigma$ and $\alpha = 13.738507$ (using red) with $\sigma = 1$; the reason for this terminology (standard versus recommended) is discussed in the last paragraph of Section 2.2. Due to its close similarity to the green graph, the plot of the blob using a more accurate version of the standard parameters is not included in figure 2.1; the importance of using the accurate values for the blob parameters is demonstrated in Section 2.3.

When using blobs as the basis functions $b_n$ in (1.1), we first select the three blob parameters $m$, $a$ and $\alpha$ in (2.1) and then define $b_n$ by shifting the center of the support of $w_{m,a,\alpha}$ to a point $\vec{x}_n$. For reasons discussed in the preceding section, the points $\vec{x}_n$ come from a bcc grid $B_\beta$. Given an image $f$ and region of space that contains its support, we first select the sampling distance $\beta$ that is appropriately small for what we wish to see in a permissible representation of $f$ and then we select the $\vec{x}_n \in \mathbb{R}^3$, for $1 \leq n \leq N$, to be the set of all grid points in $B_\beta$ that are in the given region of space.
The three blob parameters $m$, $a$ and $\alpha$ need to be set carefully in order to obtain good image representations using blobs as the basis functions. The parameter $m$ controls the continuity of the blob: for $m > 0$, the blob is a continuous function with $m - 1$ continuous derivatives [41]. It has been common practice in the literature using blobs to choose $m = 2$. Such blobs are smooth functions with continuous first derivatives. The extra smoothness with larger values of $m$ does not appear to result in better representations, but seems to increase the computational cost of obtaining representations of similar quality. Therefore, choosing $m = 2$ appears to be reasonable and we do that in this research as well. However, if in the future it appears desirable to revisit the choice of $m$, the methodology presented below for obtaining the appropriate $a$ and $\alpha$ for $m = 2$ is also applicable for other values of $m$.

For that methodology we need to make use of the Fourier transform $\hat{w}_{m,a,\alpha}$ of the basic blob $w_{m,a,\alpha}$ over $\mathbb{R}^3$ that is defined in (2.1). According to Lewitt [41],
Figure 2.2: Plot of the Bessel function $J_{7/2}$.

$$\hat{w}_{m,a,\alpha}(\vec{X}) = \begin{cases} 
\frac{(2\pi)^{\frac{3}{2}}a^3\alpha^m}{I_m(\alpha)} \frac{I_{\frac{3}{2}+m}(\sqrt{\alpha^2-(2\pi a R)^2})}{(\sqrt{\alpha^2-(2\pi a R)^2})^{\frac{3}{2}+m}}, & \text{if } 2\pi a R \leq \alpha, \\
\frac{(2\pi)^{\frac{3}{2}}a^3\alpha^m}{I_m(\alpha)} \frac{J_{\frac{3}{2}+m}(\sqrt{(2\pi a R)^2-\alpha^2})}{(\sqrt{(2\pi a R)^2-\alpha^2})^{\frac{3}{2}+m}}, & \text{otherwise},
\end{cases} \tag{2.2}$$

where $R$ denotes the norm of the vector $\vec{X}$ and $J$ denotes the Bessel function of the first kind.

We will make use of the roots (zero crossings) of $\hat{w}_{m,a,\alpha}$ for our choice of $m = 2$. We see that $\hat{w}_{2,a,\alpha}(\vec{X}) = 0$ if, and only if, $J_{7/2} \left( \sqrt{(2\pi a R)^2-\alpha^2} \right) = 0$, since $I_{7/2}$ has no roots. Figure 2.2 plots the function $J_{7/2}$ and Table 2.1 lists the values of its first nine roots rounded to six places after the decimal point.
Table 2.1: First nine roots (zero crossings) of $J_{7/2}$ with $j_r$ denoting the $r$th root.

<table>
<thead>
<tr>
<th>$j_x$</th>
<th>Root</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J_1$</td>
<td>6.987932</td>
</tr>
<tr>
<td>$j_2$</td>
<td>10.417119</td>
</tr>
<tr>
<td>$j_3$</td>
<td>13.698023</td>
</tr>
<tr>
<td>$j_4$</td>
<td>16.923621</td>
</tr>
<tr>
<td>$j_5$</td>
<td>20.121806</td>
</tr>
<tr>
<td>$j_6$</td>
<td>23.304247</td>
</tr>
<tr>
<td>$j_7$</td>
<td>26.476764</td>
</tr>
<tr>
<td>$j_8$</td>
<td>29.642605</td>
</tr>
<tr>
<td>$j_9$</td>
<td>32.803732</td>
</tr>
</tbody>
</table>

2.2 Approximation of Constant-Valued Functions

Using Blobs

Since many biomedical objects are approximately piecewise constant, it appears at first sight that it may be difficult to approximate them with a permissible representation using blob basis functions. After all, blobs have “humps” at their centers (see figure 2.1), how can we superimpose such functions to get something that is nearly constant?

Following [49], we translate this challenge into the following: Given $\beta$ and a real number $\gamma$, can we select $a$ and $\alpha$ in (2.1) and a real number $s$ in such a way that, if we define the function $g_s : \mathbb{R}^3 \to \mathbb{R}$ by

$$g_s(\vec{x}) = \sum_{\vec{x} \in B_{\beta}} sw_{2,a,\alpha}(\vec{x} - \vec{x}') = s \sum_{\vec{x} \in B_{\beta}} w_{2,a,\alpha}(\vec{x} - \vec{x}') ,$$

then the value of $g_s(\vec{x})$ will be near to the constant $\gamma$, for all $\vec{x} \in \mathbb{R}^3$? This can be trivially achieved if $\gamma = 0$, by setting $s = 0$. Now assume that $\gamma \neq 0$. If it is the case that the value of $g_1(\vec{x})$ is near to a nonzero constant $\gamma_1$, for all $\vec{x} \in \mathbb{R}^3$, then, clearly, the value of $g_{\gamma/\gamma_1}(\vec{x})$ will be near to the constant $\gamma$, for all $\vec{x} \in \mathbb{R}^3$. So, to meet the challenge, it is sufficient to show that the value of $g_1(\vec{x})$ will be near to
a nonzero constant $\gamma_1$, for all $\vec{x} \in \mathbb{R}^3$. We are now going to that. To simplify the notation we will abbreviate $g_1$ as $g$. Mathematically, the function $g$ can be defined as the convolution between the distribution of the bcc grid $B_\beta$ and the blob $w_{2,a,\alpha}$:

$$g = B_\beta \ast w_{2,a,\alpha}.$$

(2.4)

Recalling the last paragraph of Section 1.3, we see that the aim that $g$ should be a nonzero-constant-valued function is mathematically identical to the aim that its Fourier transform should be an impulse at the origin. By the convolution theorem for Fourier transforms, it follows from (2.4) and (1.12) that

$$\hat{g} = \frac{1}{4\beta^3} \Pi_{F_{1/2}\beta} \times \hat{w}_{2,a,\alpha},$$

(2.5)

which is the product of impulses at points of an fcc grid and the Fourier transform $\hat{w}_{2,a,\alpha}$ of the blob $w_{2,a,\alpha}$. The former has an impulse at the origin and the value of the latter is not zero at the origin, hence $\hat{g}$ is put together from an impulse at the origin and impulses at the other points of the fcc grid $\Pi_{F_{1/2}\beta}$ multiplied by the value of $\hat{w}_{2,a,\alpha}$ at those points. Whenever the value of $\hat{w}_{2,a,\alpha}$ is zero at an fcc grid point, it annihilates the contribution of the impulse at that grid point.

Looking at figure 1.4(c) we note that in the fcc grid $F_\phi$ there are twelve nearest neighbors to the origin, all at the distance $R_1 = \sqrt{2} \phi$ from it. Further, there are six second-nearest neighbors, all at the distance $R_2 = 2\phi$ from the origin. We generalize this notation and use $R_i$ to denote the distance from the origin of the $i$th nearest neighbors to the origin; defining, for completeness, $R_0 = 0$. Further, we introduce the notation that for all integers $i \geq 0$, $q_i$ denotes the value of $\hat{w}_{m,a,\alpha}(\vec{X})$ for the vectors $\vec{X}$ whose norm is $R_i$. If we could find an $a$ and $\alpha$ such that $q_1 = q_2 = 0$, then we would annihilate the contributions to $\hat{g}$ of the impulses at the eighteen grid points of $F_\phi$ that are nearest to the origin. Recalling that $\hat{w}_{2,a,\alpha}(\vec{X}) = 0$ if, and
only if, \( J_{7/2} \left( \sqrt{(2\pi aR)^2} - \alpha^2 \right) = 0 \), and using the notation of Table 2.1 according to which \( j_{r_1} \) and \( j_{r_2} \) are the \( r_1 \)th and \( r_2 \)th roots of \( J_{7/2} \), respectively, we see that we need to find values of \( a \) and \( \alpha \) that satisfy

\[
\sqrt{(2\pi aR_1)^2} - \alpha^2 = j_{r_1} \tag{2.6}
\]

and

\[
\sqrt{(2\pi aR_2)^2} - \alpha^2 = j_{r_2}. \tag{2.7}
\]

It is easy to show that, provided that \( r_1 < r_2 \), the solution to this problem is

\[
a = \frac{1}{2\pi} \sqrt{\frac{J_{r_2}^2 - J_{r_1}^2}{R_2^2 - R_1^2}} \tag{2.8}
\]

and

\[
\alpha = \sqrt{\frac{R_1^2 J_{r_2}^2 - R_2^2 J_{r_1}^2}{R_2^2 - R_1^2}}. \tag{2.9}
\]

Recalling that the discussion of the previous paragraph is to be applied to the fcc grid \( F_{1/2\beta} \) for which \( R_1 = 1/\sqrt{2\beta} \) and \( R_2 = 1/\beta \), we get:

\[
a = \frac{\beta}{\pi \sqrt{2}} \sqrt{J_{r_2}^2 - J_{r_1}^2}, \tag{2.10}
\]

\[
\alpha = \sqrt{J_{r_2}^2 - 2J_{r_1}^2}. \tag{2.11}
\]

Replacing \( j_{r_1} \) and \( j_{r_2} \) in (2.10) and (2.11) with the values in Table 2.1 we get the values for \( a \) and \( \alpha \), rounded to six places after the decimal point, shown in Table 2.2.

The value of the root \( j_{r_1} \) is fixed to \( j_1 \) in Table 2.2 for all entries; it can be any value as long as \( r_1 < r_2 \); the explanation for restricting \( r_1 \) to the value 1 is given
Table 2.2: Desirable blob parameters for various pairs of roots of $J_{7/2}$.

<table>
<thead>
<tr>
<th>$j_{r_1}, j_{r_2}$</th>
<th>$a$</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j_{1}, j_{2}$</td>
<td>$\beta 1.738875$</td>
<td>3.294537</td>
</tr>
<tr>
<td>$j_{1}, j_{3}$</td>
<td>$\beta 2.651778$</td>
<td>9.485434</td>
</tr>
<tr>
<td>$j_{1}, j_{4}$</td>
<td>$\beta 3.469269$</td>
<td>13.738507</td>
</tr>
<tr>
<td>$j_{1}, j_{5}$</td>
<td>$\beta 4.247117$</td>
<td>17.527826</td>
</tr>
<tr>
<td>$j_{1}, j_{6}$</td>
<td>$\beta 5.003932$</td>
<td>21.105107</td>
</tr>
<tr>
<td>$j_{1}, j_{7}$</td>
<td>$\beta 5.748062$</td>
<td>24.563319</td>
</tr>
<tr>
<td>$j_{1}, j_{8}$</td>
<td>$\beta 6.483890$</td>
<td>27.946764</td>
</tr>
<tr>
<td>$j_{1}, j_{9}$</td>
<td>$\beta 7.213964$</td>
<td>31.279745</td>
</tr>
</tbody>
</table>

in Section 2.3. Any of the pairs $j_{r_1}, j_{r_2}$ (as well as any other pairs of roots of $J_{7/2}$) results in a $\hat{g}$ for which the contributions of the impulses at the eighteen grid points of $F_{1/\beta}$ that are nearest to the origin are eliminated. That still leaves us with the nonzero contributions of the impulses at grid points that are further away from the origin. However, those contributions are multiplied by the value of $\hat{w}_{2,a,\alpha}$ at those grid points. By observing (2.2) we see that $\hat{w}_{2,a,\alpha}(\vec{X})$ decreases rapidly as $R$ increases, and so the contribution of these impulses to $\hat{g}$ can be considered to be negligible. Table 2.3 illustrates this by showing that relative to $q_0$ the values of $q_3, q_4$ and $q_5$ are negligibly small for $j_1, j_{r_2}$ with $r_2 \geq 4$. Therefore, if we select the blob parameters from Table 2.2 from a row with $r_2 \geq 4$, the resulting function $\hat{g}$ will be the sum of impulses at points of an fcc grid, with the strength of the impulse at any point other than the origin being many orders of magnitude smaller than the strength of the impulse at the origin. Thus, $\hat{g}$ will look essentially like an impulse at the origin and so its inverse Fourier transform will be nearly a nonzero-constant-valued function.

The desirable blob parameters obtained in this section were designed for blobs to be attached to points of the bcc grid with sampling distance $\beta$. Similar analysis can be carried out to derive desirable parameters for blobs to be attached to points of the sc grid $G_\sigma$ with sampling distance $\sigma$. From [25], the Fourier transform of $\|G_\sigma$ is
Table 2.3: Values $q_3/q_0$, $q_4/q_0$, and $q_5/q_0$ of $\hat{w}_{2,a,a}(R)/\hat{w}_{2,a,a}(0)$ at distances $R_3 = \sqrt{6}/2\beta$, $R_4 = \sqrt{10}/2\beta$ and $R_5 = 2/\beta$, respectively, from the origin. The $a$ and $\alpha$ are as in Table 2.2 with $\beta = 1/\sqrt{a}$.

Thus, the blob parameters $a$ and $\alpha$ can be obtained from (2.8) and (2.9) by replacing $R_1$ and $R_2$ with $1/\sigma$ and $\sqrt{3}/\sigma$, respectively; the former value is the distance from the origin of the six nearest neighbors and the latter is the location of the twelve second-nearest neighbors as can be seen from figure 1.4(a).

