COMBINATORIAL CLASSIFICATION TO SEPARATE HOMOGENEOUS SUBSETS OF HETEROGENEOUS PROJECTION SETS

by

MIROSLAW KALINOWSKI

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

2007

UMI Number: 3284420

UMI Microform 3284420

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

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. Theodore Brown, Executive Officer

Dr. Joachim Frank

Dr. Robert M. Haralick

Dr. Katherine St. John
Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

COMBINATORIAL CLASSIFICATION TO SEPARATE HOMOGENEOUS SUBSETS OF HETEROGENEOUS PROJECTION SETS

by

Miroslaw Kalinowski

Adviser: Gabor T. Herman

Some important image classification problems cannot be solved using standard techniques. The problem of classification of heterogeneous electron microscopic projections into homogeneous subsets, found in three-dimensional electron microscopy (3D-EM), belongs to this category. One important 3D-EM technique (the so-called single particle reconstruction) used to visualize complex 3D molecular structures relies on the assumption that thousands of projections of identical specimens are available. However, it is often the case that the macromolecule of interest appears in the projection set in several (not-exactly identical) conformations. The single particle method applied to such heterogeneous sets is unable to provide useful information about the encountered conformational diversity and produces reconstructions with severely reduced resolution. This limitation is a significant stumbling block to understanding molecular function (especially its dynamic aspects). One approach to solving this problem is to partition heterogeneous projection set into homogeneous components and apply existing reconstruction techniques to each of them. Due to the nature of the projection images and the high noise level present in them, this classification task constitutes a challenging computer science problem. A method is presented to achieve the desired classification by using a novel image

dissimilarity measure and finding an approximate Max k-Cut of an appropriately constructed weighted graph. Despite of the large size (thousands of nodes) of the graph and the theoretical computational complexity of finding even an approximate Max k-Cut, we propose an algorithm that finds a good (from the perspective of 3D-EM projection image classification) approximate solution within several minutes (running on a standard PC). Due to the large number of edges the task of constructing the complete weighted graph (that represents an instance of the projection image classification problems) is computationally expensive. To reduce this cost we also propose a method, which utilizes an early-termination technique, to significantly reduce the computational cost of constructing such graphs. Unlike the majority of competing approaches, the presented method employs unsupervised classification (it does not require any prior knowledge about the objects being classified) and does not involve a 3D reconstruction procedure. The performance of the proposed classification method is evaluated on synthetically generated data sets produced by projecting 3D objects that resemble biological structures.

to my Mom, my first teacher

Acknowledgments

I would like to express my great gratitude to all the people and the institutions who helped me in various ways in my pursuit of the doctorate. The encouragement, advice and support I have received from them over the years were essential to the completion of my work.

A special credit must be given to my advisor, Gabor T. Herman, for his ideas that shaped my research. I truly admire his knowledge, enthusiasm and dedication to work which all greatly contributed to all the projects we have done together. I deeply appreciate Gabor's willingness to guide me to the very end of Ph.D. candidacy. I thank him for always being accessible and responsive.

I greatly appreciate the time devoted to my work by the members of the examining committee. Specifically, I want to thank Joachim Frank for explaining to me the problem of classification in 3D electron microscopy and making sure that my research remains in touch with the practical realities of this field, Robert M. Haralick for his numerous comments about the scientific methodology, and Katherine St. John for pointing out that the generalized version of maximum capacity cut problem I have encountered in my research is known as Max k-Cut problem.

I thank Andreas Alpers, Jesse Barlow, Dan Butnariu, Yair Censor, Alain Daurat, Ivan Kazantsev, Attila Kuba, Doug Moody, Sjors Scheres for their participation in the discussions related to my research. In particular, I thank Alain Daurat for his great work on improving the performance of graph building procedure, and Sjors Scheres for teaching me Xmipp and providing one of the datasets used in Chapter 7.

I also would like to thank Tom Wesselkamper and Joe Driscoll for their invaluable assistance in my early days at CUNY.

Most importantly, I am grateful to my wife, Joanna, for her encouragement, support and infinite patience.

My research was supported by grant HL70472 from National Institutes of Health, the National Science Foundation, and fellowships from The City University of New York.

All 3D reconstructions were produced using Xmipp developed by Biocomputing Unit (BCU) of the Centro Nacional de Biotecnologia CNB (National Center for Biotechnology).

All images of 3D reconstructions were produced using the UCSF Chimera package from the Resource for Biocomputing, Visualization, and Informatics at the University of California, San Francisco (supported by NIH P41 RR-01081).

Contents

List of Tables

List of Figures

Introduction

Many engineering, scientific and other procedures involve the task of classifying images. Because of this, image classification problems have a long history in the Computer Science literature [10]. Since the objectives and difficulty of these problems varies widely, different classification criteria and methods have been developed to solve them. However, despite the wide range of methods available to us, some important problems involving image classification can be solved only by new, specially constructed, classification methods that take advantage of domain specific properties.

The analysis of macromolecular complexes and their dynamics is one of the intensively researched topics in molecular biology. The objective of this research is to understand the structure and function of molecular machines. The three-dimensional reconstruction from electron-microscopic images (3D-EM) of macromolecular complexes plays an essential role in these efforts. Since the delicate nature of macromolecular complexes puts a significant restriction on the process of obtaining projection images with an electron microscope, the task of reconstructing 3D models of macromolecular complexes from electron-microscopic images is difficult. A large body of knowledge has been developed to address many problems encountered in 3D-EM of biological molecules. Some important and challenging scientific problems related to 3D-EM of biological objects remain open.

The limited number of the projections that can be obtained from a single spec-

imen and the extremely high level of noise present in them are among the primary causes of difficulties associated with 3D-EM of macromolecular complexes. Despite these (and many other) difficulties, successful reconstruction methods that produce high-quality 3D reconstructions of macromolecular complexes have been developed. These methods employ averaging (either at the level of the 2D images or of the 3D model) to reduce the impact of noise present in the projection images on the quality of the 3D reconstructions. They solve the problem of obtaining the number of projection images needed to produce a high-quality 3D model of a complex object by treating thousands of projections obtained from different specimens as if they were projections of the same object. Such techniques are justified as long as a homogeneous projection set that contains thousands of projections of identical specimens is available. The reconstruction methods that employ these techniques have been successfully used to produce 3D structural models of many biological molecules [12].

Due to the dynamic nature of molecular processes, it is quite common that a molecule has several different conformations, that are not separable prior to the process of obtaining projections. This results in heterogeneous projection sets (sets containing projections of more than one conformation). A reconstruction procedure that involves averaging is inherently unsuitable to deal with such structural heterogeneity. The averaging across several conformations severely limits the achievable resolution and results in loss of information about the individual conformations, and hence may prevent the understanding the function of macromolecule under study.

Many limitations of the existing 3D-EM reconstruction procedures associated with heterogeneity of projection sets can be avoided by partitioning the heterogeneous projection sets into their homogeneous components and performing separate reconstructions from the homogeneous sets obtained as result of such a classification. However, due to the nature of the projection images and the high level of noise

present in them, this classification task constitutes a challenging computer science problem, which cannot be solved using the traditional classification techniques.

The work presented here focuses on utilizing the mathematical properties of projection images and a combinatorial optimization technique to devise a new method capable of classifying heterogeneous electron microscopic projections into homogeneous subsets. This classification method extends the applicability of reconstruction procedures used in 3D-EM and thereby increases the scientific value of reconstructions produced by them.

The remainder of this document is structured as follows.

Some basic information about 3D-EM processes and about methods directly related to the problem of constructing 3D models of large molecules from the images obtained by an electron microscope is provided in Chapter 1. The notion of heterogeneous projection sets is explained and the difficulties associated with them are described. The intended application environment for the proposed classification procedure and the significance of the heterogeneity problem are also discussed there.

In Chapter 2 a novel image dissimilarity measure specially designed for 2D projections of 3D objects is proposed. A mathematical justification and a description of a practical implementation are provided.

A reformulation of the image classification as a global optimization problem and its graph theoretical interpretation as a Max k-Cut problem [25] are explained in Chapter 3.

Since for the large data sets the construction of the corresponding graphs using a brute force approach is computationally expensive, significant efforts have been made to develop a technique that produces such graphs at a smaller cost. These efforts are described in the Chapter 4.

In Chapter 5 a new graph cutting algorithm, designed to find approximate so-

lutions to these instances of the Max k-Cut problem that arise from the desire to partition heterogeneous projection sets found in 3D-EM into their homogeneous subsets, is presented.

In Chapter 6 the general framework of the proposed classification method, that ties together previously introduced components, is defined.

The explanation of methodology used to evaluate the proposed classification method and the results such evaluations are provided in Chapter 7.

In Chapter 8 the main contributions of our research are discussed and important topics for future research are suggested.

Chapter 1

Background

There are many methods for reconstructing a 3D object (mathematically defined as a real-valued function of bounded support on the 3D Euclidean space) from its line integrals [21, 31]. Typically, the line integrals are estimated for a set of parallel lines using some instrument and are represented in the form of 2D projection images. A set of such images can be processed by a reconstruction procedure to produce a 3D model of the object from which the 2D images where obtained (see Figure 1.7). These type of 3D reconstruction methods have many practical applications; the best known of which is CT (computerized tomography) that is widely used in diagnostic radiology.

In some respects, the 3D-EM of macromolecular complexes is similar to other applications of 3D reconstruction methods. For example, its main objective of producing a 3D model from line integrals (in the case of 3D-EM given in form of 2D projection images) is the same as in other applications. However, there are some fundamental differences between 3D-EM of macromolecular complexes and other applications of 3D reconstruction methods. Usually, the process of obtaining projections provides precise spatial information (i.e., a common reference point for all the projection images and the Euler angles from which they were taken are known

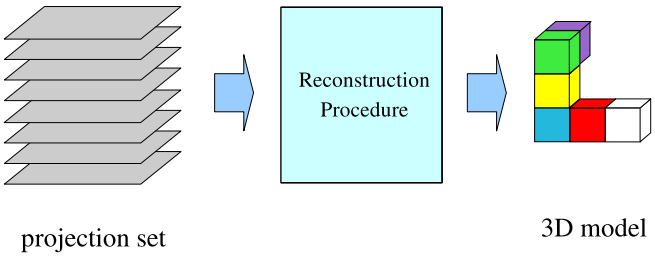


Figure 1.1: The 3D reconstruction procedure.
A set of 2D projection images is used to produce a 3D model of the object from which these projections images were obtained.

with high accuracy.) In most of the applications there is no restrictions that significantly limit the number of the projection images that can be taken, so it is relatively easy to obtain a projection sets large enough to produce high-quality reconstructions of the complex 3D object. Usually, the quality of the projection images that are used in the 3D reconstruction procedure is quite good (the images are practically free of noise and other distortions). The case of 3D-EM of macromolecular complexes is different from these points of view.

Despite the many challenges posed by the 3D-EM of macromolecular complexes, successful reconstruction methods applicable to this task have been proposed and a large body of knowledge has been developed concerning them. Detailed discussion of all the problems and methods associated with 3D-EM of macromolecular complexes is beyond the scope of this document; a comprehensive study of them can be found in [12]. However, a brief (and frequently simplified) description of issues and techniques directly relevant to our research is provided this chapter.

1.1 Single-Particle Reconstruction

There are two major strategies for reconstructing 3D models from the projection images obtained with an electron microscope: a tomographic approach (in which

the relative projection angles from which images were taken are known), and single particle approach [11] in which the projection angles are not known and must be determined by either the random-conical [29] or the angular reconstitution procedure (see [47]). Since, our classification method is designed to extend the applicability of single-particle approach, a brief description of issues related to this approach provided in this section.

One of the biggest problems associated with reconstructing 3D models of macromolecular complexes form electron-microscopic images is the destructive nature of the process of obtaining images of biological objects using an electron microscope. In order to produce a projection image that corresponds to the undamaged macromolecule the number of electrons interacting with it must be small. As result the electron-microscopic images of macromolecular complexes are very noisy and only very few of them can be produced from a single molecule.

A large number of projection images from a variety of angles is needed to produce a high-quality reconstruction of a complex object using a 3D reconstruction procedure. This number must be even larger when the projection images used in such a procedure are corrupted by a very high level of noise (as in the projection images of macromolecular complexes obtained with a electron microscope). Since only very few (very noisy) projection images can be produced from a single specimen using an electron microscope, these images cannot provide information that is sufficient for reconstructing a high resolution 3D structural model of the molecule. However, such model can be produced if the projection images obtained from many identical molecules occurring in random orientations are used in a single reconstruction process [29, 30]. This technique is refereed to as the single particle approach.

The process of obtaining projections for the single particle reconstruction method is schematically shown in Figure 1.2. In this process an image (micrograph) is pro-

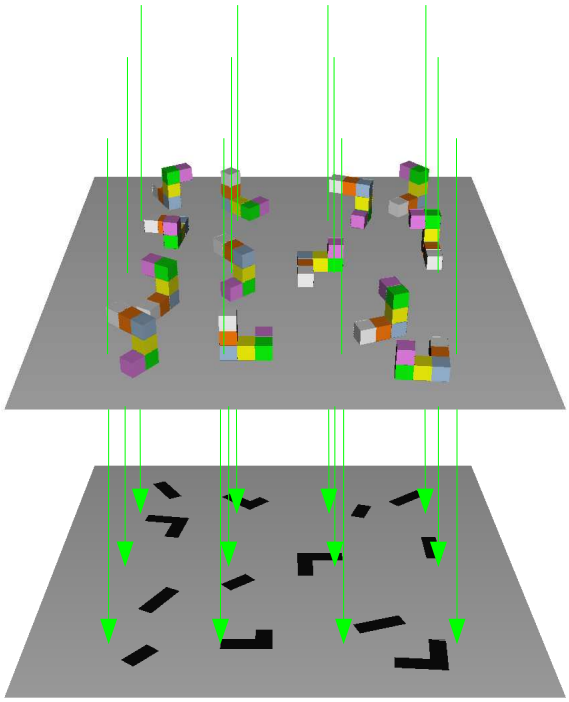


Figure 1.2: Process of obtaining projections.
Many identical randomly oriented molecules are simultaneously projected.

duced by recording changes in an electron beam that interacts with an appropriately prepared sample. An example of such a micrograph is shown in Figure 1.3.

The physics of producing micrographs using an electron microscope is complex. Many efforts have been made to understand this process and to construct instruments and procedures capable of producing high-quality micrographs. Despite of the steady progress in these efforts, the micrographs of macromolecular structures obtained by an electron microscope are far from perfect. Some of these imperfections (like high level of noise) are due to the limitations imposed by the delicate nature of macromolecules. Some are the result of less than ideal instruments and preparation procedures. Others are caused by the limitations inherent to the EM process. Apart from the high level of noise, the most significant (from the 3D reconstruction perspective) distortion that the micrographs suffer from is asso-

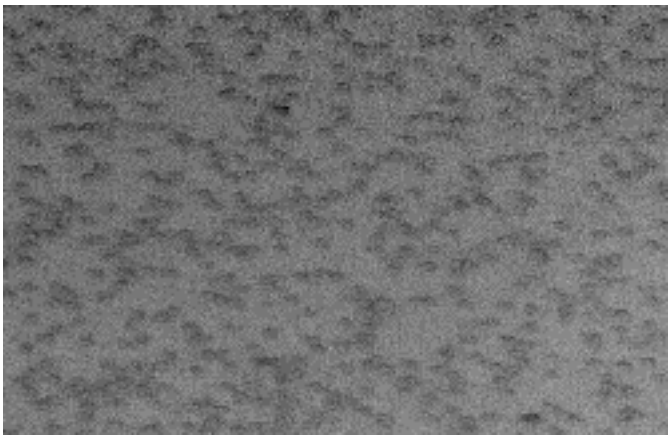


Figure 1.3: EM micrograph.
Each dark spot is a 2D projection of the molecule.

ciated with the electron microscopic contrast transfer function (CTF) [12]. Usually the impact of the CTF is significantly reduced by subjecting the micrographs to a CTF correction procedure [13]. However, some residual distortions due to CTF remain in micrographs even after such correction is performed.

Since many instances of a particular molecule are present in a preparation, many projections are simultaneously captured in a single micrograph. Small parts of the micrograph, which are likely to a contain a projection image of a single molecule, are isolated and treated as separate projection images. Ideally, each of these images should represent a perfectly aligned 2D projection of a single undamaged molecule. Due to the noise, residual CTF related distortions, discretization errors and other factors, even the best images obtained in this process only approximate the ideal projections (in the mathematical sense) of the specimens present in a sample. The high level of noise and (to much smaller extend) the discretization errors cause that a perfect alignment (i.e., one in which the center of mass of the molecule is projected exactly to the center of the image) is impossible to achieve. Due to the damage, sustained by molecules , during the preparation procedure and the overlaps of molecules in the sample, many of the projections are unsuitable for further processing. Such images should be identified and removed from the projection set.

This task is difficult to automate and is often performed manually.

In order to produce a high resolution 3D model of a macromolecule, the reconstruction procedure must combine the projection images obtained from many micrographs. One reason for this is due to fact that the information associated with some ranges of spatial frequencies is lost as result of the CTF. In order to avoid noise amplification the CTF correction procedure must be applied in a way that does not allow to recover the information associated with frequencies significantly weighed down (high frequencies and frequencies in regions near the zeros of CTF) . However, depending on the particular CTF, the information loss ours in a different ranges of spatial frequencies. Since the CTF of a microscope can be changed by adjusting the defocus, micrographs can be produced with different ranges of missing spatial frequencies. Combining in a single reconstruction procedure the projection images from different micrographs (produced with different defocus settings) allows the production of a high resolution 3D model that incorporates information associated with all spatial frequencies [36]. Another reason for using images from many micrographs in a single reconstruction procedure is due to the fact that the achievable resolution of the 3D model depends on number of the projections that are used in the reconstruction process [9]. In some case the number of image that can be obtained from a single micrograph is not large enough to achieve the desired resolution.

The main difficulty inherent to the single-particle method is the lack of information about the spatial coordinates of the projection images. Since there is now way to control the orientation of the molecules in the sample, the angles from which these projection images are taken are unknown. Depending on the shape of the molecule these angles may or may not be uniformly distributed. (Some molecules may have one or few “preferred” orientations that are more likely to be represented in a micrograph). Usually, the unknown projection angles are estimated under the

assumption that all projection images are of the same object. (Consistency with such an assumption puts mathematical restrictions on the set of angles that may be associated with the projections.) Thousands of 2D projection images and their estimated angles are used as an input to the reconstruction algorithm, which returns an estimate of the 3D structure from which the projection images were obtained.

Despite the many difficulties, reconstruction procedures based on single particle method have been successfully used to obtain the 3D structure of many biological molecules [12]. The resolution of these reconstructions has steadily improved as better instruments and techniques have become available. However, further improvements in achievable resolution are limited by the heterogeneity found in the projection sets.

1.2 Heterogeneity

In the context of single particle EM, the term heterogeneity refers to the issues associated with the samples containing molecules that are not identical. As the molecules perform their biological functions, they frequently change their 3D structure. Since such changes do not occur simultaneously in all specimens, a large sample is likely to contain molecules in different conformations.

There are different types of heterogeneity, which might occur simultaneously. In some cases differences between conformation are associated with the ability of molecules to rearrange the position of their components. Such movements are essential to the operation of many molecules, which act like “miniature mechanical devices” while performing their functions. The heterogeneity also occurs in samples that contain molecules that bind to other objects. Such molecules can be found in different stages (before and after the binding occurred or with different object attached). Usually the overall shape of the molecules remains the same. However,

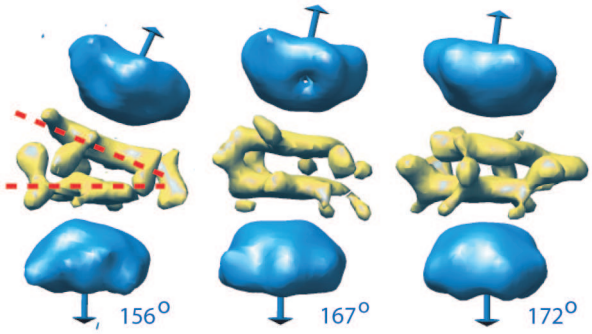


Figure 1.4: Heterogeneity - Three reconstructions Simian Virus 40 large T-antigen dodecamers with various degrees of bending along the central axes of the dodecamers.
The heterogeneous set from these models were reconstructed contained projection images of molecules with continuously varying bent [39].

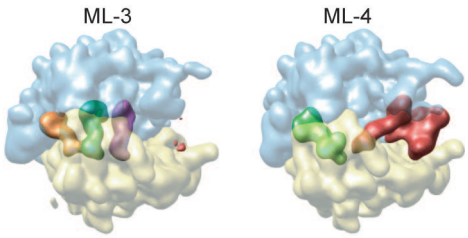


Figure 1.5: Heterogeneity - Two reconstruction of ribosomes.
The refined structures determined by maximum-likelihood optimization (ML) [39]. Unratcheted ribosomes in complex with three tRNAs (orange, green, magenta) or ratcheted ribosomes in complex with one tRNA (light-green) and EF-G (red).

the bound molecules have relatively small (~5% of the main molecule mass) objects attached to them. The example of a molecule exhibiting heterogeneity are provided by Figure 1.4 and Figure 1.5.

For many specimens there is no feasible method to create a homogeneous sample containing molecules only in one of their conformational states. Because different conformational states of a molecule are very similar (mass, size, shape, chemical properties), in the general case there is no known physical or chemical method to sort them into homogeneous subclasses. A study of ribosomal complex EF-G·70S described in [17] is an example of the reconstruction effort in which various conformational states of a "molecular machine" are represented in the sample.

The traditional reconstruction methods used in 3D-EM have been designed with the assumption that the projection set they operate on is homogeneous (i.e., was produced from a sample that contains multiple copies of identical molecules). Since heterogeneous samples contain molecules that are not identical, in theory the traditional methods are inherently unsuitable to handle them. However, frequently the general shape of the molecule in different conformational states remains the same, so the assumption that all molecules contained in a sample are identical is approximately true even for heterogeneous samples. In practice, when differences between conformations represented in the heterogeneous sample are small, the problem of heterogeneity has been ignored. In such cases, the traditional reconstruction methods produced a 3D model that represents the general shape (average over all conformations) of the molecule represented in the sample. However, since the averaging over several conformation leads to reduced resolution [50], the problem of heterogeneity cannot be ignored when high resolution models are desired. Such models can be obtained only when different conformational states are treated separately.

The main problem encountered when the traditional reconstruction methods are applied to heterogeneous projection sets is the loss of information about the individual conformations present in the sample. The single 3D model produced by traditional methods does not provide useful information about the diversity of these conformations. Frequently such information is essential to understanding the function of macromolecule under study.

Some of the problems caused by heterogeneity can be avoided by converting the heterogeneous sample into a homogeneous one by using preparation technique that forces all the molecules present in sample into the same conformational state. This method has been successfully used in some reconstruction efforts. However, the appropriate preparation techniques exists only for a few types of the molecules. Even when such techniques are available, the use of them might have undesired conse-

quences. In order to study the dynamic aspects of macromolecular machines it is necessary to capture the molecules in the variety of their natural states. The existing preparation techniques allow us to enforce only a very few states that frequently do not correspond to the natural conformations. Consequently, the 3D models reconstructed from samples in which all the specimens were forced into the same conformation by the use of such preparation technique cannot provide the information that is essential to understanding the behavior and function of the macromolecules.

To illustrate issues associated with the heterogeneous projections sets we use three 3D structures, referred to as S6, S6x, S7 in Figure 1.6, which resemble objects previously utilized in the 3D-EM literature [38, 40] as three conformations of the same object. The exact specification of their geometry is provided in Appendix A. The use of objects for which the precise mathematical description is known was very helpful in many aspects of our research (problem analysis, development of our classification method and its evaluation.)

1.3 Significance

Many of the fundamental processes in the cell involve the dynamic interaction of macromolecules transiently forming complexes that have been termed "molecular

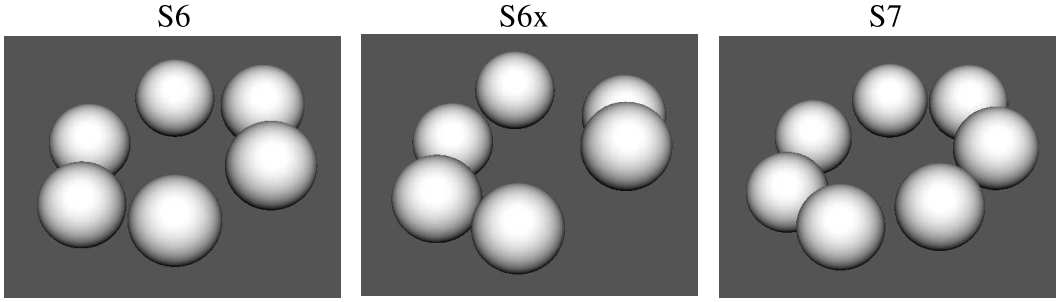


Figure 1.6: 3D objects S6, S6x, S7.

machines." Examples are protein synthesis (ribosome), transcription (RNA polymerase), protein folding (GroEL), and protein degradation (proteasome). Cryo-EM has contributed significantly to the understanding of the dynamic nature of each of these processes. One of the most recent examples is the elucidation of mRNA-tRNA translocation in eukaryotic protein synthesis [45]. The understanding of these processes is very important from the biological perspective and also may have profound implications in other disciplines (medicine, agriculture, technology etc.). However, even the complete information about the atomic level structure of a molecule may not be sufficient to understand its function and how this function is performed. Frequently, such knowledge cannot be obtained unless the changes in the shape of the molecule, essential to its function, are understood and taken into the account. The heterogeneity of samples, encountered in many 3D-EM reconstruction efforts, is an expression of the dynamic nature of the macromolecules. As indicated earlier in this chapter, such heterogeneity causes significant problem when the traditional reconstruction methods are used. However, since the diverse conformations present in heterogeneous samples are snapshots of the molecule at various stages of its operation, such samples carry valuable information. If, with the use of the appropriate tools, this information could be extracted, it would provide an important clues about the processes in which the molecule is involved and the ways in which the molecule operates.

1.4 Reconstruction from Heterogeneous Sets

Due to the limitations of the traditional methods and the difficulties associated with obtaining representative homogeneous samples, there is a need for reconstruction methods capable of handling heterogeneous samples. Such methods should convert a heterogeneous projection set into the set of 3D models that approximate confor-

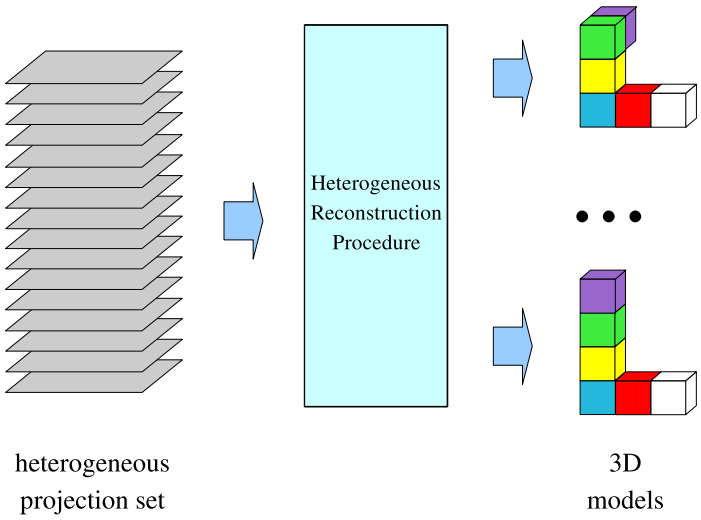


Figure 1.7: Reconstruction form heterogeneous set.
Reconstruction procedure returns 3D models of all conformations represented in the projection set.

mations represented by the projections. The process of reconstructing from heterogeneous projection set is depicted in Figure 1.7. Averaging across the conformations must be avoided in such process because it blurs the differences between models and harms their resolution.

Ideally, such reconstruction process should produce a set of 3D models that correspond to all conformations represented in the projections set. This might not be achievable in reality because some conformations might be represented by very few projections or two conformations might be very similar. The latter is especially likely when there is a continuum of conformations represented in the projection set. However combining two almost identical conformations should not be very harmful to the resolution of 3D model. Realistically, the heterogeneous reconstruction procedure should be expected to return the 3D models of all the significantly different objects that have sufficient representation in the heterogeneous projection set. Such models should not suffer from the resolution loss due to averaging over multiple conformations and should represent well the important differences between

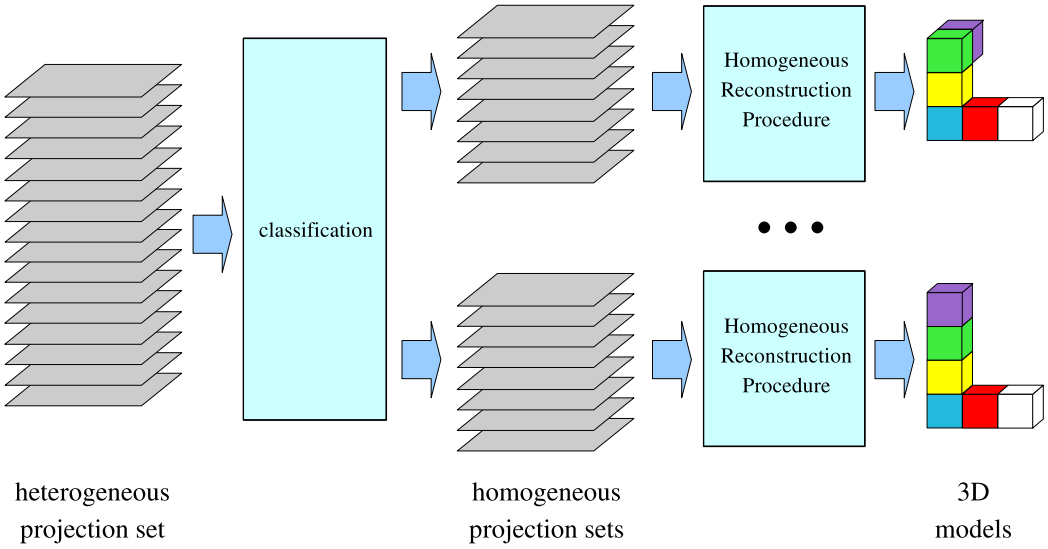


Figure 1.8: Classification-based approach to reconstruction from heterogeneous sets.

them. This will improve the achievable reconstruction quality from heterogeneous samples and give access to the information about the dynamic nature of molecules contained in them.

1.5 Classification-Based Approach

One approach to the heterogeneity problem is to is to employ a classification procedure to partition heterogeneous projection sets into homogeneous components (Figure 1.8). If a correct partitioning can be achieved, an independent reconstructions from each of the homogeneous components would produce models of all the conformations represented in the heterogeneous set. This set of models will constitute a solution to the problem of reconstruction from a heterogeneous set. This classification-based approach is appealing because it separates the issue of heterogeneity from that of reconstruction, allowing the use of existing reconstruction techniques without modifications. However, the partitioning of heterogeneous projection sets produced by EM into homogeneous components constitutes a difficult

classification problem.

The difficulty inherent in the problem of classifying projection images, comes from the fact that frequently a pair of images that belong to the same class (they are 2D projections of the same conformation) are far less similar to each other than another pair of images that belong to different classes. A traditional 2D image classification method would not consider the image pairs that are within the individual

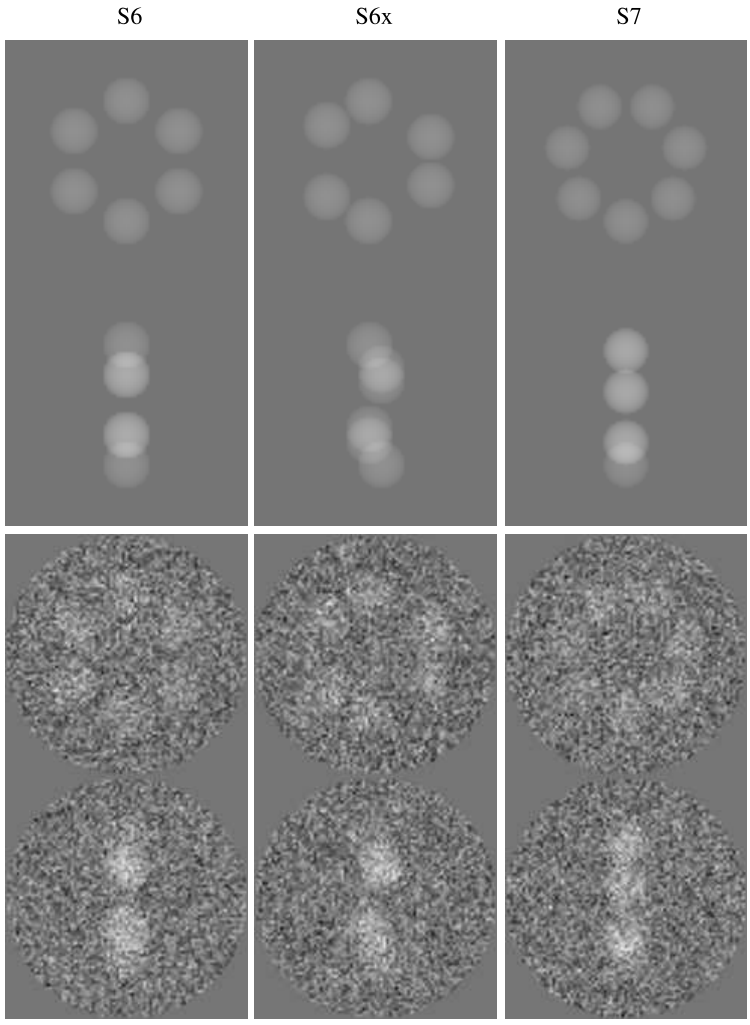


Figure 1.9: Similarity between projections of different conformations.
Top panels: top and side projections of the conformations S6, S6x and S7. Bottom panels: the corresponding top and side projections distorted by noise (SNR = 0.1). (All images use the same mapping from projection values into gray values; the two extreme projection values in the noisy images correspond to black and white.)

top panels of Figure 1.9 to be similar and would likely place them into different classes, even though the images of these pairs are 2D projections of the same 3D object in different directions. On the other hand, it would consider that the projection images in each row of Figure 1.9 as similar to each other, even though they are 2D projections of different 3D objects. When the projection images contain noise, the task of classifying them becomes even harder. The signal to noise ratio (SNR) of the EM projection images is very low. Figure 1.9 (bottom panels) shows the impact of noise (SNR = 0.1 as defined by Frank [12], p. 121) on projection image quality. Since the differences between projections in each row are even less visible, the probability of placing them into the same class increases.