The question that now arises, what is the actual value of the nonzero constant $\gamma_1$ such that the value of $g(\vec{x}) = g_1(\vec{x})$ will be near to $\gamma_1$, for all $\vec{x} \in \mathbb{R}^3$? For this we consider the average, $\bar{g}$, of the function $g$ in the Voronoi neighborhoods in $B_{\beta}$; the Voronoi neighborhood, $\mathcal{V}$, of a point of $B_{\beta}$ comprises the set of points in $\mathbb{R}^3$ that are not nearer to another point in $B_{\beta}$. It is easy to work out, based on figure 1.4(b), that the volume, $\Delta$, of the Voronoi neighborhood $\mathcal{V}$ is $4\beta^3$. From this it follows that
$\gamma_1$ is approximately the same as

$$
\bar{g} = \frac{1}{\Delta} \int g(\vec{x}) d\vec{x}
= \frac{1}{4\beta^3} \sum_{\vec{x} \in B_\beta} \int w_{2,a,\alpha} (\vec{x} - \vec{x}') d\vec{x}'
= \frac{1}{4\beta^3} \int_{\mathbb{R}^3} w_{2,a,\alpha} (\vec{x}) d\vec{x}
= \frac{\hat{w}_{2,a,\alpha}(\vec{0})}{4\beta^3},
$$

(2.13)

where $\vec{0}$ denotes the origin. Now observe from (2.2) that $\hat{w}_{2,a,\alpha}(\vec{0})$ is proportional to $a^3$, which is proportional to $\beta^3$ by (2.10). It follows therefore that the value of $\bar{g}$ does not depend on the choice of $\beta$. Its actual value when $r_1 = 1$ and $r_2 = 4$ is 2.369230, as can be obtained from the formulas for $a$ (2.10) and $\alpha$ (2.11).

We complete this section by a discussion of the justification of choosing the so-called standard values for $a$ and $\alpha$ (already referred to in Section 1.2.2) that were proposed by Matej and Lewitt [49] and have been adopted by others since; see, for example, [5, 20]. The requirement that $\hat{w}_{2,a,\alpha}$ should annihilate the impulses at the twelve grid points in $\Pi_{F_1/2\beta}$ nearest to the origin at a distance $R_1 = 1/\sqrt{2}\beta$ from it can be achieved, as discussed above, by choosing values of $a$ and $\alpha$ that satisfy (2.6), which can be rewritten for the case $r_1 = 1$ as

$$
\alpha = \sqrt{2\pi^2 \left( \frac{a}{\beta} \right)^2 - f_1^2}.
$$

(2.14)

Matej and Lewitt [49] considered it reasonable (and validated this by some reconstruction experiments) that if the result of the reconstruction is going to be displayed on an sc grid $G_\sigma$, then we should use blobs for which the radius of their support is $a = 2\sigma$. Also, as discussed in Section 1.3, $\beta = \sigma/\sqrt{2}$ for the bcc grid $B_\beta$ that is equivalent to $G_\sigma$ and is thus the one that we use for representation by blob basis.
functions. Substituting these values of $a$ and $\beta$ into (2.14), the $\sigma$ cancels out and we get, to six decimal accuracy, that $\alpha = 10.444256$. The standard values that are stated in the above-cited literature for these blob parameters are $a = 2$ (obtained by assuming that $\sigma = 1$) and $\alpha = 10.4$ (obtained by rounding the value provided by (2.14) to one place after the decimal point). As we will demonstrate in the following sections, there is a loss of accuracy when using these standard parameters for image representation or for image reconstruction as compared to the parameters that we are recommending in this research.

To avoid any possible confusion, we now make precise the uses in this research of the adjectives “standard” and “recommended”. These adjectives apply to both the choices of the blob parameters and the set of blob basis functions. We make the simplifying assumption that the representation is going to be displayed on an sc grid $G_\sigma$ with $\sigma = 1$; for alternate values of $\sigma$, other parameters will have to be scaled appropriately. Both in the standard and recommended cases, the centers of the blobs are placed at the points of the bcc grid $B_\beta$ with $\beta = 1/\sqrt{2}$ and $m = 2$. As discussed in the previous paragraph, for the standard case, $a = 2$ and $\alpha = 10.4$. The parameters for the recommended case are obtained by choosing the pair $j_1, j_4$ in Table 2.2, which results in $a = 3.469269$, $\beta = 2.453144$ and $\alpha = 13.738507$. (The basic blobs provided by these two choices are plotted in figure 2.1.) We have already seen reasons for the choice of the word “recommended” and further justification is provided in what follows. The adjectives standard and recommended are applied not only to the choice of the parameters but also to the resulting set of blobs to be used in the representation (and in reconstruction).
2.3 Representation of Images Using Blobs with Desirable Parameters

We now turn to discussing the representation using blob basis functions of images \( f : \mathbb{R}^3 \to \mathbb{R} \) as defined at the beginning of Section 1.2. Since images have bounded support, we may assume that there is a positive real number \( S \) such that if we define \( S_S \subset \mathbb{R}^3 \) by

\[
S_S = \{ \vec{x} \in \mathbb{R}^3 \mid \max \{|x_1|, |x_2|, |x_3| \} \leq \frac{S}{2} \}, \tag{2.15}
\]

then \( f(\vec{x}) = 0 \) if \( \vec{x} \notin S_S \). We note that, for any positive real number \( \beta \), the set \( B_\beta \cap S_S \) is finite; we denote by \( N \) the number of elements in it and use the notation \( \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N \) to enumerate all the elements of \( B_\beta \cap S_S \). The \( N \) basis functions to be used in the permissible image representations will be obtained by shifting a basic blob \( w_{2,a,\alpha} \) to the \( \vec{x}_n \), for \( 1 \leq n \leq N \).

In this section we discuss image representation using blobs of piecewise-constant images; i.e., images that have constant values over regions of space that are large as compared to the support of blobs (spheres of radius \( a \), see (2.1)). For any such region \( C \), we define its eroded interior to consist of all \( \vec{x} \in C \), such that the ball of radius \( a \) centered at \( \vec{x} \) is entirely within \( C \). Clearly, the value of the representation at any point in the eroded interior of \( C \) depends only on those coefficients that are assigned to basis functions that are centered at grid points in \( C \) (i.e., at points in the set \( C \cap B_\beta \)). Based on the discussion in the previous section, it follows from this statement that in order to get an approximately constant value within the eroded interior of \( C \), it suffices to assign a constant coefficient \( s \) to every basis function that is centered at a grid point in \( C \). We define, for pairs \((r_1, r_2)\) of integers such that
0 < r_1 < r_2, functions \( h_{r_1, r_2} : \mathbb{R}^3 \to \mathbb{R} \) by

\[
h_{r_1, r_2}(\vec{x}) = \sum_{\vec{x}_n \in C \cap B_\beta} sw_{2, a, \alpha} (\vec{x} - \vec{x}_n) = s \sum_{\vec{x}_n \in C \cap B_\beta} w_{2, a, \alpha} (\vec{x} - \vec{x}_n),
\]

(2.16)

where each pair \((r_1, r_2)\) generates the parameters \(a\) and \(\alpha\), according to (2.10) and (2.11); see, for example, Table 2.2. Comparison of this definition with (2.3) tells us that, for every \(\vec{x}\) in the eroded interior of \(C\), \(h_{r_1, r_2}(\vec{x}) = g_\delta(\vec{x})\). Suppose that, in the image that we are trying to represent, the constant value in the region \(C\) is \(\gamma\). The assignment of the coefficient \(s = \gamma/\bar{g}\), where \(\bar{g}\) is defined by (2.13), to every basis function that is centered at a grid point in \(C\) will result in the representation having approximately the correct value (i.e., \(\gamma\)) at all points in the eroded interior of \(C\), provided that the basic blob \(w_{2, a, \alpha}\) is selected according to the principles presented in the previous section for representing constant-valued functions. A question that now arises: How should the blob parameters be selected so that the approximation is also reasonably correct in the part of \(C\) that is not in the eroded interior?

We illustrate our answer to this question by approximating a ball with radius \(\rho = 60\); that is the image \(\Pi : \mathbb{R}^3 \to \mathbb{R}\) defined by

\[
\Pi(\vec{x}) = \begin{cases} 
1, & \text{if } ||\vec{x}|| \leq 60, \\
0, & \text{otherwise.}
\end{cases}
\]

(2.17)

For the rest of this section we adopt (without repeatedly pointing this out) the choices \(\sigma = 1\) and \(\beta = 1/\sqrt{2}\) (these are the same that were made in the last paragraph of Section 2.2). As a consequence, we have that \(R_1 = 1/\sqrt{2} \beta = 1\) and \(R_2 = 1/\beta = \sqrt{2}\).

Also we fix, except where it is otherwise stated, \(r_1 = 1\) and we investigate the consequences of the choice \(r_2\) on how well \(\Pi\) is approximated using (2.16) with \(C = \{ \vec{x} \in \mathbb{R}^3 \mid ||\vec{x}|| \leq 60 \}\) and \(s = 1/\bar{g}\). The \(h_{r_1, r_2}\) of (2.16) are indeed permissible representations as defined in (1.1); just select the \(S\) of (2.15) to be greater than 120.
and, for $1 \leq n \leq N$, set $b_n(\vec{x}) = w_{2,a,\alpha}(\vec{x} - \vec{x}_n)$ and $c_n$ to $1/\tilde{g}$ if $\vec{x}_n \in \mathbb{C}$ and to 0 otherwise.

In figure 2.3(a) we show the values of $\Pi$ at those points of $G_1$ for which $k_3 = 0$. The other eight images in figure 2.3 are values at the corresponding points of the eight representations $h_{1,r_2}$ of (2.16) that are obtained when $r_2$ is selected to be $2, 3, \ldots, 9$, respectively. The real numbers provided by these representations are translated into gray values in the images in figure 2.3 by setting all values below 0.999 to black and all values above 1.001 to white, with a linear translation in-between. This narrow window was selected so that the errors in the approximations are more visible. The red circle in each image indicates the location of the ball’s circumference; they are displayed to evaluate the edge blurring effect for each approximation. The images indicate that the representations using $r_2 = 2$ or $r_2 = 3$ are inaccurate, but all representations with $r_2 \geq 4$ appear to be acceptable. This conclusion is similar to the one we drew previously based on Table 2.3. To distinguish between the images using $r_2 \geq 4$, we plotted the values of the phantom (2.17) and the blob representations $h_{1,r_2}$, using $r_2 = 4$ and $r_2 = 9$, along the line $(r, 0, 0)$ for $54 \leq r \leq 66$ in figure 2.4(a). The plots of the representations using $r_2 = 5$ to $r_2 = 8$ lie between these two graphs and are not shown. Clearly, using $r_2 = 4$ captures the edge better than using $r_2 = 9$; the size of the gap between the approximation’s edge and the red circle, in figure 2.3, also illustrates this concept. On the other hand, the blob representation using $r_2 = 9$ is much smoother in the flat part ($r \leq 56$) than using $r_2 = 4$. This is illustrated in figure 2.4(b), where we see that the plot of the representation using $r_2 = 4$ has small oscillations in the flat part (the order of their magnitude is $10^{-5}$), but no such oscillations are apparent for $r_2 = 9$. Thus, both representations have advantages and disadvantages, but the inaccuracy of $r_2 = 4$ does not have a significant effect since the oscillations in the flat part are very small, to the extent that they do not interfere with human visual judgment; losing the sharp
Figure 2.3: Central slices of (a) the ball phantom and its blob representations using (b) $r_2 = 2$, (c) $r_2 = 3$, (d) $r_2 = 4$, (e) $r_2 = 5$, (f) $r_2 = 6$, (g) $r_2 = 7$, (h) $r_2 = 8$ and (i) $r_2 = 9$. The blob parameters $a$ and $\alpha$ are as in Table 2.2 with $\beta = 1/\sqrt{2}$. The red circles indicate the location of the ball’s circumference. Display thresholds are 0.999 and 1.001.
edge in the representation using $r_2 = 9$ may matter more. For the sake of comparison with blobs using standard parameters, similar plots are displayed in figures 2.5(a) and 2.5(b) with $h_{std}$ used to denote the blob representation in (2.16) using the standard parameters $a = 2\sigma$ and $\alpha = 10.4$. In terms of edge preservation, we notice from the plots that the blob representation using standard parameters is slightly better than using the recommended parameters, but the oscillations in the flat part of the former representation are significantly larger than in the latter one.

The choice of the recommended parameters, as discussed in this research, depends on a trade-off between preservation of the edges of the objects to be represented (or reconstructed) and of the smoothness of the homogeneous parts of the objects. In general, the choice depends on the type of the application at hand. For example, if the goal of the application is segmenting anatomical structures or detecting lesions, then the object edges will be more important than their uniform parts, in this case we may recommend blob parameters obtained from low zero-crossing orders (even the pairs $j_1, j_2$ or $j_1, j_3$ in Table 2.2, which may outperform the standard parameters with regard to preservation the edges); but if the uniform parts of the object are meaningful for the application, we may recommend blob parameters obtained from high zero-crossing orders. A relevant application for the latter is industrial computerized tomography (see, for example, [52]) for the detection of undesirable inhomogeneities in manufactured machine parts.

We now provide an explanation, using Fourier analysis, for the different behaviors of the two representations that are illustrated in figure 2.4. For the ball, the sum in (2.16) can be rewritten as

$$\sum_{\vec{x}_n \in \mathcal{C} \cap \mathbb{B}_\beta} w_{2,a,\alpha}(\vec{x} - \vec{x}_n) = \left[ (\Pi \times \mathbb{B}_{\beta}) * w_{2,a,\alpha} \right](\vec{x}), \quad (2.18)$$
Figure 2.4: Plot of the ball phantom (blue) and the blob representations using $r_2 = 4$ (red) and $r_2 = 9$ (black) along the line $(r, 0, 0)$. The blob parameters $a$ and $\alpha$ are as in Table 2.2 with $\beta = 1/\sqrt{2}$. 
Figure 2.5: Plot of the ball phantom (blue) and the blob representations using the recommended parameters $a = 2.453144$ and $\alpha = 13.738507$ (red) and the standard parameters $a = 2$ and $\alpha = 10.4$ (green) along the line $(r,0,0)$. 
and thus,

\[ h_{r_1, r_2}(\vec{x}) = \left[ (\Pi \times \Pi_{B_{\beta}}) * \frac{1}{g} w_{2,a,\alpha} \right](\vec{x}). \tag{2.19} \]

Taking the Fourier transform of both sides, we get

\[ \hat{h}_{r_1, r_2}(\vec{x}) = \left[ \hat{\Pi} * \left( \frac{1}{4\beta^3} \Pi_{F_{1/2\beta}} \right) \right](\vec{x}) \times \frac{1}{g} \hat{w}_{2,a,\alpha}(\vec{x}). \tag{2.20} \]

Moving the weighting factor of the distribution \( \Pi_{F_{1/2\beta}} \) to the second term of the multiplication in (2.20) leads to

\[
\begin{align*}
\hat{h}_{r_1, r_2}(\vec{x}) &= \left[ \hat{\Pi} * \Pi_{F_{1/2\beta}} \right](\vec{x}) \times \frac{1}{g} \hat{w}_{2,a,\alpha}(\vec{x}) \times \frac{1}{4\beta^3} \\
&= \left[ \hat{\Pi} * \Pi_{F_{1/2\beta}} \right](\vec{x}) \times \frac{\hat{w}_{2,a,\alpha}(\vec{x})}{\hat{w}_{2,a,\alpha}(\vec{0})}; \tag{2.21}
\end{align*}
\]

the second equality follows from (2.13).