1.6 Supervised versus Unsupervised Classification

The use of prior knowledge can significantly simplify some aspects of the reconstruction process and improve the quality of the results. In some 3D-EM reconstruction efforts, prior knowledge is available in the form of approximate 3D density maps. Frequently such maps are used to group together the projections taken from similar direction and to determine approximate values for the Euler angles associated with the projections. They also provide an initial reference for iterative reconstruction procedures. Many image classification procedures used in 3D-EM are designed with the assumption that prior knowledge is available. The type of classification that involves the use of prior knowledge is known as supervised classification. The supervised classification methods are relatively simple and efficient. However, they have two major disadvantages. The first is their limited applicability. The supervised classification methods can only be applied to those problems for which appropriate prior knowledge is available. Many important classification problems do not belong to this category. The second is associated with the strong

dependence of the results on the accuracy of the prior knowledge (initial models). Even small inaccuracies in this knowledge (a bad choice of initial models) might be amplified by the classification procedure and lead to incorrect results.

In many classification problems no prior knowledge is available. In such cases the classification must be based exclusively on the information contained in the data. This type of classification is called unsupervised classification. The unsupervised classification techniques do not suffer from the limitations of supervised classification techniques. However, since less information is provided, the problem of unsupervised classification is inherently harder than the one of supervised classification.

1.7 Existing Approaches

Since the 3D visualization of molecular machines in their various conformations is essential to understanding the dynamic processes they undergo, significant efforts have been made to develop techniques capable of reconstructing 3D models of different conformations represented in heterogeneous projection sets. The majority of the proposed methods make use of supervised classification techniques to expand the applicability of single particle approach to heterogeneous projection sets (Penczek, Frank and Spahn [35] provide a recent example). The expansion typically involves estimating the projection directions and the conformations that gave rise to the projections using a set of initial models (or 'references,' and hence such approaches are referred to as multireference 3D projection alignment). Once each projection image is assigned an estimated projection angle and is associated with the specific reference model, a series of 3D reconstructions is performed. Each of these reconstruction involves a subset of the projection images that have been associated with the same reference model. Such series of reconstructions results in a set

of refined (second generation) 3D models. A further refinement can be achieved by using the second generation models as new references and repeating this process. The final 3D models produced by multireference reconstruction methods are the result of several such iterations. (The number of performed iterations depends on a stopping condition used by the specific method and the dataset being processed.)

Methods based on the multireference approach involve either implicitly or explicitly supervised classification of projection images. This inherently difficult task is easier to perform when prior knowledge is used. In case of multireference reconstruction methods, such prior knowledge is provided by the initial models that are approximations (frequently very coarse) of conformations represented in the heterogeneous projection set. Depending on the particular reconstruction problem, different approaches have been taken to obtain models that are required to initiate the multireference refinement procedure.

In some reconstruction efforts that are based on supervised classification [3, 17, 23, 46] the approximate 3D models of some known (frequently pre- and post-transition) states have been used to partition projection set into the subsets representing all conformations present in the heterogeneous projection set. The 3D volumes reconstructed from these subsets are used as an initial models in a multireference reconstruction procedure. Another way to obtain initial models for multireference classification is based on flexibility analysis of a known molecular structure. In some cases an approximate 3D model of the molecular structure is available. This model may represent a single conformation or an average of several conformations (for example, may have been obtained in the reconstruction procedure that ignores the heterogeneity). The structures obtained by elastic deformations of such 3D model can be used as a set of initial references in a multireference reconstruction procedure [4]. Similarly, elasticity analysis of atomic level models can be used to construct a set of 3D references [43, 49]. For example, elasticity analysis of

two known (frequently pre- and post-transition) atomic level models, which represent different conformations, can be used to determine the structure of intermediate states in conformational transition.

The approaches described above share the same problem. In all of them, the initial models of unknown conformations present in the heterogeneous projection set are being construed on the basis of an incomplete knowledge about the 3D structure of the molecule and/or the type of deformations encountered in nature. As a result these approaches may produce sets of initial models that include models of states that are never realized.

The procedures based on multireference approach have been successfully used to produce high-resolution 3D models from heterogeneous projection sets. However, many of them are "custom-tailored" for a particular reconstruction problem and are ad-hoc in nature. The strong dependence on prior knowledge by these methods limits their applicability to that subset of real life cases in which such knowledge is available.

Until recently a comprehensive approach to the analysis of heterogeneous projection set in absence of initial 3D templates has not been worked out [35]. However, a new generation of three-dimensional reconstruction procedures, capable of handling heterogeneous projection sets, is emerging. Unlike the older (at-hoc methods), the new methods are designed to handle a wide variety of the heterogeneous projection sets. Some of them attempt to deal with heterogeneity without relying on any prior knowledge, others significantly reduce the need for such knowledge [16, 39].

The recently proposed cluster tracking method of Fu, Gao and Frank [16] attempts to partition heterogeneous projection sets into homogeneous components. This classification-based approach makes use of the similarity between those 2D projections of the 3D object that were obtained from close projection angles. The

performance of the cluster tracking method has been so far demonstrated only on a relatively small, highly sampled region of angular space. Since cluster tracking cannot be used in regions that are sparsely populated with projections, this classification method can be directly applied only to data sets containing a large number reasonably densely distributed projections. Merging of homogeneous subsets obtained in disjoint regions maybe required if cluster the tracking method is used to classify datasets in which there are regions of angular space with a low density projection coverage.

Another, recently proposed method is based on the maximum likelihood (ML) approach [39]. This method employs multireference refinement process to search for a solution that maximizes the probability of observing the measured data (the projection set given as input). In order to solve this optimization problem, the ML method estimates at the same time each of the 3D conformations, the angles determining each of the projections, the level of noise in the data and the level of misalignment between the projections. The ML method has been demonstrated to produce 3D models of several conformations present in real heterogeneous projection sets. However, the computational costs of doing this were very high; equivalent to several months of computation on a typical desktop computer in current use. This time can be reduced by the use of multiple processors, but significant computational resources are required to complete the reconstruction process within acceptable time.

1.8 Objectives

A classification-based approach (described in Section 1.5) has the potential to become a universal procedure for reconstructing high-resolution 3D models of macromolecular structures from heterogeneous sets of EM projection images. Such a re-

construction procedure would be capable of handling a wide variety of reconstruction problems, including those for which no prior knowledge is available. However, the success of this approach depends on the availability of an appropriate unsupervised image classification procedure. Since, as indicated in Section 1.5, the task such procedure would be required to perform is inherently difficult and cannot be handled by the traditional classification methods, the classification-based approach, in which the projection images are classified directly (without involvement of any reconstruction procedure) is feasible only if a new classification method capable of partitioning the heterogeneous projection sets of EM projection images into their homogeneous components can be constructed.

The main objective of the work presented here is to demonstrate that by utilizing mathematical properties of the projection images and combinatorial optimization techniques, one can construct such a method. Ideally, this method should be capable of producing a high-quality partitioning of the entire heterogeneous projection set that can be directly used in series of reconstructions (one per partition class) to produce 3D models of all conformations present in the projection set. It also should be demonstrated that an implementation of the proposed method, efficient enough to handle classification problems encountered in 3D-EM, is possible. Specifically, such implementation must be able to handle data sets that contain thousands EM projection images that represent more than one conformation and must complete the classification process within acceptable time running on standard computers.

However, if this goal cannot be achieved, a classification method capable of handling only a subset of heterogeneous projection set that produces less than perfect separation of homogeneous subsets can also be useful in 3D-EM reconstruction procedures. As long as the classification produced by such a method captures the essential differences between conformations represented in the projection set, the 3D reconstructions from the individual classes can be used as the initial templates

in a traditional multi-reference reconstruction procedure.

Chapter 2

Projection Image Dissimilarity Measure

The classification-based approach to the heterogeneity problem (described in Section 1.5) has been used in many 3D-EM reconstruction efforts. The classification of electron microscopic projection images, required by this approach, is frequently performed by a supervised classification procedure. In such a procedure, a set of 3D reference models is used to guide the classification process. Since it makes use of additional information (provided by the initial models), a supervised classification is inherently easier than unsupervised one. However, methods based on supervised classification have some significant limitations (see Chapter 1).

As indicated in Section 1.5, the classification-based approach has a potential to become a basis of universal method for reconstructing 3D models of macromolecular structures from heterogeneous projection sets. However, due to the limitations of supervised classification methods, the full potential of the classification-based approach cannot be realized if such methods are used in the process. A method for reconstructing 3D models of macromolecular structures from heterogeneous projection sets that utilizes the classification based approach can be applicable to wide

range of reconstruction problems only if the classification of the projection images is performed by an unsupervised classification procedure. However, unsupervised classification techniques are rarely used to partition heterogeneous projection sets into homogeneous subsets [35]. The required classification task is difficult because, due to nature of the projections and the distortions found in them, it is not possible to determine reliably if two projection images represent the same or two different objects. If a mathematical function that takes two projection images as arguments and returns one value when the images belong to the same class and a different value otherwise could be constructed, the partitioning of heterogeneous projection sets into homogeneous subsets would be easy.

In practice, such a binary decision is not necessary. The desired unsupervised classification can also be achieved when a reliable function that measures dissimilarity of images is available. Such a function takes two images as arguments and returns a value that corresponds to the likelihood that the given images do not belong to the same class (that they are 2D projections of different conformations). The difficulty associated with determining the dissimilarity of 2D projection images defined in this way comes from he fact that frequently a pair of images that belong to the same class are far less similar (in the traditional sense) to each other than another pair of images that belong to different classes. Consequently, the traditional measures cannot be applied to the projection image classification problem.

A new image dissimilarity measure, specifically designed to deal with 2D projections of 3D objects, is proposed in this chapter. This measure utilizes a property of images that are 2D projections of the same 3D object and, consequently, it is well suited to the problem of partitioning heterogeneous projection sets into homogeneous subsets.

2.1 Mathematical Background

Our projection image dissimilarity measure is based on the following mathematical argument. The value at a point in a perfect (i.e., noiseless) projection image is the line integral of the 3D object to be reconstructed along a line that goes through that point and is orthogonal to the plane of the projection image. An immediate consequence of this is that if we take any line in the projection image and we integrate the projection values along that line, the resulting line integral will have the same value as the planar integral of the original 3D object over a plane that contains the given line in the projection plane and is orthogonal to the projection plane.

Consider now two perfect projection images $\bar{x}$ and $\bar{y}$ (in different directions) of the same 3D object (Fig. 2.1). The planes (p and q) of these projections intersect in a line (cl). This line occurs in both of the projection planes, and it is therefore referred to as the common line. Take any point (a) on the common line and consider the two lines ($l_{a,p}$ in p and $l_{a,q}$ in q) that are perpendicular to the common line and include this point. These two lines lie in the same plane (P_a, which is perpendicular to the common line cl and goes through the selected point a). It follows therefore that the line integrals that are obtained by integrating in the projection images ($\bar{x}$ and $\bar{y}$) along the lines perpendicular to the common line (cl) going through the selected point (a) have the same value (namely, that of the planar integral of the original 3D object over the plane P_a).

An important (and well known [8]) consequence of this is the following property of any two perfect projection images of the same 3D object. Let us assume that we can identify in the projection planes (p and q) of the two projections the points (call them o_p and o_q, respectively) onto which the origin of the assumed coordinate system of the 3D Euclidean space project. We can now translate the property described in the previous paragraph into the following: there is a line (call it l_p) in p going through o_p and a line (call it l_q) in q going through o_q such that, for any

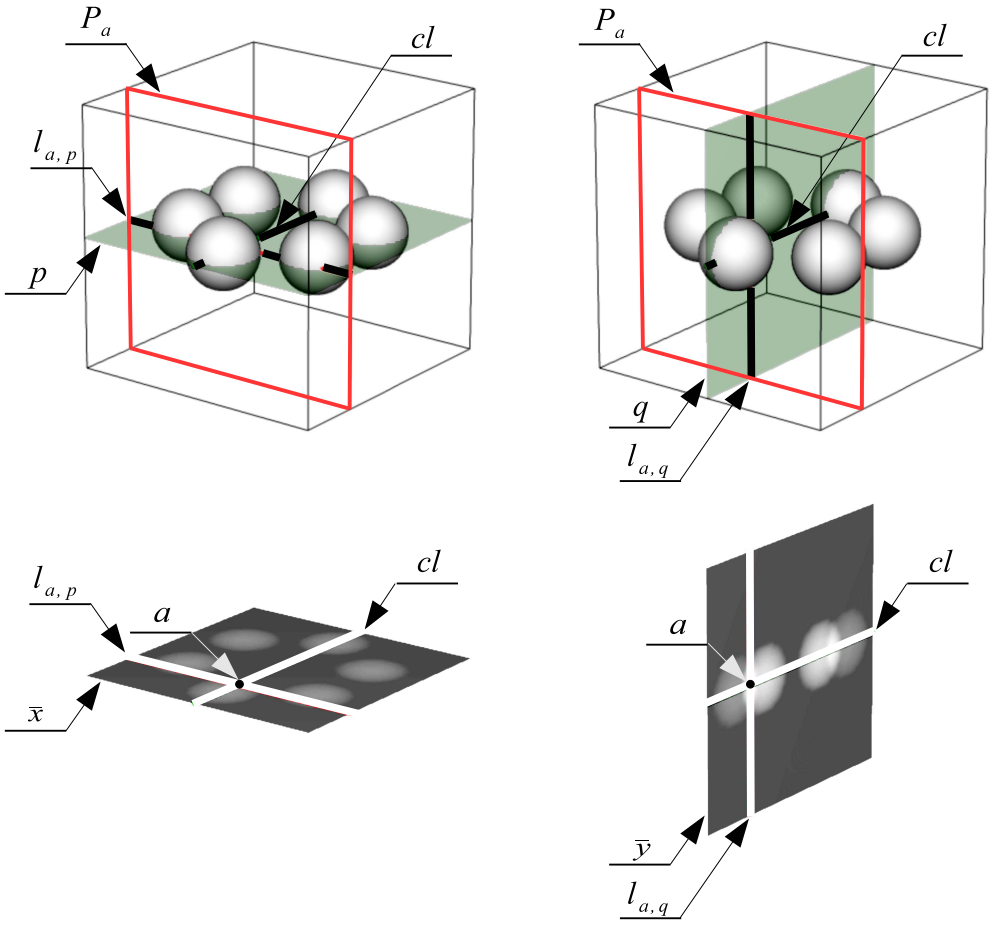


Figure 2.1: Two projections $\bar{x}$ and $\bar{y}$ of object S6.
The line integrals in projection planes (p and q) along the indicated lines ($l_{a,p}$ and $l_{a,q}$) perpendicular to the common line (cl) are both equal to the planar integral over the indicated plane of integration (P_a). (P_a - plane of integration; $l_{a,p}$, $l_{a,q}$ - lines perpendicular to the common line; cl - common line; $\bar{x}$, $\bar{y}$ - projections onto projection planes; p, q - projection planes; a - point on the common line).

real number d, the value of the line integral of the projection $\bar{x}$ onto p along the line perpendicular to l_p at a distance d from o_p is the same as the value of the line integral of the projection $\bar{y}$ onto q along the line perpendicular to l_q at the distance d from o_q.

2.2 Definition

In practice, we have to deal with the discrete and noisy nature of our data (which implies that the planar integrals of the 3D object can be determined from the projection images only approximately) and computational reality does not allow us to look at all lines in the projection planes. Let L be the number of evenly distributed lines at which we will look in each projection plane p, we index them by l, $1 \leq l \leq L$. Each of these lines go through o_p. On each of them we pick N points (these points are picked at matching distances from o_p in all projection planes p and for all lines l). For each projection image $\bar{x}$ and for each such line l we define an N-dimensional vector X_l whose n-th component (for $1 \leq n \leq N$) is the estimated line integral in the projection image along the line perpendicular to l going through the n-th point. If errors due to noise and discretization are ignored, then (according to the property described in Section 2.1) two projection images $\bar{x}$ and $\bar{y}$ of the same 3D object must have identical vectors X_l and Y_m for some pair of indexes l and m. This can also happen if $\bar{x}$ and $\bar{y}$ are projection images of different 3D objects, but only under very special circumstances. In reality, due to discretization error and noise, there is practically no pair of indexes l and m for which vectors X_l and Y_m are identical. However there is an increased probability of finding two 'similar' vectors X_l and Y_m, if the projections $\bar{x}$ and $\bar{y}$ came from the same object. Let us assume that 'dissimilarity' of vectors can be measured by a function s that returns 0 given a pair of identical vectors and a positive value indicative of the differences between the

vectors otherwise. We define the dissimilarity of any two projection images $\bar{x}$ and $\bar{y}$ as

$$\bar{s}(\bar{x},\bar{y}) = \min_{1\le l,m\le L} s\,(X_l, Y_m)\,. \tag{2.1}$$

Note: The definition of function s will be provided in Section 2.3.2.

2.3 Application to EM Projection Images

With the exception of some special cases, the value of dissimilarity measure defined in Section 2.2 reliably determines whether two noiseless 2D projection images represent the same 3D object. However, before this measure can be applied to the images obtained by an electron microscope, some prepossessing of the images is necessary. When rectangular, pixels images are used to produce 1D projections, the

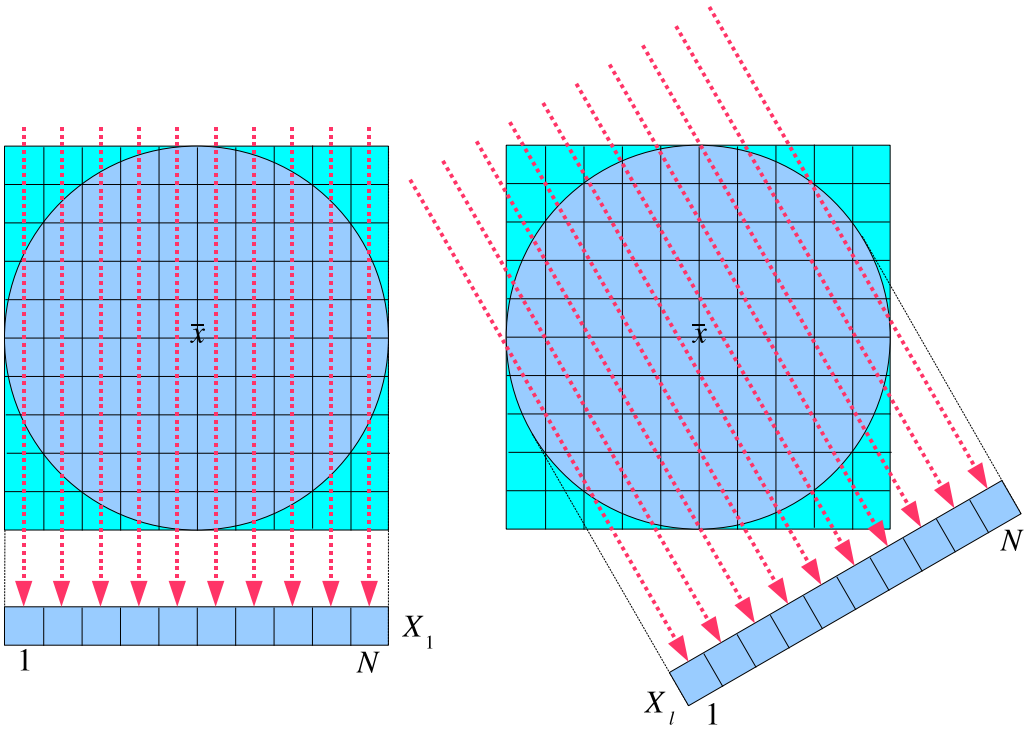


Figure 2.2: The 10 x 10 image and its two 1D projections.
Circular mask and 1D projections of 2D image (the values of pixels with centers outside of the masking circle are set to zero).

number of pixels over which the integration occurs depends on the direction from which the 1D projection is taken. In case of noiseless images this can be ignored because the values of pixels in 2D image that are not covered by the support of 3D object are equal to zero. When 1D projections are produced from a noisy image the varying number of pixels over which integration occurs results in undesired artifacts. To avoid this problem, before the 1D projections are produced the 2D projection image is masked with a circular mask large enough to contain the footprint of the molecule. Setting the values outside the mask to zero ensures that, for each ray, the integration occurs over approximately the same length of nonzero values regardless of the direction (see Figure 2.2).

In the second prepossessing step, the values of the pixels within the masking circle are normalized by subtracting from each of them a constant value so that their sum after subtraction is zero. Such a normalization is justified by the nature of the contrast transfer function of an electron microscope (the information about the sum of densities is essentially lost [12]) and the facts that it does no harm in the noiseless case but eliminates the average of the noise in the noisy case.

After the prepossessing is applied to a pair of images $\bar{x}$ and $\bar{y}$, the value of

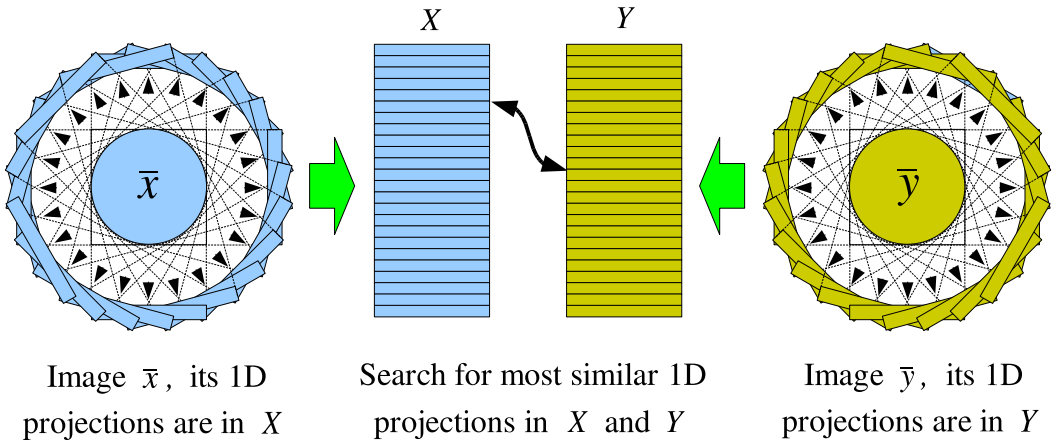


Figure 2.3: Process of calculating the value of dissimilarity measure for two images $\bar{x}$ and $\bar{y}$.

dissimilarity measure for them can be found by the process illustrated by Figure 2.3. First 1D projections of 2D image $\bar{x}$ are calculated for L (parameter of the process) equally spaced directions. These L 1D projections are stacked on top of each other to form an image X (a sinogram of $\bar{x}$). Each horizontal line of the sinogram X is a vector (X_l) that represents a single 1D projection of the 2D image $\bar{x}$. In an identical procedure, a sinogram Y of the image $\bar{y}$ is produced.

Once both sinograms X and Y are available, the calculation of the dissimilarity measure of (2.1) is preformed by searching for a pair of lines X_l and Y_m (in sinograms X and Y correspondingly) for which the value of function $s(X_l, Y_m)$ is minimal. A very similar procedure is routinely used in the common-line approach to the "angular reconstitution" (a process that finds the angular relationship between the projection images) [47].

Figures 2.4, 2.5 and 2.6 illustrate the process of calculating the value of dissimilarity measure for three pairs of images.

Figure 2.4 illustrates this process for two dissimilar (in the traditional sense)

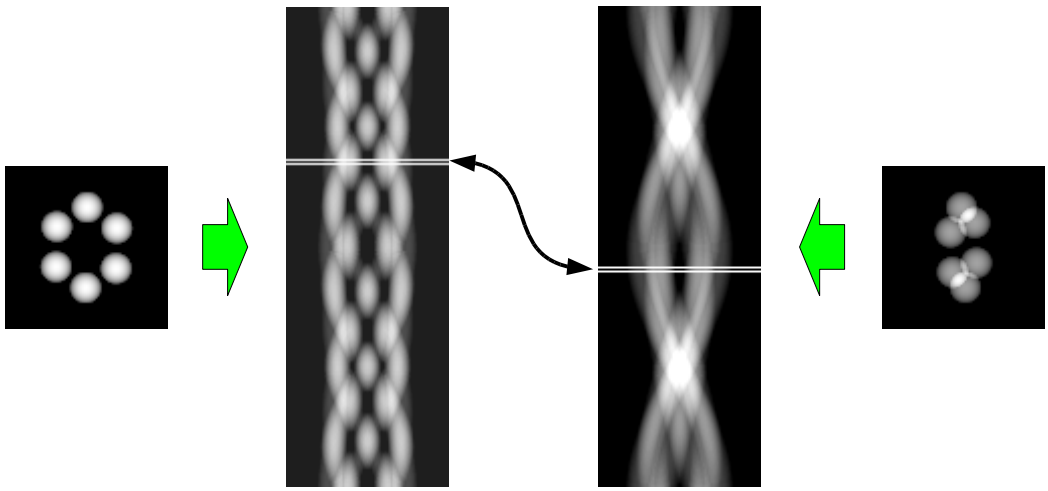


Figure 2.4: Matching line in the sinograms of two noiseless projections images that originate from the same 3D object.

projection images of the same 3D object. Despite the differences between the two images, a pair of similar lines can be found in their sinograms.

When the dissimilarity measure is calculated for two projection images obtained from different 3D object for some pairs of 2D projections (like these in Figure 2.5) the differences between sinogram lines are more visible than for others (like these in Figure 2.6). However no pair of matching sinogram lines can be found in either case.

The dissimilarity measure becomes much less reliable when high level of noise is present in the 2D projection images. The lines in the sinograms become blurred and hence it is much harder to determine if a pair of lines from two different sinograms differ because of the impact of noise or because of the differences between the noiseless versions of 1D projections. The Figure 2.7 illustrates this problem.

2.3.1 Alignment

The proposed dissimilarity measure relies on the assumption that it is possible to identify in the projection planes (p and q) of the two projections ($\bar{x}$ and $\bar{y}$) the points (call them o_p and o_q, respectively) onto which the origin of the assumed coordinate system of the 3D Euclidean space project. In practice, this is equivalent to aligning projection images $\bar{x}$ and $\bar{y}$ so that the values of the central pixels in both images correspond to line integrals calculated along two lines that cross each other. Such *alignment* can be achieved by the translation of both images in their corresponding projection planes in such a way that, after the translation, the center of the mass of the 3D object is projected onto the central pixels of both images.

The problem of aligning electron microscopic projection images is not new. The methods for reconstructing 3D models from the projection images require that all the images used in reconstruction are placed in a common coordinate system. The electron microscope does not provide information about the spatial coordinates of

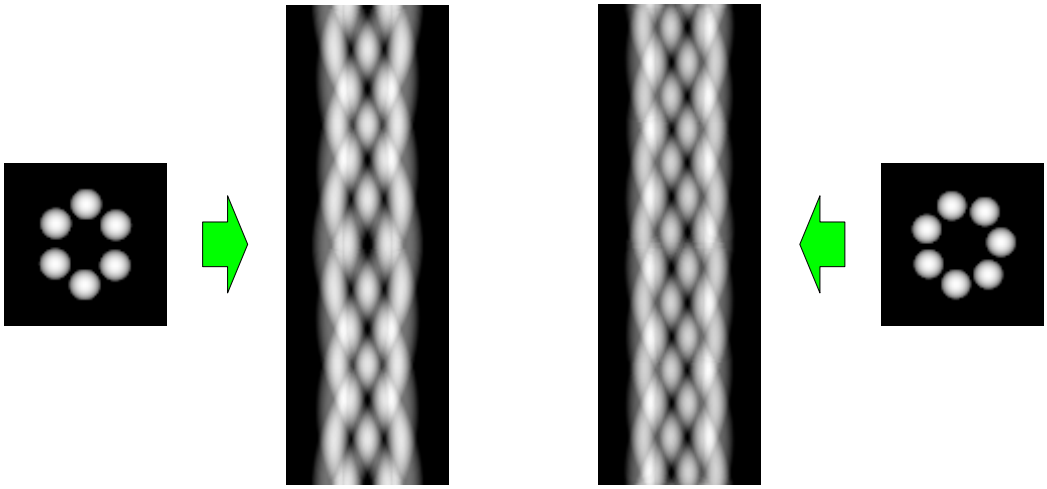


Figure 2.5: The sinograms of two noiseless projections images that originate from different 3D object.
(No matching line can be found.)

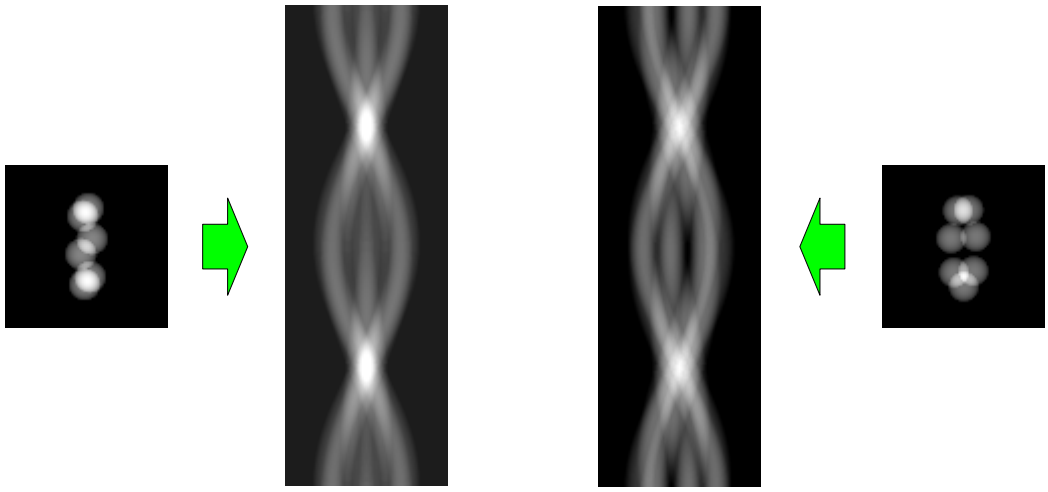


Figure 2.6: The sinograms of two noiseless projections images that originate from different 3D object.
(Some of the lines are quite similar, however no matching line can be found.)

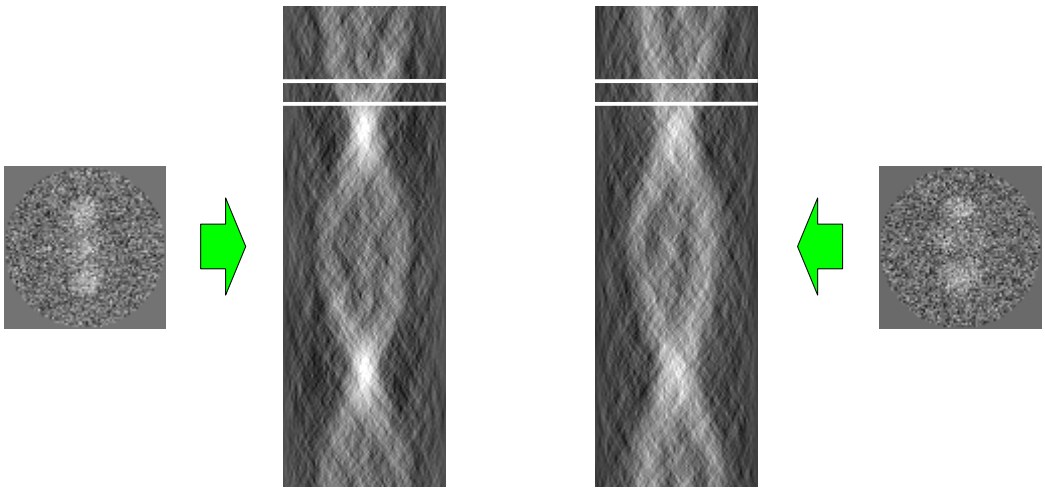


Figure 2.7: The sinograms of two noisy projections images that originate from different 3D objects.
(The noise significantly blurs the differences between the lines in sinograms, The lines in marked regions are hard to distinguish.)

the images it produces. Therefore image alignment is an integral part of any 3D reconstruction procedure that involves EM images (see [32, 27]for examples). Due to the high level of noise present in the EM projection images and other distortions these images suffer from, this task is not trivial and perfect alignment of EM projection images cannot be achieved.