To explain the influence of the choice of the blob parameters on the ability of the representation to reproduce edges as illustrated in figure 2.4(a), we plot in figure 2.6 the first term of the right hand side of (2.21), along with the second term for both representations \( r_2 = 4 \) and \( r_2 = 9 \). The blue graph corresponds to the Fourier transform of the sampled ball that comprises repeats of the Fourier transform of the ball \( \Pi \) centered at points of an fcc grid \( F_{1/2\beta} \). The main idea of the analysis is that we wish to eliminate the repeats (except for the one centered at the origin, which is the true Fourier transform of \( \Pi \)), but at the same time we wish to attenuate the oscillating values near to the origin as little as possible, because those values carry information that is important for reproducing the edge of the ball. The difference between the compared blobs comes from the order of the zero crossing used to eliminate repeats at the second nearest neighbors, the red graph uses the fourth zero crossing \( j_4 \) and the black one uses the ninth zero crossing \( j_9 \). These graphs
Figure 2.6: Plot of the first term of the right hand side of (2.21) (blue with values on the left vertical axis) along with plots of the second term for the two representations using \( r^2 = 4 \) and \( r^2 = 9 \) and the representation using standard parameters (red, black and green, respectively, with values on the right vertical axis). The three graphs plot the values of these terms for the points \((R/\sqrt{2}, R/\sqrt{2}, 0)\). The blob parameters \( a \) and \( \alpha \) for \( r^2 = 4 \) and \( r^2 = 9 \) are as in Table 2.2 with \( \beta = 1/\sqrt{2} \); the standard parameters are \( a = 2 \) and \( \alpha = 10.4 \).

indicate that the edge is better preserved when using \( j_4 \) than when using \( j_9 \), because the values of the Fourier transform of \( \Pi \) near the origin are less attenuated when \( r^2 = 4 \). The second term of the right hand side of (2.21) for the representation using standard parameters is plotted as well; the green graph that corresponds to this representation explains why in figure 2.5(a) the edge is preserved slightly better when using the representation with the standard parameters than when using the representation with the recommended parameters.

Now we turn our attention to explaining the differences between the two representations in figure 2.4(b), for values of \( r \leq 56 \). For this purpose, in figure 2.7 we plot yet again the second term of the right hand side of (2.21) for the two representations, but using new scales that are more appropriate for our current discussion. These plots demonstrate that it is indeed the case, as it should be by design, that the Fourier transforms of the basic blobs have zero values at both \( R = R_1 = 1 \) and...
Figure 2.7: Plots of the second term of the right hand side of (2.21) (a) and its base 10 logarithm (b) for $r_2 = 4$ (red), for $r_2 = 9$ (black) and for the standard parameters (green). The blob parameters $a$ and $\alpha$ for $r_2 = 4$ and $r_2 = 9$ are as in Table 2.2 with $\beta = 1/\sqrt{2}$; the standard parameters are $a = 2$ and $\alpha = 10.4$. 
\( R = R_2 = \sqrt{2} \), with \( R_1 \) at the first zero crossing and \( R_2 \) at the fourth (for \( r_2 = 4 \)) or the ninth (for \( r_2 = 9 \)). In the uniform case, this would be considered sufficient, since in the Fourier transform of a sampled uniform function there is only an impulse that repeats and so, if we cancel out the values at the locations of the repeats, there is nothing left except an impulse at the origin. But in the Fourier transform of the sampled ball, it is the Fourier transform of the ball that repeats and the Fourier transform of the ball has large values near the origin (see figure 2.6). The repetitions of these values need to be zeroed out by the multiplication with the second term on the right hand side of (2.21). That the choice of \( r_2 = 9 \) is much better than the choice of \( r_2 = 4 \) from this point of view is clear from figure 2.7. It is this difference that results in oscillations in the latter case, but not in the former; see figure 2.4(b).

Similar considerations explain why the quality of the representations obtained using the so-called standard values for the blob parameters discussed in the paragraph containing (2.14) are inferior to the quality obtained using blobs with the parameters obtained above and also the reason why the quality of representations is very sensitive to using the exact blob parameters that we have derived above. The value of the Fourier transform of the ball is large at the origin; it is the volume of the ball. Due to sampling, such large values repeat at all points of the grid \( F_{1/2} \) in the first term of the right hand side of (2.21) and, unless the second term on the right hand side of (2.21) is extremely small at those locations, this will lead to oscillations in the representations.

The essential problem with calculating the blob parameters based on (2.14) alone is that such an approach eliminates only the repeats nearest to the origin and the amelioration due to the small size of the Fourier transform of the blob at the repeats second-nearest to the origin is not quite sufficient to get rid of the artifact due to the high value at those points of the repeated Fourier transform of the ball.

A similar argument explains the sensitivity of the quality of the representations
to the exact values of the blob parameters. Incorrect values of the blob parameters will lead to a shift in the zero crossings of the Fourier transform of the blob and this will result in non-negligible values at the repeat locations and will thus adversely affect the quality of the representation.

Figure 2.8 displays a central slice of three different representations of the ball adopting the same narrow window used in figure 2.3: (a) using our recommended parameters (parameters making the Fourier transform of the blob having zero crossings at the desired roots), (b) using the standard blob parameters \( (a = 2\sigma \text{ and } \alpha = 10.4) \) [49], and (c) using a more accurate version of the parameters provided by (2.14) \( (a = 2\sigma \text{ and } \alpha = 10.444256) \) with \( \sigma = 1 \). The oscillatory patterns in the second and third cases are due to the sampling error being less-well eliminated than by the blobs that we advocate in this research. The oscillations in (c) are a consequence of not eliminating the repeats that are second-nearest to the origin. The oscillations in (b) are more serious due to the additional rounding that is performed to obtain the so-called standard values for the blob parameters. The importance of using the exact values for the blob parameters had been demonstrated previously; for example, in [31], figure 6.4, which illustrates a serious deterioration in the quality of representation due to moving from an accuracy of four places after the decimal point to an accuracy of two places after the decimal point.

Recalling that the image representation we discussed in the beginning of this section concerns piecewise-constant images. Let us now consider an arbitrary image \( f : \mathbb{R}^3 \rightarrow \mathbb{R} \) and analyze the nature of its blob representation

\[
f_{r_1, r_2} (\vec{x}) = \left[ \left( f \ast \Pi_{B_{\theta}} \right) \ast \frac{1}{g} w_{2,a,a} \right] (\vec{x}), \tag{2.22}
\]

which is a generalization of (2.19) with \( \Pi \) replaced by \( f \). Just as we derived in
Figure 2.8: A central slice through three ball representations: (a) using our recommended parameters $a = 2.453144$ and $\alpha = 13.738507$, (b) using the standard parameters $a = 2$ and $\alpha = 10.4$ (these are obtained from (2.14) by rounding to one place after the decimal point), (c) using an improved version of the standard parameters $a = 2$ and $\alpha = 10.444256$ (obtained by rounding to six places after the decimal point). Display thresholds are 0.999 and 1.001.

(2.19)-(2.21), we can derive that

$$
\hat{f}_{r_1,r_2}(\vec{X}) = \left[ \hat{f} * \Pi F_{1/2\beta} \right](\vec{X}) \times \frac{\hat{w}_{2,a,\alpha}(\vec{X})}{\hat{w}_{2,a,\alpha}(\vec{0})}.
$$

(2.23)

We now analyze the quality of such a representation as a function of the choice of the pair $(r_1,r_2)$.

Considering (2.23) we see that the first term of the product on its right hand side consists of the Fourier transform of $f$ itself and its repetitions shifted to all points of the grid $F_{1/2\beta}$. As discussed in Section 2.2, the nearest neighbors to the origin in $F_{1/2\beta}$ are at a distance $1/\sqrt{2}\beta$ from it. From this it follows (see also figure 2.6) that in order for $f_{r_1,r_2}$ to be a good approximation of a bandlimited version of $f$, it is desirable that the second term in (2.23) should have the value 1 if $|\vec{X}| < 1/2\sqrt{2}\beta$ and the value 0 if $|\vec{X}| > 1/2\sqrt{2}\beta$. Considering the fact that this is over three-dimensional space, we define the following error function that measures how badly, for a particular choice of $\beta$, a pair $(r_1,r_2)$ will result in the violation of the above-stated requirements.
(recall that $\beta$, $r_1$ and $r_2$ determine $a$ and $\alpha$, according to (2.10) and (2.11)).

\[
E_{\beta, \theta, r_1, r_2} = \int_{|\vec{X}| < 1/2\sqrt{2\beta}} \left( 1 - \frac{\hat{w}_{2,a,\alpha}(\vec{X})}{\hat{w}_{2,a,\alpha}(\vec{0})} \right)^2 d\vec{X}
+ \theta \int_{|\vec{X}| > 1/2\sqrt{2\beta}} \left( \frac{\hat{w}_{2,a,\alpha}(\vec{X})}{\hat{w}_{2,a,\alpha}(\vec{0})} \right)^2 d\vec{X}.
\]  

(2.24)

where $\theta$ is a weighting coefficient that depends on the goal of the application; small values of $\theta$ can be used to enhance the importance of the object edges, while large values of it can be employed to promote the significance of the uniform parts of the object. We choose $\theta = 4$ here because we are more interested in artifacts in the uniform regions.

The values of the errors $E_{\beta, \theta, r_1, r_2}$ for $\beta = 1/\sqrt{2}$, $\theta = 4$ and various pairs $(r_1, r_2)$ are shown in Table 2.4. The empty cells in the table correspond to $(r_1, r_2)$ pairs leading to complex values for the blob parameter $\alpha$ (2.11). Clearly, using the pair $(1,4)$ reduces the error better than using any other pair. Although the table demonstrates this only for $\beta = 1/\sqrt{2}$, the conclusion is valid for all $\beta$. This is because examination of (2.24) combined with (2.2) reveals that, for any fixed pair $(r_1, r_2)$, $E_{\beta, \theta, r_1, r_2}$ is inversely proportional to $\beta^3$. (To see this, just consider the consequences of the change of variables $\vec{Y} = \beta \vec{X}$. Observing the effects of this on the fractions and limits occurring in (2.24), we see that the only dependence on $\beta$ that does not cancel out is due to the change from $d\vec{X}$ to $d\vec{Y}$, which -since we are in 3D space- results in a division by $\beta^3$.) We note that in this particular case ($\beta = 1/\sqrt{2}$ and $\theta = 4$), the value of the error (2.24) using blobs with standard parameters is $E_{\beta, \theta, STD} = 0.2352$. Similar superiority of our parameters to the standard parameters for various values of $\theta$ is demonstrated in Table 2.5.

Having justified the choice of using the pair $(1,4)$, we now turn our attention to
Table 2.4: Values of the error $E_{\beta, \theta, r_1, r_2}$ for the $(r_1, r_2)$ pairs ($1 \leq r_1 < r_2 \leq 9$) with $\beta = 1/\sqrt{2}$ and $\theta = 4$.

<table>
<thead>
<tr>
<th>$r_1 \backslash r_2$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4737</td>
<td>0.2455</td>
<td>0.2345</td>
<td>0.2633</td>
<td>0.2972</td>
<td>0.3280</td>
<td>0.3542</td>
<td>0.3760</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.5: The pairs $(r_1, r_2)$ that minimize the error $E_{\beta, \theta, r_1, r_2}$ for various values of $\theta$ with $\beta = 1/\sqrt{2}$, together with $E_{\beta, \theta, STD}$ (which is greater than the optimal $E_{\beta, \theta, r_1, r_2}$ in all cases).

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>$(r_1, r_2)$</th>
<th>$E_{\beta, \theta, r_1, r_2}$</th>
<th>$E_{\beta, \theta, STD}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/4</td>
<td>(1, 2)</td>
<td>0.1035</td>
<td>0.1639</td>
</tr>
<tr>
<td>1</td>
<td>(1, 3)</td>
<td>0.1705</td>
<td>0.1782</td>
</tr>
<tr>
<td>2</td>
<td>(1, 3)</td>
<td>0.1955</td>
<td>0.1972</td>
</tr>
<tr>
<td>4</td>
<td>(1, 4)</td>
<td>0.2345</td>
<td>0.2352</td>
</tr>
<tr>
<td>8</td>
<td>(2, 4)</td>
<td>0.2559</td>
<td>0.3113</td>
</tr>
</tbody>
</table>

assigning values to the coefficients $c_n$. The straight forward method simply assigns

$$c_n = f(\mathbf{x}_n)/g.$$  \hspace{1cm} (2.25)

As shown above, the resulting image will be very nearly constant with the correct value in the eroded interior of constant valued regions. It will also blur edges and have overshoots near edges. Is there a better way of assigning values to the coefficients?

In 2D, the SNARK09 program [40]\(^1\) attempted to improve the image by using an algebraic reconstruction technique [31, Chapter 11] in an effort to achieve sharper edges without greatly increasing the overshoot. Some details of this approach are presented and illustrated in [16, Section 5]. Without going into the details, the algorithm starts with the initial coefficients obtained by (2.25). It then considers that the value of the blob image at a pixel center should be approximately equal to the value

\(^1\)http://www.dig.cs.gc.cuny.edu/software/snark09/
of \( f \) at the same point. Using all the pixel centers gives rise to a set of equations. A tolerance was assigned to each equation that varied from equation to equation. The tolerance was a function of the difference between the value at a pixel center and the average value of the image at nearby pixel centers. If the average value of the image at nearby pixel centers is near the value at the central pixel, then the equation has a small tolerance. As the difference increases, so does the tolerance. This choice of tolerances keeps the value of coefficients inside the eroded centers of constant density regions from changing very much while allowing those near edges more freedom to better approximate the edge. The algorithm was translated into 3D and implemented in the jSNARK program.\(^2\) When it was applied to realistic images, we found that the representations obtained by this sophisticated-looking method were visually indistinguishable from those obtained by the simpler method of (2.25). This comparison was then tried in 2D (by modifying the SNARK09 program) with the same result. It is possible that there is a better way to assign blob coefficients than equation (2.25), but the SNARK09 algorithm is not it. For now, equation (2.25) provides an acceptable and efficient method of assigning the blob coefficients.

Figure 2.9 shows a 256 × 256 slice of the FORBILD\(^3\) abdomen phantom at \( x_3 = 0.0 \). The images have been thresholded with 0.0 as black and 1.5 as white and linear interpolation in-between. Subfigure (a) is the original voxel image and subfigure (b) is the corresponding re-sampled blob image using our recommended parameters. Even though only single slice is shown, the conversion was done in 3D from voxels to blobs. Figure 2.10 reports in a similar manner on the slice of the FORBILD thorax phantom at \( x_3 = 0.0 \). These images have been thresholded with 0.18 as black and 0.3175 as white in order to enhance the contrast of the heart.

\(^2\)http://jsnark.sourceforge.net/  
\(^3\)http://www.imp.uni-erlangen.de/phantoms/
Figure 2.9: Central slices of (a) the FORBILD abdomen phantom and (b) its blob representation using our recommended parameters $a = 2.453144$ and $\alpha = 13.738507$; display thresholds are $0.0 \text{ cm}^{-1}$ and $1.5 \text{ cm}^{-1}$.

Figure 2.10: Central slices of (a) the FORBILD thorax phantom and (b) its blob representation using our recommended parameters; display thresholds are $0.18 \text{ cm}^{-1}$ and $0.3175 \text{ cm}^{-1}$. 
2.4 Image Reconstruction Using Blobs with Desirable Parameters

The next experiments on which we report are ball reconstructions from projections using ART with blobs [31, 44]. 180 noiseless parallel ray projections were taken of a ball with radius 60 onto a planar detector that is taken through a series of 180 tilts in 1 degree increments around the $x_2$ axis. The detector had $207 \times 207$ elements, where each element was a 1 unit square. Five iterations of the ART algorithm were performed using a constant relaxation factor of 0.05. When we do reconstructions instead of representations, the differences between using our recommended and standard blob parameters are not that great, but they are still there; see figure 2.11 for an illustration: (a,c) using our recommended parameters and (b,d) using the same standard parameters we used for the standard approximation in the previous section. (The images in (a) and (b) have been thresholded with 0.99 as black and 1.01 as white with a linear translation in-between.) It is clearly seen in figure 2.11 that the artefactual oscillations in the interior of the reconstructed images are smaller when using our recommended parameters than what they are when using the standard parameters.

We also performed reconstructions of the FORBILD abdomen and thorax phantoms. Projection taking was done by integrating the density of the mathematically-described phantom (rather than its digitization) along lines between the x-ray source positions and the detectors in a two-dimensional array. The data-collection geometry was similar to that used in Section 13.2 of [31]. For every source position, data were collected for 385 equally spaced detectors in each of 17 rows in the array. The size of each detector was $0.425 \text{ cm} \times 0.425 \text{ cm}$, thus at the center of the reconstruction region the rays were $0.2125 \text{ cm}$ apart. The reconstruction was performed on a $256^3$ grid with each voxel $0.1953 \text{ cm}$ on each side (i.e., $\sigma = 0.1953$...
Figure 2.11: Central slices ($x_3 = 0$) through the ball reconstructions and the plots of their values, as $x_2$ varies from -90 to +90, along the line for which $x_1 = 10$ and $x_3 = 0$: (a,c) using our recommended parameters $a = 2.453144$ and $\alpha = 13.738507$; and (b,d) using the standard parameters $a = 2$ and $\alpha = 10.4$. Display thresholds in (a) and (b) are 0.99 (black) to 1.01 (white).
cm). To model the blurring effect of the imaging device in our simulated data collection, we subdivided each detector into four sub-detectors and defined the number of photons detected by a detector to be the sum of the numbers of photons detected by the four sub-detectors. Data were collected using 16,800 pulses of the x-ray source in a helical orbit around the $x_3$ axis with 30 turns going from -8.64 cm to 8.64 cm. The radius of the helix was 57 cm. The distance from the source to the detector was 104 cm. Integrals of the density were collected for the 109,956,000 rays (16,800 pulses times 17 rows times 385 detectors). The effect of photon statistics was simulated using a quantum mean of 1,000,000 and quantum calibration of 300,000,000. The reconstructions were performed using a single cycle of ART with a constant relaxation factor of 0.05, similarly to what is reported in Section 13.3 of [31]. For comparison purposes, the reconstructions were performed using both our recommended blobs (with $\beta = 0.1953/\sqrt{2} = 0.1381$ cm) and voxels as the basis functions. Figures 2.12 (a,b) are the reconstructions of the abdomen phantom at the same display thresholds as used for figures 2.9 (a,b). Figures 2.13 (a,b) are the reconstructions of the thorax phantom at the same display thresholds as used for figures 2.10 (a,b). It is clear that the blob reconstructions are superior to the voxel reconstruction due to the better noise suppression properties.

We point out that in our reconstructions we did not model in the system matrix the blurring effect of the imaging device, we used instead the approximate model in which the measurements are assumed to have been obtained along lines between the source and the centers of the detectors. Thus we did not attempt to use resolution recovery methods that have been demonstrated to lead to significant improvements in emission tomography; see, e.g., [1]. Our reason for leaving such an investigation for future work is that the blurring effect in x-ray CT (just as in electron tomography) is quite different from that in emission tomography. In the latter case, the integral associated with the whole detector is approximately the sum of the integrals
associated with the sub-detectors (see, e.g., the first equation in [1]). On the other hand, x-ray CT presents a strong non-linearity that makes the relationship between such integrals much more complicated; see, for example, figure 3.2 of [31] and the associated discussion in that book. Essentially, the modeling of sub-sampling of detectors in CT mandates nonlinear operators, and so it is not clear how it should be treated with an algorithm such as ART that assumes a linear model between the object to be reconstructed and the measurements.

We do not show reconstructions using the standard blobs. The reason for this is that while the superiority of our proposed blobs as opposed to the standard blobs is easy to illustrate for reconstructions from noiseless data (see figure 2.11), when the simulated data incorporates noise due to limited photon statistics (as it is the case for our FORBILD phantoms experiments) the deleterious effects of such noise overwhelms the relatively small improvement due to using our recommended blob parameters.
Figure 2.13: Central slices of the reconstructions of the FORBILD thorax phantom: (a) using voxels and (b) using blobs with our recommended parameters. Display thresholds are 0.18 - 0.3175 cm$^{-1}$. 
Chapter 3

Influenza Virus Reconstruction

In this chapter we list the mechanisms we used for the data collection and the data processing prior to the 3D reconstruction. The purpose of processing the projection data is to obtain high quality reconstructions which they are the essential foundations for successful classifications. We note that the material of this chapter appeared in a book chapter written together with Paul Gottlieb, Al Katz, Stuart W. Rowland, and Doris Bucher [3]. A closely-related publication by us is [39].

Influenza virus type B (B/Lee/40) was grown and amplified in embryonated chicken eggs. The original “seed” allantoic fluids containing B virus were diluted 1 : 1000 in phosphate buffered saline containing 250 µg/ml aminoglycoside antibiotic gentamicin. Each egg was inoculated with 0.1 ml of the diluted allantoic fluid and incubated at 33°C for 75 hours. A step gradient was utilized to purify the virus particles to approximately 1.6 mg/ml protein total viral mass.

Three µl of a suspension of influenza virus type B sample was placed onto glow-discharged, perforated Quantifoil grids, blotted, and plunge-frozen in liquid ethane. The virus sample was pre-mixed with a suspension of 10 nm gold beads in order to add fiducial markers to aid in tomographic alignment. Input data for the reconstruction were collected, using the single-axis tilt geometry, and processed further.
(after the tomographic alignment described in the next paragraph) as specified in the following subsections.

For the tomographic alignment that makes use of the fiducial markers we used the software package IMOD\(^1\). The aim of the alignment is to produce geometrically consistent micrographs. We perceive each micrograph as having a 2D coordinate system such the 3D tilt axis projects onto the \(x_2\) axis in the micrograph, see figure 1.3. Furthermore, the micrographs are vertically aligned so that the \(x_2\)-coordinates of the projections of any particular gold bead are the same in all projections.

### 3.1 Data Collection

Images were recorded at 50000× magnification (this resulted in projection images in which the edge of a pixel is 0.44 nm) and an underfocus of \(8 \pm 0.5 \mu m\), with a JEOL 3200FSC electron microscope (JEOL, West Chester, PA) operating at 300 kV. An energy filter, with a slit width of 20 eV, was inserted to eliminate non-elastically scattered electrons and thereby enhance contrast. CTF correction was performed using IMOD. Tilt series were recorded using the SerialEM software [46] on a 4096 × 4096 pixel CCD camera (Gatan Inc, Pleasanton, CA). The size of the images was reduced to 2048 × 2048 pixels by binning; the edges of the resulting pixels are 0.88 nm. Specimen angles ranged approximately between \(-60^\circ\) and \(+60^\circ\) with approximate 2° steps, producing 61 projection images, the exact angles are recorded by the microscope software. The low-dose imaging mode limited total specimen dosage to \(60 e/\AA^2\) over the entire tilt series. Figure 3.1 displays three projection images of the specimen at angles \(-60.32^\circ\), \(-0.10^\circ\) and \(+59.66^\circ\).

\(^1\)http://bio3d.colorado.edu/imod
Figure 3.1: Three projection images of the specimen at angles (from left to right): $-60.32^\circ$, $-0.10^\circ$ and $+59.66^\circ$.

3.2 Data Processing

Since the viruses are suspended in a layer of ice, the physically-obtained line integrals have nonzero contributions both from ice and from the biological structure of the virus. The mathematical theory of image reconstruction from projections requires that the object to be reconstructed can be represented by a function $f$ of finite support, which means that there is a real number $E > 0$ such that the value of $f$ is zero at all points that are farther than $E$ from the origin (in other words, the object to be reconstructed is inside a ball of radius $E$ centered at the origin); see [31, Section 6.1]. This implies that the aim of recovering a physical parameter, such as the Coulomb potential, from a tilt series of EM projections is inconsistent with the mathematical assumptions of image reconstruction approaches. This can be seen in figure 3.2; since the Coulomb potential of ice is not zero, there is no reasonable way of identifying the required ball of radius $E$. We now show that by subtracting the contribution of ice from the line integrals, we can define a function $f$ that satisfies the requirement for image reconstruction.

Figure 3.2 indicates two lines, $i$ and $i'$, that contribute to a projection image in the tilt series; line $i$ goes through a virus, while $i'$ goes through ice only. Line $i$ first intersects the top edge of the ice layer at point $a$, meets the virus at point $b$ and
leaves it at point \( c \), and finally exits the ice layer at point \( d \). Line \( i' \) intersects the top and bottom edges of the ice layer at points \( a' \) and \( d' \) respectively.

The physically-measured line integral \( p_i \) for line \( i \) can be expressed as

\[
p_i = \int_a^b \gamma \, dl + \int_b^c v(l) \, dl + \int_c^d \gamma \, dl,
\]

(3.1)

where \( \gamma \) and \( v(l) \) are the Coulomb potentials of ice (assumed to be constant) and virus (assumed to vary with the distance \( l \) along the line), respectively. If we assume that the thickness of ice is locally uniform, then the line integral \( p_{i'} \) for line \( i' \) is

\[
p_{i'} = \int_{a'}^{d'} \gamma \, dl = \int_a^d \gamma \, dl.
\]

(3.2)

Since, the Coulomb potential inside a protein is greater than it is in ice, we expect
to be smaller than \(p_i\). This is indeed the case as it can be observed in figure 3.1, in which the parts of the images that correspond to the virus particles are darker than their surroundings (since in these images larger projection values are mapped into smaller gray values). Subtracting (3.2) from (3.1) we get

\[
p_i - p'_i = \int_b^c [v(l) - \gamma] \, dl.
\] (3.3)

Now define \(f\) as follows: If a point is inside the virus and the Coulomb potential at that point is \(v\), then the value of \(f\) at that point is \(v - \gamma\); for all other points, the value of \(f\) is zero. Clearly, this \(f\) is of finite support, since its value is zero outside any ball that contains the whole of the virus. Furthermore, integrating \(f\) along a line \(i\) that goes through the virus, we get

\[
\int_{-\infty}^{\infty} f(l) \, dl = \int_b^c [v(l) - \gamma] \, dl.
\] (3.4)

The value of this integral can be obtained from the physical-obtained line integrals, by the use of (3.3). The integral of \(f\) along a line \(i'\) that does not go through the virus is clearly zero. This means that the line integral of \(f\) is available to us for a line whether or not that line goes through the virus and, so, \(f\) (being of finite support) can be estimated from the physically obtained projection measurements by methods of image reconstruction from projections.

One particular aspect of our approach is that we reconstruct each virus individually, as opposed to the alternative approach of reconstructing all the virions that appear in the tilt series simultaneously. There are several advantages to our approach. A minor one is that due to the smaller sizes of the data sets we can do the reconstructions more rapidly. More importantly, we also believe that we can do the reconstructions more accurately for two reasons. One is that the subtraction of the contribution of the ice layer from the physically-measured line integrals be-
comes more reliable, since what is to be subtracted is now estimated based only on those lines that go near (but not through) the specific virus we wish to reconstruct. Second, as we will see, the data that are plugged into the reconstruction algorithm will be complete and consistent in the sense that, for all projection directions, we will have an estimate of the line integral of the same 3D object for all lines in that direction.

In order to achieve what is stated in the last sentence, we aim at identifying a ball in 3D space that is large enough to contain the virus in question and at transforming the physically-collected projection data into projection data of the contents of that ball in the given projection directions. In order to do this, we need to make a coordinate transformation: essentially, we wish to identify the locations of the lines of integration in a coordinate system that has its origin at the center of the ball. Since each virus is only of a finite size, we know that there is such an enclosing ball (in fact there is a multitude of them), but in order to perform the required coordinate transformation we need to know where the center of the ball is and, having only the projections, such information is not directly available to us. The projection of the ball in each micrograph is a disk that contains the projection of the virus, and the centers of those disks are the projections of the center of the ball. If we could identify those disks in the projections, then the center of the ball would lie at the intersection of all the projection lines that go to the centers of the disks.