The misalignment of the projection images reduces the reliability of the proposed dissimilarity measure. Since the spatial density distribution of biological objects does not change rapidly, in case when two projection images $\bar{x}$ and $\bar{y}$ of the same object are slightly misaligned, a similar pair of 1D projections X_l and Y_m can be found. Such pair would produce a low value of the function $s(X_l, Y_m)$, indicating that projection the images $\bar{x}$ and $\bar{y}$ are likely to originate from the same object. However, when the misalignment of images $\bar{x}$ and $\bar{y}$ is large, the corresponding line integrals in the vectors X_l and Y_m will be shifted with respect to each other and, unless a shift invariant function s is used, the estimated likelihood that images $\bar{x}$ and

$\bar{y}$ originated from the same 3D object will be low.

We conducted a set of experiments to test the impact of image misalignment on the usefulness of the proposed image dissimilarity measure to the task of classification of EM projection images. The results of these tests are presented in the later chapters.

2.3.2 Vector Dissimilarity Measure

The definition of image dissimilarity measure proposed in Section 2.2 is based on the assumption that a 'dissimilarity' of vectors can be measured by a function s that returns 0 given a pair of identical vectors and a positive value indicative of the differences between the vectors otherwise. So far, the function s has not been formally defined. Since there are many ways of measuring the dissimilarity of two vectors, we conducted some initial experiments to evaluate different functions. These experiments have shown that some functions are much better from the perspective of the proposed image dissimilarity measure than others. The problem of selecting a "good" vector dissimilarity measure is not trivial. A systematic (analytical or experimental) approach to this problem would require a significant amount of work and a better understanding of distortions present in the projection images. Ideally, the selection of the function s should take into consideration the knowledge about the statistical characteristics of both the noise and the signal present in the EM images.

Since the main focus of our work was to demonstrate feasibility of classification-based approach to the heterogeneity problem, only limited experimental effort has been made to find a good (from the projection image classification perspective) measure of vector dissimilarity. We experimentally tested performance of several functional as measures of vector dissimilarity. (Among others cross correlation, 1-norm of the difference, 2-norm of the difference and 3-norm of the difference were tested.) Based on these experiments the squared 2-norm of the difference

$$s(x,y) = \|x-y\|^2 \tag{2.2}$$

was selected (x and y are N-dimensional vectors). There is no evidence that this function is optimal for the classification of EM projection images. However, it has been experimentally demonstrated that it performs sufficiently well to demonstrate the feasibility of the proposed classification framework (and better than other functions that we tried).

A more systematic approach (either analytical or experimental) to finding optimal vector dissimilarity function, incorporating knowledge about the statistical properties of the signal and the noise in the EM projection images, is likely to improve the reliability of the proposed dissimilarity measure.

2.3.3 Implementation

Since the EM images are relatively small (60 by 60 to 200 by 200 pixels), the computational cost and storage necessary to compute the dissimilarity measure for a single pair of images is quite manageable (it depends on the cost of calculating the value of the function s, the values of the parameters L and N, and the size of the projection images). However, the the CPU time necessary to compute image dissimilarities for many pairs of images is not negligible. However, some simple implementation techniques can be employed to keep down the cost of calculating the values of dissimilarity measure.

When the dissimilarity of projection images is calculated for many pairs of images in which one of the images remains the same, a significant reduction of the computational cost can be achieved by pre-calculating 1D projections (the sinogram) of that image and using them in all distance calculations. Even if an image appears in many pairs for which the dissimilarity measure must be calculated, its

1D projections need be calculated only once. It is therefore beneficial to divide the process of calculating images distances into two phases. In the first phase, sinograms are produced for all the images for which the distances must be calculated. These sinograms are stored and used in the second phase of the process. In the second phase, the dissimilarity measure of images $\bar{x}$ and $\bar{y}$ is calculated by finding the minimal value of function s for pairs of 1D projections X_l and Y_m (horizontal lines in the sinograms X and Y) produced in the first phase.

By avoiding recalculation of sinograms, the two-phase approach to the calculation of image dissimilarity measure significantly reduces the computational cost of calculating distances for many pairs of images. Nevertheless, when the distances must be calculated between many pairs of images in a large projection set, the amount of storage necessary to save all sinograms can be very substantial. A significant (50%) reduction of storage requirements and computational cost can be achieved by utilizing the fact that two vectors representing 1D projections obtained from angles that differ by 180 degrees are reversed (in terms of order of elements) copies of each other. This property allows to calculate the dissimilarity of two images by using only halves of their sinograms (covering angles within the [0, 180) range). The remaining halves can be easily recovered by reversing the order of elements in the calculated 1D projections.

Another important consequence of the sinogram symmetry described above is the reduction of the number of pairs X_l and Y_m for which the value of function s must be calculated in order to determine the value of $\bar{s}(\bar{x},\bar{y})$. As long as

$$s(X_l, Y_m) = s(X'_l, Y'_m),$$

where vector X'_l contains elements of vector X_l in reversed order and vector Y'_m contains elements of vector Y_m in reversed order, the dissimilarity measure $\bar{s}(\bar{x},\bar{y})$ between two images $\bar{x}$ and $\bar{y}$ can be calculated by finding the minimum of the func-

tion s for $\frac{1}{2}L^2$ pairs of vectors X_l and Y_m (L^2 pairs must be considered in a brute force approach).

Chapter 3

Projection Image Classification as Optimization Problem

The specific nature of the projection images imposes some restrictions on a method that can be used to solve the problem of partitioning heterogeneous projection sets into their homogeneous components. It appears that our measure of dissimilarity between two projection images (proposed in Section 2.2) is the only mathematical concept, defined in the domain of the projection images, that can be used to solve this classification problem. Since in this domain there is no meaningful from our classification perspective definitions of other mathematical notions (e.g., average of several projection images), some traditional approaches (e.g., clustering algorithms like k-means) are not applicable. Without such definitions, a method for solving the problem of partitioning heterogeneous projection sets into their homogeneous components must rely exclusively on the distances between the pairs of images. Under these restrictions, a simple approach to the problem of finding the desired classification is to use a thresholding method. However, such a method can be successful only if the homogeneous subsets are well separated (the distances between the members of the same class are much smaller than the distances between members of two

different classes). When the separation of classes is not so well defined, a more sophisticated approach must be taken. Frequently, in such cases the classification of objects is achieved by solving an optimization problem (finding an arrangement of the objects that optimizes some objective function). As we demonstrate in this chapter, the homogeneous classes of EM projection images are not well separated and cannot be isolated using a thresholding based method. Therefore we propose a reformulation of the projection image classification as an optimization problem.

3.1 Similarity of EM Projection Images

The high level of noise present in the projections sets obtained by EM has this consequence that even when an appropriate dissimilarity measure (as defined by (2.1)) is used, the task of identifying homogeneous components in such sets is difficult. The value of the dissimilarity measure between two projections produced from the same conformation is only statistically smaller than the value for two projections of different conformations. In fact, when dissimilarity measures are calculated for all the pairs of projections in a realistic heterogeneous projection set, the range of values for pairs originating from the same conformation is practically the same as the one for pairs originating form different conformations. (For example, using the data set generated for the experiment of Table 7.7, the first range is from 0.84×10^6 to 2.21×10^6 with mean 1.57×10^6 and standard deviation 0.12×10^6, while the second range is from 0.79×10^6 to 2.44×10^6 with mean 1.63×10^6 and standard deviation 0.13×10^6. The histograms of distances in these sets are shown in Fig. 3.1.) The problem of partitioning heterogeneous projection set into their homogeneous components clearly cannot be solved by a simple thresholding method. Only an optimization-based approach that simultaneously considers many (possibly all) projections has the potential to produce the correct partitioning.

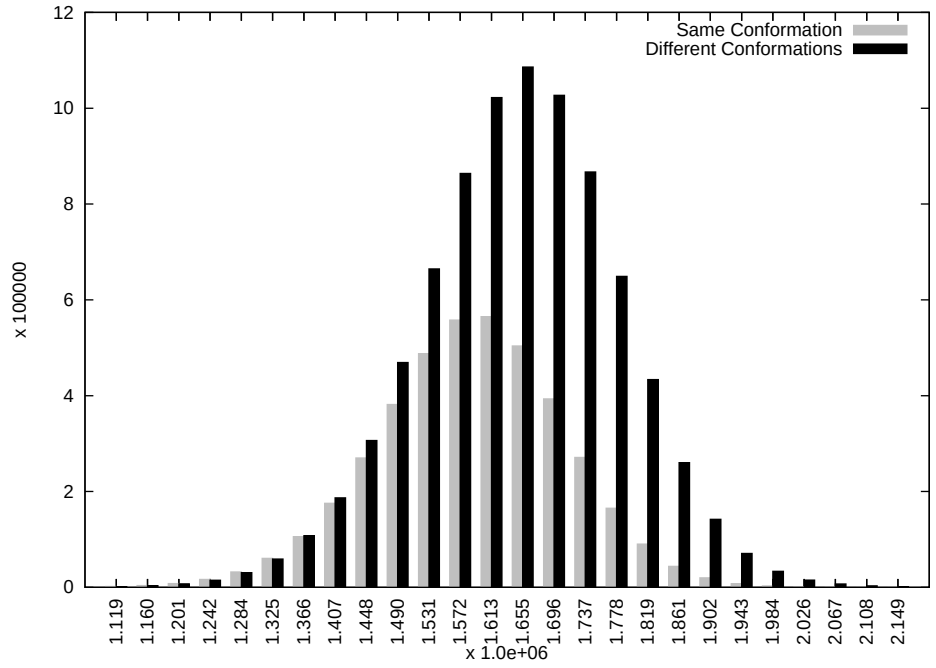


Figure 3.1: Histograms of distances between pairs of projection images in a heterogeneous set for the pairs originating from the same and from different conformations.

3.2 Formal Statement of the Optimization Problem

In order to obtain a reliable classification of noisy projection images (such as those produced by EM), we propose the reformulation of the original classification problem as the following optimization problem. Let V denote the heterogeneous projection set. For any positive integer K, a K-partition A of V is a set $\{A_1, \cdots, A_K\}$ of K nonempty subsets of V such that the union of these subsets is the whole of V and no two subsets have any element in common. Our reformulated problem is:

GIVEN a set V of 2D projections and a positive integer K,

FIND a K-partition $A = \{A_1, \cdots, A_K\}$ of V,

SUCH THAT

$$\sum_{k=1}^{K} \sum_{\bar{x},\bar{y} \in A_k} \bar{s}(\bar{x},\bar{y}) \tag{3.1}$$

is as small as possible.

The translation of the biological classification problem into our proposed opti-

mization problem is appropriate for the situations in which the projection images being classified contain a significant amount of noise, because the classification of the individual images is not made independently. In fact, the classification of each projection image is determined by the optimal arrangement of all the images and therefore it is much less likely to be influenced by noise.

3.3 Graph-Theoretical Interpretation

The classification problem posed in Section 3.2 can also be restated in the context of graph theory [19]. The projections in V are represented by nodes of a complete weighted graph G; the weight of the edge between nodes $\bar{x}$ and $\bar{y}$ is the distance $\bar{s}(\bar{x},\bar{y})$, defined by (2.1). In such a graph, the edges between the nodes representing projections of the same object are more likely to have lower weights. The problem of separating the homogeneous subsets of a heterogeneous projection sets becomes a graph cutting problem, in which the objective is to find a separation of the graph G into K complete subgraphs $G_1,...,G_K$ such that the sum of all edge weights in the subgraphs $G_1,...,G_K$ is minimal. This problem is known as Max k-Cut [25], and in case $K = 2$ it is equivalent to the maximum capacity cut problem [41]. Fig. 3.2 shows a small (10 images) instance of heterogeneous projection set classification problem interpreted as maximum capacity cut problem of the corresponding complete graph.

3.4 Computational Complexity

The reformulation of the optimization problem stated in Section 3.2 as a known graph cutting problem allows to take advantage of graph-theoretical results to estimate its computational complexity. Both Max k-Cut and maximum capacity cut problems have been shown to be NP-complete [25, 41]. It also has been demon-

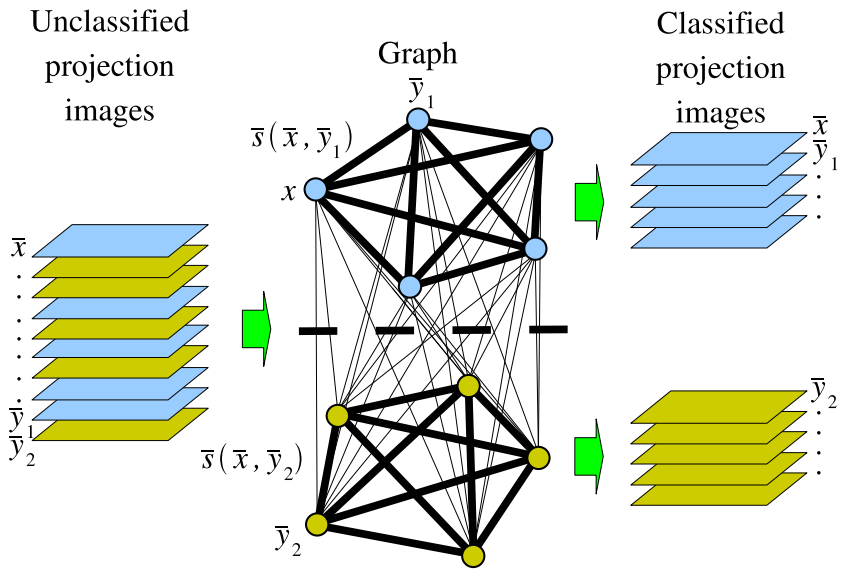


Figure 3.2: Classification by graph cutting.
Images represented by nodes of each graph component belong to the same class. The thick edges connect the graph nodes that represent pairs of similar images (those for which the value of (2.1) is low). The thin edges connect the graph nodes that represent pairs of dissimilar images (those for which the value of (2.1) is higher).

strated that finding even approximately optimal solution to the Max k-Cut is NP-complete [25]. This means that, according to theory, finding an optimal (or even approximately optimal) solution to a large instance of the Max k-Cut or the maximum capacity cut problem is extremely computationally expensive, and so for larger graphs one cannot in general expect that an algorithm will find an approximate minimizer of the expression of (3.1) in an acceptable computer time.

The theoretical complexity of Max k-Cut and maximum capacity-cut problems describes the relationship between the size of the problem and the cost of solving it. It is calculated with the assumption that all instances of the problem can be encountered. Since the instances that represent a real life process are not randomly generated, they may form a specific subclass of all possible instances that exhibit some regularities. In some cases, the cost of solving instances of problems that belong to this subclass may be reduced by utilizing their special properties. Some-

times, for instances that belong to such subclasses, it is possible to find a good (from the practical perspective) approximation of the optimal solution at relatively low cost.

The most important factor affecting the computational effort required to solve an NP-complete problem is its size. In fact, solutions to small instances of such problems can be found relatively quickly. In order to estimate the relationship between the size of a realistic (from EM image classification perspective) graph and the computational cost of finding its maximum capacity cut (partitioning into two classes), we conducted several experiments using a standard algorithm (called DSDP [1]) for finding the maximum capacity cut. Due to the long run-time, solution for graphs with more than 2,000 nodes were not attempted. The run-times of the experiments with smaller graphs are summarized in Table 3.1. The 2,000 node instance of the problem took over 50 hours to solve. The estimated (based on the values in Table 3.1) run time for solving the 5,000 node instance of the graph cutting problem using DSDP is approximately one month. It is important to notice the a data set that contains 5,000 image is relatively small by 3D-EM standards. In many reconstruction efforts tens or even hundreds of the projection images are used (for example see [45]).

No. of nodes	500	1,000	2,000
Run time (sec.)	615	17,326	187,825

Table 3.1: Timings of DSDP for solving the maximum capacity cut problem for graphs of different sizes.

The results of our experiments confirmed theoretical predictions that the computational cost of finding the maximum capacity cut for large graphs is very high. However it is important to notice that the algorithm used in these experiments (DSDP) has not been optimized for the instances of the problem that originate from 3D-EM. Therefore our results provide just an estimate of the computational cost of

finding the maximum capacity cut using an off-the-shelf algorithm and do not prove that these instances cannot be solved faster. In fact, as we demonstrate in Chapter 5, an efficient algorithm capable of producing good (from our classification problem perspective) estimates of Max k-Cuts for graphs originating from 3D-EM can be constructed. This algorithm produces an acceptable approximate solutions to large (5,000 nodes) instances of Max k-Cut problem in two minutes, provided that they arose from the problem of classification of heterogeneous 2D projections.

Chapter 4

Construction of the Distance Graph

As described in Chapter 3, the problem of partitioning heterogeneous projection sets into their homogeneous components can be reformulated as a graph-cutting problem. In this reformulation, an instance of the image classification problem is represented as a complete weighted graph in which each node represents a single projection image. The weight of each edge describes the dissimilarity of the images represented by the nodes it connects. The desired image classification is achieved by finding a Max k-cut of such graph. Frequently the datasets encountered in 3D-EM reconstruction efforts contain tens of thousands or even hundreds on thousands of projection images (for example see [45]). In order to classify such datasets a large graphs must be constructed. Since the topology of these graphs is fixed (complete graph with each node corresponding to a single 2D projection image) the process of constructing them is simple (only the weights of the edges must be calculated). However, due to the large number of nodes in these graphs (equal to the number of images in the heterogeneous set) and their topology (complete graphs), the number of edge weights that need to be calculated is very large (for a graph with 5,000 nodes, 12,497,500 edge weights must be calculated). A significant amount of computer time must be dedicated to calculating edge weighs in a realistically sized

graph. For example (without optimizations) it takes approximately 24 hours on a single processor (Intel Xeon 1.7 GHz) to construct a graph for a data set that contains 5,000 images. Since the calculations of edge weights between different nodes of the graph are mutually independent, the task of constructing the graph can be easily parallelized. However, the cost of constructing such graphs increases proportionally to the square of the number of projection images. So for larger datasets that contain tens of thousands projection images, which are not unusual in 3D-EM of macromolecules, significant resources are required to construct the corresponding graphs. The computational cost of constructing such graphs can be reduced by the use of an efficient procedure for calculating edge weights.

The weight of an edge, which connects a pair of nodes that represents images $\bar{x}$ and $\bar{y}$, is equal to the value of the dissimilarity measure $\bar{s}(\bar{x},\bar{y})$ (defined by (2.1)) between images $\bar{x}$ and $\bar{y}$. If dissimilarity s between two vectors is defined as in 2.2, then the process of calculating a single edge weight can be interpreted as finding a squared distance between two sets of points in multi-dimensional Euclidean space. The problem of finding distances between sets of points in Euclidean space is not new. It has been encountered in other applications and efficient algorithms to solve this problem have been proposed. However, some of these algorithms do not perform well in the high-dimensional spaces[51] and consequently cannot be applied to the problem of building graphs for 3D-EM image classification.

In order to reduce computational costs associated with the construction of such graphs, we adapted some known techniques to construct two algorithms for finding the distances between sets of points in high-dimensional Euclidean spaces. The first of these algorithms is based on the AESA [48] algorithm and finds the distance between two sets of points by solving a series of Nearest Neighbor Search (NNS) problems. The second, our contribution [24], utilizes an early-termination technique to accelerate the computation of the set distances. We implemented these

algorithms and experimentally evaluated their applicability to our graph building problem.

4.1 Formal Problem Statement

Let N be the dimension of the Euclidean space $\mathbb{R}^N$. The distance $d(x,y)$ between two points $x, y \in \mathbb{R}^N$ is

$$d(x,y) = \sqrt{\sum_{n=1}^{N} (x^{(n)} - y^{(n)})^2}. \tag{4.1}$$

We define the distance $D(X,Y)$ between two nonempty finite sets of points X and Y as

$$D(X,Y) = \min_{x \in X, y \in Y} d(x,y). \tag{4.2}$$

When the dissimilarity $s(x,y)$ between two N-dimensional vectors x and y is defined by (2.2), then finding a value of dissimilarity $\bar{s}(\bar{x},\bar{y})$ (defined by (2.1)) between two images $\bar{x}$ and $\bar{y}$ can be interpreted as finding a distance $D(X,Y)$ of (4.2) between two sets of points X and Y in N-dimensional Euclidean space and squaring the result. In this interpretation, the L horizontal lines in each of the sinograms X and Y are treated as points in Euclidean space.

Note that $d(x,y) = \sqrt{s(x,y)}$ and, hence, $S(X,Y) = D(X,Y)^2$, where $S(X,Y) \equiv \bar{s}(\bar{x},\bar{y})$. Using this interpretation, our graph-building problem can be stated as follows. Given a set Ω, whose elements are nonempty subsets of $\mathbb{R}^N$ (each containing L points), find $D(X,Y)$ (equivalently, $S(X,Y)$), for all X and Y in Ω.

The practical difficulty associated with this problem is that Ω typically consists of thousands of elements, each one of which contains hundreds of points in a high-dimensional space. Due to large number of point-set pairs (equal to the number of

edges in the graph) for which the distance must be computed, the computational cost of solving this problem is strongly dependent on the efficiency with which a distance between a single pair of point-sets can be found.

4.2 Brute-Force Method

According to the definition (4.2), one method of finding the distance $D(X,Y)$ between two sets of points X and Y in $\mathbb{R}^N$ is to compute distance $d(x,y)$ for all pairs of points $(x,y) \in X \times Y$ and find their minimum. This method is simple but expensive. Even when the simplistic approach that involves exhaustive examination of all pairs (x,y) is taken, the cost of finding $D(X,Y)$ can be reduced. Since $D(X,Y) = \sqrt{S(X,Y)}$, the distance $D(X,Y)$ between two sets of points X and Y can be found by computing for all pairs of points $(x,y) \in X \times Y$ the values of $s(x,y)$, finding their minimum $S(X,Y)$ and then calculating $D(X,Y)$ as $\sqrt{S(X,Y)}$ (in our image classification problem the edge weight is equal to the value of $S(X,Y)$ so there is no need for this last operation). Since the calculation of $s(x,y)$ is simpler than the calculation of $d(x,y)$ (calculation of $s(x,y)$ does not require the expensive operation of taking a square root of the sum), this method is computationally less expensive than a method derived directly from the definition of $D(X,Y)$. We refer to the method that finds value $D(X,Y)$ as the square root of minimum of values $s(x,y)$ calculated for all pairs of points $(x,y) \in X \times Y$ as brute-force method. The performance of the brute-force method (see Algorithm 1) provides the baseline for evaluation of more advanced methods that are designed to calculate the distance of two point-sets in high-dimensional Euclidean space with higher efficiency.

Algorithm 1 Brute-force method

1: {**Input:** }
2: { X, Y: nonempty sets of points in $\mathbb{R}^N$ }
3: {**Output:** }
4: { $D(X,Y)$: the distance between sets of points X and Y }
5: $s_{min} \leftarrow +\infty$
6: **for all** $x \in X$, $y \in Y$ **do**
7: $\quad s_{min} \leftarrow \min(s_{min}, s(x,y))$
8: **end for**
9: **return** $\sqrt{s_{min}}$

4.3 Triangle Inequality Based Method

4.3.1 Nearest Neighbor Search

An alternative method for computing the distance $D(X,Y)$ between two sets of points $X, Y \in \mathbb{R}^N$ (that is defined by (4.2)) is suggested by the formula

$$D(X,Y) = \min_{1 \le l \le L} D'(X_l, Y), \text{ where } D'(X_l,Y) = \min_{1 \le m \le L} d(X_l, Y_m). \tag{4.3}$$

The computation of $D'(x,Y)$ is equivalent to finding the distance between a point $x \in \mathbb{R}^N$ and its *nearest neighbor* $y_x \in Y$ (because, by the definition of nearest neighbor of point x in point-set Y, $d(x,y_x) = D'(x,Y)$). The problem of finding the nearest neighbor $y_x \in Y$ of a given point x is known as the Nearest Neighbor Search (NNS) problem. Since this problem is frequently encountered in database searching, pattern recognition, and data compression [6, 33], a number of algorithms have been proposed for solving it efficiently. However, when the dimensionality of the space is high, the computation time of NNS can be reduced only by the use of metric properties of the distance [51]. Therefore some advanced methods for solving NNS problem cannot be used to accelerate the process of building graphs for EM image classification.

One technique used to reduce the cost of solving NNS problem, which is appli-

cable to high-dimensional spaces, is based on the triangle inequality

$$d(a,b) \geq |d(a,c) - d(b,c)|, \text{ for } a,b,c \in \mathbb{R}^N. \tag{4.4}$$

In many cases, the use of triangle inequality allows to significantly reduce the computational cost of solving NNS problem. This is so because, according to the following mathematical argument (based on triangle inequality), during the search for the nearest neighbor $y_x \in Y$ of a given point x, the calculation of distances $d(x,y)$ between many pairs (x,y) with $y \in Y$ can be avoided.

Let us assume that the distances $d(Y_l, Y_m)$ are precalculated for all pairs of vectors $Y_l, Y_m \in Y$ and the distances $d(x,Y_1)$, $d(x,Y_2)$ for $Y_1, Y_2 \in Y$ have been already computed. Now, if for some point $Y_3 \in Y$ the inequality

$$d(x,Y_1) \leq |d(x,Y_2) - d(Y_2,Y_3)| \tag{4.5}$$

holds, then the computation of distance $d(x,Y_3)$ is unnecessary for the purpose of calculating $D'(x,Y)$, since from (4.5) follows that $d(x,Y_3) \geq d(x,Y_1)$.

4.3.2 Algorithm

Our attempt to apply NNS techniques to the graph building problem is based on the AESA [48] algorithm. AESA was chosen because it is one the fastest ways of finding the nearest neighbor when the number of points that must be considered is small (which is the case in our graph building problem) and because it is efficient when the number of spatial dimensions is high (also the case in our problem). AESA is based on the technique described in Section 4.3.1.

Our generalization of AESA (Algorithm 2) to compute the distance between two sets (X and Y) of points in $\mathbb{R}^N$ assumes that the distance between each pair of points in Y has been precomputed. (In our application, for all $Y \in \Omega$, distances between all pairs of points (y_1, y_2), were $y_1, y_2 \in Y$, are calculated during the initialization of the

Algorithm 2 AESA-based Algorithm.

1: {**Input:** }
2: { X, Y: nonempty sets of points in $\mathbb{R}^N$ }
3: {**Internal objects and functions:** }
4: { Y': the set that contains points $y \in Y$, for which $d(x,y)$ has not been }
5: { calculated and may be smaller than d_{min} (current estimate of $D(X,Y)$) }
6: { $g(y)$: the vector that for current $x \in X$ associates with each vertex $y \in Y$ }
7: { a lower bound $g(y)$ of $d(x,y)$ }
8: {**Output:** }
9: { $D(X,Y)$: the distance between sets of points X and Y }
10: $d_{min} \leftarrow +\infty$
11: **for all** $x \in X$ **do**
12: $Y' \leftarrow Y$
13: **for all** $y \in Y'$ **do**
14: $g(y) \leftarrow 0$
15: **end for**
16: $y \leftarrow$ an arbitrary element of Y'
17: **while** $Y' \neq \emptyset$ **do**
18: $Y' \leftarrow Y' \setminus \{y\}$**;** $d \leftarrow d(x,y)$
19: $d_{min} \leftarrow \min(d_{min}, d)$
20: $g_{min} \leftarrow +\infty$
21: **for all** $y' \in Y'$ **do**
22: $g(y') \leftarrow \max(g(y'), |d - d(y',y)|)$
23: $\{d(y',y)$ is precomputed, because $y', y \in Y\}$
24: **if** $g(y') > d_{min}$ **then**
25: $Y' \leftarrow Y' \setminus \{y'\}$
26: **else if** $g(y') < g_{min}$ **then**
27: $g_{min} \leftarrow g(y')$**;** $y_{min} \leftarrow y$
28: **end if**
29: **end for**
30: $y \leftarrow y_{min}$
31: **end while**
32: **end for**
33: **return** d_{min}

graph building process. Since these distances are repeatedly used during calculation of many edge weights, the overhead associated with precalculating them does not have a significant impact on the overall cost of building the graph.) The algorithm processes one point $x \in X$ at the time (lines 12-31) and for this point maintains two structures: a set Y' that contains points $y \in Y$, for which $d(x,y)$ has not been calculated and may be smaller than d_{min} (current estimate of $D(X,Y)$) and a vector g that associates with each vertex $y \in Y$ a lower bound $g(y)$ of $d(x,y)$. Initially Y' contains all the elements of Y and all $g(y)$ are set to zero (lines 12-15). The size of Y' is gradually reduced by the following process (lines 18-30), which is repeated until Y' is empty, and starts with an arbitrarily selected point y.

The point y is removed from Y' and the value $d(x,y)$ is calculated (line 18). If $d(x,y)$ is smaller than d_{min}, then d_{min} is set to $d(x,y)$. For each $y' \in Y'$, the value of $g(y')$ is updated by setting it to $\max(g(y'), |d(x,y) - d(y',y)|)$. All elements $y' \in Y'$ for which $g(y')$ is larger than d_{min} are removed from Y'. The element $y' \in Y'$ for which $g(y')$ is smallest becomes the next point y.

When Y' is empty, the distance $d(x,y)$ has been either calculated or determined to be larger than d_{min} for all $y \in Y$. Therefore the algorithm is ready to process next point x. When all $x \in X$ have been processed, $d_{min} = D(X,Y)$.

The use of the triangle inequality (4.4) sometimes significantly reduces the number of distances $d(x,y)$ that must be computed, and this lowers the cost of finding the distance $D(X,Y)$. However, for some datasets, only few calculations of $d(x,y)$ can be avoided using this technique. (Unfortunately, this happens to be the case in our application area.) In such cases, the cost of finding the distance $D(X,Y)$ may increase due to the overhead associated with computation and testing of the values $g(p)$.

4.3.3 Applicability to the Graph Building Problem

In order to evaluate applicability of the triangle inequality based method to our graph building problem, we conducted a series of experiments designed to compare the performance of the triangle inequality based method to the performance of brute-force method (described in Section 4.2). All data sets used in these experiments were synthetically generated by a process designed to produce sets closely reassembling those found in 3D-EM (see [24] for the details). According to the results of our experiments, the triangle inequality based method is more than 5 times slower than the brute-force method. These disappointing result can be explained as follows.

The AESA based algorithm attempts to lower the computational cost of finding $D(X,Y)$ by reducing the number of pairs $x \in X$, $y \in Y$ for which the computation of the distance $d(x,y)$ is necessary. The distance $d(x,y)$ is calculated only for these pairs x, y, for which the lower bound of $d(x,y)$ (calculated using triangle inequality) smaller than the current estimate of $D(X,Y)$ (the smallest $d(x,y)$ encountered so far). Since the cost of calculating and testing the lower bound of distance $d(x,y)$ is lower than the cost of calculating $d(x,y)$, the time is saved on pairs x, y for which the calculation of $d(x,y)$ is avoided. However, the cost of calculating and testing the lower bound of distance $d(x,y)$ adds to the cost of calculating $d(x,y)$ for those pairs x, y for which the calculation $d(x,y)$ is necessary. The total cost of finding $D(X,Y)$ is reduced only if the computation of $d(x,y)$ is avoided for a sufficiently large number of pairs x, y. The poor performance of AESA based algorithm can be explained by the fact that the computation of $d(x,y)$ was avoided only for less than 20% of the pairs x, y. Clearly, the spatial distribution of points in sets representing EM images is such that AESA based approach cannot be applied to them.

4.4 Early-Termination Method

4.4.1 Early-Termination

Early-termination is an optimization technique that has been used in many applications (see [2], for example). The basic principle behind this technique is to terminate the computation of some value, when based on partial result (calculated so far) and some application-dependent rule, it can be determined that the computation of this value is unnecessary. When the cost of the overhead associated with termination condition checking is smaller than the cost of the avoided computations the overall cost is reduced. The following illustrates how the early-termination can be used to reduce the cost of finding the distance $D(X,Y) = \sqrt{S(X,Y)}$ between two sets of points $X,Y \in \Omega$.

Let Q be such that $Q \neq \varnothing$ and $Q \subseteq \{1,\ldots,N\}$. We define the partial sum $s_Q(a,b)$, for any two points $a,b \in \mathbb{R}^N$, as

$$s_Q(a,b) = \sum_{n \in Q} (a_n - b_n)^2, \tag{4.6}$$

where a_n and b_n are correspondingly n-th coordinates of points a and b in N-dimensional Euclidean space. Using this definition, $s(a,b)$ can be expressed as

$$s(a,b) = s_Q(a,b) + s_{\bar{Q}}(a,b), \text{ where } \bar{Q} = \{i | 1 \leq i \leq N \text{ and } i \notin Q\}. \tag{4.7}$$

Now let us assume that we have already calculated the value $s(x',y')$ for a pair of points $x' \in X$ and $y' \in Y$. Then this value constitutes an upper bound for the calculation of $S(X,Y)$ because

$$S(X,Y) \leq s(x',y'). \tag{4.8}$$

If the value of $s_Q(x,y)$ in (4.7), for some pair of points $(x,y) \in X \times Y$, is greater or equal to $s(x',y')$, then the calculation of $s_{\bar{Q}}(x,y)$ for these points is not necessary (the value of $s(x,y)$ for this pair must be greater or equal to $s(x',x')$). If a tight upper bound of $S(X,Y)$ can be found early, then the calculation of $s_{\bar{Q}}(x,y)$ may not be necessary for many pairs (x,y), which results in a significant reduction of the computational cost of finding $S(X,Y)$.