A question that now arises is: How to determine the exact location of the above-mentioned disk-centers in each micrograph? If the virus to be reconstructed is very close to the tilt axis, then the answer is simple; we just need to choose a ball whose center is on the tilt axis and contains the virus; the projection of that ball will be the same disk in all micrographs. For viruses that are away from the tilt axis, the answer is provided by the following more complicated reasoning.

Consider a ball that contains the virus; let the coordinates of its center be
(\(C_1, C_2, C_3\)). Let \(u\) be used to index micrographs in our tilt series (in our case, \(1 \leq u \leq 61\)) and let the tilt angle of the \(u\)th micrograph be \(\alpha_u\) (these angles are provided to us by the microscope’s software). Let \((C_{1u}, C_{2u})\) denote the projection of \((C_1, C_2, C_3)\) in the \(u\)th micrograph. We do not know what \((C_1, C_2, C_3)\) and the \((C_{1u}, C_{2u})\) are, but we can find out things about them based on the micrographs, such as the one in figure 1.3.

First, since those micrographs are aligned in the previously-described-manner, it is the case that \(C_{2u} = C_2\), for all \(u\). To estimate this common value of all the \(C_{2u}\), we make use of interactive software that allows us to pick a radius \(\delta\) and a \(x_2\)-coordinate \(C_2\) and, for any selected micrograph, slide a circle of radius \(\delta\) and a center \(x_2\)-coordinate \(C_2\) horizontally over the micrograph, see figure 3.3. We use this software to find a \(\delta\) that is as small as possible for which there exists a \(C_2\) and, for each index \(u\), a \(C_{1u}'\), such that the circle with center \((C_{1u}', C_2)\) and radius \(\delta\) properly surrounds the projection of the virus in the \(u\)th micrograph. We take the \(C_2\) that we find in this fashion the common value of \(C_{2u}\), for all \(u\). This estimation of \(C_2\) is quite robust, since it is based on all the micrographs. Furthermore, even if a small mistake were made in the value of \(C_2\), we would get good reconstructions, since (due to vertical alignment of the projection images) an error in \(C_2\) would still provide us with consistent projections of a ball that encloses the virus.

The situation is quite different for the \(C_{1u}'s\) obtained by the process of the previous paragraph. First, each one of them is based on one micrograph only and so they are less reliable than the estimate of \(C_2\) based on all the micrographs. Second, inaccuracies in using \(C_{1u}'s\) as estimators of the unknown \(C_{1u}s\) will result in geometrical inconsistencies between the projection images extracted from the micrographs (as indicated by the small square in figure 1.3), resulting in inaccuracies in the reconstructions. However, we can overcome these difficulties by the following least squares approach to estimating simultaneously (and more accurately) all \(C_{1u}s\) from
the collection of all the (less accurate) $C'_{1u}$s.

As illustrated in figure 3.4, if the values $C_1$ and $C_3$ were $x_1$ and $x_3$, respectively, then the value of $C_{1u}$ would be

$$
C_{1u}(x_1, x_3) = \sqrt{x_1^2 + x_3^2} \times \cos \left( \alpha_u + \arctan \left( \frac{x_3}{x_1} \right) \right),
$$

(3.5)

for all $u$. Let

$$
d(x_1, x_3) = \sum_u \left( C_{1u}(x_1, x_3) - C'_{1u} \right)^2,
$$

(3.6)

where the $C'_{1u}$s are the manually determined values using the software illustrated in figure 3.3. In figure 3.5 we plot the values of $d(x_1, x_3)$ for ranges of values of $x_1$ and $x_3$. The pair $(x_{1d}, x_{3d})$ that minimizes (3.6) is the least squares estimator of the coordinates $(C_1, C_3)$. Using this estimator and (3.5), we estimate, for all $u$, that
Figure 3.4: Projection of \((x_1, x_3)\) after rotation by tilt angle \(\alpha_u\).

\[
C_{1u} = \sqrt{x_{1d}^2 + x_{3d}^2} \times \cos \left( \alpha_u + \arctan \left( \frac{x_{3d}}{x_{1d}} \right) \right) .
\]

(3.7)

Once the disk centers \((C_{1u}, C_2)\) were determined for each of the micrographs, we created square-shaped input images in such a way that the center of each square coincides with the pixel nearest to the determined disk center. The actual size of the square-shaped input images is specified in the next paragraph.

The effective thickness \((d - a = d' - d'\) in figure 3.2) of the frozen liquid layer that interacts with the electron beam varies with the tilt angle, resulting in differences in gray values between micrographs since the Coulomb potential of ice is not zero; this is clearly illustrated in figure 3.1. Before conducting the 3D reconstruction, we processed the contents of the physically-obtained projection images with two purposes in mind: (i) to obtain line integrals of \(f\) as described in (3.4) and (ii) to create a single coordinate system for the specification of the locations of the lines along which \(f\) is integrated for the various projection images. First we determined
a circle, \( C \), surrounding the virus in each of the images; that circle is considered to be the circumference of the disk that is the shadow of the support of the function \( f \) to be reconstructed. Then for each image we averaged the line integrals outside the circle \( C \) (a region that contains only ice); those averages correspond to the line integrals of the Coulomb potential of ice, \( \gamma \), as described in (3.2). After that we subtracted, for each image, the average from the line integrals inside \( C \) to obtain the line integrals of \( f \) as stated in (3.3). Finally we set the line integrals to zero outside the circle \( C \) in all images to make the support of \( f \) finite. The underlying assumption is that the virus is enclosed in a ball surrounded by ice; in each projection the shadow of that ball is a disk. All the pixels outside the disk have values determined by lines that go through only ice. By subtracting the average value of ice from each pixel value in each projection image, the resulting values represent line integrals of what we wish to reconstruct (assuming that the thickness of ice is
uniform).

The following is worth emphasis, since it validates our claim that the projection data set provided by the procedure just described (which is the data set that is plugged into the reconstruction algorithm) is complete and consistent in the sense that, for all projection directions, it contains an estimate of the line integral of the same 3D object for all lines in that direction. The point is that the only required property of the estimator of the coordinates \((C_1, C_3)\) together with the estimator for the radius of the ball is that the resulting ball encloses the virus of interest and all points of the ball that are not occupied by that virus are occupied by ice. If our procedure achieves (as it is likely) to produce such a ball and if the thickness of ice is uniform (at least locally), then our method provides a complete and consistent projection data set for the virus we wish to reconstruct. Note that although the virus on which we are illustrating our procedure has approximately circular projections, this is not necessary for our procedure to function as desired; we just need to enclose the virus of interest (of whatever shape) inside a ball that contains only that virus and nothing else but ice. From such a data set we are able to reconstruct the entire interior of the ball, including the virus of interest.

In the experiments we are reporting, the radius, \(\delta\), of the smallest circle surrounding the virus was 77.44 nm (see figure 3.3); the radius of the enclosing ball was 83.60 nm (under 110% of \(\delta\)), and its center is located at \(C_1 = -99.75\) nm, \(C_2 = 108.24\) nm, and \(C_3 = 5.37\) nm. The size of the edge of the square-shaped input images was 176 nm. The red square drawn in figure 1.3 is the exact boundary of the input image generated from the micrograph with tilt angle \(-0.10^\circ\). In the line integral averaging process, we excluded the ice pixels located in the top right quadrant of that square, since that region overlaps with an adjacent virus in some projection images (namely, in the projections with low and high tilt angles).

Three processed projection images of the virus surrounded by the red square in
3.3 Virus Reconstruction

In this section we demonstrate the potential of the algorithm ART (1.4) with blobs (2.1) to produce more efficacious 3D reconstructions from cryo-EM images of influenza virus particles than those produced by alternative methods. The reconstruction parameters were set as follows:

- ART parameters: The initial estimate $c^{(0)}$ is the zero vector, the relaxation parameter is $\lambda = 0.05$, the function $i^{(k)}$ is determined by the orthogonal data access ordering as described in the Subsection 1.2.1 (and, in greater detail, in [31], page 209, where it is referred to as the efficient ordering) and the number of iterations is 1.

- Blob parameters: The order of the Bessel function is $m = 2$, the support is $a = 2.158767\text{nm}$ and the shape parameter is $\alpha = 13.738507$; these are the parameters used in the blob plot in figure 2.1.

- Grid parameters: The grid $B_\beta$ of figure 1.4 is used with $\beta = \frac{1}{\sqrt{2}} \times s$, where $s = 0.88\text{nm}$ is the size of the edge of the pixel in the projection image after binning, see Subsection 3.1.
• 3D scene parameters: the range of the integers \((k_1, k_2, k_3)\) that are used to index the voxels in the 3D scene as described following (1.7), is \([1, 200]\) and the sampling distance \(\sigma = s = 0.88\) nm.

For comparison, we reconstructed the same virus from the same data using SIRT and WBP. The software package we used for all reconstructions was Xmipp\(^2\). For SIRT we used the same blobs as for ART, but the number of iterations was 10 (rather than 1, as in ART). This implies that the computer cost of our use of SIRT was ten times what was needed for our use of ART. This is necessary, since fewer iterations of SIRT lead to inferior results; indeed, it is stated on the IMOD web-page regarding SIRT\(^3\) that “The desired number of iterations is usually in the range of 8-15 for cryotomograms”. For the WBP reconstructions we used the default parameters provided by Xmipp. This by itself leads to reconstructions with noise located in the high spatial frequencies, which can be eliminated by the application of a low-pass filter. Application of such a filter is not needed for the outputs of the series expansion methods (such as ART or SIRT) using blobs, since (for reasons explained in Subsection 1.2.2) the outputs of those reconstruction methods are essentially bandlimited without any further filtering. To make the output of WBP comparable from this point of view with the outputs of the series expansion methods using blobs, we low-pass filtered the output of WBP using the SPIDER\(^4\) Butterworth filter (option 7) with lower and upper limiting frequencies 0.08 and 0.18.

Figure 3.7 shows three different slices, all perpendicular to the \(x_3\) axis, each reconstructed using ART, SIRT and WBP; (a), (b) and (c) are slices from the ART reconstruction, (d), (e) and (f) are matching slices from the SIRT reconstruction and (g), (h) and (i) are matching slices from the filtered WBP reconstruction. The edge of each pixel in these reconstructions is of the same length as the edge of a pixel

\(^{2}\)http://xmipp.cnb.csic.es/twiki/bin/view/Xmipp/Reconstruct_art_v3

\(^{3}\)http://bio3d.colorado.edu/imod/doc/tomoguide.html#SIRTtomogram

Figure 3.7: Three different slices \((k_3 = 89, 90 \text{ and } 91)\) from virus reconstructions: (a), (b) and (c) are from the ART reconstruction; (d), (e) and (f) are corresponding slices from the SIRT reconstruction and (g), (h) and (i) are corresponding slices from the filtered WBP reconstruction; all from the same projection data.
in the projection images after binning; i.e., 0.88 nm. In accordance with results in earlier literature [13, 53], we observe that the WBP reconstruction is inferior. In the reconstructed slices shown in figure 3.7, ART is seen to have performed somewhat better than SIRT. However, experience reported in the literature ([53] and [31, Chapter 12]) indicates that, with more iterations, the quality of the SIRT reconstruction would approach that of the ART reconstruction, but that would add to its computational expense that is already an order of magnitude greater than what is required by ART.

3.4 Results and Discussion

The examination of a conventional projection image (figures. 1.3 and 3.1) of frozen-hydrated Influenza B/Lee/40 virus shows an intact particle, surrounded by a distinct envelope that contains the surface glycoprotein spikes, HA and NA.

A midsection slice of the reconstructed 3D scene, obtained using ART, is shown in figure 3.8(left) for a 120 nm diameter virion. The slice clearly shows (14 nm long) surface spikes, the (8 nm thick) envelope-matrix, and discrete RNPs inside the virion. The surface proteins penetration into the matrix is resolved and close visual inspection reveals two distinct surface protein morphologies: (1) near uniform density and thickness; and (2) “club-like” and carrying a denser top. The bi-lobed HA trimer is distinguished from the club-like NA tetramer with reasonable accuracy by evaluation of the contour spike density in the reconstructed slice. In figure 3.8(left), high density, i.e. HA, stalks are indicated by solid yellow and club-like, i.e. NA, stalks by red arrows. Individual RNPs with different orientations are evident in the slice of the ART with blobs reconstruction. In figure 3.8(right), the X-ray crystal structure of HA and NA are docked to two spikes considered to be HA and NA from the morphology contour. The correspondence of the docked
This discussion demonstrates that ART (with its free parameters appropriately tuned) provides 3D reconstructions of viruses from tomographic tilt series that allow reliable quantification of the surface proteins. By looking at figure 3.7, we conclude that the same is not achieved using WBP.
Chapter 4

Virus Spikes Classification

Human classification is subjective because it is based on an individual’s judgment, which in turn depends on his/her knowledge, experience and mindset. For that reason we decided to automate the process of classifying the surface proteins. Automated classification outputs more reproducible results by making the same choices under the same conditions; it can be defined as the process by which a computer assigns a category to a protein spike on the basis of its shape. The objectives of automating the classification are to make the process more reliable and reproducible and also to save time and costs by making the process much faster and more efficient than the human process. The following subsections outline the steps we performed in order to automate the classification procedure.

4.1 Identification of the Region of Interest

Our first step toward the classification of the two kinds of spikes is the removal from the output of the 3D reconstruction all voxels outside a region that contains the spikes. Based on figures 1.1 and 1.2, we assume that this region of interest lies between two spheres. The outer one is the surface of the ball (of radius $E$ centered at the origin $(C_1,C_2,C_3)$) that is identified as containing the whole virus
Figure 4.1: Gray-value image of the slice displayed on the left of figure 3.8: (a) superimposing the inner and outer circles on it and (b) setting the values of all pixels outside the annulus between the circles to zero.

by the method described in Section 3.2. The inner sphere is centered at the origin as well; its radius was obtained from the central slice \((k_3 = C_3)\) by using an interactive software similar to the one used in Section 3.2. The radius we picked produces an inner sphere that is large enough to include all of the bilayer envelope surrounding the matrix, but small enough so as not to cut off what are referred to as stems of the HA and NA spikes in figure 1.2. The radii of the outer and inner spheres were 83.60 nm and 59.84 nm, respectively.

This method is illustrated in figure 4.1. On its left is the gray-value image of the slice displayed on the left of figure 3.8 with two concentric circles superimposed on it. The outer and inner circles are the intersections of the spheres specified in the previous paragraph with the plane of the slice. On the right of figure 4.1 is the gray-value image that is obtained by setting the values of all pixels whose centers are outside the annulus between the circles to zero.

It has been our experience that in slices that are far from the central ones, the reconstructed values are not as reliable as for the more central slices. Accordingly,
we limited our region of interest to the central slices only. The number of the slices that we included in the region of interest slab was 21. This number was chosen to ensure that full spikes are contained in the slab at different orientations. The width of one full spike extends approximately over 10 slices. The outcome of this process is the input to the further processing that is described in the following subsections.

4.2 Segmentation

The segmentation that we use in our project takes the 3D scene produced as described in the previous paragraph as input and provides as output a structure system, which is a finite collection of structures; see [56]. The intent is that each of these structures is a finite collection of voxels that taken together overlap with the extent of exactly one of the surface spikes as closely as possible.

In our research we first attempted to use the simplest method of segmentation, namely the thresholding method. This method creates binary 3D scenes from the original ones by assigning the value zero to all voxels whose value is below some threshold and the value one to all other voxels. Due to the presence of noise in the input data, the outcome of this segmentation method was not promising. To overcome this, we decided to use a more sophisticated segmentation method, namely the fuzzy connectedness technique [14, 18, 32], which is described in the following subsections.

4.2.1 Description of the Approach

Prior to getting into the theory we give a picturesque description of the approach. The model takes the form of a military exercise. It involves $M$ competing armies (one corresponding to each object) and a number of castles (one corresponding to each voxel within the region of interest) such that there is a one-way road from
every castle to every other castle.

The exercise proceeds in discrete iterative steps. Initially, armies have full strength and they occupy their respective castles. All armies try to increase their respective territories, but the moving from one castle to another reduces the strength of the soldiers to be the minimum of their strength at the previous castle and the affinity for that army between the castles. At any given time, a castle will be occupied by the soldiers of the armies who were not weaker than any other soldiers who reached that castle at that time; the strength of a castle is set to the strength of its new occupiers. The output of the algorithm provides, for each castle, the strength of the castle and the armies that occupy it at the end of the exercise.

4.2.2 Theory and Algorithm

In this very general approach, we deal with an arbitrary set $V$, whose elements are referred to as spels (short for spatial elements). These spels can represent many different things, such as pixels of an image, or voxels in a 3D volume. (In our specific application, $V$ is the set of all the voxels in the region of interest as defined in Section 4.1.) We desire to partition $V$ into a number of objects, but in a fuzzy way; i.e., in addition to a spel being judged to belong to a particular object, it is also assigned a grade of membership in the object (that is, a number between 0 and 1, where 0 indicates that the spel definitely does not belong to the object, and 1 indicates that it definitely does). We call a sequence of spels a chain, its links are the ordered pairs of consecutive spels in the sequence. The strength of a chain is the strength of its weakest link. The fuzzy connectedness of $c$ to $d$ in a set $A$ is then defined as the strength of the strongest chain in $A$ from $c$ to $d$.

Each of the $M$ objects has its own definition of strength for the links (we use $\psi_m$ to denote the strength of the links, alternatively called the affinity function, for the $m$th object) and its own set of seed spels $V_m$. Each of the objects is then defined
as the collection of those spels that are connected entirely within the object to one of its own seed spels in a stronger way than to any of the other seed spels. The detailed specification of the MOFS (multi-object fuzzy segmentation) algorithm [14] is given below. We note that a fuzzy segmentation of $V$ is a function $\sigma$ that maps each $c \in V$ into an $(M + 1)$-dimensional vector $\sigma^c = (\sigma^c_0, \sigma^c_1, \ldots, \sigma^c_M)$. $\sigma^c_m$ represents the grade of membership of the spel $c$ in the $m$th object, and $\sigma^c_0$ is always $\max_{1 \leq m \leq M} \sigma^c_m$.

**Algorithm 4.1 MOFS algorithm.**

1. for $c \in V$ do
2.    for $m \leftarrow 0$ to $M$ do
3.        $\sigma^c_m \leftarrow 0$
4.    $H \leftarrow \emptyset$
5.    for $m \leftarrow 1$ to $M$ do
6.        $U_m \leftarrow V_m$
7.    for $c \in U_m$ do
8.        if $\sigma^c_0 = 0$ then do $H \leftarrow H \cup \{c\}$
9.        $\sigma^c_0 \leftarrow \sigma^c_m \leftarrow 1$
10.   $r \leftarrow 1$
11. while $r > 0$ do
12.    for $m \leftarrow 1$ to $M$ do
13.        while $U_m \neq \emptyset$ do
14.            remove a spel $d$ from $U_m$
15.            $C \leftarrow \{c \in V \mid \sigma^c_m < \min (r, \psi_m (d, c))$ and $\sigma^c_0 \leq \min (r, \psi_m (d, c))\}$
16.            while $C \neq \emptyset$ do
17.                remove a spel $c$ from $C$
18.                $t \leftarrow \min (r, \psi_m (d, c))$
19.                if $r = t$ and $\sigma^c_m < r$ then do $U_m \leftarrow U_m \cup \{c\}$
20.                if $\sigma^c_0 < t$ then do
21.                    if $\sigma^c_0 = 0$ then do $H \leftarrow H \cup \{c\}$
22.                        for $n \leftarrow 1$ to $M$ do
23.                            $\sigma^c_n \leftarrow 0$
24.                    $\sigma^c_0 \leftarrow \sigma^c_m \leftarrow t$
25.                while Maximum-Key($H$) = $r$ do
26.                    Remove-Max($H$)
27.                    $r \leftarrow$ Maximum-Key($H$)
28.    for $m \leftarrow 1$ to $M$ do
29.        $U_m \leftarrow \{c \in H \mid \sigma^c_m = r\}$

The essential feature of this approach is that the MOFS algorithm calculates, for
every spel, the grade of membership of that spel to each of the individual object and then assigns the spel to the objects for which its grade of membership is maximal.

4.2.3 Experiments

In the specific experiments on which we report below, the MOFS algorithm is applied to the 3D region of interest determined in Section 4.1 to identify the virus spikes in it. The number of objects to be segmented is $M = 3$. They are the spikes (foreground), the background, and the remaining pieces of the envelope-matrix (interior). The fuzzy spel affinity function is defined based on statistical properties of the links within regions identified by the user as belonging to the three objects.

The way we specify the affinity functions $\psi_m$ and sets of seeds $V_m$ ($1 \leq m \leq 3$) is the following. We make use of an interactive seed-selection application (see figure 4.2) that allows us to click on some spel, in each slice, to identify them as belonging to the $m$th object, and $V_m$ is formed by these points and those of their twenty-six neighbors (8 from the same slice and 18 from the adjacent ones) that are within the region of interest. We define $g_m$ to be the mean and $h_m$ to be the standard deviation of the average brightness for all face-adjacent pairs of spel in $V_m$ and $e_m$ to be the mean and $f_m$ to be the standard deviation of the absolute differences of brightness for all face-adjacent pairs of spel in $V_m$. Then $\psi_m(c,d)$ is defined to be 0 if $c$ and $d$ are not face adjacent and to be $\left[ \rho_{g_m,h_m}(g) + \rho_{e_m,f_m}(e) \right]/2$ if they are, where $g$ is the mean and $e$ is the absolute difference of the brightnesses of $c$ and $d$ and the function $\rho_{r,s}(x)$ is the probability density function of the Gaussian distribution with mean $r$ and standard deviation $s$ multiplied by a constant so that the peak value becomes 1.

The interactive seed-selection application allows us to select a slice and then an object $m$ is activated by pressing the button labeled with the object’s name; the mouse cursor is reshaped once the button is pressed to indicate that the seeds selec-
Figure 4.2: Interactive seed-selection application used in the segmentation of spikes in the region of interest defined in Section 4.1.

Three different colors are used to mark the locations of the clicked seeds (green, red, and yellow for the foreground, the background, and the interior objects, respectively). Figure 4.2 illustrates this for the 11th slice of the region of interest. The brightness statistics of the object \( m \), to be used in the affinity functions (namely, \( g_m \), \( h_m \), \( e_m \) and \( f_m \)) are computed on the fly and displayed in the right panel of the interactive application.

Manual seed selection is a tedious and time-consuming task (which, by the way, also suffers from questionable reproducibility); in our illustrative example more than 3,000 seeds need to be assigned for the following reason: The region of interest of the virus has 44 spikes and around 4 seeds have to be chosen for each spike (2 for the foreground and 2 for the background) in each of the 21 slices generated by the method described in Section 4.1. To overcome the time-consuming nature of manually selecting over 3,000 seeds, we incorporated into the interac-
tive application a feature for the automated generation of many seed spels. The objectives of automating the seed spels generation are to save time and costs by making the seeds selection much faster and also to make the process more reliable and reproducible. The usefulness of this feature will be much appreciated in the evaluation methodology, described in Chapter 5, where the seeds selection has to be performed for 30 different simulated viruses.

We now discuss in some detail how the seed spels to be used in the MOFS algorithm were selected for our illustrative example. We applied the interactive seed selection application to each slice of the region of interest (this is illustrated for Slice 11 in figure 4.2). Most of the foreground and background seeds that we eventually used were automatically selected by pressing the button Generate Seeds, which results in a foreground seed for each 2D local brightness minimum and a background seed for each 2D local brightness maximum. A few such seeds were manually added especially for the background object; but all the seeds in the interior object were manually selected (around 500 seed spels in the total for the 3D region of interest). We note that, in the automatic generation of seeds, some seed spels in the interior object were initially (automatically) selected as foreground seeds; this happened because the brightness values for those spels were local minima in the interior object; the interactive application has the ability to turn those spels into interior seeds by clicking on their locations with the Interior Select Seeds button activated; the color of those locations are then changed from green to yellow. In figure 4.2, the green and most of the red marks indicate seeds that were automatically generated, they are local minima and local maxima of the brightness values, respectively; very few red marks were added manually. The yellow marks pinpoint the locations of manual reassignments from Foreground to Interior inside the envelope-matrix.

Another feature of our interactive application is the ability to save seeds and
Figure 4.3: Surface-smoothed display of the structure system of virus spikes segmented within the region of interest by the MOFS algorithm using the seeds whose selection is illustrated in figure 4.2.

load them later in order to make further modifications; this feature is useful in case it becomes desirable to add more seeds to the ones previously selected.

Inputs to the MOFS algorithm are the set $V$ of spels in the 3D region of interest, the sets of seed spels $V_m$ and the affinity functions $\psi_m$ ($1 \leq m \leq 3$). The output is an array that associates to each spel $c \in V$ a 4-dimensional vector $\sigma^c = (\sigma^c_0, \sigma^c_1, \sigma^c_2, \sigma^c_3)$, where $\sigma^c_m$ ($1 \leq m \leq 3$) is the grade of membership of the spel $c$ in the $m$th object, and $\sigma^c_0 = \max (\sigma^c_1, \sigma^c_2, \sigma^c_3)$. The foreground object (structure system of virus spikes) produced as a result of applying the MOFS algorithm is the finite collection of spels $c$ whose vectors $\sigma^c$ are characterized by the property $\sigma^c_0 = \sigma^c_1$; these spels are displayed in figure 4.3 using the molecular visualization.
software UCSF Chimera\(^1\) with its surface smoothing feature.

### 4.3 Feature Extraction and Classification

#### 4.3.1 Reorientation of Spike Structures

In order to be able to achieve our main aim, which is the classification of the surface spikes, we need to extract the individual spike structures from the whole foreground object. Looking at figure 4.3, we conjecture that the individual spike structures can be obtained by partitioning the foreground object into components using face-adjacency (see [30]). In more difficult cases, in which the separation between spikes is not sufficient to extract them individually, we can remedy this by returning to our interactive seed selection application and inserting additional background seeds to ensure separation.

As a preliminary step to the classification of the individual surface spikes, we rotate each of the spike structures so that they end up to be in an approximately upright position (by which we mean that the direction from the stem to the head will be approximately that of the positive \(x_2\) direction of the coordinate system). The details of how this is done (in particular for our illustrative example), using a program that we designed for this purpose, is as follows.

1. The user selects a slice perpendicular to the \(x_3\) axis in which the spike to be extracted is clearly identifiable. Let \(s\) \((1 \leq s \leq 44)\) be the index of the spike \((44\) is the number of spike structures that appear in the foreground \((1 \leq k \leq 21,\) see figure 4.2). We use \(I_k\) and \(c_k\) to denote the chosen slice and its \(x_3\) coordinate, respectively. \(I_k\) is a 2D binary \((\{0,1\}\)-valued) image with \(200 \times 200\) pixels in our illustrative example; with the value 1 indicating that the corresponding location is in the foreground and, hence, in one of the spike structures.

\(^1\)http://www.cgl.ucsf.edu/chimera/
structures. This is demonstrated in figure 4.4 in which the blue cross indicates the location of the point \((0,0,c_k)\), which will be referred to as the slice center in what follows.

2. The user clicks on a pixel inside the spike \(s\) in the 2D image \(I_k\).

3. The application calculates the angle \(\alpha_s\) of counterclockwise rotation around the slice center that is needed to bring the clicked pixel to the positive side of the vertical line through the slice center; more precisely to \((0, p, c_k)\), where \(p\) is the distance between the rotation center and the center of the clicked pixel.

4. The application rotates the whole 2D image, \(I_k\), with the angle \(\alpha_s\) around the slice center. The rotation makes the output image large enough to contain the entire rotated image. It uses nearest neighbor interpolation and sets the values of pixels in the output image that are outside the input image to zero.

5. The output image is then cropped to the original size \(200 \times 200\) pixels in such a way that the center of the final image coincides with the slice center. Let \(J_k\)
be that final slice. After this transformation the spike \( s \) will appear in the 12 o’clock position in the slice \( J_k \).

6. Steps (4) and (5) are repeated for the remaining slices, \( I_i \) \((i \neq k)\), of the foreground object.

7. The final slices, \( J_i \), are stacked together, in the original order of \( i \), to shape the rotated volume of the foreground object. The outcome of this step is identical to a counterclockwise rotation, with angle \( \alpha_s \), of the whole 3D foreground object around the \( x_3 \) axis.