4.4.2 Algorithm

Our algorithm (see Algorithm 3) implements, with a slight modification, the early-termination technique that is described in Section 4.4.1. The computation of the value $s(x,y)$ is performed according to the formula:

$$s(x,y) = s_{Q_1}(x,y) + s_{Q_2}(x,y) + s_{Q_3}(x,y), \tag{4.9}$$

where Q_1, Q_2, Q_3 form a partition of $\{1,\ldots,N\}$. The algorithm takes these three subsets Q_1, Q_2, Q_3 and an integer R as parameters. The first stage (lines 13-27) of Algorithm 3 computes and saves the values of $s_{Q_1}(x,y)$, for all pairs $(x,y) \in X \times Y$, and identifies R pairs (x,y) for which the partial sums $s_{Q_1}(x,y)$ are the smallest (the values x, y and $s_{Q_1}(x,y)$ for these pairs are stored in vectors $\hat{X}$, $\hat{Y}$ and $\hat{S}$). In the second stage (lines 29-32) of Algorithm 3, the values of $s(x,y)$ are computed for the R pairs (x,y) identified during the first stage. This is done using (4.9), in which the value $s_{Q_1}(x,y)$ has been already computed in the first stage. The minimum of values $s(x,y)$ computed in this stage is used in the third stage as an upper bound s_{min} for value of $S(X,Y)$. In the third stage (lines 33-39) of Algorithm 3, the search for the $S(X,Y)$ is conducted. For each of the pairs $(x,y) \in X \times Y$ the algorithm checks if the value of $s_{Q_1}(x,y)$ (calculated in first stage) is greater than s_{min} (the current estimate of upper bound of $S(X,Y)$). If the value of $s_{Q_1}(x,y)$ for a pair (x,y) is

Algorithm 3 Early-Termination.

{**Parameters:** }
{ Q_1, Q_2, Q_3: subsets of $\{1, \ldots, N\}$ that form its partition (N is the number }
{ of dimensions) }
{ R: number of pairs (x, y) with the lowest partial sums $s_{Q_1}(x, y)$ that are }
{ stored }
{**Input:** }
{ X, Y: nonempty sets of points in $\mathbb{R}^N$ }
{**Internal objects and functions:** }
{ $\hat{X}, \hat{Y}, \hat{S}$: vectors that store x, y and the partial sums $s_{Q_1}(x, y)$ for R }
{ pairs (x, y), for wich the values $s_{Q_1}(x, y)$ are the smalest }
{**Output:** }
{ $D(X, Y)$: the distance between sets of points X and Y }
$r_{last} \leftarrow 0$; $s_{min} \leftarrow \infty$
for all $(x, y) \in X \times Y$ **do**
  $s \leftarrow s_{Q_1}(x, y)$
  if $s < s_{min}$ **then**
    $r_{last} \leftarrow \min(r_{last} + 1, R)$
    $r \leftarrow r_{last}$
    while $r > 1$ and $\hat{S}[r-1] > s$ **do**
      $\hat{X}[r] \leftarrow \hat{X}[r-1]$; $\hat{Y}[r] \leftarrow \hat{Y}[r-1]$; $\hat{S}[r] \leftarrow \hat{S}[r-1]$; $r \leftarrow r-1$
    end while
    $\hat{X}[r] \leftarrow x$; $\hat{Y}[r] \leftarrow y$; $\hat{S}[r] \leftarrow s$
    if $r = R$ **then**
      $s_{min} \leftarrow \hat{S}[r]$
    end if
  end if
end for
{Here $(\hat{X}[1], \hat{Y}[1]), \ldots, (\hat{X}[R], \hat{Y}[R])$ are the R pairs with smallest $s_{Q_1}(x, y)$}
$s_{min} \leftarrow \infty$
for $r = 1, \ldots, R$ **do**
  $s_{min} \leftarrow \min(s_{min}, s_{Q_3}(\hat{X}[r], \hat{Y}[r]) + s_{Q_2}(\hat{X}[r], \hat{Y}[r]) + s_{Q_1}(\hat{X}[r], \hat{Y}[r]))$
end for
for all $(x, y) \in X \times Y$ **do**
  if $s_{Q_1}(x, y) < s_{min}$ **then**
    if $s_{Q_2}(x, y) + s_{Q_1}(x, y) < s_{min}$ **then**
      $s_{min} \leftarrow \min(s_{min}, s_{Q_3}(x, y) + s_{Q_2}(x, y) + s_{Q_1}(x, y))$
    end if
  end if
end for
return $\sqrt{s_{min}}$

greater or equal to the value of s_{min}, then the algorithm examines next the pair. If the value of $s_{Q_1}(x,y)$ for a pair (x,y) is smaller than the value of s_{min}, then the algorithm computes the sum $s_{Q_1}(x,y)+s_{Q_2}(x,y)$. Since the value of $s_{Q_1}(x,y)$ is already known (it was calculated in the first stage), only $s_{Q_2}(x,y)$ must be calculated. If the value of sum $s_{Q_1}(x,y)+s_{Q_2}(x,y)$ is larger than or equal to the value of s_{min}, then the algorithm examines the next pair. Otherwise, the algorithm computes the value of $s(x,y)$ as $s(x,y)=s_{Q_1}(x,y)+s_{Q_2}(x,y)+s_{Q_3}(x,y)$ (the values of the sums $s_{Q_1}(x,y)$ and $s_{Q_2}(x,y)$ are reused from previous computations). If the computed value of $s(x,y)$ is smaller than s_{min}, then the value of s_{min} is set to that value. The algorithm terminates after examining all pairs (x,y) in the third stage and returns the square root of s_{min} as the distance $D(X,Y)$.

4.4.3 Applicability to the Graph Building Problem

The process of evaluating the applicability of the early-termination method to our graph building problem was very similar to the one used for the evaluation of the triangle inequality based method. In both, the same synthetically generated datasets were used. As before, the brute-force method (described in Section 4.2) was used as the benchmark.

However, since the early-termination method has four free parameters (Q_1, Q_2, Q_3 and R), an additional step was necessary in the evaluation to determine their optimal values. The number of dimensions, N, for the datasets used in the evaluation was equal to 81. Recognizing that most of the useful information is concentrated in the center of the image, we decided to divide $\{1,\ldots,81\}$ among sets Q_1, Q_2, Q_3 in the following way:

$$Q_1 = \{41-n_1,\ldots,41+n_1\}$$

$$Q_2 = \{41-n_2,\ldots,41-n_1-1,41+n_1+1,\ldots,41+n_2\}$$

$$Q_3 = \{1,\ldots,41-n_2-1,41+n_2+1,\ldots,81\} = \{1,\ldots,81\} \setminus (Q_2 \cup Q_3).$$

In order to determine the optimal values of the parameters we tested run-times of the early-termination algorithm with different values of n_1, n_2, R on a randomly selected, small subset of the projection images. Based on the results of these tests, we chosen $n_1 = 13$, $n_2 = 22$, $R = 20$ for the reminder of the evaluation procedure (see [24] for the details).

The experimental evaluation has shown that, when applied to datasets that closely resemble those found in 3D-EM, the early-termination method is more than 45% faster than the brute-force method. The good performance of the early-termination method indicates that the assumption about the concentration of useful information in the center of the image is correct. The calculation of the sum $s(x,y)$ (4.9) was necessary only for approximately 4% of the pairs $(x,y) \in X \times Y$. For approximately 65% of pairs, only the sum $s_{Q_1}(x,y)$ was computed.

4.5 Further Improvements

Our results indicate that the cost of constructing graphs to classify EM projection images can be significantly reduced by using the method of early-termination. An additional reduction of the graph building cost has been achieved by the implementation of this method, which takes advantage of the parallel processing capabilities available on practically all personal computers (MMX and SSE instructions of Intel based processors). This implementation allows us to calculate simultaneously four elements in the summation in (4.6). By dividing $\{1,\ldots,81\}$ into three sets $Q_1 = \{25,\ldots,56\}$, $Q_2 = \{1,\ldots,24\}$, $Q_3 = \{57,\ldots,81\} = \{1,\ldots,81\} \setminus (Q_2 \cup Q_3)$ we were able to maximize the use of parallel capabilities of our hardware (since the numbers of dimensions Q_1 and Q_2 are divisible by four and sum $s_{Q_2}(x,y)$ is computed infrequently, almost always four elements of the summation in (4.6) are calculated simultaneously). This parallel implementation of the early-termination

method allowed us to further reduce the cost graph construction approximately by an additional 50% (no rigorous performance evaluation of this implementation has been conducted, therefore only an approximate value can be provided).

Since even with this reduction, significant computer resources must be dedicated to the graph construction, it may be desirable to explore applicability of other optimization techniques to the process of constructing graphs to classify EM projection images. Our method significantly reduces the cost of constructing graphs to classify EM projection images by relatively simple means. Future research is necessary to test applicability of other, more advanced methods to this task. Since the problem of searching for nearest neighbor has been intensively studied in many domains, the number of methods which could be considered is quite large. Several nearest neighbor searching algorithms have been developed to quantize image vectors (an early example of such algorithm can be found in [20]). Since these algorithms were designed for searching in spaces with small number of dimensions, their applicability to our problem may be limited. As indicated by Yianilos [51], the methods based on kd-trees or on constructions of computational geometry become inefficient as the number of dimensions increases. However, some methods, such as the one proposed by Lai et. al. [26], are better suited for high-dimensional spaces and are more likely to reduce the computational cost of constructing graphs to classify EM projection images. The results of our initial experiments with the algorithm of [26], adapted to the computation of $D(X,Y)$, suggest that, when applied to the data sets that are discussed above, this algorithm is approximately 13% faster than the brute-force algorithm and 63% slower than the early-termination algorithm.

Chapter 5

Graph Cutting Algorithm

Despite the computational complexity suggested by the theory, solutions or sufficiently good, from the practical perspective, approximations of the solutions for some large instances of the hard optimization problem that are encountered in practice can be found within acceptable time without employing extraordinary computing power. Since these instances are not generated by an unbiased random process, they may preferentially belong to a subset of all possible manifestations of the problem that has some special proprieties, which can be used to reduce the cost of solving the optimization problem. The performance of an algorithm that has been designed to take advantage of such properties can be significantly better for such instances than the performance of a general purpose algorithm.

As we explained in Section 3.4, the Max k-Cut problem encountered in our reformulation of the EM image classification problem is NP-complete. We also demonstrated that, due to the large size of our datasets, the cost of finding an exact solution (using an off-the-shelf algorithm) for the corresponding instances of the Max k-Cut problem is very high.

Since a some small number of misclassified projections does not significantly impact on the quality of the 3D reconstructions, from the perspective of our appli-

cation an approximate solution of the Max k-Cut problem is acceptable. As illustrated by Figure 3.1, the distribution of edge weights in the graph representing our instances of the Max k-Cut problem has some regularities (e.g., most of the edges have weights that fall into a relatively small range). A significant effort has been made to construct an algorithm that exploits these regularities to produce a desirable partitioning of the EM projection images (one that approximately minimizes the expression in (3.1)) at a reasonable computational expense. After many attempts, a successful algorithm (utilizing a tabu search, a standard method of combinatorial optimization [34]) was constructed.

5.1 Concept

One of the simplest method for finding an approximate solution to a combinatorial optimization problem is greedy search. When applied to a minimization problem, this method attempts to find a configuration for which the value of the objective function is minimal by gradually evolving some initial configuration in a sequence of modification steps. At each step the method selects and executes, from all allowed modifications, the one that results in the largest decrease of the objective function. When a configuration is reached for which none of the allowed modifications results in a decrease of the objective function, the method stops. The final configuration reached is considered to be an approximation of the optimal arrangement. The deficiency of this method is that it stops after encountering a local minimum of the objective function. Consequently, it is likely that both the configuration and the corresponding value of the objective function returned by this method are far from optimal.

Tabu search [34] is a superior variation of greedy search. It incorporates a mechanism that allows continuation of the search for the global minimum even af-

ter a local minimum has been reached. A slightly different rule for selecting the next modification is used, because at a local minimum there is no modification that decreases the value of the objective function. In such circumstances, the modification that results in the smallest increase of the objective function is selected. However this modified rule by itself is not sufficient to allow for the continuation of the search. Having selected at a local minimum the modification that results in a new configuration with a minimally higher value of the objective function, it is likely that in this configuration the best (according to the selection rule) modification is the one that results in the configuration that corresponds to the just visited local minimum. In this situation, unless an additional mechanism is provided, the method would alternate between these two configurations infinitely. In a tabu search, this problem is solved by using the tabu list. A tabu list has a finite length (a parameter of the method) and is used to keep track of recently executed modifications. With the exception of some special cases (see the next section for details) the modifications recorded on the tabu list are not allowed to be reversed. This ensures that the method does not repeatedly return to the same configuration.

The tabu search concept can be applied in the following way to our graph cutting problem. A cut of the complete graph into the user-provided number of subgraphs corresponds to what was referred to in the description of the tabu search method as a configuration. In each iteration of the algorithm, the current cut is modified by reassigning a single node to a different subgraph (class). The edges between the reassigned node and the other nodes of class from which it was removed are cut. Edges between the reassigned node and nodes of class to which it is reassigned are added. In order to escape local minima, the nodes affected by recent reassignments are recorded on the tabu list. The detailed description of an algorithm based on this concept is provided in the following section.

5.2 Operation

The algorithm has three parameters K, I and t. The value of K determines the number of subgraphs into which the graph should be cut (this is the number of classes in our classification problem). The parameter I defines the number of iterations to be executed. The length of the *tabu list* is specified by the value of the parameter t.

During the operation of the algorithm the following structures are maintained. A mapping vector M that defines the current cut (this vector is updated in every iteration to incorporate changes introduced by it). A mapping vector M_{min} that defines the best cut found by the algorithm. The tabu list T that keeps track of recent node reassignments and the objective function values associated with them (the length of this list is determined by the value of parameter t).

The algorithm frequently calls two functions $C(G,M)$ and $C_{new}(G,M,v,k)$. Both are used to calculate the value of the objective function. The function $C(G,M)$, for the given graph G and the mapping vector M, returns the sum of the weights w of edges (v_1,v_2,w) in graph G such that $M(v_1) = M(v_2)$ (i.e., the sum of weights of all inter-class edges). Similarly, the function $C_{new}(G,M,v,k)$ returns the sum of the weights w of edges (v_1,v_2,w) in graph G such that $M_{new}(v_1) = M_{new}(v_2)$, where $M_{new}(x) = \begin{cases} M(x), \text{ for } x \neq v \\ k, \text{ otherwise} \end{cases}$ (i.e., the sum of weights of all inter-class edges assuming that node v have been reassigned to class k).

The flowchart of the algorithm is shown in Figure 5.1. The pseudo code is provided in Algorithm 4.

The algorithm starts with randomly assigning each the graph nodes to one of the classes (lines 20-21 of pseudo code Algorithm 4). This defines the value of vector M (the initial cut).

This initial and subsequent classifications are modified in the main loop of the algorithm by reassigning one of the nodes to a different class. Before entering the

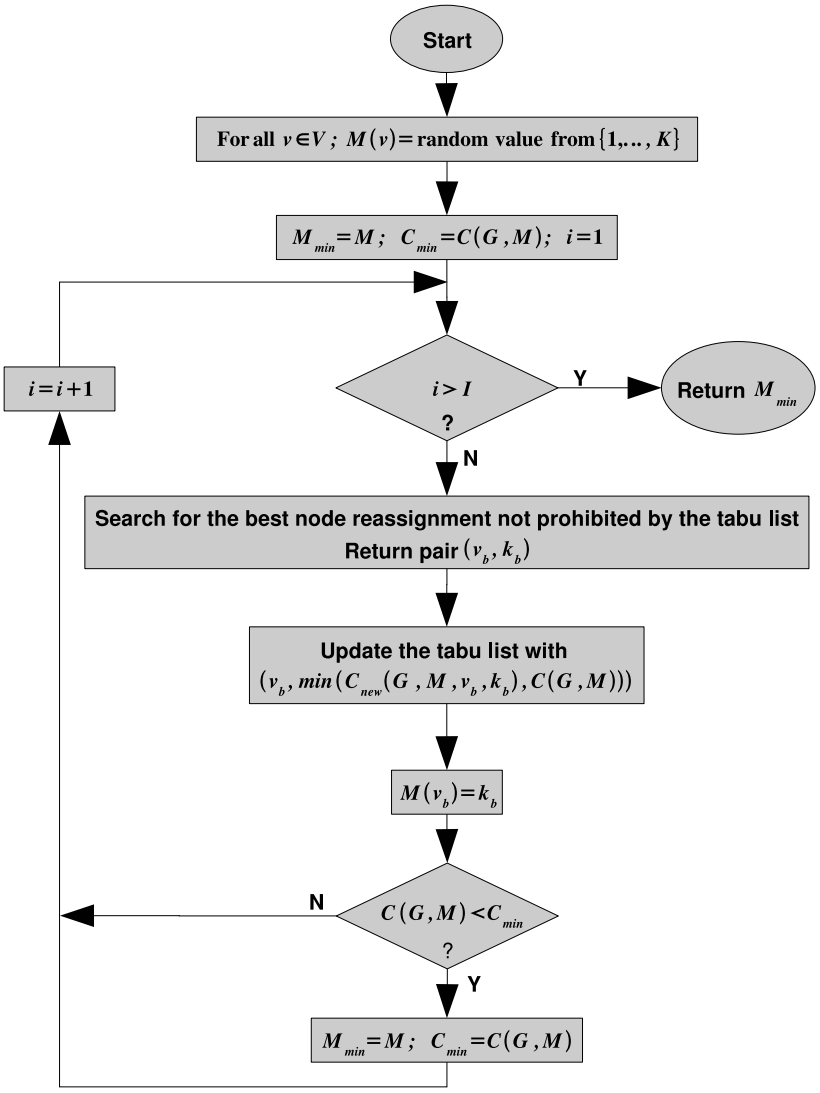


Figure 5.1: Flowchart - Graph cutting algorithm.

Parameters: K - number of classes, I - number of iterations, t - length of tabu list; Input: G - compete weighted graph with nodes V end edges E; Used objects and functions: T - tabu list, M - mapping vector that assigns to each element $v \in V$ a label $M(v) \in \{1,\ldots,K\}$, $C(G,M)$ - function that returns the sum of weights of all inter-class edges, $C_{new}(G,M,v,k)$ - function that returns the sum of weights of all inter-class edges assuming that node v have been reassigned to class k; Output: M_{min} - best mapping vector found by the algorithm. Details of "Search for the best node reassignment not prohibited by tabu list" and "Update tabu list" are provided by additional flowcharts (Figures 5.2 and 5.3 respectively).

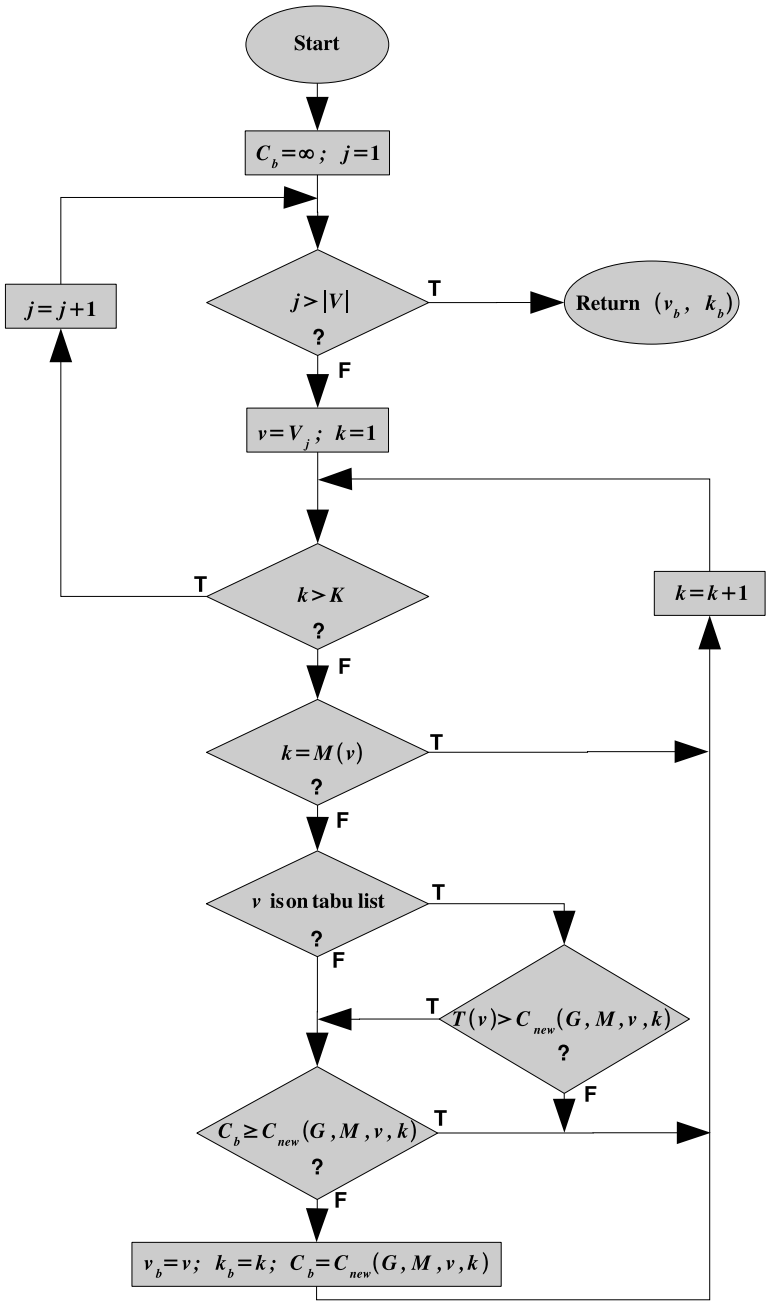


Figure 5.2: Flowchart - Graph cutting algorithm (Search for best node reassignment).
Used objects and functions: T - tabu list, M - mapping vector that assigns to each element $v \in V$ a label $M(v) \in \{1,\ldots,K\}$, $C(G,M)$ - function that returns the sum of weights of all inter-class edges, $C_{new}(G,M,v,k)$ - function that returns the sum of weights of all inter-class edges assuming that node v have been reassigned to class k; Output: (v_b,k_b) - found reassignment (v_b is the node to be reassigned, k_b is its new assignment). (Note: V_j is used to denote j-th element of set V.)

Algorithm 4 Graph Cutting Algorithm.

{**Parameters:** }
{ K: number of classes }
{ I: number of iterations }
{ t: length of tabu list }
{**Input:** }
{ G: complete weighted graph with nodes V and edges }
{ $E = \{(v_1, v_2, w) \mid v_1, v_2 \in V, w \in \mathbb{R}\}$ }
{**Internal objects and functions:** }
{ M: mapping vector that assigns to each element $v \in V$ a label }
{ $M(v) \in \{1, \ldots, K\}$ }
{ (this label determines the partition of V to which v belongs) }
{ T: tabu list of length t that contains pairs (v, c), where $v \in V$, $c \in \mathbb{R}$ }
{ $C(G, M)$: objective function (to minimize), $C(G, M)$ returns the sum of }
{ the weights w of edges (v_1, v_2, w) in graph G such that $M(v_1) = M(v_2)$ }
{ $C_{new}(G, M, v, k)$: function that returns the sum of the weights }
{ w of edges (v_1, v_2, w) in graph G such that $M_{new}(v_1) = M_{new}(v_2)$, }
{ where $M_{new}(x) = \begin{cases} M(x), & \text{for } x \neq v \\ k, & \text{otherwise} \end{cases}$ }
{**Output:** }
{ M_{min}: best mapping vector found by the algorithm }
for all $v \in V$ **do**
  $M(v) \leftarrow randomvalue\, from \{1, \ldots, K\}$
end for
$M_{min} \leftarrow M$; $C_{min} \leftarrow +\infty$
for $i = 1, \ldots, I$ **do**
  find a pair $v_b \in V$ and $k_b \in \{1, \ldots, K\}$ such that
    $k_b \neq M(v_b)$
    and
    v_b is not on the tabu list or $T(v_b) > C_{new}(G, M, v_b, k_b)$
  for which $C_{new}(G, M, v_b, k_b)$ is minimal
  if v_b is on the tabu list **then**
    $T(v_b) \leftarrow min\{C_{new}(G, M, v_b, k_b), C(G, M)\}$
  else
    if the tabu list reached its maximum size t **then**
      remove last item from the tabu list
    end if
    add pair $(v, min\{C_{new}(G, M, v_b, k_b), C(G, M)\})$ to the front of the tabu list
  end if
  $M(v_b) \leftarrow k_b$
  if $C(G, M) < C_{min}$ **then**
    $M_{min} \leftarrow M$; $C_{min} \leftarrow C(G, M)$
  end if
end for
return M_{min}

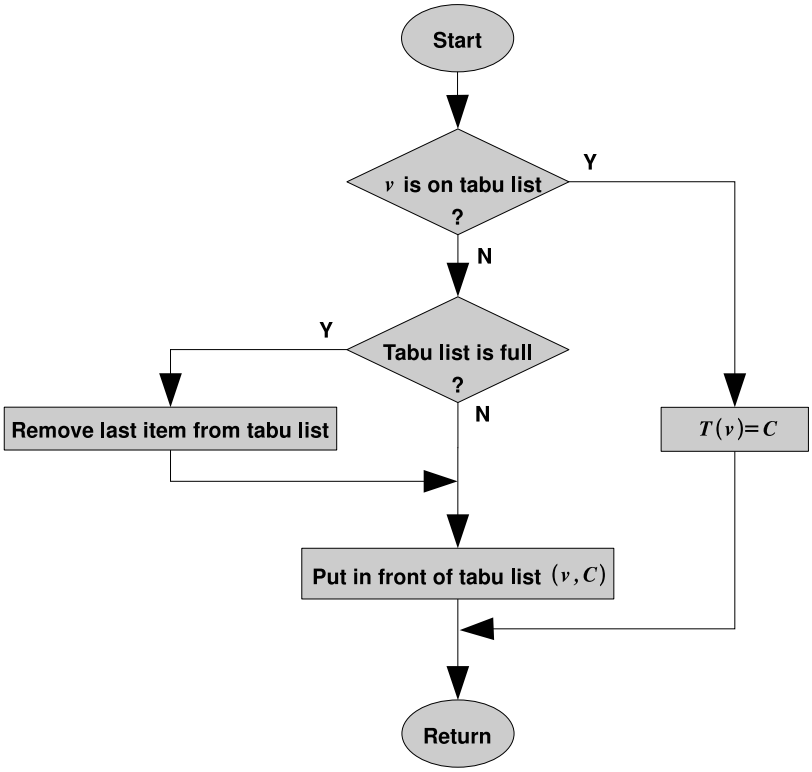


Figure 5.3: Flowchart - Graph cutting algorithm (Update tabu list).
Parameters: v - recently reassigned node, C - minimum of the objective function values for cuts before and after the reassignment of v; Used objects and functions: T - tabu list.

main loop, the algorithm sets the value of M_{min} to be the same as M (line 23). The body (lines 25-41) of the main loop (lines 24-42) is executed I times. Each of these iterations starts with the search for the node reassignment that is not prohibited by the tabu list mechanism and results in the lowest value of the objective function (lines 25-29 see also Figure 5.2). All possible reassignments of nodes $v \in V$ in graph G are considered. The tabu list is checked to prohibit the reassignments which involve nodes on the tabu list unless they result in value of the objective function lower than the one recorded on the list. The reassignment that results in the smallest value of the objective function is selected. This process returns node v_b and its new class (subgraph) assignment k_b. Once such reassignment is identified the tabu list is updated (lines 30-37, see also Figure 5.3). Next, the current cut M is modified by reassigning node v_b to class k_b. If the value of the objective function that corresponds to the new cut is lower than any observed so far (C_{min}), then value mapping vector M_{min} is set to the value of vector M and the value of C_{min} is updated. After exiting the main loop vector M_{min} represents the cut which corresponds to the lowest encountered value of the objective function (the best encountered approximation of the Max k-cut).

5.3 Parameters

As indicated in the previous section, our graph cutting algorithm takes three parameters (K, I and t). The selection of the correct values of these parameters for a particular data set is very important. It has an impact on both the operation of the algorithm and the usefulness of the results from the EM image classification perspective. To ensure that the algorithm successfully avoids getting trapped in the vicinity of some local minimum and that it finds a desirable cut, an appropriate length of the tabu list (parameter t) and a sufficiently large number of iterations

(I) must be selected. If the tabu list is too short, the algorithm may get stuck in a loop that includes some local minimum and consequently will be unable to examine many configurations that potentially can result in a lower value of the objective function. However, an excessively long tabu list can significantly slow down the operation of the algorithm. Since the algorithm executes the number iterations predetermined by the value of parameter I, this value must be large enough to allow the algorithm to explore a sufficiently large part of the search space. If a too small value of I is selected, the algorithm may be unable to reach a near-optimal configuration, because it may require more than I reassignments to reach it from the randomly selected initial configuration. On the other hand, too large a value of I causes many unnecessary iterations to be executed.

The selection of the correct number of classes (parameter K) is less important from the algorithm's operation perspective. However, it has a significant impact on the applicability of the produced graph cuts to the underlying image classification problem. As explained in the following chapters, it might be desirable to experiment with different values of K to find the one that best matches the number of conformations represented in the dataset and/or one that compensates for the algorithm's bias towards even cuts in the situations when conformations are unevenly represented.

5.4 Implementation

The performance of our algorithm depends greatly on the efficiency of the implementation. The large number of nodes in our graphs (equal to the number of EM images being classified) causes that many possible reassignments must be considered in each iteration. For each of them the value $C_{new}(G,M,v,k)$ of the objective function must be calculated to select the best reassignment. The cost of calculating

values of $C(G,M)$ and $C_{new}(G,M,v,k)$ can be significantly reduced by maintaining edge sum tables. The complete contents of these tables need to be calculated only once (during initialization) and only local updates are necessary after each iteration.

5.5 Multiple Runs

The cut produced by our algorithm is an approximation of the Max k-Cut that depends on the initial random classification (lines 20-21 of Algorithm 4) of the nodes. From some initial configurations, the path (sequence of reassignments) to a good approximate Max k-Cut solution may be long and a very large number of iterations may be necessary to complete it. If the value of parameter I is not sufficiently large, a good approximate solution may never be reached. The chances of finding a good approximation of the Max k-Cut can be significantly increased by running the core algorithm (Algorithm 4) several times. Since each of the runs starts from different randomly selected initial cut, the likelihood that all of them are many reassignments away from a good approximations decreases. In addition to that, by comparing the results from different runs, one can estimate the robustness of the produced solution. Figure 5.4 provides a flowchart for the multi-run process of finding approximate Max k-Cut.

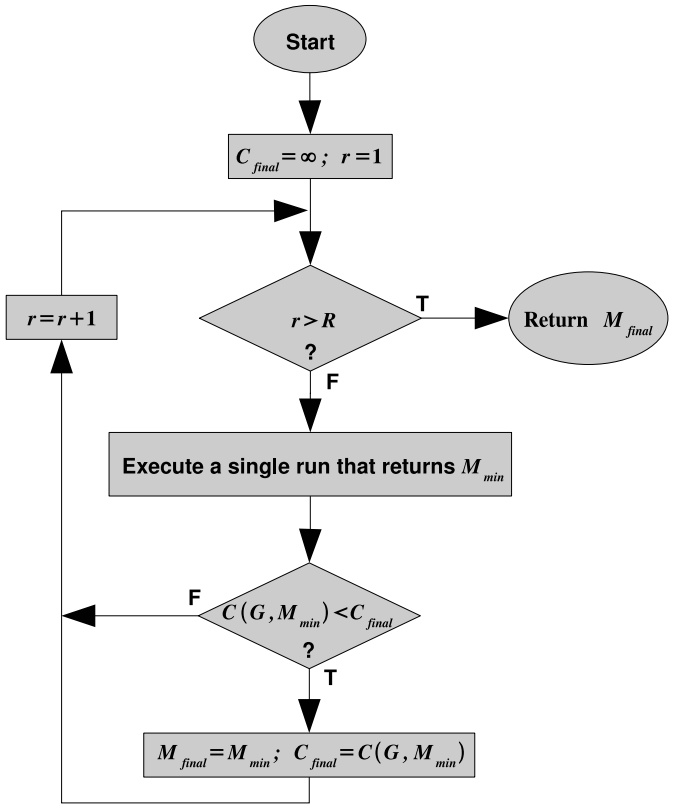


Figure 5.4: Flowchart - Finding approximate Max k-Cut by multiple runs of the core algorithm.
Parameter: R - number of runs; Input: G - graph; Used objects and functions: $C(G,M)$ - function that returns the sum of weights of all inter-class edges; Output: M_{final} - approximation of Max k-Cut is found by R runs of the core algorithm.

Chapter 6

Reconstruction Framework

The unsupervised classification of the projection images is just one among many procedures that must be preformed in order to reconstruct 3D models that accurately represent various conformations found in heterogeneous projection sets. To achieve this goal, many processing steps have to be organized in a coherent framework that determines the sequence in which these steps of the process are performed and ensures compatibility of data structures exchanged by them. The framework for reconstructing 3D models from heterogeneous projection sets that incorporates our projection image classification procedure is schematically shown in Figure 6.1. In the following sections, the processing steps of this framework are described. Some details provided in these descriptions refer to specific methods and implementations that have been used in our evaluation process (see Chapter 7). However, this does not imply that a particular method or implementation is required. Within this reconstruction framework, different methods and implementations can be used in the individual processing steps. In fact redesign of some processing steps may result in improved quality and/or efficiency of the entire reconstruction process.

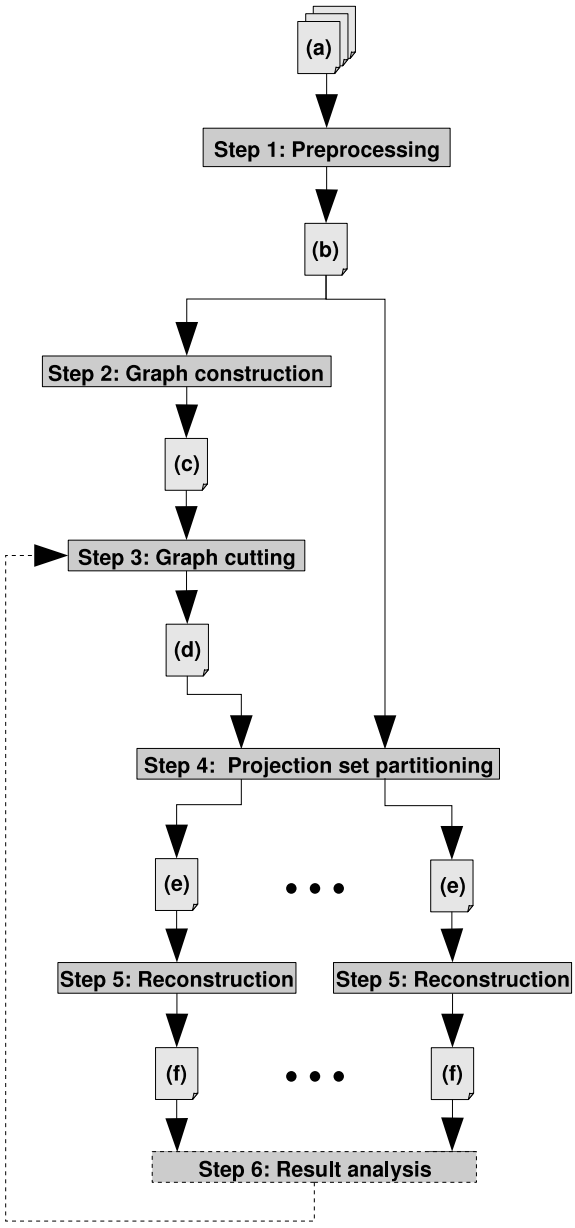


Figure 6.1: The framework for reconstructing 3D models from heterogeneous projection sets.
(a) Heterogeneous projection set
(b) File containing preprocessed (CTF corrected, aligned, masked and normalized) projection images
(c) File containing complete weighted graph that represents the heterogeneous projection set
(d) File containing class assignment of each image
(e) Homogeneous projection sets
(f) 3D reconstructions of conformations present in the heterogeneous projection set

6.1 Preprocessing

Purpose

The main objective of the preprocessing step is to remove or reduce distortions present in the projection images and prepare them for further processing. The operations performed in this step include CTF correction, masking and normalization. It is also important that the initial alignment of all the projection images to the common spatial reference is preformed in this step. Despite the high likelihood that such initial alignment is not very accurate, this is essential for successful classification of the images.

Input

A heterogeneous set of raw projection images.

Output

A heterogeneous set of CTF-corrected projection images that are masked with a circular mask, have normalized pixel values, and are approximately aligned to a common spatial reference.

Parameters

The parameters of this step vary depending on the specific method used.

Operation

A standard CTF correction procedure is applied to all images in the projection set. Each of them is masked with a circular mask using the following procedure. The

radius of the masking circle is selected to be large enough to enclose all the projections of the supports of the 3D objects. The center of the masking circle is placed in the center of the image. The pixel values outside the masking circle are set to zero. The normalization of the pixel values of each image is accomplished by subtracting a constant value from the pixel values within the masking circle so that their sum after subtraction is zero (see Section 2.3 for details).

The initial alignment of the projection images can be achieved by placing the center of the mass of the projection as close as possible to the center of the image.

6.2 Graph Construction

Purpose

The purpose of this step is to construct a graph that represents particular instance of the EM image classification problem.

Input

Preprocessed projection images.

Output

A complete weighted graph that represents an instance of the projection image classification problem.

Parameters

The number of elements (N) in vectors that used represent discrete versions of 1D projections of 2D projection images. The number of 1D discrete projections (L)

to be produced from a 2D image (this number is equal to the number of lines in sinogram).

Operation

In the current implementation (that is described in Chapter 4) the graph construction is done in two steps. In the first, sinograms are produced for each of the images found in the preprocessed heterogeneous projection set. This can be done by variety of tools. (In our initial experiments this task was performed using the code from SNARK05 [5]. In later experiments we used a dedicated program to produce 1D projection of 2D images. This task can also be performed by one of the popular 3D reconstruction packages like Xmipp [42] and SPIDER [14, 15].) In the second, the edge weights of the complete graph that represents an instance of projection image classification problem are calculated. This two step approach and use of some optimization techniques are essential to the efficiency of the process (see Chapter 4 for details).

6.3 Graph Cutting

Purpose

The purpose of this step is to find an approximate K-cut of the graph that represents an instance of the EM projection image classification problem. According to our heuristic, each subgraph defined by this cut defines a class that is approximately homogeneous. The collection of all the classes constitutes the solution to the EM projection image classification problem.

Input

A graph that represents an instance of the projection image classification problem.

Output

Approximate Max k-Cut of the graph provided as input. This cut defines an approximately homogeneous partition of the heterogeneous projection set.

Parameters

The number and meaning of parameters for this step may vary depending on the particular implementation. The proposed implementation requires that the following parameters are specified. The number of subgraphs (K) into which the graph is to be cut (this number is also the number of homogeneous subsets). The number of iterations (I) to be executed by the graph cutting algorithm. The length of the tabu list (t). The number of runs (R). See Sections 5.3 and 5.5 for detailed descriptions of these parameters.

Operation

In our implementation of this step, an approximate Max k-Cut of the graph that represents an instance of EM projection image classification problem is produced using a tabu search based graph cutting algorithm. For detailed description of this process, see Chapter 4.

6.4 Projection Set Partitioning

Purpose

Preparation of approximately homogeneous projection sets for the reconstruction procedure.

Input

The original heterogeneous set of projection images and node mapping produced by the graph cutting algorithm.

Output

A set of approximately homogeneous projection sets in a format required by the 3D reconstruction method used in the next step.

Parameters

None

Operation

This is a relatively simple step in which the projection images are grouped based on the results of graph cutting into approximately homogeneous subsets. Since there is a one to one relationship between graph nodes and projection images, this step is very easy. The exact actions performed in this step depend on particular reconstruction tool used to obtain 3D models (the particular implementation of the next step). For some reconstruction tools, the operations in this step are limited to running simple scripts that are based on the results produced by the graph cutting algorithm

to create appropriately formatted lists of 2D images that have been classified as originating from the same conformation.

6.5 Reconstruction

Purpose

Construction of a 3D model that is consistent with the projection images provided as an input. Since the projection set used to construct this model is approximately homogeneous, this model should closely approximate a conformation found in the original heterogeneous set.

Input

Approximately homogeneous projection set.

Output

A 3D model of the object represented by the projection images provided as an input.

Parameters

The parameters depend on reconstruction method used

Operation

The operations performed in this step vary depending on the specific reconstruction method used. One of the important goals of the framework design was to allow use of existing methods and tools in this step. Usually this step is performed by one of the popular 3D-EM reconstruction packages like Xmipp or SPIDER.

6.6 Result Analysis

Purpose

Since currently used classification method requires that the number of classes be provided (parameter K of graph cutting procedure), is is possible that the initially selected number of classes is not optimal. In the following chapters, we demonstrate that when operating on datasets with uneven representation of conformations, it is beneficial to run classification procedure with value of K greater than number of conformations represented in heterogeneous projection set (see the discussion in Section 7.2.4). This allows us to achieve better classification results. However, the drawback of this approach is that some conformations are represented by more than one 3D model. In such situations it maybe desirable to either produce an average of 3D object that represent the same conformation or merge the classes of the projection images that were used to produce them and repeat the reconstruction procedure on the merged projection sets. Since in both cases the obtained models are produced from larger number of projection images, their quality should be better than the quality of the models produced from smaller sets. This merging process also removes the ambiguity of results that may occur when two models of the same conformation are not identical.

In some circumstances (for example when the inappropriate value of the parameter K is used by the graph cutting procedure), not all the image sets that correspond to the results of graph cutting are sufficiently pure (they contain a mixture of projection images in which no one conformation is represented by significantly larger number of images than other conformations). Due to the inconsistencies between the images used to produce them, the 3D models reconstructed from such sets may exhibit recognizable artifacts. These models may also suffer from detectable loss of quality. If such symptoms of residual heterogeneity in the classes used to pro-

duce 3D models are identified during the result analysis step, then an additional run of the graph cutting algorithm with adjusted parameter values (which may lead to better results) should be executed.

Input

The 3D models produced by reconstruction procedures from approximately homogeneous projection sets. In some cases, prior knowledge about the molecule being reconstructed and general knowledge about the biology, chemistry and physics of the molecular structures may also be used.

Output

Adjusted values of the parameters (for example, the parameter K) for the use in additional runs. A new smaller set of 3D models in which no conformation is represented by more than one model. An estimate of some quantitative measure that reflects the quality of obtained 3D models.

Parameters

The parameters used in evaluation procedure vary depending on the particular implementation.

Operation

The operation of this step varies depending on particular implementation. It also may be influenced by the amount of available prior knowledge about the 3D structure that is being reconstructed.

Chapter 7

Evaluation

A significant effort has been made to ensure the applicability of our image classification method to 3D-EM reconstruction problems that involve heterogeneous projection sets. Since dealing simultaneously with all the issues associated with the classification of real EM projection sets is very difficult, an incremental approach to method development and evaluation was taken. Throughout the development process, many experiments on increasingly realistic datasets were performed. Gradually, as the problems associated with more idealized datasets were solved, additional complexity was introduced. This approach required a fast evaluation procedure and a high level of control over the parameters of the datasets. Our evaluation methodology and the experiments conducted to validate the proposed classification method are described in this chapter. Some of our evaluation experiments have been previously reported in [22].

7.1 Methodology

From the perspective of 3D-EM, the main objective of the heterogeneous reconstruction process is to produce high-quality 3D models of conformations present in the projection set produced with an electron microscope. In order to become a

part of this process, an image classification procedure must be capable of dealing with EM projection images and produce classifications that result in high-quality 3D models. However, it is not practical to base the performance evaluation of the classification procedure on outcome of the complete 3D reconstruction procedure that involves real EM projection images. If the performance evaluation of such a procedure is based on a figure of merit that measures the quality of the reconstructed 3D models, it may produce results that depend on the specific reconstruction procedure used to construct the models. It also may be unable to detect small differences between the number of misclassifications that occurred in various experiments. The impact of such differences on the quality of 3D models may be small; however, the information about them may be useful (e.g., it might help in the development of an improved classification method). Due to the low level of control over the parameters of datasets obtained with an electron microscope, the range of experiments that can be conducted using such sets is very limited. Therefore a comprehensive evaluation of an image classification method is only possible when synthetically generated projection sets are used. Various characteristics of synthetic datasets are fully controlled and can be independently changed. Since for such sets the "perfect" classification is known, they can be used in an efficient and unbiased evaluation procedure that does not include 3D reconstructions.

We conducted many experiments to evaluate the performance of our classification procedure (some of them are described later in this chapter). Since these experiments were designed to emulate various problems encountered in 3D-EM and were conducted using different tools, they did not follow exactly the same procedure. However, in the majority of them, the dataset generation and the results evaluation were performed in a similar way.

The dataset generation was done in two steps. First, noiseless heterogeneous projection sets were constructed by mixing noiseless 2D projection images. The

2D projection images used in this step were obtained using either a specially designed program or one of the popular software packages (for details see below the description of the experiments) from several 3D models of structures that resemble object encountered in 3D-EM. The noise insertion was performed in the second step of data generation. In order to emulate the high level of noise present in EM projections, the intensity of each pixel in every image was modified by adding to it the value of a random variable selected from a Gaussian distribution with zero mean and appropriately selected variance σ_N^2. The following procedure was used to estimate this variance.

A (large) number of noiseless projection images with different (either evenly distributed or randomly selected) projection angles was generated from each of the 3D models used in the set of experiments. Each of these images was masked with an identical circular mask whose center was aligned with the center of the projection image. The signal variance σ_S^2 was calculated as the intensity variance of pixels (within the masking circle) in all images. Following the definition of SNR in [12] on p. 121, the noise variance σ_N^2 was calculated as $\sigma_N^2 = \sigma_S^2/SNR$. In most of our experiments reported below $SNR = 0.1$; if it is not stated otherwise, this is the value used in a reported experiment.

To illustrate some points we provide the images of selected 3D reconstructions in Section 7.4 and appendixes B, C, D, E, F, G, H, I, J. However, in order to simplify the process and to avoid bias associated with the use of a particular reconstruction method, the figure of merit used in our performance evaluation is exclusively based on the outcome of the classification procedure and does not take into account the quality of the 3D models produced at the end of the heterogeneous reconstruction process from the identified classes. Such simplification is justified to some extent since a perfect separation into homogeneous subsets clearly leads to the correct results of the following reconstructions. To measure the classification quality we

use the figure of merit called classification purity [44].

We calculated the classification purity [44] in % as follows. We create an array whose rows correspond to the conformations and whose columns correspond to the classes produced by our algorithm; an entry in the array is the number of projection images from the corresponding conformation that are put by our algorithm into the corresponding class. For example, Table 7.1 exhibits such array. Ideally, all elements of a class should come from the same conformation. We therefore define the *classification purity* in % as: 100 times the sum over columns of the maximum of the entries in the column, divided by the sum of all the entries in the table. For the three arrays in Tables 7.1, 7.2 and 7.3, the classification purities are 99.32%, 86.18% and 80.00%, respectively.

In order to increase the statistical significance of our results, many of our experiments involve groups of data sets. For each group of data sets used in the experiment, we report the mean classification purity in % (m) and the standard error of the mean (e) using the notation $m \pm e$; see, for example, Table 7.6.

7.2 Experiments with Aligned Projection Images

The main purpose of the experiments with aligned data was to test whether our method has a chance at all of solving real EM projection image classification problems. The first group of experiments was designed to test the performance of our algorithm on datasets with equal representation of two conformations. Since there is no reason to assume that in real heterogeneous projection sets objects are evenly represented, we designed a second set of experiments to evaluate the performance of our method when it is applied to sets with uneven representation of conformations. The third set of experiments was designed to test the performance of our method on datasets that contain projection images of three equally represented objects.

7.2.1 Datasets

To produce datasets for the initial experiments we used three 3D objects S6, S6x, S7, which have been described earlier (see Figure 1.6 and corresponding text in Chapter 1). First, from each of these three objects we generated 2,562 noiseless 2D projection images (81 $\times$ 81 pixels) with evenly distributed projection angles. This produced three noiseless homogeneous projection sets. The heterogeneous sets containing projections of two conformations were generated by randomly selecting (with possible repetitions) N_1 projections from the homogeneous set for the first object and then randomly selecting (with possible repetitions) N_2 projections from the homogeneous set of the second. The total number of projections in resulting heterogeneous set is $N = N_1 + N_2$. For each pair of objects (S6-S7, S6x-S7, S6-S6x), we used three different ratios (50:50, 35:65, 20:80) between the values N_1 and N_2 to produce the heterogeneous projection sets with possibly uneven representation of the objects. In our experiments we set $N = 5,000$ and $N_1 = 2,500$, $N_2 = 2,500$ for the 50:50 sets, $N_1 = 1,750$, $N_2 = 3,250$ for the 35:65 sets, and $N_1 = 1,000$, $N_2 = 4,000$ for the 20:80 sets. In order to obtain statistically reliable results, for each pair of objects (S6-S7, S6x-S7, S6-S6x) at each object representation ratio (50:50, 35:65, 20:80), the random selection was executed five times. This produced 45 noiseless heterogeneous projection sets, each containing projections of two 3D objects, in nine groups (S6-S7_50:50, S6x-S7_50:50, S6-S6x_50:50, S6-S7_35:65, S6x-S7_35:65, S6-S6x_35:65, S6-S7_20:80, S6x-S7_20:80, S6-S6x_20:80). The data sets in each group were produced from the same pair of objects and had the same representation ratio.

A similar procedure was used to produce heterogeneous sets containing 2D projections of all three 3D object. In the experiments reported here, we used $N = 5,000$ and $N_1 = 1,667$, $N_2 = 1,667$, $N_3 = 1,666$. As in the case of sets containing two 3D object, the random selection was executed five times. This resulted in five sets

(referred to as group S6-S6x-S7_33:33:33), each containing 2D projections of three 3D objects (with equal object representation).

The noise insertion procedure described in Section 7.1 have been applied to all fifty noiseless datasets. The signal variance σ_S^2 for this procedure have been estimated using all 7,686 projection images generated in the first step of data generation process. This noise variance σ_N^2 was calculated assuming $SNR = 0.1$.

7.2.2 Parameters and Settings

In order to calculate the dissimilarities between images (edge weights in the graph that represents a particular instance of our classification problem), for each 2D projection image in a given dataset we produced 240 1D projections at 1.5° angular increments with line integrals calculated for 81 equally-spaced lines for each 1D projection (240 is the value of L in (2.1), 81 is the value of N in Figure 2.2). In all the reported experiments, the classification algorithm was executed 10 times (each time starting with different randomly selected initial assignment). In all these runs, the length of the tabu list (t) was set to 3,000 and the algorithm was terminated after executing $I = 10,000$ iterations.

7.2.3 Results

In our first set of experiments, two conformations were evenly represented in the projection sets. For these experiment we used data set groups S6-S7_50:50, S6x-S7_50:50, S6-S6x_50:50 and configured our classification algorithm to identify two distinct classes. Table 7.1 provides a typical result. The results of this experiment are summarized in Table 7.6 (column: 50:50→2). For all data sets in these experiments the classification purity was 98.5% or higher. The majority of the time required by each experiment was dedicated to the graph building process (approximately 24 hours of computation on Intel Xenon 1.7 GHz with unoptimized method).

Projections of object	No. of projections assigned to	
	class 1	class 2
S6x	33	2467
S7	2499	1

Table 7.1: Example of the results from the two-class classification experiments with conformation representation ratio 50:50.
(The 3D reconstructions based on this classification are pictured in Figure B.3.)

The graph cutting process required only less than 20 minutes of computation per experiment (10 runs of graph cutting algorithm) on the same hardware.

The second set of experiments was designed to test the performance of our classification method on datasets with uneven representation of conformations. In these experiments, we used two types of sets with representation ratio 35:65 (data set groups: S6-S7_35:65, S6x-S7_35:65, S6-S6x_35:65) and 20:80 (data set groups: S6-S7_20:80, S6x-S7_20:80, S6-S6x_20:80). We set the classification algorithm to identify two distinct classes. A typical result for the representation ratio 35:65 is provided in Table 7.2, a corresponding example for the ratio 20:80 is provided

Projections of object	No. of projections assigned to	
	class 1	class 2
S6x	0	1750
S7	2559	691

Table 7.2: Example of the results from the two-class classification experiments with conformation representation ratio 35:65.
(The 3D reconstructions based on this classification are pictured in Figure C.3.)

Projections of object	No. of projections assigned to	
	class 1	class 2
S6x	8	992
S7	2535	1465

Table 7.3: Example of the results from two-class classification experiments with conformation representation ratio 20:80.
(The 3D reconstructions based on this classification are pictured in Figure D.3.)

in Table 7.3. These results clearly indicate that our method, when applied to realistic (from the EM perspective) projection sets with uneven representation, has a bias towards creating evenly sized classes. This has the effect on the produced classification that one of the classes contains almost exclusively projections of the conformation from which the majority of the projections in heterogeneous set was obtained and the other contains a mixture of the projections of both conformations. This classification successfully isolates a sufficiently large homogeneous projection set for one of the conformations to allow its reconstruction. However, it is insufficient to produce a 3D model for the conformation with the smaller representation in the heterogeneous dataset, because the object reconstructed from the class of lower purity will likely exhibit artifacts indicating that it was reconstructed from a heterogeneous set.

Our attempts to remove the bias towards even splits by modifying the objective function resulted only in limited success. We tested many different objective functions; among them:

$$\sum_{k=1}^{K} \frac{1}{|A_k| - 1} \sum_{x,y \in A_k} \bar{s}(\bar{x}, \bar{y}). \tag{7.1}$$

This function is designed to minimize the sum of the weight means of internal edges attached to each node. The use of mean in this function reduces the impact of uneven splits on its value. By treating the nodes separately and summing over their individual means, it also reinforces the fact that, when nodes are correctly classified, the mean weight of the internal edges for each of them should be as small as possible. This function reduces the bias towards even splits. Its use allowed us to achieve good classifications for all datasets groups (S6-S7_35:65, S6x-S7_35:65, S6-S6x_35:65) with representation ratio 35:65 and for a single datasets group (S6x-S7_20:80) with representation ratio 20:80. However, for all datasets in groups S6-S7_20:80, S6-S6x_20:80 even splits resulted in the lowest value of the objective

function of (7.1).

Our approach to correctly identifying the conformation represented by a minority of the projections is to partition the original projection set into a larger than the anticipated number of classes.

In next two sets of experiments we evaluated the performance of our classification algorithm when setting the number of classes to three for sets with representation ratio 35:65 and to five for sets with representation ratio 20:80. Typical results in these experiments are provided in Tables 7.4 and 7.5. The classification purities for all data set groups in these two experiments are summarized in Table 7.6 (columns: 35:65→3 and 20:80→5). Despite of the uneven representation, our method (when using the appropriate number of classes) was able to produce a classifications with classification purity 95.34% or higher, which allows for reconstruction of all the conformations represented in the heterogeneous projection set. Of course, in case of a real heterogeneous projection set, the correct number of classes is not known. However, it can be determined experimentally by using our classification procedure

Projections of object	**No. of projections assigned to**		
	class 1	**class 2**	**class 3**
S6x	18	95	1637
S7	1674	1575	1

Table 7.4: Example of the results from the three-class classification experiment with conformation representation ratio 35:65.
(The 3D reconstructions based on this classification are pictured in Figure C.4.)

Projections of object	**No. of projections assigned to**				
	class 1	**class 2**	**class 3**	**class 4**	**class 5**
S6x	958	1	3	35	3
S7	9	1015	1004	961	1011

Table 7.5: Example of the results from the five-class classification experiment with conformation representation ratio 20:80.
(The 3D reconstructions based on this classification are pictured in Figure D.5.)

Data sets	Mean classification purity %		
	50:50→2	**35:65→3**	**20:80→5**
S6-S7	98.73±0.06	95.97±0.11	95.66±0.12
S6x-S7	99.38±0.02	97.77±0.01	98.91±0.03
S6-S6x	98.71±0.05	95.80±0.07	96.09±0.52

Table 7.6: Mean classification purity when the number of classes is appropriate for the conformation representation ratio.
(Based on five data sets for all nine groups.)

to produce several different (2, 3, 4, 5, 6, ... -fold) partitions of the heterogeneous projection set, producing 3D reconstructions from each of them and finding one for which reconstructed objects do not exhibit artifacts indicating that averaging over multiple conformations has occurred. It is important to notice that once the graph is constructed for given a data set, it can be used in many classification runs. Therefore when experimenting with different number of partitions on given dataset, the computationally expensive part of our classification process (graph building) must be performed only once. This combined with the high speed of the our graph cutting algorithm provides a feasible solution to the uneven representation problem, by experimenting with several different partitions of the heterogeneous projection set.

In the following set of experiments we tested the performance of our classification procedure when applied to a heterogeneous projection set with even representation of three conformations. In these experiments, our classification procedure was configured to produce three classes. A typical result of this experiment is provided by Table 7.7. The mean classification purity based on five data sets was 98.3% with the standard error of the mean 0.11%.

Projections of object	No. of projections assigned to		
	class 1	class 2	class 3
S6	24	1637	6
S6x	7	29	1631
S7	1654	12	0

Table 7.7: Example of the results from the three-class classification experiment with three equally-represented conformations.
(The 3D reconstructions based on this classification are pictured in Figure E.3).

7.2.4 Discussion

The experiments involving data set groups S6-S7_50:50, S6x-S7_50:50, S6-S6x-_50:50 clearly demonstrate that the proposed method produces very good results when applied to heterogeneous data sets with even representation of two conformations.

A bias of the method toward even splits (solutions in which all classes contain approximately the same number of projections) was observed in the experiments with data set groups S6-S7_35:65, S6x-S7_35:65, S6-S6x_35:65, S6-S7_20:80, S6x-S7_20:80, S6-S6x_20:80, when the number of classes was set to two. This is caused by the overlap in the ranges of the dissimilarity measure (see the histograms in Figure 3.1). Since all the edges in the graph have approximately the same weight, the cut that removes largest number of edges tends to minimize the sum of remaining edges. The largest number of edges is removed when the subgraphs produced by the cut have the same number of nodes, therefore the method tends to produce classes with similar size. In the additional experiments, in which the algorithm was set to produce a larger number of classes (three for the datasets with representation ratio 35:65 and five for the ratio 20:80), a much higher classification purity was achieved. This indicates that our classification algorithm can handle heterogeneous datasets with uneven conformation representation. However, for such datasets, repeated experimentation might be necessary to determine the best number of classes.

When applied to datasets with uneven conformation representation, our algorithm produces a classification in which a single conformation is represented by several classes. Therefore an additional procedure to merge classes that represent the same conformation should be introduced to the reconstruction process.

Our approach to handling datasets with uneven conformation representation based partitioning the projection set into a larger number of classes than the expected number of conformations has a significant advantage over the one based on (7.1). In the approach that uses a larger number of classes and the function of (3.1), bias toward even splits is known and is the same for any representation ratio. Therefore, in situations where the representation ratio of the heterogeneous datasets is unknown (as in 3D-EM), the procedure for classifying that uses a larger number of classes and merges those that came from the same object may well be preferable to using fewer classes and the function of (7.1) (that produces results, which, depending on the representation ratio may, or may not be influenced by the bias).

The high classification purity achieved in the experiments with dataset group S6-S6x-S7_33:33:33 indicates that our algorithm can be successfully applied to heterogeneous dataset in which more than two conformations are represented.

7.3 Experiments with Misaligned Projection Images

The method used to produce datasets for the experiments of Section 7.2 guarantees that all the projections within a projection set are perfectly aligned (the geometric center of each configuration is projected exactly onto the center of the projection image). In practice, a perfect alignment of the noisy EM projections cannot be achieved. Since the misalignment of the projection images has a potentially negative impact on the performance of proposed method, we conducted additional experiments to test the impact of such misalignment on the quality of classifica-

tions produces by our method. The datasets used in these experiments have been produced by a procedure that emulates the misalignment encountered in datasets produced by an electron microscope.

7.3.1 Datasets

To test the impact of misalignment on classification quality, we used fifty datasets. To produce them we used the same conformation mixtures (S6-S7, S6x-S7, S6-S6x, S6-S6x-S7) and representation ratios (50:50, 35:65, 20:80 and 33:33:33, respectively) as those in the experiments with aligned projection images. However, the process by which these sets where produced was slightly different.

First, for each of the heterogeneous projection sets five thousand projection angles where randomly (with potential repetitions) selected in such a way that all the projection angles were equally likely. Each of these angles was used to project one of the objects to be represented in the projection set being constructed. In 50:50 case, the first 2,500 angles were used to project the first object and the remaining 2,500 angles were used to project the second object (in the case of the 20:80 and 35:65 data sets, appropriately adjusted numbers were used). However, before producing a projection image, the center of the 3D object was shifted in a randomly selected direction parallel to the projection plane by a distance selected from the Gaussian distribution with mean zero and standard deviation equal to 1/81 of the diameter of the masking circle (the misalignment in practice may very well be larger than this). For each pair of objects (S6-S7, S6x-S7, S6-S6x) at each object representation ratio (50:50, 35:65, 20:80), the random selection of projection angles and shifts was executed five times. This produced 45 noiseless heterogeneous projection sets, each containing misaligned projections of two 3D objects, in nine groups (S6-S7_Sh_50:50, S6x-S7_Sh_50:50, S6-S6x_Sh-_50:50, S6-S7_Sh_35:65, S6x-S7_Sh_35:65, S6-S6x_Sh_35:65, S6-S7_Sh_20:80,

S6x-S7_Sh_20:80, S6-S6x_Sh_20:80). A similar procedure was used to produce five heterogeneous sets (data set group S6-S6x-S7_Sh_33:33:33) containing misaligned 2D projections of all three 3D objects.

The noise insertion was performed in exactly the same way as for the experiments with aligned projection images (see Section 7.2.1). This noise variance σ_N^2 was calculated assuming $SNR = 0.1$.

7.3.2 Parameters and settings

As in the experiments with aligned projection images, we produced $L = 240$ 1D projections at 1.5° increments, each with $N = 81$ equally-spaced lines from every image in the dataset. We also used the same configuration of the classification method ($t = 3,000$, $I = 10,000$, 10 algorithm runs per experiment).

7.3.3 Results

The results of experiments with the projection sets containing misaligned projection images of two objects (data set groups: S6-S7_Sh_50:50, S6x-S7_Sh_50:50, S6-S6x_Sh_50:50, S6-S7_Sh_35:65, S6x-S7_Sh_35:65, S6-S6x_Sh_35:65, S6-S7_Sh-_20:80, S6x-S7_Sh_20:80, S6-S6x_Sh_20:80) are summarized in Table 7.8. (The 3D reconstructions produced from representative datasets in groups S6-S7_Sh_50:50, S6x-S7_Sh_50:50, S6-S6x_Sh_50:50 are provided in Appendix F, Appendix H, Appendix I correspondingly.) With the exception of the one data set (the 3D reconstructions produced from this set are provided in Appendix G), the classification purity was above 89%. This resulted in high (larger than 88%) mean classification purities (reported in Table 7.8) for all data set groups. However, in a single case our method failed to separate conformations present in the dataset. In this case the classification purity was 53.66% (which is reflected in Table 7.8 by the lowest mean classification purity and the highest standard error of the mean for corresponding

Data sets	Mean classification purity %		
	50:50→2	**35:65→3**	**20:80→5**
S6-S7_Sh	88.52±7.80	93.11±0.21	92.92±0.14
S6x-S7_Sh	98.59±0.09	97.34±0.05	98.08±0.13
S6-S6x_Sh	97.76±0.09	90.34±0.43	91.71±0.17

Table 7.8: Mean classification purity for the misaligned projection sets when the number of classes is appropriate for the conformation representation ratio.
(Based on five data sets for all nine groups.)

data set group). Such low value indicates that in this case the algorithm failed to identify meaningful (from the EM image classification perspective) classes, because both classes produced by the algorithm contained a practically identical mixture of the projection images originating from the two conformations represented in the heterogeneous projection set.

The mean classification purity based on five data sets containing misaligned projections of three objects (data set group S6-S6x-S7_Sh_33:33:33) was 96.1%, with the standard error of the mean 0.08%. (The 3D reconstructions produced from the representative datasets in this group are provided in Appendix J.)

7.3.4 Discussion

The results of experiments with misaligned projection sets demonstrate that our method is to some extent robust when applied to such data sets. The classification purity achieved in the vast majority of the experiments (all but one) is sufficient to produce high-quality 3D models of the conformations represented in the heterogeneous sets. However, the single case for which the method failed indicates that the issue of misalignment can not be ignored in practice.

There are several ways of dealing with projection image misalignment that should be considered and evaluated. One of the possible approaches is to prealign the projection images before they are used in the classification procedure. Such prealignment can be based on the use of the centers of the mass of projection im-

ages. Due to high level of noise present in the projection images, it is unlikely that a procedure based on this approach will produce a high-quality alignment. However, even such coarse alignment might be sufficient to classify projection images well enough to produce acceptable 3D models. Once these models are available, an improved alignment can be achieved by using them as references. Another way to deal with the misalignment problem is to modify the classification procedure in such a way that it becomes immune to the in-plane shifts. This can be achieved by the use of a shift-invariant distance measure for the 1D vectors. If, for example, the distance between two 1D vectors is measured as a function of magnitudes of their Fourier transforms, then our classification method becomes inherently immune to in-plane shifts; however it is possible that the resulting loss of phase information would strongly interfere with the classification efficiency. At this point it is not clear what the impact of ignoring phase information would be on classification quality. The third approach to the misalignment problem is to incorporate a search through multiple shifted versions of the projection images into the classification method. In this approach the graph building procedure would determine the weight of each edge by finding a minimum distance between several shifted versions of the the images represented by the nodes this edge connects. The computational cost associated with such a search greatly depends on the number of shifted versions of each image to be considered. The methods for dealing with the misalignment problem are part of our ongoing research. It is possible that the best solution to this clearly nontrivial problem will be achieved by a combination of the methods proposed above.