8. The rotated spike \( s \) is defined as that component of the foreground object when partitioned using face-adjacency (see [30]) that contains rotated pixel \((0, p, c_k)\). Since both HA and NA spikes extend radially from the membrane [29], we believe that the so-obtained spike structure will be in an approximately upright position (i.e., the positive \( x_2 \) direction will be approximately from the stem to the head of the spike).

9. Steps (1) to (8) are to be repeated for each individual spike structure. We note that most of the above computation is done by the application; the user’s duty is to choose a slice and click on a spike to obtain its structure. A user may use a single slice for extracting all spike structures; behaving this way will limit the task to making 44 clicks in the chosen slice (one click for each spike). Figure 4.4 illustrates the concept. In that figure we show a slice in which the 23\(^{rd} \) spike is clearly in evidence. Figure 4.4(a) is the original slice and figure 4.4(b) is the rotated one. This process was applied for each individual spike to obtain its rotated structure; outcomes of this exercise are the inputs to the next processing task that is described in the following subsection.
4.3.2 Feature Extraction

To perform automated classification of the segmented structures that correspond to individual spikes, we need to identify features that distinguish the two kinds of spikes from each other. In our search for suitable features, we looked at the work of Harris et al. [29] that reports the following. Both HA and NA spikes extend radially from the membrane to terminate in bulbous heads; see figure 1.2. NA is slightly longer than HA and may be distinguished in longitudinal sections by its shorter head (the head of HA has a characteristically bi-lobed “peanut” shape) and longer stem (compare figures 1.2(b) and (c)) and in transverse sections by its square profile, as opposed to the triangular profile of HA (see figure 1.2(d)) [29]. In principle, such considerations can be translated into a method of associating with each structure a feature vector, which may then be used for the binary classification of spikes as being either HA or NA.

However, discussions with biologists suggested to us that there may be a single feature that can be used reliably for the classification process. That feature is the ratio of the width of the spike’s head to the width of its stem. The ratio appears to be greater for NA than it is for HA, as can be seen by comparison of figure 1.2(c) to figure 1.2(b). The methodology for obtaining an efficacious numerical value of this feature for each of the segmented spike structures is similar in its nature to the methodology presented in Subsection 9.7.1 of [56] for distinguishing various kinds of bone structures in the human foot and ankle. The approach consists of first finding the centroid and the first principal axis for each of the spike structures obtained by the reorientation process described in Subsection 4.3.1. The first principal axis for each spike structure was determined using the principal component analysis described in [36, 37]; and their centroids were obtained using Matlab; see the Example\textsuperscript{2} for reading a binary image into workspace and calculating centroids

\textsuperscript{2}http://www.mathworks.com/help/images/ref/regionprops.html
for connected components in the image. We now discuss the technical details of obtaining, for any spike, the “the ratio of the width of the spike’s head to the width of its stem”.

As stated above, we assume that the spike structure has been already reoriented as described in the previous subsection. Consequently, the positive $x_2$ direction is approximately from the stem to the head of the spike. We also assume that we have identified the first principal axis and the centroid for the spike structure in question; the first principal axis is illustrated in figure 4.5(a). The location of the centroid is indicated by a blue cross in figures 4.5(b), (c) and (d).

Below we describe a way of rotating the spike structure so that centroid does not move but the first principal axis moves into a position parallel to the $x_2$ axis, with the positive $x_2$ direction from the stem to the head of the spike. This is done in two stages that are illustrated in figures 4.5(b) and (c); details are given below. For any point on the first principal axis, the area of intersection of the spike with a plane perpendicular to the first principal axis and containing that point is defined to be the cross-sectional area of the spike associated with that point. The maximal value of these cross-sectional areas is then defined to be the “width of a spike’s head”. The “width of the stem” is defined to be the median value of all cross-sectional areas between the centroid and the extreme bottom (furthest from the head) point on the first principal axis of the spike structure. The locations of the cross-sections at which these widths are measured are indicated in figure 4.5(d) in green for the head and yellow for the stem.

We now give the details of the procedure for rotating a reoriented spike structure so as to make its first principal axis lie in the direction parallel to the $x_2$ axis without moving its centroid. The procedure is a sequence of two actions, illustrated in figure 4.5. The actions are specified in a local coordinate system whose origin is the centroid and whose axes $x'_1, x'_2, x'_3$ are parallel to the global axes $x_1, x_2, x_3$. 
Figure 4.5: Feature extraction steps for the 23rd spike structure.

The first action is a rotation of the spike structure by less than $\pi/2$ around the $x_1^l$ axis so as to bring the first principal axis to the plane $(x_1^l, x_2^l)$. This rotation (of a set of voxels in the cubic grid into a new set of voxels in the cubic grid) is done in a manner similar to how it is done for reorientation, see the previous subsection. Figure 4.5(b) displays the $(x_1^l, x_2^l)$-slice of the 23rd spike containing first principal axis and centroid (blue) after the first action. Note that due to the reorientation of
Figure 4.6: Feature extraction for (a) the 11th, (b) the 12th, and (c) the 16th spike structures.

the spike structure prior to the first action and the rotation being less than $\pi/2$, it is still the case after the first action that the positive $x_2$ direction is approximately from the stem to the head of the spike.

The second action is a rotation of the resulting spike structure by less than $\pi/2$ around the $x_3$ axis so as to align the first principal axis with the $x'_2$ axis. Figure 4.5(c) displays the $(x'_1,x'_2)$-slice of the 23rd spike that contains the first principal axis after the second action.

Let $R_{\text{head/stem}}(s)$ denote the feature we are seeking for the spike $s$; it is the ratio of the spike’s head cross-sectional area to its stem cross-sectional area. The green and yellow lines in figure 4.5(d) show the locations of the cross-sectional areas of the spike’s head and the spike’s stem for the 23rd spike structure. Results for other spike structures (namely, for the 11th, 12th and 16th) are displayed in figure 4.6. The feature values for the four spike structures illustrated in figures 4.5 and 4.6 are $R_{\text{head/stem}}(23) = 4.42$, $R_{\text{head/stem}}(11) = 18.95$, $R_{\text{head/stem}}(12) = 19.66$, and $R_{\text{head/stem}}(16) = 3.43$. 
4.3.3 Classification

The feature presented in the last subsection seems to be sufficient for the task at hand. The feature associates with each spike structure a single number and, we believe, the numbers associated with the NA spikes are greater than the numbers associated with the HA spikes. If this is correct, then the classification becomes a straightforward matter. We should be able to classify by finding a threshold that optimally separates these two sets of numbers. A methodology for finding such a threshold is described below.

We use the Fisher’s linear discriminant to set the optimal threshold. We arrange all distinct numerical values of the feature obtained in the previous subsection in increasing order and then we choose as threshold candidates the values that are halfway between two consecutive values in that order. Each threshold candidate divides the set of all feature values into two classes (class 1 for below the candidate and class 2 for above the candidate); we select as the threshold \( \tau \) to be used for classification the candidate that maximizes the Fisher’s linear discriminant [24], which is defined by

\[
J = \frac{|m_1 - m_2|^2}{s_1^2 + s_2^2},
\]

where \( m_i \) and \( s_i^2 \) represent the mean and the variance of the class \( i \), respectively. The Fisher’s linear discriminant values are plotted for each of threshold candidates in figure 4.7(a) and the \( R_{\text{head/stem}} \) numerical values, in increasing order, are displayed in figure 4.7(b). The maximizer of the Fisher’s linear discriminant is the threshold \( \tau = 8.29 \); it is marked by the red line in figure 4.7(b). Figure 4.8 displays a reoriented vertical planar section (see Subsection 4.3.1) for each spike structure, \( s \), along with its feature numerical value \( R_{\text{head/stem}}(s) \).

Using the feature \( R_{\text{head/stem}} \) and the threshold \( \tau \), spike structures can be divided
Figure 4.7: (a) Fisher’s linear discriminant $J$, see (4.1), plotted against the threshold candidates. (b) $R_{\text{head/stem}}$ numerical values in the increasing order. The threshold $\tau$ that maximizes the Fisher’s linear discriminant is marked by the red line in (b).

into two classes $c_1$ and $c_2$:

$$c_1 = \{ s | R_{\text{head/stem}}(s) \leq \tau \} ,$$

(4.2)

$$c_2 = \{ s | R_{\text{head/stem}}(s) > \tau \} .$$

(4.3)

From figure 4.7(b) we notice that ten (out of forty-four) spike structures have the $R_{\text{head/stem}}$ values greater than the threshold $\tau = 8.29$; these structures are classified as being in class $c_2$; their numbers can be deduced from figure 4.8; they are the 4th, 11th, 12th, 14th, 17th, 25th, 32nd, 35th, 38th and 40th spikes. The remaining structures are then classified as being in class $c_1$. The classification for all spike structures using the threshold $\tau = 8.29$ is illustrated in figure 4.9; spikes that belong to the class $c_1$ are displayed in yellow while spikes from the class $c_2$ are in red. Taking into account discussions with biologists and analyzing the shapes and amounts of spikes in each class, we strongly believe that the classes $c_1$ and $c_2$ correspond to HA and NA spikes, respectively. This claim will be further strengthened in the next chapter (in Subsection 5.1.4).
Figure 4.8: Reoriented vertical planar sections of all spike structures along with their feature values $R_{head/stem}$. 
Figure 4.9: Virus spikes classification. Spikes that belong to the classes $c_1$ and $c_2$ are colored with yellow and red, respectively.
Chapter 5

Evaluation

5.1 Evaluation Methodology

Many things have been proposed and discussed in this dissertation. A number of them are brought together in this chapter, in which we investigate, for the ultimate task of virus spike classification, as described in Chapter 4, the relative efficacy of various 3D electron microscopic reconstruction methods that were the subject matter of Chapters 1-3 (specifically, the methods ART, SIRT, and WBP and, in the case of ART with the use of three different kinds of basis functions: voxels, blobs with the standard parameters and blobs with the desirable parameters). For this we adopt the same statistical hypothesis testing methodology that is used in Chapter 5 of [31]. This methodology consists of four steps:

1. Generation of 3D phantoms of viruses and computer simulation of the data collection.

2. Reconstruction from the data so generated by each of the algorithms.

3. Assignment of a figure of merit (FOM) to each reconstruction. The FOM should be a measure of the helpfulness of the reconstructed image for solving the classification problem.
4. Calculation of statistical significance by which the null hypothesis that two of the reconstruction methods are equally efficacious for classification can be rejected in favor of the alternative hypothesis that the one with the higher average FOM is more efficacious.

5.1.1 Generation of Phantoms and Their Projection Data

We created 3D test phantoms of viruses with randomly arranged HA and NA conformations; the proportions of HA and NA spikes in each phantom are approximately 75% and 25%, respectively. Phantoms were created using the already mentioned-software package jSNARK. Just as suggested in Section 1.2, the size of the reconstruction region is $200 \times 200 \times 200$ voxels. (Since in this chapter we deal with purely computer simulations, there is no need to specify the exact physical size of a voxel. As before, we use $\sigma$ to denote the length of the edge of a voxel; see, (1.5), (1.6) and figure 1.4(a).) The matrix envelop was simulated as a ball of radius $R = 70\sigma$, with the spikes comprising cylinders and ellipsoids as follows. HA heads were simulated as ellipsoids of semi-axes $3.5\sigma$, $3.5\sigma$ and $6\sigma$ (see figure 5.1(a)), while cylinders of radius $2\sigma$ and height $10\sigma$ were used to simulate the HA stems. Similarly, ellipsoids of semi-axes $6\sigma$, $6\sigma$ and $3\sigma$ were used to simulate NA heads (see figure 5.1(b)), while cylinders of radius $1.5\sigma$ and height $16\sigma$ were used for NA stems.

As mentioned earlier in Section 1.1, the number of surface spikes in a typical 120 nm diameter type A influenza virion is lower than 375. For that reason we decided to use 337 random spikes in each phantom; all of them extend radially from the virus surface. The distribution of these spikes was treated as follows. Spikes were placed in the central slice at 45 equidistant points over the slice’s perimeter of length $2\pi R$. Spikes in the other slices were placed in a proportional manner (the ratio of the number of spikes in a slice to its perimeter is approximately the same as
Figure 5.1: Ellipsoids simulating (a) HA spike’s head with semi-axes $u = 3.5\sigma$, $v = 3.5\sigma$ and $w = 6\sigma$ and (b) NA spike’s head with semi-axes $u = 6\sigma$, $v = 6\sigma$ and $w = 3\sigma$.

Figure 5.2: Spike distribution: (a) Lateral views of 3 slices where spikes are equidistantly placed and (b) 3D locations of random spikes in an illustrative example. The locations of HA and NA spikes are colored with yellow and red, respectively.
that for the central slice). This concept is illustrated in figure 5.2(a); the red line is a lateral view of the central slice and the blue lines are the locations of slices at polar angles $\pm \theta$. The spikes we used were placed in 11 slices; their locations are $\theta = 0^\circ$, $\theta = \pm 15^\circ$, $\theta = \pm 30^\circ$, $\theta = \pm 45^\circ$, $\theta = \pm 60^\circ$, and $\theta = \pm 75^\circ$. Figure 5.2(b) displays the exact 3D locations of the random spikes in each slice of an illustrative example; the locations of HA and NA spikes are colored with yellow and red, respectively.

After that, the same software package jSNARK was used to generate projection images from the phantoms that are similar to the kind of projection images that are obtained in practice by the data collection and data processing mechanisms used prior to 3D reconstruction, as described in Sections 3.1 and 3.2. In particular, specimen angles ranged between $-60^\circ$ and $+60^\circ$ with $2^\circ$ steps, producing 61 projection images of $200 \times 200$ pixels each (as intimated in Section 1.2). Gaussian noise of mean 0 and variance 1.1 was added to the purely mathematical projections in order to make the quality of the spike reconstructions from the simulated noisy data (figure 5.3) resemble that from real data (figure 3.7).