7.4 A Case Study Involving Externally Obtained Projection Data

To verify the correctness of the procedures used in our previous experiments (reported in Section 7.2 and Section 7.3) we conducted a case study on an independently developed heterogeneous dataset (provided by Sjors H. W. Scheres). This dataset was created with different tools and procedures than those we used to produce the datasets for previous experiments. The object represented in this dataset (two conformations of the Simian Virus 40 large T-antigen) are not even similar to those represented in the datasets that we used in the development of our method and the experiments reported in Section 7.2 and Section 7.3. Due to these characteristic, our case study has a potential to detect any unintended dependence of our method on the particular data generation procedure or the type of the object that is represented in our datasets.

As in our previous experiments we used the classification purity to measure the quality of classifications produced by our method. In addition to that we used ART [18] to reconstruct two 3D models from the classes produced by our method. These models are compared with three reference 3D models. Two of the reference models were reconstructed from two homogeneous subsets of the heterogeneous projection set (perfectly classified heterogeneous set). The third was produced by reconstruction from all images contained in the heterogeneous set.

All 3D models in our study were reconstructed using the implementation of ART [28] available in Xmipp [42]. These models were produced under unrealistic assumption that the projection angles for all the images are perfectly assigned. The quality of the models reconstructed from these images with imperfect (more realistic) angular assignment may be significantly lower. All images of the 3D volumes were rendered with UCSF Chimera [37].

7.4.1 Dataset

The dataset used in the case study consisted of 5,124 projection images (100×100 pixels) of two evenly represented conformations of Simian Virus 40 large T-antigen (2,562 images with 10° bent and 2,562 images with 30° bent). These images were produced by projecting 3D models of each conformation from 2,562 approximately evenly distributed projection angles. Since in this process the geometrical center of the model was always projected onto the center of the image, the projections in the dataset are perfectly aligned to a common spatial reference. Examples of the projection images in the set are shown in Figure 7.1. Since the dataset was independently generated in another laboratory (the Centro Nacional de Biotecnologia, Spain), not all the details about the procedure used to produce it are known to us. However, it should be emphasized that its characteristics are considered to be reasonable for the propose of evaluating the methods that are intended for use in 3D-EM.

7.4.2 Parameters and Settings

As in our previous experiments we produced $L = 240$ 1D projections at 1.5° increments from every image in dataset. However, in order to match the larger image

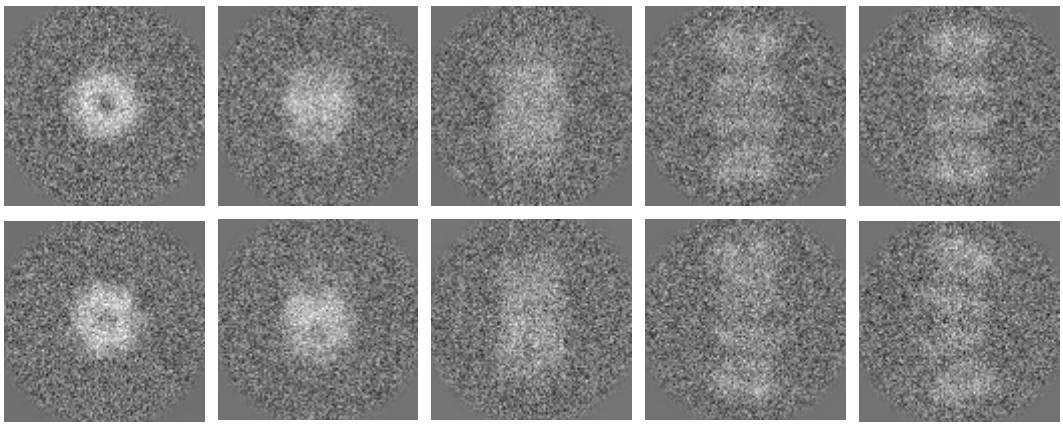


Figure 7.1: Examples of Simian Virus 40 large T-antigen projection images. Top row: projections of conformation with 10° bent, bottom row projections of conformation with 30° bent

Projections of object	No. of projections assigned to	
	class 1	class 2
10°	636	1926
30°	1904	658

Table 7.9: Classification results for dataset that contains two conformations of the large T-antigen.

size, we increased the number of lines in each 1D projection to $N = 100$. The settings of the classification algorithm were identical to those in the previous experiments ($t = 3,000$, $I = 10,000$, 10 runs).

7.4.3 Results

The 3D models obtained by two reconstruction from perfectly classified projection images are shown in Figure 7.2. Despite the high level of noise present in the projection images, the quality of 3D models produced from them is good and differences between the conformations are clearly visible.

The 3D model obtained by reconstruction from heterogeneous set is shown in Figure 7.3. The bent angle of this model is an average of bent angles of models of Figure 7.2. This reconstruction provides no information about the diverse conformations present in the projection set.

The result of the classification preformed by our method is in Table 7.9. The classification purity is 74.75%. Although this value is not as high as the classification purity values obtained in our previous experiments, the 3D models reconstructed from the produced classes well capture the differences between conformations present in the projection set (see Figure 7.4). Figure 7.5 illustrates the differences between the models obtained from classes produced by our method and and those defined by the perfect classification.

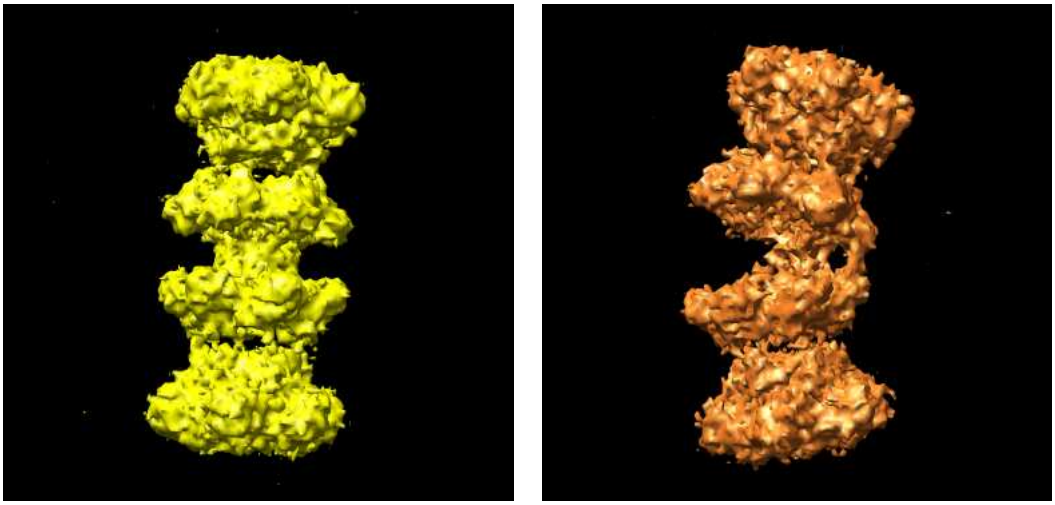


Figure 7.2: 3D models obtained by reconstructing from perfectly classified projection images of two conformations of the Simian Virus 40 large T-antigen.

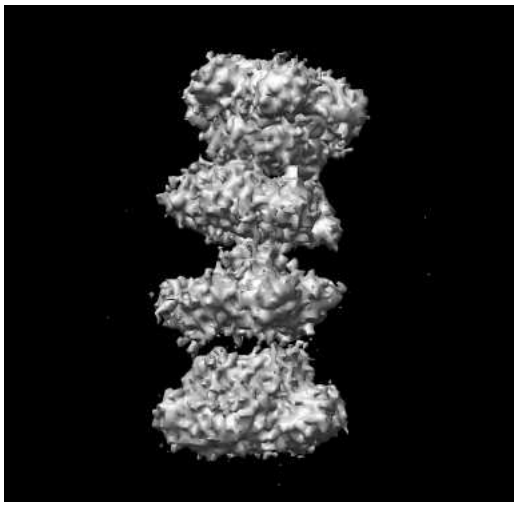


Figure 7.3: 3D model obtained by reconstructing from heterogeneous projection set that contains two conformations of the large T-antigen.

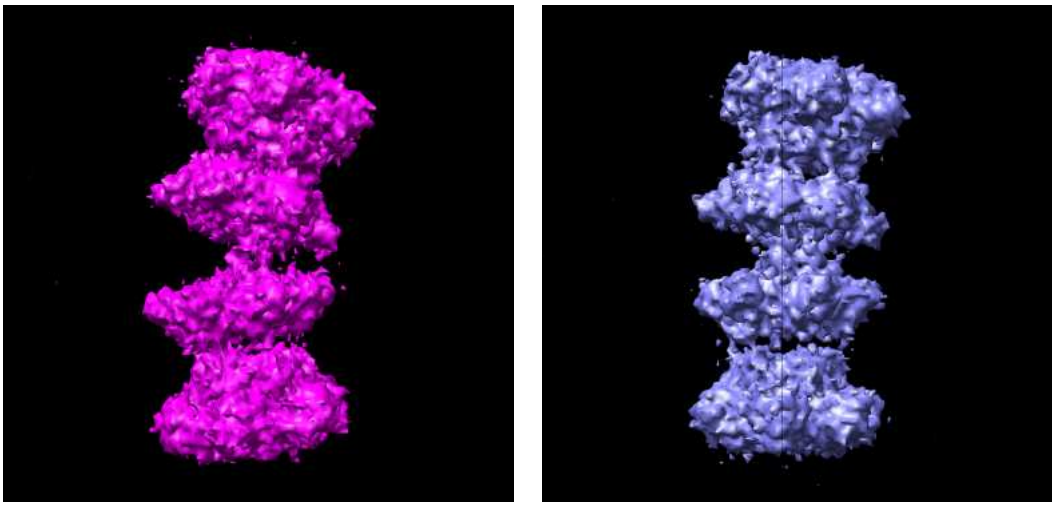


Figure 7.4: 3D models obtained by reconstructing from the projection images of two conformations of the large T-antigen classified by the proposed method.

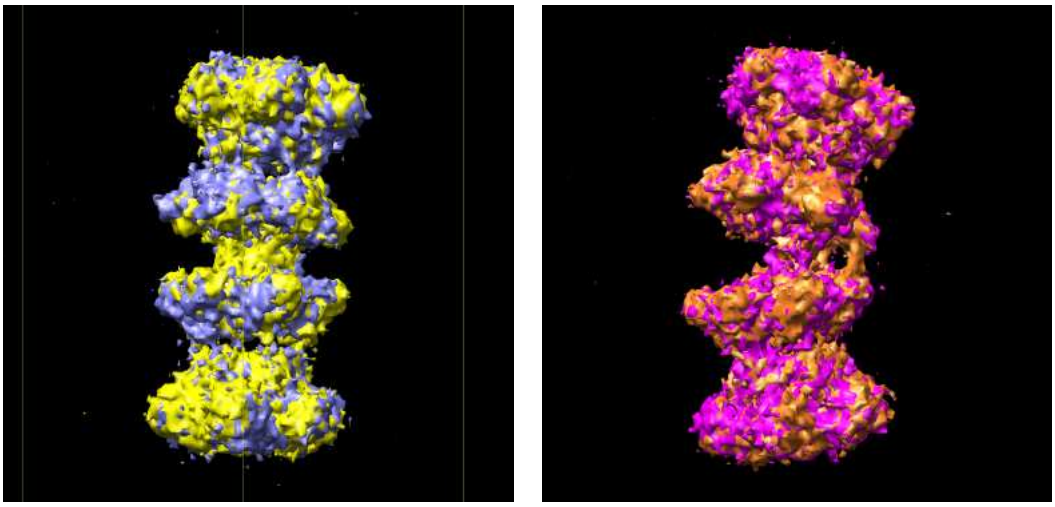


Figure 7.5: Differences between 3D models obtained by reconstructing from perfectly classified projection images of two conformations of the large T-antigen and corresponding 3D models obtained by reconstructing from these images classified by the proposed method.
The yellow and the orange models are 3D reconstructions from perfectly classified projection images. The violet and blue models are 3D reconstructions obtained from images classified by the proposed method.

7.4.4 Discussion

Since our case study is based on only one dataset, it does not have statistical significance that is expected from rigorous evaluation. Despite its limitations this study provides an important insight into some issues associated with use of projection image classification method in the heterogeneous reconstruction process.

The classification purity (74.75%) achieved in this study was significantly lower than the one achieved in our previous experiments. However, the reconstructed models well capture important differences between the conformations present in the heterogeneous dataset (one of them has a significantly smaller bent angle than the other). This confirms our expectation that even a significantly less than perfect classification of the projection images may result in acceptable 3D models. It is not clear if the quality of the models obtained with our method is sufficient for detailed structural analysis of the conformations present in the heterogeneous set. However, since these models capture essential differences between the conformations, they can be used as the initial models in multireference refinement procedure that will produce high quality reconstructions.

The classification of the projection images used in this study is inherently hard from our method's perspective. Assuming that two conformation from which they where produced differ only by the bent angle (which is approximately true), there are many pairs of projection images with the first image obtained from one conformation and the second from the other that (if impact of noise is ignored) share identical 1D projections. This is a consequence of the fact that all planar integrals over planes parallel to the plane of the bent angle are identical.

The reasonably good quality of the 3D models that was achieved despite the low value of the classification purity demonstrates a non-trivial relationship between the classification quality and the quality of the resulting 3D models. In some circumstances, a misclassification of even a large number of the projection images does not

have a significant impact on the model quality. Specifically, if two conformations have identical 2D projections, then the assignment of this projection to the class associated with either of the conformations is correct from the reconstruction perspective, despite the fact that one of these assignments is technically a misclassification. In a less extreme case, when two conformations have very similar projections, the misclassification of them is not very harmful, since each of them contributes almost the same values to the reconstruction process. When noisy projections are considered, it is possible that due to the noise a projection is less consistent with the other projections that originated from the same conformation than with the projections that originated from a different conformation. The misclassification of such a projection can lead to an improvement in reconstruction quality.

The dataset used in our case study was produced independently using different tools and procedures from the ones used in our other experiments. The 3D structure of the objects used to produce the projection images is significantly different from the structures used in our other experiments. Therefore a successful reconstruction from this dataset confirms that the mathematical principles behind our method are applicable to a range of projection image classification problems.

7.5 Evaluation of the Impact of Differences Between Conformations and Noise on Classification Quality

The difficulty of the particular projection image classification problem clearly depends on the level of noise present in the involved images. As the level of noise present in the projection images increases the classification task becomes harder. However, the noise is not the only factor that must be considered to estimate the

difficulty of such problem. The degree to which the objects represented in the heterogeneous dataset differ must also be taken into account. For example, it is more likely that a set that contains noisy projections of two significantly different 3D structures can be successfully partitioned into its homogeneous components than a set that contains noisy projections of two almost identical structures, even if the level of noise that is present in both datasets is identical.

In order to evaluate the impact of both the differences between the objects represented in a dataset and the noise level present in the images it contains on the quality of classifications produced by our method, we included in the evaluation process a specially designed set of experiments. The data sets used in these experiments are constructed to represent a wide range of differences between the objects and a wide range of noise levels. As in the previous experiments, we used classification purity to measure the quality of the results produced by our method.

7.5.1 Dataset

To evaluate the impact of noise present in the projection images and the impact of differences between conformations represented in a heterogeneous projection set on achievable classification quality, we used 500 heterogeneous datasets. To emulate varying level of differences between different conformations we generated eleven 3D objects (O1, ..., O11). The detailed description of the geometry of these objects is provided in Appendix K. We used objects O1, O3, O5, O7, O9 and O11 to produce the dataset for our experiments. The datasets we produced were divided into two sets. In the first set of datasets, the projection of object O1 are always present in the dataset (O1 is used as a reference). In the second set of datasets, the projection of object O11 are always present in the dataset (O11 is used as a reference). In order to emulate the effects of increasing differences between 3D structures of object represented in dataset, the heterogeneous datasets in both sets

were produced by combining the projection images of the reference model (O1 or O11) with one of four other objects (O3, O5, O7, O9, O11 or O9, O7, O5, O3, O1 respectively). Following these rules, we produced the heterogeneous datasets by mixing projection images of the following pairs of objects: O1-O3, O1-O5, O1-O7, O1-O9, O1-O11 and O11-O9, O11-O7, O11-O5, O11-O3, O11-O1. From each of these pairs, we produced the heterogeneous datasets at ten different noise levels (*SNR*=0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.14, 0.16, 0.18 and 0.20). This resulted in hundred dataset groups (O1-O3_0.02, ..., O1-O11_0.20 and O11-O1_0.02, ..., O11-O1_0.20). In each of these groups we generated five datasets using the following process.

First, for each of the heterogeneous projection sets two thousand projection angles were randomly (with potential repetitions) selected in such a way that all the projection angles were equally likely. Next, the first 1,000 angles were used to project the first object to be represented in the projection set being constructed and the remaining 1,000 angles were used to project the second object to be represented in the projection set being constructed. This resulted in a heterogeneous projection set containing 2,000 noiseless projection images of two evenly represented objects. Since in this process the geometrical center of the model was always projected onto the center of the image, the projections in the dataset are perfectly aligned to a common spatial reference.

To emulate the impact of noise at the desired level, the noise insertion procedure described in Section 7.1 was applied to this set. The signal variance σ_S^2 for this procedure was estimated using 11,000 projection images generated from 11 objects O1, O2, ..., O11 (from each object, a 1,000 projection images was generated from evenly distributed, randomly selected projection angles).

Note: When referring to all dataset groups within the set that contain projections of the same pairs of objects we put * in the name of the dataset group at the

position where the SNR value is normally specified (e.g., all dataset groups within the set that contain projections of objects O1 and O2 are referred to as O1-O3_*). Similarly, when referring to all dataset groups within the set that contain projections with the same noise level we put * in the name of dataset group at the position where names of the objects are normally specified (e.g., all dataset groups within the set for $SNR = 0.02$ are referred to as *_0.02).

7.5.2 Parameters and Settings

As in the experiments with aligned projection images and in the experiments with misaligned projection images, we produced $L = 240$ 1D projections at 1.5° increments, each with $N = 81$ equally-spaced lines from every image in the dataset. The classification algorithm was executed 10 times for each of the datasets. In all these runs, the length of the *tabu list* (t) was set to 1,200 and the algorithm was terminated after executing $I = 4,000$ iterations.

7.5.3 Results

The results of two sets of experiments (one in which object O1 in used as the reference and one in which O11 is used as the reference) were analyzed separately. These results of are summarized in Table 7.10 and Table 7.11 correspondingly. The plots that correspond to these tables are presented in Figure 7.6 and in Figure 7.7.

As anticipated, our results indicate that the proposed method produces classifications with higher purity when applied to datasets that contain projection images of objects with large structural differences. Similarly, higher classification purity was achieved when the method was applied to datasets with higher SNR. So, an informal analysis of the results summarized in Table 7.10 and Table 7.11 seems to indicate that the difficulty of the classification task (measured by mean classification purity achievable with our method) depends on both, the noise level and the

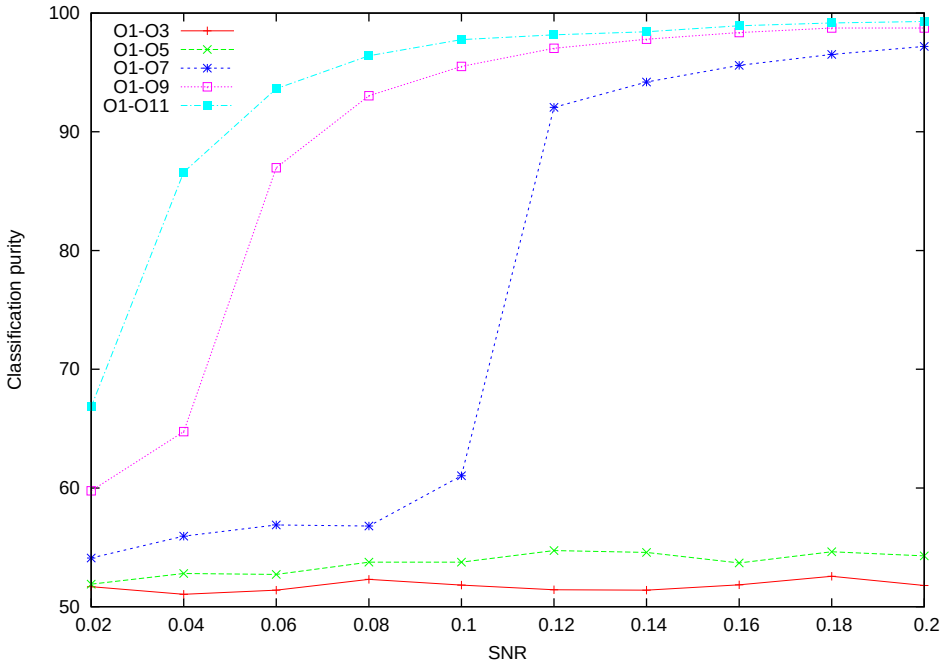


Figure 7.6: Mean classification purity for a dataset group as a function of noise level and difference between conformations (dataset groups O1-O11_0.02, ..., O1-O3_0.20, ..., O1-O11_0.02, ..., O1-O3_0.20).
(Based on five data sets in each of the 50 groups.)

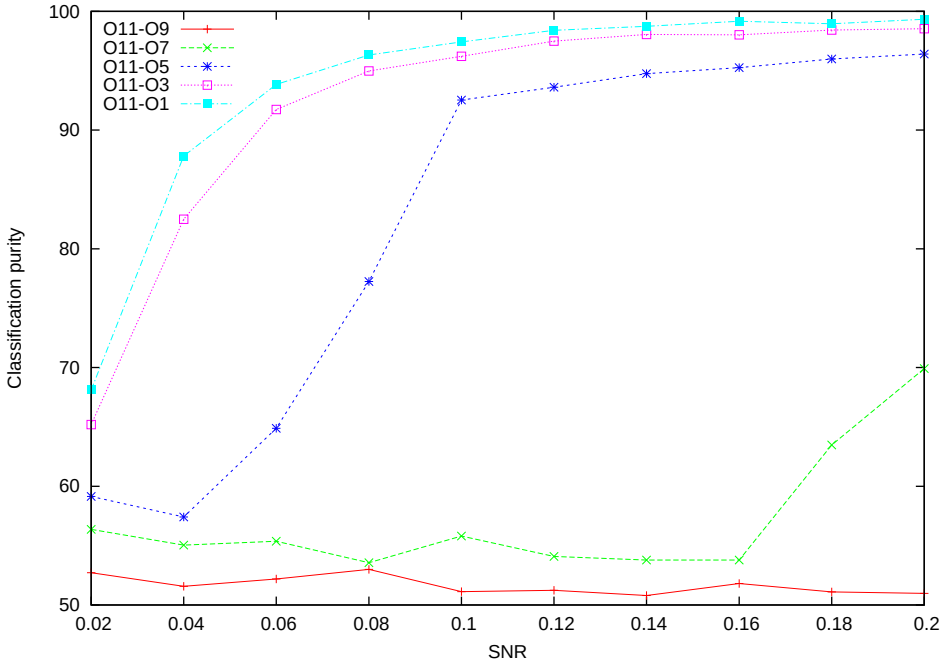


Figure 7.7: Mean classification purity for a dataset group as a function of noise level and difference between conformations (dataset groups O11-O9_0.02, ..., O11-O9_0.20, ..., O11-O1_0.02, ..., O11-O1_0.20).
(Based on five data sets in each of the 50 groups.)

	O1-O3_*	O1-O5_*	O1-O7_*	O1-O9_*	O1-O11_*
*_0.02	51.69±0.49	51.90±0.48	54.10±0.68	59.76±0.52	66.84±0.87
*_0.04	51.05±0.33	52.80±0.49	55.94±0.91	64.75±1.03	86.60±0.50
*_0.06	51.39±0.43	52.72±0.39	56.89±0.94	86.97±0.64	93.63±0.22
*_0.08	52.30±0.54	53.75±0.83	56.79±0.57	93.04±0.34	96.41±0.12
*_0.10	51.82±0.37	53.75±0.65	61.03±0.67	95.51±0.24	97.76±0.10
*_0.12	51.43±0.52	54.73±0.60	92.06±0.67	97.03±0.13	98.17±0.14
*_0.14	51.39±0.32	54.57±0.90	94.19±0.31	97.80±0.18	98.43±0.12
*_0.16	51.84±0.59	53.68±0.72	95.60±0.31	98.36±0.09	98.93±0.12
*_0.18	52.56±0.52	54.63±0.80	96.52±0.45	98.74±0.09	99.17±0.08
*_0.20	51.79±0.45	54.28±1.62	97.19±0.18	98.74±0.11	99.29±0.07

Table 7.10: Mean classification purity for the first set of dataset groups (O1-O3_*, O1-O5_*, O1-O7_*, O1-O9_* and O1-O11_*).
(Based on five data sets in each of the 50 groups.)

	O11-O9_*	O11-O7_*	O11-O5_*	O11-O3_*	O11-O1_*
*_0.02	52.72±0.86	56.37±0.56	59.14±0.51	65.19±0.55	68.11±0.61
*_0.04	51.57±0.38	55.04±0.62	57.41±1.67	82.49±0.80	87.81±0.67
*_0.06	52.19±0.75	55.37±0.16	64.88±4.53	91.73±0.37	93.85±0.28
*_0.08	53.00±0.34	53.56±1.05	77.25±6.85	94.98±0.24	96.33±0.26
*_0.10	51.12±0.32	55.81±0.79	92.53±0.59	96.21±0.09	97.43±0.21
*_0.12	51.24±0.63	54.09±0.74	93.61±0.22	97.50±0.12	98.40±0.08
*_0.14	50.80±0.46	53.78±1.09	94.75±0.30	98.06±0.11	98.74±0.13
*_0.16	51.81±0.42	53.78±1.16	95.26±0.23	98.02±0.14	99.17±0.08
*_0.18	51.10±0.35	63.48±5.28	95.99±0.23	98.43±0.11	98.96±0.04
*_0.20	50.98±0.33	69.91±8.06	96.41±0.24	98.54±0.09	99.34±0.05

Table 7.11: Mean classification purity for the second set of dataset groups (O11-O9_*, O11-O7_*, O11-O5_* and O11-O1_*).
(Based on five data sets in each of the 50 groups.)

magnitude of structural difference between the objects that are represented in the heterogeneous set.

By making some untested assumptions abut the distributions from which our samples are drawn, the dependence of the classification task difficulty on the noise level present in the projection images and the magnitude of structural difference between the objects that are represented in the heterogeneous set, can be put to statistical test using two-way ANOVA [7]. In this analysis the following statistical hypotheses are made:

Factor: Noise

H_{0_Noise} The level of noise has no effect on difficulty of the classification task.

H_{A_Noise} The level of noise has an effect on difficulty of the classification task.

Factor: Structural difference between the objects that are represented in the heterogeneous set

$H_{0\_3D\_diff}$ The difficulty of the classification task is unrelated to the structural difference between the objects that are represented in the heterogeneous set.

$H_{A\_3D\_diff}$ The difficulty of the classification task is related to the structural difference between the objects that are represented in the heterogeneous set.

Interaction of factors: SNR and structural difference between the objects that are represented in the heterogeneous set

$H_{0\_Noise\_\times\_3D\_diff}$ The effect of SNR on difficulty of the classification task does not depend on the structural difference between the objects that are represented in the heterogeneous set.

$H_{A\_Noise\_\times\_3D\_diff}$ The effect of SNR on difficulty of the classification task depends on the structural difference between the objects that are represented in the heterogeneous set.

The results of two-way ANOVA for the experiment summarized in Table 7.10 and Table 7.11 are presented in Table 7.12 and Table 7.13, respectively.

The classification purity scores were subjected to 10×5 independent-groups ANOVA with noise and level of structural difference between the objects as between-experiments factors. When object O1 was used as a reference (see Table 7.12) and

Source of variation	SSq	DF	MSq	F	p
Noise	15707.652	9	1745.295	1104.81	<0.0001
3D_diff	75712.222	4	18928.055	11981.87	<0.0001
Noise × 3D_diff	17015.174	36	472.644	299.19	<0.0001
Within cells	315.945	200	1.580		
Total	108750.993	249			

Table 7.12: Results of two-way ANOVA for experiments that used object O1 as a reference.
Factors:
Noise - the level of noise present in the projection images signal to noise.
3D_diff - the level of structural difference between the objects that are represented in the heterogeneous set.

Source of variation	SSq	DF	MSq	F	p
Noise	11681.628	9	1297.959	74.35	<0.0001
3D_diff	78508.598	4	19627.150	1124.32	<0.0001
Noise × 3D_diff	11110.539	36	308.626	17.68	<0.0001
Within cells	3491.372	200	17.457		
Total	104792.137	249			

Table 7.13: Results of two-way ANOVA for experiments that used object O11 as a reference.
Factors:
Noise - the level of noise present in the projection images signal to noise.
3D_diff - the level of structural difference between the objects that are represented in the heterogeneous set.

when object O11 was used as a reference (see Table 7.13), both main factors and the interaction between them were extremely significant. Based on both sets of experiments (with object O1 and with object O11 used as a reference) the null hypotheses H_{0_Noise}, $H_{0\_3D\_diff}$, $H_{0\_Noise\_\times\_3D\_diff}$ can be rejected with $p < 0.0001$.

In a further analysis we compared the results obtained for appropriately selected pairs of dataset groups, to determine how large must be the change in one of the factors (either the level of noise or the structural difference between the objects that are represented in the set), to observe statistically significant changes in the achieved classification purity. To demonstrate the statistical significance of the differences between the results that we obtained for two dataset groups, we used the Kruskal-Wallis test [7], which is a nonparametric method whose use is valid for our experiments. For each pair of dataset groups selected for testing we made the following hypothesis:

H_0 The results obtained for the selected groups are identical.

H_A The results obtained for the selected dataset groups are not identical.

For all our tests that involved five results in both dataset groups the critical values of Kruskal-Wallis statistic at $\alpha = 0.01$ and $\alpha = 0.05$ are $H_{crit\_0.01} = 6.635$ and $H_{crit\_0.05} = 3.841$ correspondingly. We report the values of the Kruskal-Wallis statistic H for a given pair of dataset groups using *italics* if it is lower $H_{crit\_0.01}$ and ***bold italics*** if it is lower than $H_{crit\_0.05}$. (Upright numbers indicate that H is higher than $H_{crit\_0.01}$, an unlikely occurrence if the null hypotheses were true.)

In the first set of Kruskal-Wallis tests, for each of the noise levels (from 0.02 to 0.20) we tested if the null hypothesis H_0 can be rejected for the pairs of dataset groups that include the dataset group produced from the most similar objects (the reference group) and (one at the time) all other dataset groups at the same noise level. The values of Kruskal-Wallis statistic for these pairs are shown in Table 7.14

	O1-O3_*	O1-O5_*	O1-O7_*	O1-O9_*	O1-O11_*
*_0.02	reference	***0.535***	*4.811*	6.818	6.818
*_0.04	reference	*3.938*	6.818	6.818	6.818
*_0.06	reference	*3.556*	6.818	6.818	6.818
*_0.08	reference	***1.844***	6.818	6.818	6.818
*_0.10	reference	*3.938*	6.818	6.860	6.818
*_0.12	reference	6.818	6.818	6.818	6.818
*_0.14	reference	6.818	6.818	6.818	6.818
*_0.16	reference	***2.470***	6.860	6.860	6.902
*_0.18	reference	*3.962*	6.818	6.860	6.818
*_0.20	reference	***0.884***	6.818	6.860	6.860

Table 7.14: The values of Kruskal-Wallis test statistics H for the first set of dataset groups (dataset groups with the smallest difference between the represented objects are used as the references).
All the values in a row are calculated using the same dataset group as the reference (one marked as "reference" in that row).
Note: The difference between the object represented in the sets increases from left to right.

	O11-O9_*	O11-O7_*	O11-O5_*	O1-O3_*	O11-O1_*
*_0.02	reference	5.806	6.818	6.860	6.818
*_0.04	reference	6.818	6.818	6.860	6.818
*_0.06	reference	6.818	6.818	6.818	6.818
*_0.08	reference	***0.098***	6.818	6.818	6.818
*_0.10	reference	6.818	6.818	6.860	6.860
*_0.12	reference	*4.811*	6.860	6.818	6.818
*_0.14	reference	*3.938*	6.818	6.818	6.818
*_0.16	reference	***1.320***	6.818	6.860	6.818
*_0.18	reference	6.818	6.818	6.818	6.860
*_0.20	reference	6.818	6.860	6.818	6.902

Table 7.15: The values of Kruskal-Wallis test statistics H for the second set of dataset groups (dataset groups with the smallest differences between the represented objects are used as the references).
All the values in a row are calculated using the same dataset group as the reference (one marked as "reference" in that row).
Note: The difference between the object represented in the sets increases from left to right.

and Table 7.15). This values indicate hypothesis H_0 can be rejected with $\alpha < 0.05$ for all the pairs that include groups O1-O7_*, O1-O9_*, O1-O11_* (Table 7.14) and groups O11-O5_*, O1-O3_*, O11-O1_* (Table 7.15). The null hypothesis can be rejected with $\alpha < 0.01$ for all the pairs that include groups O1-O9_*, O1-O11_* (Table 7.14) and groups O1-O3_*, O11-O1_* (Table 7.15). This suggests that at all the noise levels a sufficiently large increase of the structural difference between the objects represented in the heterogeneous projection set has a statistically significant impact on the quality of classifications produced by our method. (A comment on this conclusion and those in the following paragraphs; Because we have used the same test for many cases, the individual rejection levels must be taken with a grain of salt, since some of them may be accidental outliers. However, the overall-nonstatistical-conclusion that there is a general dependence on task difficulty is undeniable.)

In the second set of Kruskal-Wallis tests we used the dataset groups produced from the most dissimilar objects as the references. As in our previous tests, at each noise level (from 0.02 to 0.20) we tested if the null hypothesis H_O can be rejected for the pairs of dataset groups that include the reference group and each of the other groups at the same noise level. The values of Kruskal-Wallis statistic for these pairs are shown in Table 7.16 and Table 7.17). This values indicate hypothesis H_0 can be rejected with $\alpha < 0.05$ for all the tested pairs. Also according to these results the null hypothesis can be rejected with $\alpha < 0.01$ for all the pairs with the exception of the pairs that include some groups in O1-O9_* (Table 7.14) and the exception of the pairs that include some groups in O1-O3_* (Table 7.14). This suggests that at all the noise levels even a small decrease of the structural difference between the objects represented in the heterogeneous projection set has a statistically significant impact on the quality of classifications produced by our method.

In the third set of Kruskal-Wallis tests we used the dataset groups with $SNR =$

	O1-O3_*	O1-O5_*	O1-O7_*	O1-O9_*	O1-O11_*
*_0.02	6.818	6.818	6.818	6.818	reference
*_0.04	6.818	6.818	6.818	6.818	reference
*_0.06	6.818	6.818	6.818	6.818	reference
*_0.08	6.818	6.818	6.818	6.818	reference
*_0.10	6.818	6.818	6.818	6.860	reference
*_0.12	6.818	6.818	6.818	6.818	reference
*_0.14	6.818	6.818	6.818	*4.390*	reference
*_0.16	6.902	6.860	6.860	6.860	reference
*_0.18	6.818	6.860	6.818	*5.914*	reference
*_0.20	6.860	6.860	6.860	*5.842*	reference

Table 7.16: The values of Kruskal-Wallis test statistics H for the first set of dataset groups (dataset groups with the largest differences between the represented objects are used as the references).
All the values in a row are calculated using the same dataset group as the reference (one marked as "reference" in that row).
Note: The difference between the object represented in the sets increases from left to right.

	O11-O9_*	O11-O7_*	O1-O5_*	O11-O3_*	O11-O1_*
*_0.02	6.818	6.860	6.818	*4.840*	reference
*_0.04	6.818	6.818	6.818	6.860	reference
*_0.06	6.818	6.818	6.818	*5.771*	reference
*_0.08	6.818	6.818	6.818	6.322	reference
*_0.10	6.860	6.860	6.860	6.902	reference
*_0.12	6.818	6.818	6.860	6.818	reference
*_0.14	6.818	6.818	6.860	6.818	reference
*_0.16	6.818	6.818	6.818	6.860	reference
*_0.18	6.860	6.860	6.860	6.860	reference
*_0.20	6.902	6.902	6.944	6.902	reference

Table 7.17: The values of Kruskal-Wallis test statistics H for the second set of dataset groups (dataset groups with the largest differences between the represented objects are used as the references).
All the values in a row are calculated using the same dataset group as the reference (one marked as "reference" in that row).
Note: The difference between the object represented in the sets increases from left to right.

	O1-O3_*	O1-O5_*	O1-O7_*	O1-O9_*	O1-O11_*
*_0.02	reference	reference	reference	reference	reference
*_0.04	***0.702***	***1.098***	***1.844***	6.818	6.818
*_0.06	***0.099***	***3.153***	***3.153***	6.818	6.818
*_0.08	***0.535***	***2.455***	*5.771*	6.818	6.818
*_0.10	***0.098***	*3.938*	6.818	6.860	6.818
*_0.12	***0.884***	*5.771*	6.818	6.818	6.818
*_0.14	***0.098***	*4.811*	6.818	6.818	6.818
*_0.16	***0.099***	***3.153***	6.818	6.818	6.860
*_0.18	***1.844***	*4.840*	6.818	6.860	6.818
*_0.20	***0.176***	***0.535***	6.818	6.860	6.860

Table 7.18: The values of Kruskal-Wallis test statistics H for the first set of dataset groups (dataset groups with the smallest SNR are used as the references).
All the values in a column are calculated using the same dataset group as the reference (one marked as "reference" in that column).
Note: The difference between the object represented in the sets increases from left to right.

	O11-O9_*	O11-O7_*	O11-O5_*	O11-O3_*	O11-O1_*
*_0.02	reference	reference	reference	reference	reference
*_0.04	***0.884***	***3.172***	***1.844***	6.902	6.818
*_0.06	***0.395***	***2.470***	***2.455***	6.860	6.818
*_0.08	***0.535***	***2.516***	*4.811*	6.860	6.818
*_0.10	***1.844***	***0.889***	6.818	6.902	6.860
*_0.12	***1.844***	*4.840*	6.860	6.860	6.818
*_0.14	***3.153***	***1.855***	6.818	6.860	6.818
*_0.16	***0.884***	***3.578***	6.818	6.902	6.818
*_0.18	***2.151***	***3.172***	6.818	6.860	6.860
*_0.20	***3.153***	***3.172***	6.860	6.860	6.902

Table 7.19: The values of Kruskal-Wallis test statistics H for the second set of dataset groups (dataset groups with the smallest SNR are used as the references).
All the values in a column are calculated using the same dataset group as the reference (one marked as "reference" in that column).
Note: The difference between the object represented in the sets increases from left to right.

0.02 as the references and tested, for each of level of differences between represented objects, if the null hypothesis H_0 can be rejected for the pairs of dataset groups that include the reference group and (one at the time) all other dataset groups with the same level of differences between the represented objects. The values of Kruskal-Wallis statistic for these pairs are shown in Table 7.18 and Table 7.19. This values indicate that the hypothesis H_0 can be rejected with $\alpha < 0.01$ for all the pairs that include groups in O1-O9_*, O1-O11_* (Table 7.18) and for all the pairs that include groups in O11-O3_*, O11-O1_* (Table 7.19). This suggests that when the structural difference between the objects represented in the heterogeneous projection set is high, there is a statistically significant impact of noise on the quality of classifications produced by our method (the method fails on the sets with low SNR but works on sets with lesser noise). Our test results also show that the hypothesis H_0 cannot be rejected even with $\alpha < 0.05$ for all the pairs that include groups in O1-O3_* (Table 7.18) and with the pairs that include groups in O11-O9_* (Table 7.19) and that the hypothesis H_0 cannot be rejected with $\alpha < 0.01$ for all the pairs that include groups in O1-O3_*, O1-O5_* (Table 7.18) and with the pairs that include groups in O11-O9_*, O1-O7_* (Table 7.19). This means that when the structural difference between the objects represented in the heterogeneous projection set is small there is no statistically significant impact of noise on the quality of classifications produced by our method (the method fails on all such sets regardless of the level of noise).

In the last set of Kruskal-Wallis tests we used the dataset groups with $SNR = 0.20$ as the references and tested, for each of level of differences between represented objects, if the null hypothesis H_0 can be rejected for the pairs of dataset groups that include the reference group and (one at the time) all other dataset groups with the same level of differences between the represented objects. The values of Kruskal-Wallis statistic for these pairs are shown in Table 7.20 and Table 7.21. This

	O1-O3_*	O1-O5_*	O1-O7_*	O1-O9_*	O1-O11_*
*_0.02	***0.176***	***0.535***	6.818	6.860	6.860
*_0.04	***1.098***	***0.535***	6.818	6.860	6.860
*_0.06	***0.395***	***0.273***	6.818	6.860	6.860
*_0.08	***0.273***	***0.273***	6.818	6.860	6.860
*_0.10	***0.011***	***0.011***	6.818	6.902	6.860
*_0.12	***0.535***	***0.011***	6.818	6.860	6.860
*_0.14	***0.176***	***0.011***	6.818	6.860	6.860
*_0.16	***0.011***	***0.011***	6.818	*4.417*	*4.444*
*_0.18	***1.320***	***0.011***	***1.844***	***0.099***	***1.370***
*_0.20	reference	reference	reference	reference	reference

Table 7.20: The values of Kruskal-Wallis test statistics H for the first set of dataset groups (dataset groups with the largest SNR are used as the references).
All the values in a column are calculated using the same dataset group as the reference (one marked as "reference" in that column).
Note: The difference between the object represented in the sets increases from left to right.

	O11-O9_*	O11-O7_*	O11-O5_*	O11-O3_*	O11-O1_*
*_0.02	***3.153***	***3.172***	6.860	6.860	6.902
*_0.04	***1.320***	*3.938*	6.860	6.860	6.902
*_0.06	***1.320***	***3.153***	6.860	6.818	6.902
*_0.08	*5.771*	*5.771*	6.860	6.818	6.902
*_0.10	***0.176***	***2.455***	6.860	6.860	6.944
*_0.12	***0.011***	*4.811*	6.902	6.818	6.902
*_0.14	***0.395***	*4.811*	6.860	6.322	6.902
*_0.16	***2.151***	*4.811*	*5.806*	*5.806*	***1.630***
*_0.18	***0.044***	***0.535***	***1.353***	***0.884***	6.944
*_0.20	reference	reference	reference	reference	reference

Table 7.21: The values of Kruskal-Wallis test statistics H for the second set of dataset groups (dataset groups with the largest SNR are used as the references).
All the values in a column are calculated using the same dataset group as the reference (one marked as "reference" in that column).
Note: The difference between the object represented in the sets increases from left to right.

values indicate that the hypothesis H_0 cannot be rejected even with $\alpha < 0.05$ for all the pairs that include groups in O1-O3_*, O1-O5_* (Table 7.18) and cannot be rejected with $\alpha < 0.01$ for all the pairs that include groups in O11-O9_*, O11-O9_* (Table 7.19). This confirms the results from the previous set of test. When the structural difference between the objects represented in the heterogeneous projection set is small there is no statistically significant impact of noise on the quality of classifications produced by our method (the method fails on all such sets regardless of the level of noise). The results in Table 7.18 and Table 7.19 also show that the hypothesis H_0 cannot be rejected even with $\alpha < 0.05$ for pairs that include groups in O1-O7_0.18, O1-O9_0.18, O1-O9_0.18 (Table 7.18) and for the pairs that include groups in O11-O5_0.18, O11-O3_0.18, O11-O3_0.16 (Table 7.19). This suggests that when the structural difference between the objects represented in the heterogeneous projection set is large and the noise level is low, the small increase of noise level does not have a statistically significant impact on the quality of classifications produced by our method (there is no statistically significant decrease in the quality of classifications produced by our method, for datasets with large difference between the represented object, when low level of noise is slightly increased).

7.5.4 Discussion

The results obtained with both statistical methods (two-way ANOVA and Kruskal-Wallis test) indicate that both factors, the level of noise and the degree of structural differences between the objects represented in the heterogeneous set, have a significant impact on the quality of classification produced by our method. They also provide a strong evidence of the interaction between the two factors.

Our statistical tests confirmed that when the structural difference between the object represented in the projection set is sufficiently large, our method is able to produce an acceptable classifications for a wide range of noise levels. They also

show that when this difference is small, acceptable classification cannot be produced even for dataset with relatively low level of noise. This has the following practical implications. Small differences between conformations (that may be important from the biological perspective), are not detectable using our classification method even when it is applied to datasets with high SNR, large however conformation differences can be successfully detected with our method for dataset with very significant level of noise.

7.6 Summary

The main goal of the conducted evaluation experiments was to verify the soundness of the proposed method and to demonstrate its potential to classify heterogeneous projection sets that are encountered in 3D-EM. The results of our experiments show that partitioning (with high classification purity) of heterogeneous projection sets, similar to those encountered in 3D-EM, into homogeneous subsets is possible, despite of severe noise present in them. They also demonstrate that the proposed classification method can be successfully used in various scenarios involving heterogeneous projection sets containing images of molecules in a few conformations, even when the the number of images associated with each conformation strongly varies from conformation to conformation.

Despite the level of noise present in the projection images (which significantly reduces the similarity between the 1D projections onto the common line), our method was able to successfully separate homogeneous subsets by the simultaneous use of the common lines between many images. Based on the histogram of edge weights (Figure 3.1), it is clear that an attempt to determine whether one particular pair of images belongs to the same object using the common line approach would fail. Our method was able to produce promissing results by utilizing simultaneously

all images in the set.

In order to allow for a large number of experiments and to focus on the classification step of the heterogeneous reconstruction procedure we made some simplifying assumptions. The impact of these assumptions should be considered when the results are analyzed. The use of the synthetically generated datasets in the evaluation process provided us with high level of control over the experimental conditions. Since, in case of such sets, the conformation from which a projection image was produced is known, it was possible to measure the quality of classifications produced by our method without reconstructing the 3D models. The parameters of the datasets used in the evaluation experiments have been carefully selected to capture the most important characteristics of datasets encountered in 3D-EM. However, it must be recognized that the datasets we used resembled real projection sets only to some extent. Consequently, not all the issues associated with the classification of EM projection images that may have impact on the outcome of heterogeneous reconstruction procedure were accurately reflected in the experiments.

The use of a figure of merit (classification purity) that measures the quality of classification instead of one that measures the quality of the 3D reconstructions allowed us to obtain unambiguous results, which are not influenced by differences between various reconstruction procedures. However, since classification purity does not directly translate into the quality of reconstruction results (see Section 7.4.4), some caution should be taken when interpreting these results. As our case study in Section 7.4 have shown, when the complete 3D reconstruction procedure is considered, less than perfect results of the classification procedure may lead to acceptable results.

Our experiments show that despite the computational complexity of the Max k-Cut problem, it is possible to find acceptable (from the 3D-EM perspective) approximate solutions of Max k-Cut within minutes of computation time for large

graphs that represent heterogeneous projection sets similar to those encountered in 3D-EM. (Our algorithm required approximately two minutes to find the approximate Max k-Cuts for 5,000 node graph). As demonstrated in Section 3.4, when a general-purpose algorithm is used, there is a huge computational cost associated with finding the maximum capacity cuts for graphs of this size. Since many datasets encountered in 3D-EM contain tens of thousands projection images, this ability to handle large datasets is very important. To overcome the computational intractability of the NP-complete max k-Cut problem, we relied on the fact that the graphs that we need to cut are not arbitrary but are derived from a physical process (EM). Such graphs have some special properties (for example, as it can be seen in Fig. 3.1, the edge weights in these graphs fall into a relatively small range and within that range they tend to center close to some fixed value). Due to such properties, the proposed algorithm is much faster for the application at hand than the general-purpose algorithms. However, the performance of our algorithm when applied to an arbitrary graph is likely to be poor.

Since the combinatorial optimization required can be performed very fast, the initial graph construction task is the only time-consuming step in the proposed method. When the early-termination based method was used (see Section 4.4) in the graph building process, a single experiment required approximately 12 hours of computation per data set on an Intel(R) Xeon(TM) CPU running at 1.70 GHz and approximately 8 hours of computation per data set on an AMD Athlon(TM) 64 Processor 3200+ running the code with SSE optimizations. The ten runs of our graph partitioning algorithm took only approximately twenty minutes of this time.

The fact that the graph construction takes the majority of time required to classify heterogeneous projections and the graph partitioning can be done very fast has some useful consequences. Once the graph is constructed, many runs of the partitioning algorithm can be conducted at a very low cost. Since the solutions found by

the graph partitioning algorithm are only approximations (which depend on the initial random classification of nodes), the multiple runs starting from different initial configurations significantly increase the likelihood of finding a good approximation to the global minimum of the objective function (see Section 5.5). Another reason for running the graph partitioning algorithm several times on the same graph is to determine the optimal number of classes into which to partition given the projection set. The graph partitioning algorithm requires that the number of classes is provided as a parameter (this does not imply that the number of objects represented in heterogeneous projection set is known). Several runs of the algorithm with different number of classes as a parameter, followed by 3D reconstruction and evaluation of obtained models, might be necessary to optimize the results (application of this technique to datasets with uneven representation of conformations have been described in Section 7.2.4).

The results of our experiments demonstrate that the proposed method can be successfully used in various scenarios involving datasets with characteristics similar to those found in datasets produced by an electron microscope. Consequently, they provide some level of confidence that the proposed method (perhaps with modifications and enhancements) can successfully classify actual electron microscopic projection images. However, more testing is advisable to demonstrate the applicability of our method to 3D-EM. The experiments conducted so far constitute an initial step of a comprehensive evaluation process in which the method is tested on increasingly realistic datasets. Since these experiment have been preformed on datasets affected only by idealized (Gaussian) noise and free of other distortions, they should be followed by additional tests (on more realistic datasets), which capture the impact of various distortions, affecting electron microscopic projection images, on both classification purity and the quality of 3D models.

Chapter 8

Conclusions

The main objectives of our research were to construct a classification method to separate homogeneous subsets of noisy heterogeneous projection sets and to demonstrate its ability to become a component of a reconstruction method that is capable of producing high-quality 3D models from heterogeneous projection sets that are similar to those encountered in 3D-EM. These goals have been achieved. We designed a novel classification procedure that utilizes properties of 2D projection images obtained from 3D objects and employs a combinatorial optimization technique to separate homogeneous subsets. We also proposed a framework for reconstructing 3D models from heterogeneous projection sets in which our classification method can be used. We demonstrated the capabilities of our method by conducting evaluation experiments on appropriately constructed synthetic datasets. However, some additional research is needed to conclusively demonstrate that the proposed method can be efficaciously used on real 3D-EM data.

8.1 Contributions

We have constructed an unsupervised classification method to separate homogeneous subsets of very noisy heterogeneous projection sets and experimentally demon-

strated that it has the potential to become a superior alternative to existing methods of addressing the problem of heterogeneity. We have shown that, incorporated into a heterogeneous reconstruction procedure, our method (perhaps with some enhancements proposed in the next section) will produce representative 3D models of various conformations represented by heterogeneous projection sets obtained with an electron microscope. Depending on their quality, these models can be used either directly or indirectly (after refining with a multireference method) in the analysis of various biological structures. We have conducted extensive research to develop, optimize and test the key components of our classification method.

We have proposed a new image dissimilarity measure, specifically designed to deal with 2D projections of 3D objects. This measure utilizes a property of images that are 2D projections of the same 3D object and, therefore, it is well suited to our classification problem. The proposed measure provides the foundation for our combinatorial classification method. However, its use should not be limited to this particular application.

After exploring a number of options, we have devised a procedure that builds the graphs that represent an instance of the projection image classification problem at a significantly reduced the computational cost. Since the cost of building such a graph increases proportionally to the square of its size (measured by the number of nodes), this reduction is important when large datasets are processed with our classification method.

We have constructed an algorithm that is able to find good (from the classification perspective) approximate Max k-Cuts for graphs that represent instances of heterogeneous projection sets similar to those that originate from 3D-EM. Despite of the computational complexity of Max k-Cut problem and the large size of these sets (containing thousands of nodes) our algorithm required only several minutes of run-time on a standard personal computer to produce such approximations.

We proposed and implemented a framework that allowed us to demonstrate the use of our classification method in the process of reconstructing 3D models from heterogeneous projection sets. This framework is an essential tool in future developments of classification based heterogeneous reconstruction procedures. Since the individual components of this framework can be modified and even replaced without changing others, it allows us to improve the quality of the results achievable with classification based methods and to extend the range of problems to which such methods can be applied by focusing research on specific steps of the reconstruction process.

We demonstrated that it is possible to devise an efficient implementation of our method that does not require extraordinary computing resources to handle the type of problems encountered in practice. This confirms that, despite of the large size and complexity of these problems, the use of the classification based approach in our target domain (3D-EM) is feasible.

8.2 Future Works

In order to extend the range of applicability of our classification based method and to improve its ability of to deal with some of the problems identified by our research, various extensions and enhancements to both the classification method and the other components of the proposed reconstruction framework should be considered.

For practical reasons, the vector dissimilarity measure used by our classification method has been selected experimentally without a thorough analysis of the statistical characteristics of noise and other distortions present in the EM projection images. Therefore, a careful study of these distortions (which was beyond the scope of our research) will likely lead to a vector dissimilarity measure better suited to such images. The use of different (appropriately constructed) vector dissimilar-

ity measure may also improve handling of the misaligned projection images by our classification method. Future research should explore if a shift invariant vector dissimilarity measure (e.g., one that compares the amplitudes of corresponding spatial frequencies in the Fourier transforms of the vectors being compared) can be used for this purpose.

We have demonstrated that, by finding an approximate solution to a particular optimization problem, one can produce acceptable classifications of the projection images. However, there is no evidence that the objective function we have used is best possible from the perspective of the original classification problem. Future research may show that a different objective function is more appropriate. The use of such a function may lead to the improvement of the classification results. It also may remove or suppress the bias towards even splits exhibited by our current method (some research to explore this possibility have been already conducted).

Our classification algorithm requires that the number of classes be specified as a parameter. Depending on how even is the representation of various conformations in the projection set this number should be either equal to (when conformations are evenly represented) or larger than (when the representation is uneven) than number of conformations. In practice, the representation ratio between conformations and even the number of them is unknown. Therefore, selection of the appropriate number of classes in not trivial. We have proposed an experimental approach to this problem in which results of classifications with different values are used to produce 3D models. Based on the evaluation of these models, the best value of parameter can be selected. Another way of handling this problem is to conduct the classification with a number of classes that is significantly larger than the number of conformations and merge those classes that correspond to the same conformation. Additional research is needed to explore the construction of an automated procedure that performs this task.

Our research has shown that, despite high level of noise, it is possible to separate homogeneous subsets of a heterogeneous set by finding a configuration that minimizes the differences between appropriately selected pairs 1D projections of images that are assigned to the same class. The best matching lines for each pair of images treated as a common line in two 2D projections. Our method does not check if such lines obtained from different pairs of sinograms form a geometrically consistent a set. The discovery of a method that ensures such consistency should be undertaken in the future. Such a method can be potentially better suited to handle extremely noisy datasets. It would also be able to determine relative projection angles of all images in the dataset, which can be used in the associated reconstruction procedures.

We have made a significant effort to ensure that our evaluation methodology is objective and well replicates conditions encountered in practice. It must be recognized that we have achieved this goal only to the limited extent. The design of a comprehensive evaluation process was beyond the scope of our research. It would require extensive study of issues related to the physics of image creation with EM and therefore it could not be completed within our time limits. In future, an effort should be made to define a standardized, comprehensive procedure that will allow us to evaluate and compare existing and emerging methods for dealing with the heterogeneity problem. To ensure its thoroughness and objectivity, the evaluation process must incorporate a figure of merit that measures the quality of the 3D models produced by the method under test from many benchmark datasets.

The benchmark datasets should well represent the range of problems that are encountered in practice. The evaluation tests should be preformed on both synthetic data sets and sets produced by an electron microscope. The synthetic data sets should be as realistic as possible and should well reflect the complex relationship between various distortions (noise, CTF, misalignment, etc.) encountered in EM

projection images. The use of such sets allows us to conduct tests, with high level of control over experimental conditions. The inclusion of tests with actual EM projection images in the evaluation process will ensure that the issues not captured by the synthetic sets are represented in the evaluation process.

Appendix A

Objects Used in Experiments with Aligned and Misaligned Projection Images

The objects S6 and S6x are two configurations of six identical spheres. The object S7 is constructed from seven identical spheres. The coordinates of sphere centers (measured using the size of the pixel edge in the projection image as a unit and assuming the origin to be at the geometrical center of the configuration) are:

S6: A (0, 20, 0), B (17.32, 10, 0), C (17.32, -10, 0), D (0, -20, 0), E (-17.32, -10, 0), F (-17.32, 10, 0).

S6x: A (-2.24, 19.87, 2), B (18.33, 8, -2), C (18.33, -8, 2), D (-2.24, -19.87, -2), E (-16.09, -11.87, 2), F (-16.09, 11.87, -2).

S7: A (0, 20, 0), B (15.64, 12.47, 0), C (19.50, -4.45, 0), D (8.68, -18.02, 0), E (-8.68, -18.02, 0), F (-19.50, -4.45, 0), G (-15.64, 12.47, 0).

The radius of spheres in objects S6 and S6x (in the same unit) is 8. The radius of spheres in object S7 (measured in the same unit) is 7.6. The density of all the spheres is 1.0.

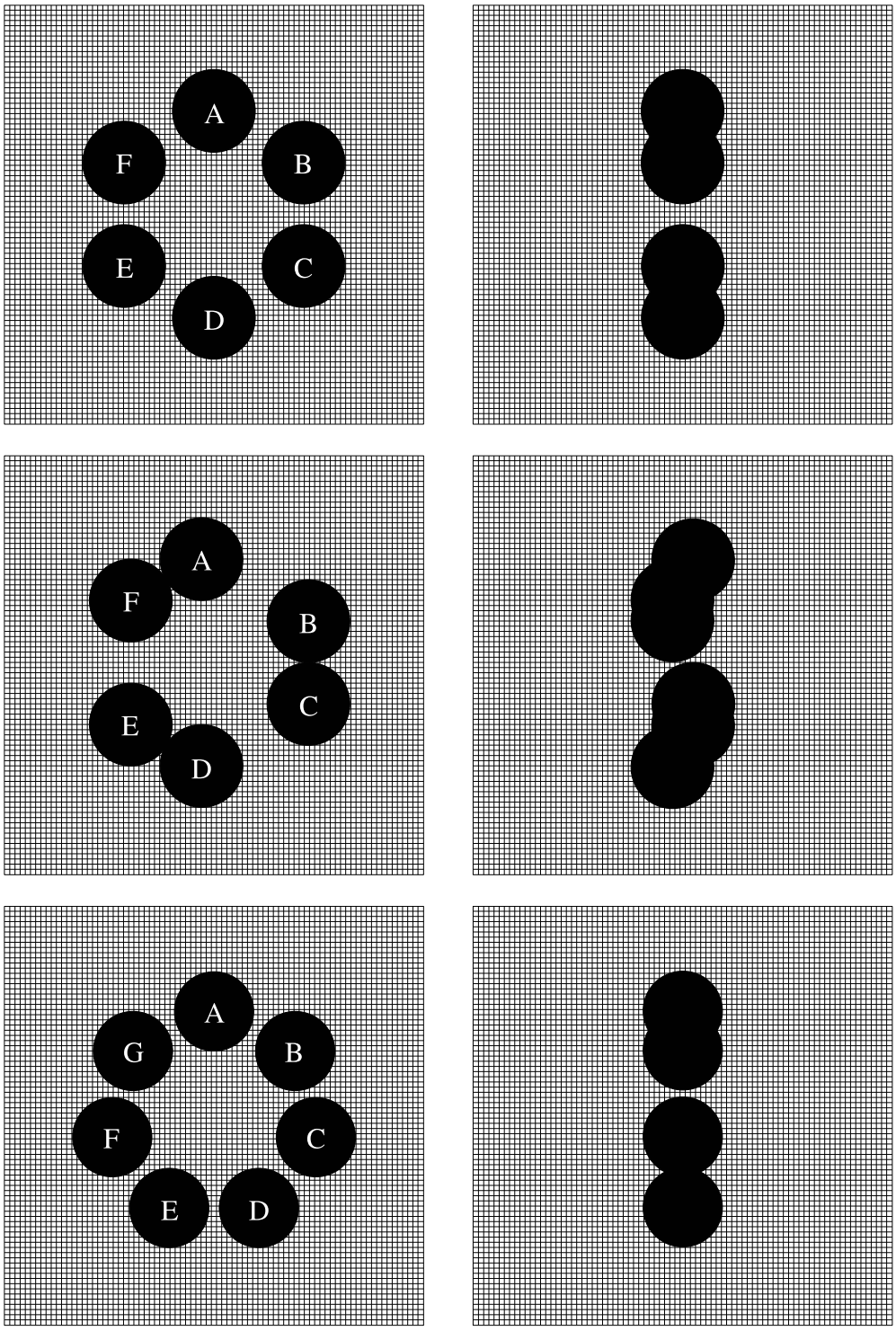


Figure A.1: Geometries of the objects S6, S6x, S7 and pixel grid of the projection images.
Top (left column) and side (right column) views of S6 (top row), S6x (center row), S7 (bottom row).

Appendix B

Example of the Results from Experiment with Aligned Projection Images I

Note: All 3D models presented in this appendix were reconstructed using the implementation of ART [28] available in Xmipp [42]. These model were produced under the (unrealistic) assumption that the projection angles for all the images are perfectly assigned. The quality of the models reconstructed form these images with more realistic angular assignment may be significantly lower. All images of the 3D volumes were rendered with UCSF Chimera [37].

Dataset

Dataset group: S6x-S7_50:50

Objects represented in the set: S6x, S7

Number of projections of object S6x: 2500

Number of projections of object S7: 2500

SNR: 0.1

Alignment: perfect

Reconstructions

No.	Description	Figure
1.	Reconstruction from all images in heterogeneous set	B.1
2.	Reconstruction from perfectly classified projection images Number of classes: 2 Classification Purity: 100%	B.2
3.	Reconstruction from the images classified by our method Number of classes: 2 Classification Purity: 99.32%	B.3

Note: The differences between reconstructions 2. and 3. are shown in Figure B.4.

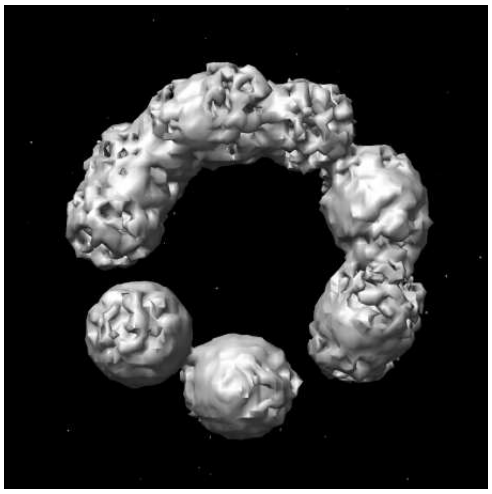
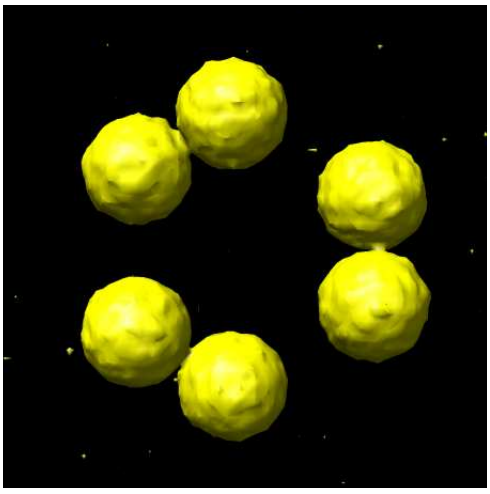
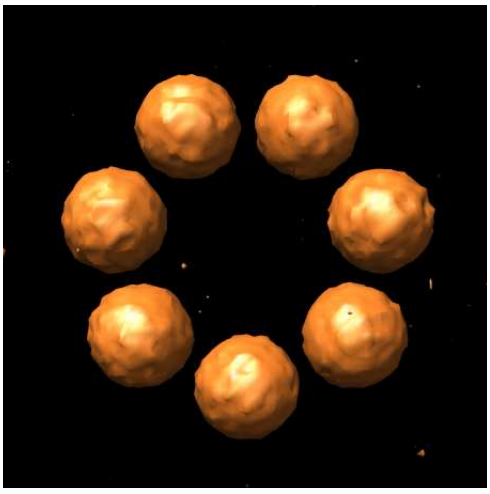


Figure B.1: 3D model obtained by reconstructing from heterogeneous projection set that contains aligned projection images of objects S6x and S7. (Representation ratio 50:50.)

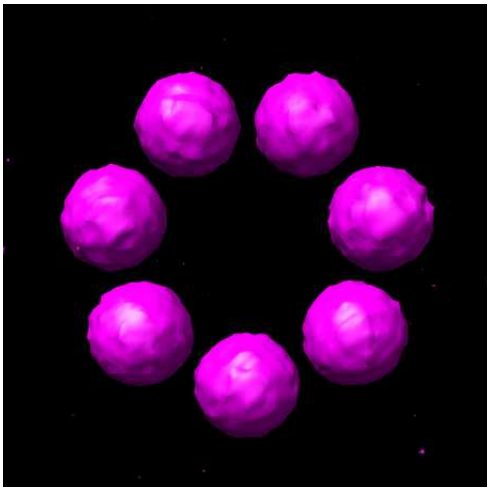


(a) Reconstruction from projections of object S6x.

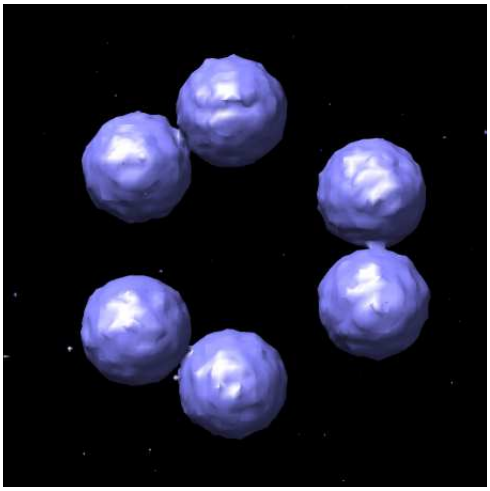


(b) Reconstruction from projections of object S7.