### 5.1.2 Reconstructions by the Algorithms to be Compared

We used the software package Xmipp to calculate 3D reconstructions from the data so generated by each of the following algorithms: ART using blobs with the desirable parameters as discussed in Chapter 2, ART using blobs with the standard parameters, SIRT using blobs with desirable parameters, ART using voxels, and WBP using the default parameters provided by Xmipp. A sample of a random phantom along with its 3D reconstructions are illustrated in figure 5.3. (Three of these algorithms were compared on real data of similar nature in Section 3.3, see figure 3.7.)
5.1.3 Assignment of an FOM to Each Reconstruction

It is, of course, possible to apply the virus spikes classification procedure of Chapter 4 to the outputs of each of the reconstruction algorithms; that will result in classifying the spike structures in the reconstructions as belonging to either class $c_1$ or $c_2$. Intuitively, we see that each of the spike structures in the reconstruction “come from” a spike of the test phantom that was created by us, and thus we can label it either as an HA spike or as an NA spike; this labeling provides us with the ground truth for the evaluation methodology.

Technically, the labeling of a spike structure in the reconstruction as HA or NA can be done as follows. The spike structure is one of the components of the foreground object (using face-adjacency) that was obtained by the segmentation
method described in Section 4.2. Consider now all the voxels of the 3D scene that belong to this component and look at the phantom definition; if more of those voxels are defined to be within an HA spike than within a NA spike, then we label the spike structure in the reconstruction HA, otherwise we label it NA.

The FOM we assign to each reconstruction is called classification purity ($CP$) [55]. Based on the description in [33], this evaluation measure is computed as follows. For every reconstruction, we create a $2 \times 2$ array of numbers, see Table 5.1, whose rows correspond to the two kinds of spikes (HA and NA) and whose columns ($c_1$ and $c_2$) correspond to the classes produced by the classifier, as described in Subsection 4.3.3. The numbers in the table are the number of spikes that have both properties (indicated by the row and the column) simultaneously. Table 5.1 summarizes the results of the classification applied to the output of the WBP algorithm that appears in figure 5.3(f).

Ideally, all elements of a class should come from the same kind of spike. We therefore define the classification purity $CP$ in % as: 100 times the sum over columns of the maximum of the entries in each column, divided by the sum of all the entries in the array. A more efficacious classification procedure should result in a higher value of classification purity. For the array in Table 5.1, the classification purity is $CP = \frac{100 \times (32 + 8)}{32 + 2 + 3 + 8} \%$, that is 88.89 %.

### 5.1.4 Calculation of Statistical Significance

In order to obtain statistically significant results, we sampled the ensemble of phantoms and the generated projection data a number of (namely, $C$) times. For the

<table>
<thead>
<tr>
<th></th>
<th>$c_1$</th>
<th>$c_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>HA</td>
<td>32</td>
<td>2</td>
</tr>
<tr>
<td>NA</td>
<td>3</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 5.1: Example of an array created to calculate the classification purity of a reconstruction.
Table 5.2: Average values of the classification purity over 30 data sets.

<table>
<thead>
<tr>
<th>Reconstruction</th>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ART using blobs with desirable parameters ($ART_{OPT}$)</td>
<td>97.48%</td>
</tr>
<tr>
<td>ART using blobs with standard parameters ($ART_{STD}$)</td>
<td>97.04%</td>
</tr>
<tr>
<td>SIRT using blobs with desirable parameters ($SIRT_{OPT}$)</td>
<td>94.07%</td>
</tr>
<tr>
<td>ART using voxels ($ART_{VXL}$)</td>
<td>92.07%</td>
</tr>
<tr>
<td>WBP using the default parameters ($WBP_{DFLT}$)</td>
<td>83.93%</td>
</tr>
</tbody>
</table>

experiments reported here, we used $C = 30$. We applied each of the five reconstruction algorithms that we listed in Subsection 5.1.2 to the same 30 data sets. For each reconstruction we calculated its classification purity and then, for each reconstruction algorithm, we averaged the 30 classification purity values provided by the 30 data sets. We report on the average values of this FOM over all phantoms and associated noisy projection data in Table 5.2. This table suggests that from the point of view of their efficacy for classifying surface spikes of viruses, the five algorithms may be rank-ordered as follows:

1. ART using blobs with desirable parameters;
2. ART using blobs with standard parameters;
3. SIRT using blobs with desirable parameters;
4. ART using voxels;
5. WBP using the default parameters.

A side observation here is that the high (such as 97.48 %) classification purity values in Table 5.2 of the virus spikes classification procedure we applied to the outputs of reconstruction algorithms strengthens the claim we made at the end of Subsection 4.3.3 (namely that the classes $c_1$ and $c_2$ correspond to HA and NA spikes, respectively).

We have not yet come to the main point of this subsection, which is the statistical significance of the results. The 30 data sets that were generated in order to produce
Table 5.2 were random: both in the choice of HA vs NA spikes and the noise in the simulated projections. The question arises: how confident are we that the reported relative ranking of two of the algorithms is in their essential nature (and thus very likely be observed over repeated experiments of the same kind) rather than just some accidental choice in the random data set used for the experiment. The following is a standard method for answering this question in a statistically rigorous fashion; see, for example, [31, Section 5.2].

Suppose that we wish to compare the performance of two (of the five) reconstruction algorithms from the point of view of their efficacy for spike classification. Let us assume, that the average classification purity value (call it $CP^1$) in Table 5.2 for the first algorithm is higher than that (call it $CP^2$) for the second algorithm. For $1 \leq c \leq C$, let $CP^1(c)$ and $CP^2(c)$ denote the values of the classification purity of the reconstructions by the first and second algorithms, respectively, from the $c$th data set. The null hypothesis that the two reconstruction methods are equally good for the task at hand translates into the statistical statement that each value of $CP^1(c) - CP^2(c)$ is a sample of a continuous random variable whose mean is 0 and whose probability density function is unknown. However, the central limit theorem tells us that, for a sufficiently large $C$ (and 30 is generally considered large enough),

$$s = \sum_{c=1}^{C} (CP^1(c) - CP^2(c)) = C (CP^1 - CP^2),$$

(5.1)

can be assumed to be a sample from a Gaussian random variable $S$ with mean $\mu_S = 0$ and variance

$$V_S = \sum_{c=1}^{C} \left( CP^1(c) - CP^2(c) \right)^2.$$

(5.2)

It is a consequence of the null hypothesis that $s$ is a sample from a zero-mean random variable. Recalling our assumption that $CP^1 > CP^2$, we have that $s > 0.$
This makes us suspect that in fact the first algorithm is better than the second one for the task at hand and so the null hypothesis may be false. The question is: how significantly large is the observed value \( s \) for rejecting the null hypothesis? To answer this question we consider the so-called P-value, which is defined to be

\[
P_S(s, \infty) = \int_s^\infty p_S(x) \, dx,
\]

(5.3)

where \( p_S \) is the probability density function of the Gaussian random variable \( S \)

\[
p_S(x) = \frac{1}{\sqrt{2\pi V_S}} \exp \left( -\frac{x^2}{2V_S} \right).
\]

(5.4)

The P-value is the probability of a sample of \( S \) being as large or larger than \( s \).

If the null hypothesis were correct, we would not expect to come across an \( s \) defined by (5.1) for which the P-value is very small. Thus, the smallness of the P-value is a measure of significance by which we can reject the null hypothesis that the two reconstruction algorithms are equally good for our task in favor of the alternative hypothesis that the first one is better than the second one.

### 5.2 Comparisons of Algorithm Efficacy for Spike Classification

Table 5.3 provides the P-values for pairwise comparisons algorithms in Subsection 5.1.4 measuring the significance by which we can reject the null hypothesis that the two reconstruction algorithms are equally good for classification in favor of the alternative hypothesis that the one with higher ranking is better.

Nearly all the P values in the table are very small, which means that the observed results are very significant because they are extremely unlikely to occur by chance if the null hypothesis were true. In particular, there is no question that one should
choose $ART_{OPT}$ rather than $ART_{VXL}$, $SIRT_{OPT}$ or $WBP_{DFLT}$, since in all three cases the P-values for rejecting the null-hypothesis of equally good performance (in favor of the alternative hypothesis of $ART_{OPT}$ being superior) are all less than $10^{-4}$.

On the other hand, the reported results for the $ART_{OPT}$ and $ART_{STD}$ reconstructions are not statistically significant. The P-value is 0.1444, which means that even if the null hypothesis that the two reconstructions are equally good were correct, there would be a 14.4% chance of observing $ART_{OPT}$ performing better than $ART_{STD}$. However, there is no reason for not using $ART_{OPT}$ as opposed to $ART_{STD}$ and so, at least for now, $ART_{OPT}$ is our recommended algorithm for the purpose of computerized classification of surface spikes in three-dimensional electron microscopic reconstructions of viruses.

<table>
<thead>
<tr>
<th></th>
<th>$ART_{STD}$</th>
<th>$SIRT_{OPT}$</th>
<th>$ART_{VXL}$</th>
<th>$WBP_{DFLT}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ART_{OPT}$</td>
<td>$1.4441 \times 10^{-1}$</td>
<td>$5.0598 \times 10^{-5}$</td>
<td>$1.1678 \times 10^{-6}$</td>
<td>$8.9435 \times 10^{-8}$</td>
</tr>
<tr>
<td>$ART_{STD}$</td>
<td>$1.3036 \times 10^{-4}$</td>
<td>$8.0137 \times 10^{-7}$</td>
<td>$1.7856 \times 10^{-7}$</td>
<td></td>
</tr>
<tr>
<td>$SIRT_{OPT}$</td>
<td></td>
<td>$5.5433 \times 10^{-3}$</td>
<td></td>
<td>$7.5194 \times 10^{-7}$</td>
</tr>
<tr>
<td>$ART_{VXL}$</td>
<td></td>
<td></td>
<td></td>
<td>$4.3642 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

Table 5.3: P-values for pairwise comparisons of reconstruction algorithms.
Chapter 6

Discussion

In this final chapter we discuss the work that has been done for this dissertation, including indications of its significance.

6.1 Printed Publications Based on Work for This Dissertation


6.2 Workshop Presentations Based on Work for This Dissertation


6.3 Summary of the Contributions of this Dissertation

Influenza is a rapidly changing virus that manifests itself seasonally in the human population. Every few years a new strain of the influenza virus appears and causes a serious global pandemic. Knowledge of the structure and density of the virus surface proteins is of critical importance in a vaccine candidate. Each season, the

³http://www.cuny.edu/research/news-events/StructuralBiologyWorkshop.html
vaccine must be re-engineered to match the current influenza strains with rapid production capability.

The purpose of this research is to develop computer techniques for improved 3D reconstruction of viruses from electron microscopic images of them and for the subsequent improved classification of the surface spikes in the resulting reconstruction. Therefore, our first task was to develop a reconstruction procedure that can produce high quality reconstructions of viruses from electron-tomographic data and our second task was to develop a classification procedure that can be applied to such reconstructions to provide a reliable classification of the surface spikes.

With respect to the first task, we demonstrated in our research the usefulness of blobs in 3D reconstructions of viruses from electron microscopic images. We investigated the selection of blob parameters using an extra degree of freedom that has been ignored previously. Using that extra degree of freedom, we produced a family of blob parameters for accurate representation of images. We then investigated how well the various members of this family can be utilized for representing the image of a ball. We showed that there is a trade-off between the blurring of the edge of the ball and the magnitude of local oscillations in the representation. We generalized this approach to representation by blobs first piecewise-constant and then arbitrary 3D images. Based on a deeper mathematical analysis and experimental demonstrations of various choices we ended up with providing a new technique for optimizing parameters for 3D image representation and reconstruction using blob basis functions. It has been demonstrated that, with the recommended data-processing steps performed on the projection images prior to reconstruction, the reconstruction algorithm ART with the blobs that we advocate provides 3D reconstructions of viruses from tomographic tilt series that allow reliable quantification and identification of the surface proteins, which is a valuable tool for the selection of useful viral strains for successful manufacture of vaccines.
Regarding the classification aspect, the process we built assigns a category to a protein spike on the basis of its shape. The objectives of automating the classification are to make the process more reliable and reproducible by making the same choices under the same conditions. The quantification of the influenza surface spikes was made using a fuzzy connectedness technique; this sophisticated technique has been successfully used to segment an object from a background especially when the image is corrupted by noise and/or shading. Individual spike structures were extracted by partitioning the segmented object into components using face-adjacency; then reoriented to settle all of them into the same framework. The differentiation between the two types of surface spikes, HA and NA, was achieved by using a single feature; which is the ratio of the width of the spike’s head to the width of its stem in 3D space; the ratio appears to be greater for NA than it is for HA. The proposed classifier was tested on different types of 3D reconstructions derived from simulated data. A statistical hypothesis testing based methodology allowed us to evaluate the relative suitability of reconstruction methods for the given classification task.

We note that in this research we limited the 3D scene of the region that contains spikes (to be classified) to the central slices only. A direction for further investigation is to extend the region of interest to include slices that are far from the central ones.

In this dissertation, we brought together several tasks related to image processing and the computational aspects of electron microscopy; these tasks include image representation, reconstruction from projection images, data alignment, segmentation, feature extraction, classification and algorithm evaluation. Major contributions of the thesis are (1) a new approach to how blob parameters should be selected for both 3D image representation and reconstruction and (2) a complete set of programs to get us from electron-microscopic projections of a virus to a classification
of its surface spikes.
Bibliography


Index

3D image, 9
3D reconstruction techniques, 2
3D scene, 1, 15

A
affinity function, 71
algebraic reconstruction technique, 2, 11, 44
antigenic determinant, 5
ART, 2, 11, 47, 63
atomic force microscopy, 7

B
basic basis function, 9
basis function, 3, 9, 12, 23, 30
bcc grid, 15, 20, 24
blobs, 3, 4, 13, 15, 19, 20, 23, 30, 47, 63
body-centered cubic grid, 15

C
centroid, 80
chain, 71
classification, 84
classification purity, 93
coefficients, 3, 9, 11, 44

coulomb potential, 2
cryo-EM tomography, 6
cryogenic electron-microscopic tomography, 6

electron microscopy, 1, 13
electron tomography, 2
EM, 1
ET, 2

F
face-centered cubic grid, 15
fcc grid, 15, 24, 37
feature, 80
Fisher’s linear discriminant, 84
fuzzy connectedness, 70

G
Gaussian random variable, 95
glycoprotein spikes, 5, 66
gray value, 60
grid, 9, 15, 30
transform method, 2

U
UCSF Chimera, 77

V
virus, 5
voxel, 1, 12, 15, 19, 68
voxel basis function, 12, 14

W
WBP, 2, 11, 64, 67
weighted backprojection, 2

X
Xmipp, 14, 64
X-ray crystallography, 5