Figure B.2: 3D models obtained by reconstructing from perfectly classified aligned projection images of objects S6x and S7. (Representation ratio 50:50.)

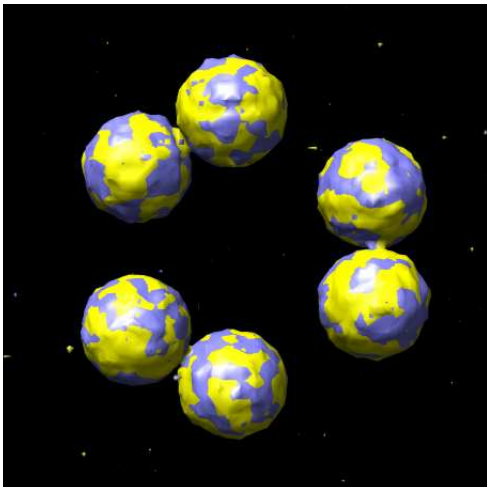


(a) Reconstruction from projections in Class 1.

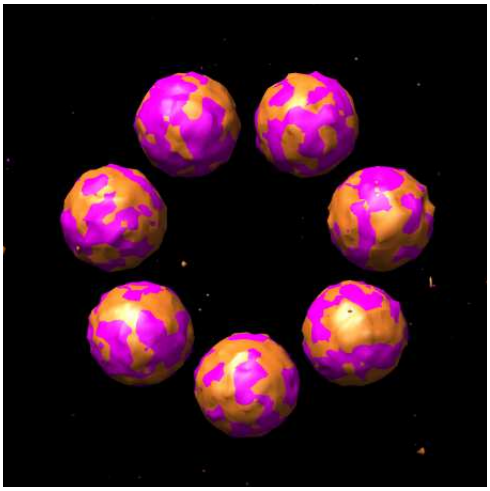


(b) Reconstruction from projections in Class 2.

Figure B.3: 3D models obtained by reconstructing from the aligned projection images of objects S6x, S7 classified by the proposed method. (Representation ratio 50:50.)



(a) Reconstruction from projections of object S6x and reconstruction from projections in Class 2.



(b) Reconstruction from projections of object S7 and Reconstruction from projections in Class 1.

Figure B.4: Differences between 3D models obtained by reconstructing from perfectly classified aligned projection images of objects S6x, S7 and corresponding 3D models obtained by reconstructing from these images classified by the proposed method. (Representation ratio 50:50.)
The yellow and orange models are 3D reconstructions from perfectly classified projection images. The violet and blue models are 3D reconstructions obtained from images classified by the proposed method.

Appendix C

Example of the Results from Experiment with Aligned Projection Images II

Note: All 3D models presented in this appendix were reconstructed using the implementation of ART [28] available in Xmipp [42]. These model were produced under the (unrealistic) assumption that the projection angles for all the images are perfectly assigned. The quality of the models reconstructed form these images with more realistic angular assignment may be significantly lower. All images of the 3D volumes were rendered with UCSF Chimera [37].

Dataset

Dataset group: S6x-S7_35:65

Objects represented in the set: S6x, S7

Number of projections of object S6x: 1750

Number of projections of object S7: 3250

SNR: 0.1

Alignment: perfect

Reconstructions

No.	Description	Figure
1.	Reconstruction from all images in heterogeneous set	C.1
2.	Reconstruction from perfectly classified projection images. Number of classes: 2 Classification Purity: 100%	C.2
3.	Reconstruction from the images classified by our method. Number of classes: 2 Classification Purity: 86.18%	C.3
4.	Reconstruction from the images classified by our method. Number of classes: 3 Classification Purity: 97.72%	C.4
5.	Reconstruction from the images classified by our method. Number of classes: 3 (2 classes corresponding to the same object were merged) Classification Purity: 97.72%	C.5

Note: The differences between reconstructions 2. and 5. are shown in Figure C.6.

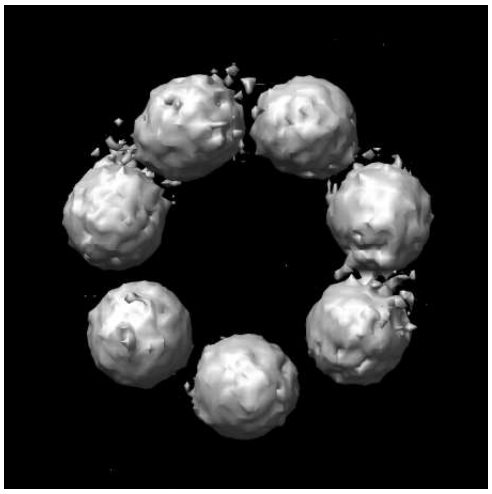
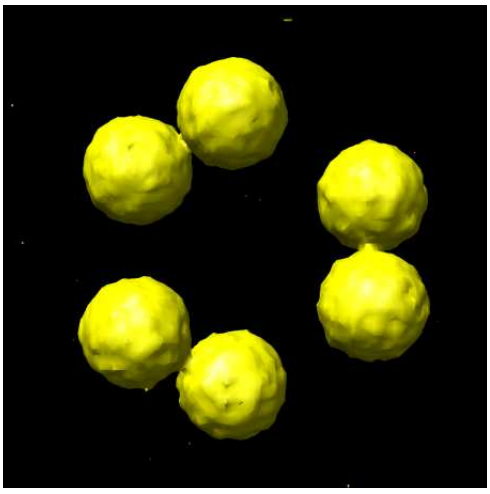
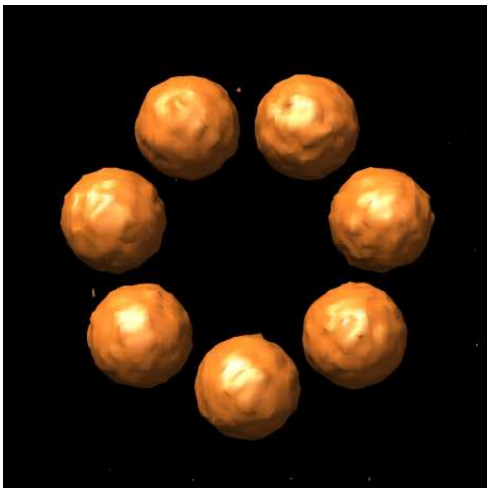


Figure C.1: 3D model obtained by reconstructing from heterogeneous projection set that contains aligned projection images of objects S6x and S7. (Representation ratio 35:65.)

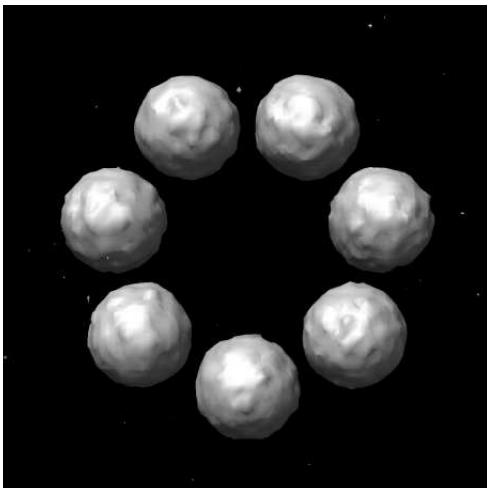


(a) Reconstruction from projections of object S6x.



(b) Reconstruction from projections of object S7.

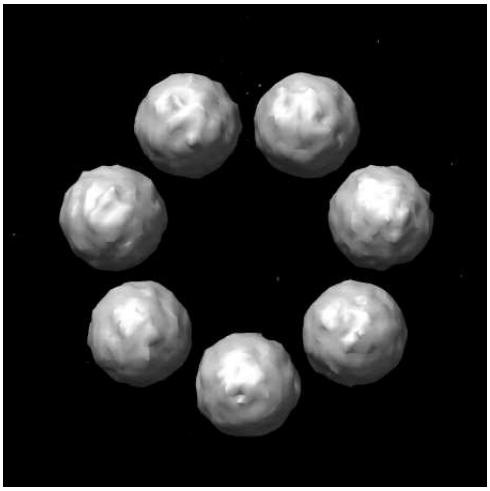
Figure C.2: 3D models obtained by reconstructing from perfectly classified aligned projection images of objects S6x and S7. (Representation ratio 35:65.)



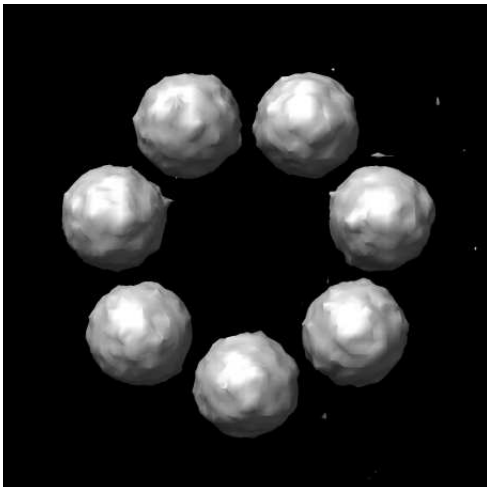
(a) Reconstruction from projections in Class 1.

(b) Reconstruction from projections in Class 2.

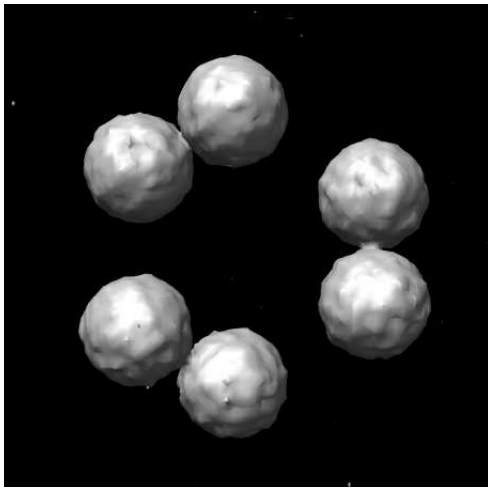
Figure C.3: 3D models obtained by reconstructing from the aligned projection images of objects S6x, S7 classified by the proposed method into two classes. (Representation ratio 35:65.)



(a) Reconstruction from projections in Class 1.

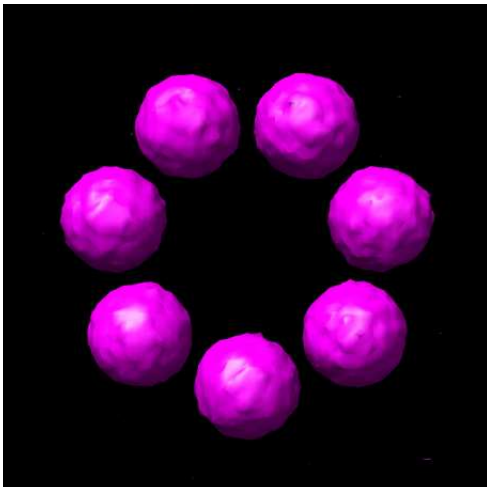


(b) Reconstruction from projections in Class 2.

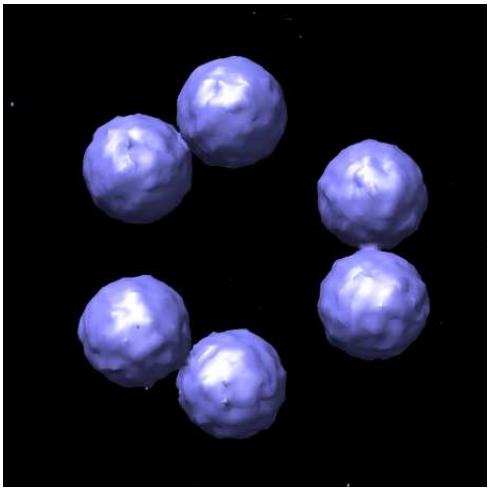


(c) Reconstruction from projections in Class 3.

Figure C.4: 3D models obtained by reconstructing from the aligned projection images of objects S6x, S7 classified by the proposed method into three classes. (Representation ratio 35:65.)

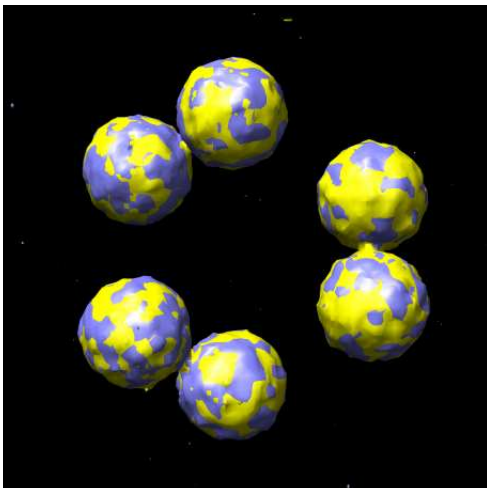


(a) Reconstruction from projections in Class 1 and Class 2.

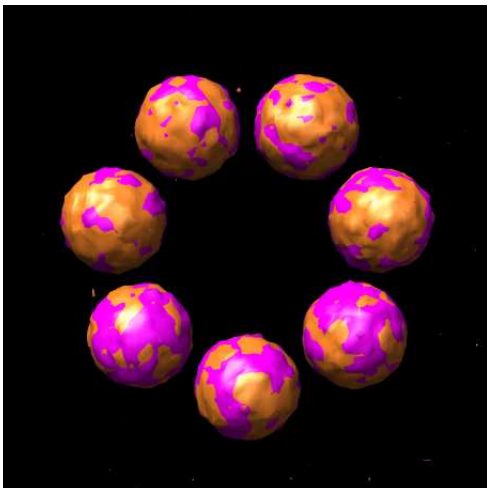


(b) Reconstruction from projections in Class 3.

Figure C.5: 3D models obtained by reconstructing from the aligned projection images of objects S6x, S7 classified by the proposed method into three classes. The classes corresponding to the same object were merged. (Representation ratio 35:65.)



(a) Reconstruction from projections of object S6 and reconstruction from projections in Class 3.



(b) Reconstruction from projections of object S7 and reconstruction from projections in Class 1 and Class 2.

Figure C.6: Differences between 3D models obtained by reconstructing from perfectly classified aligned projection images of objects S6x, S7 and corresponding 3D models obtained by reconstructing from these images classified by the proposed method. (Representation ratio 35:65.)
The yellow and the orange models are 3D reconstructions from perfectly classified projection images. The violet and the blue models are 3D reconstructions obtained from images classified by the proposed method.

Appendix D

Example of the Results from Experiment with Aligned Projection Images III

Note: All 3D models presented in this appendix were reconstructed using the implementation of ART [28] available in Xmipp [42]. These model were produced under the (unrealistic) assumption that the projection angles for all the images are perfectly assigned. The quality of the models reconstructed form these images with more realistic angular assignment may be significantly lower. All images of the 3D volumes were rendered with UCSF Chimera [37].

Dataset

Dataset group: S6x-S7_20:80

Objects represented in the set: S6x, S7

Number of projections of object S6x: 1000

Number of projections of object S7: 4000

SNR: 0.1

Alignment: perfect

Reconstructions

No.	Description	Figure
1.	Reconstruction from all images in heterogeneous set.	D.1
2.	Reconstruction from perfectly classified projection images. Number of classes: 2 Classification Purity: 100%	D.2
3.	Reconstruction from the images classified by our method. Number of classes: 2 Classification Purity: 80%	D.3
4.	Reconstruction from the images classified by our method. Number of classes: 5 Classification Purity: 98.98%	D.5
5.	Reconstruction from the images classified by our method. Number of classes: 5 (4 classes corresponding to the same object were merged) Classification Purity: 98.98%	D.6

Note: The differences between reconstructions 2. and 5. are shown in Figure D.7.

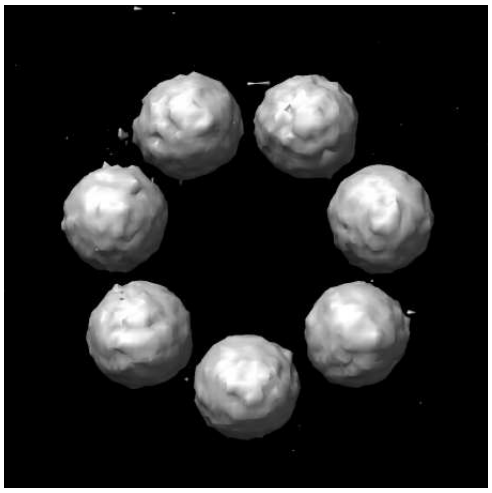
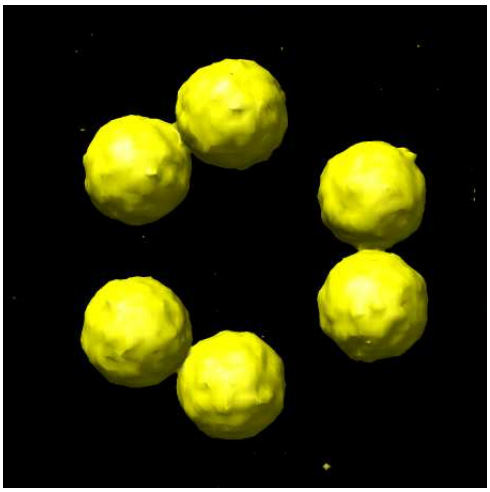
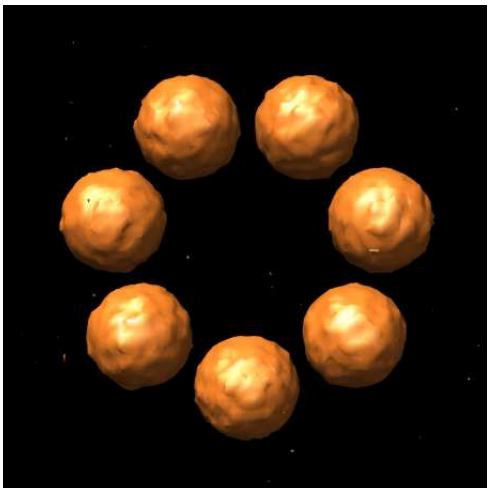


Figure D.1: 3D model obtained by reconstructing from heterogeneous projection set that contains aligned projection images of objects S6x and S7. (Representation ratio 20:80.)

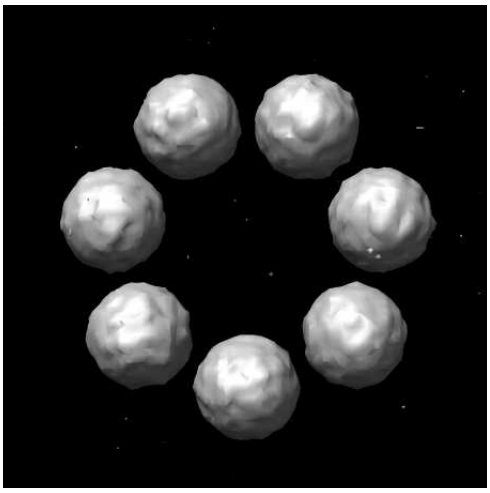


(a) Reconstruction from projections of object S6x.

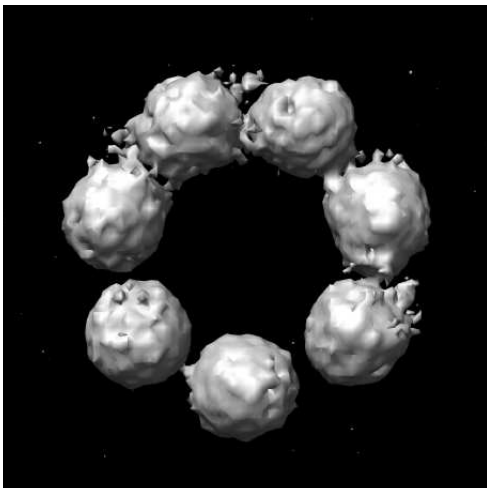


(b) Reconstruction from projections of object S7.

Figure D.2: 3D models obtained by reconstructing from the aligned projection images of objects S6x, S7 classified by the proposed method into two classes. (Representation ratio 20:80.)

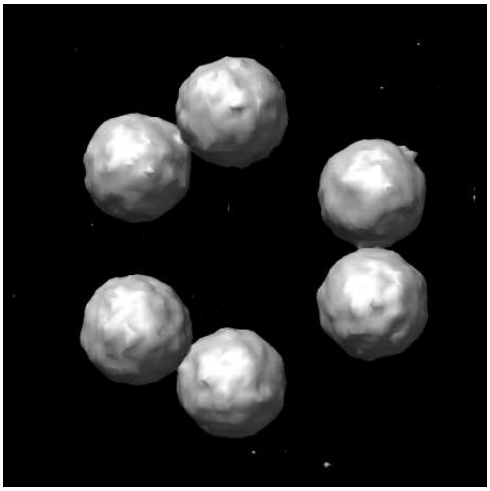


(a) Reconstruction from projections in Class 1.

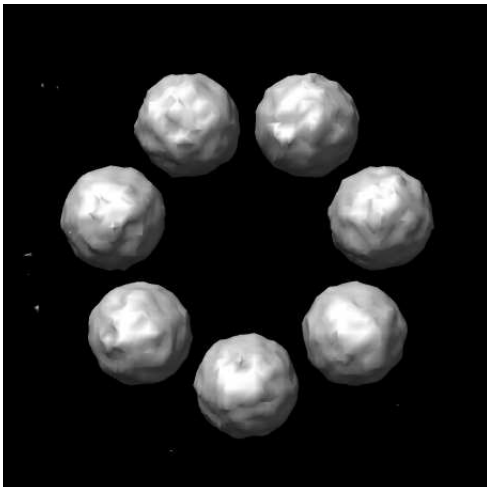


(b) Reconstruction from projections in Class 2.

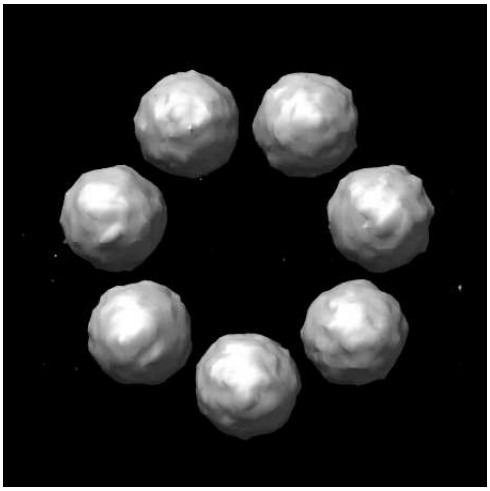
Figure D.3:
3D models obtained by reconstructing from the aligned projection images of objects S6x, S7 classified by the proposed method into two classes. (Representation ratio 20:20.)



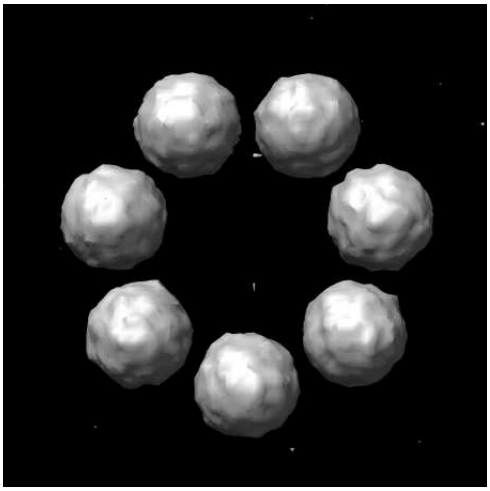
(a) Reconstruction from projections in Class 1.



(b) Reconstruction from projections in Class 2

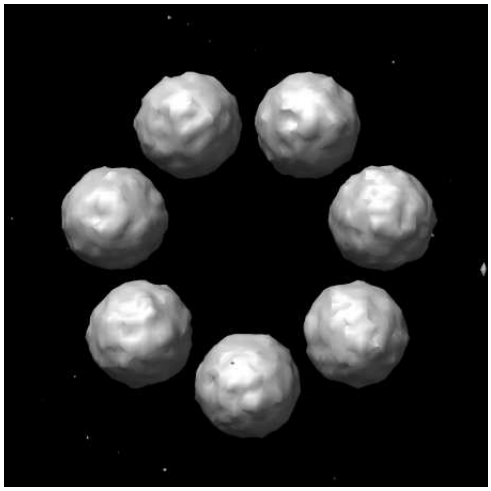


(c) Reconstruction from projections in Class 3.



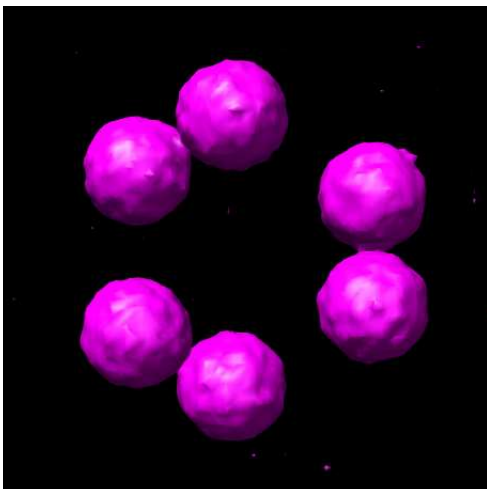
(d) Reconstruction from projections in Class 4.

Figure D.4: 3D models obtained by reconstructing from the aligned projection images of objects S6x, S7 classified by the proposed method into five classes (the fifth model is shown on the next page). (Representation ratio 20:80.)

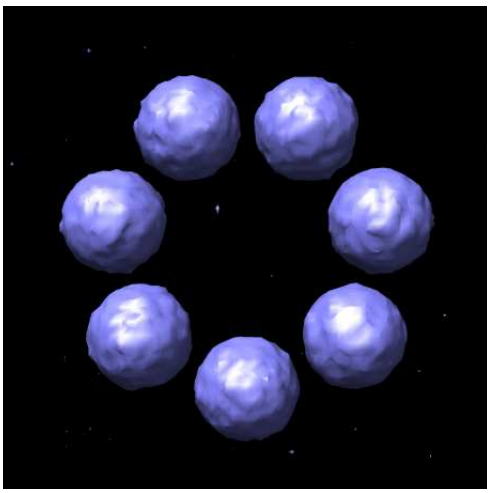


(a) Reconstruction from projections in Class 5.

Figure D.5: 3D models obtained by reconstructing from the aligned projection images of objects S6x, S7 classified by the proposed method into five classes (continuation from the previous page). (Representation ratio 20:80.)

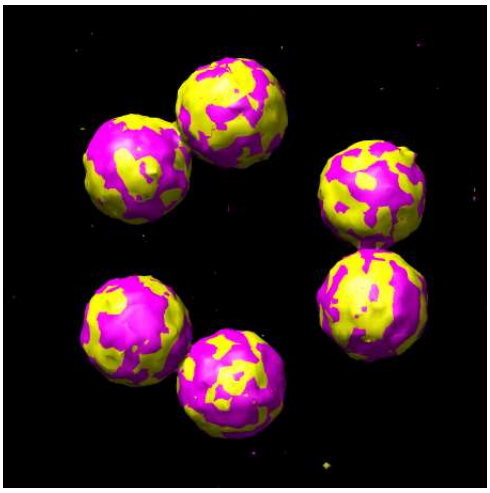


(a) Reconstruction from projections in Class 1.

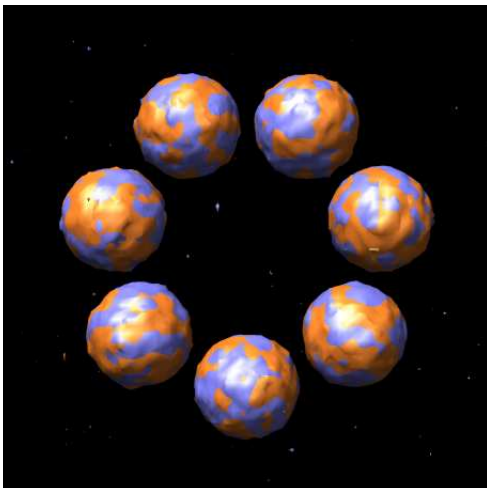


(b) Reconstruction from projections in Class 2, Class 3, Class 3 and Class 4.

Figure D.6: 3D models obtained by reconstructing from the aligned projection images of objects S6x, S7 classified by the proposed method into three classes. The classes corresponding to the same object were merged. (Representation ratio 20:80.)



(a) Reconstruction from projections of object S6x and reconstruction from projections in Class 1.



(b) Reconstruction from projections of object S7 and reconstruction from projections in Class 2 , Class 3, Class 3 and Class 4.

Figure D.7: Differences between 3D models obtained by reconstructing from perfectly classified aligned projection images of objects S6x, S7 and corresponding 3D models obtained by reconstructing from these images classified by the proposed method. (Representation ratio 20:80.)
The yellow and the orange models are 3D reconstructions from perfectly classified projection images. The violet and the blue models are 3D reconstructions obtained from images classified by the proposed method.

Appendix E

Example of the Results from Experiment with Aligned Projection Images IV

Note: All 3D models presented in this appendix were reconstructed using the implementation of ART [28] available in Xmipp [42]. These model were produced under the (unrealistic) assumption that the projection angles for all the images are perfectly assigned. The quality of the models reconstructed form these images with more realistic angular assignment may be significantly lower. All images of the 3D volumes were rendered with UCSF Chimera [37].

Dataset

Dataset group: S6-S6x-S7_33:33:33

Objects represented in the set: S6, S6x, S7

Number of projections of object S6: 1666

Number of projections of object S6x: 1666

Number of projections of object S7: 1667

SNR: 0.1

Alignment: perfect

Reconstructions

No.	Description	Figure
1.	Reconstruction from all images in heterogeneous set.	E.1
2.	Reconstruction from perfectly classified projection images. Number of classes: 2 Classification Purity: 100%	E.2
3.	Reconstruction from the images classified by our method. Number of classes: 2 Classification Purity: 98.44%	E.3

Note: The differences between reconstructions 2. and 3. are shown in Figure E.4.

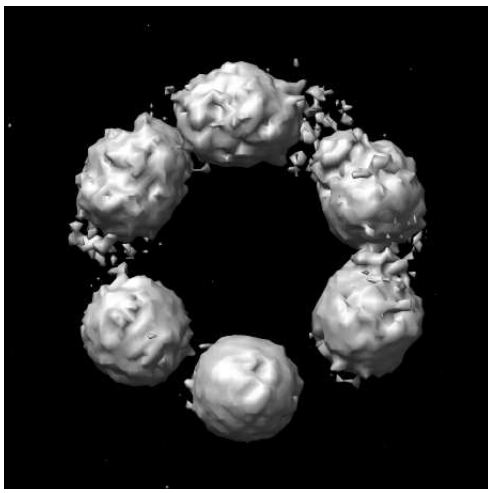
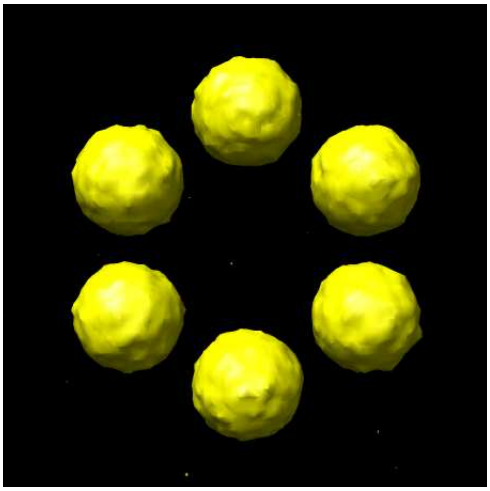
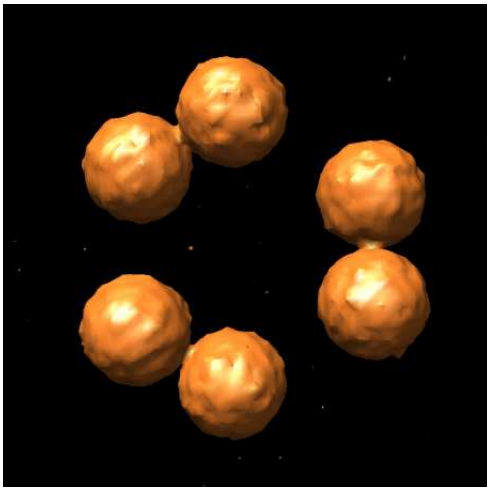


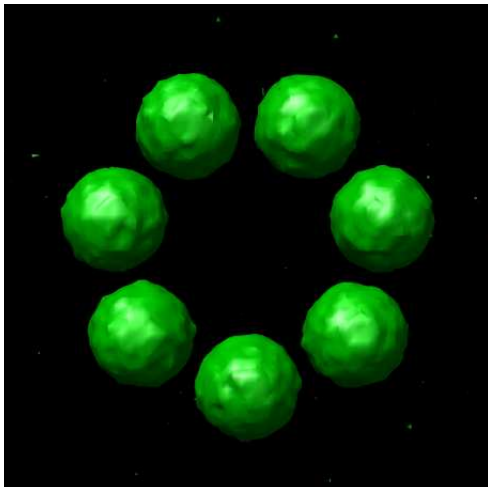
Figure E.1: 3D model obtained by reconstructing from heterogeneous projection set that contains aligned projection images of objects S6, S6x and S7.



(a) Reconstruction from projections of object S6.

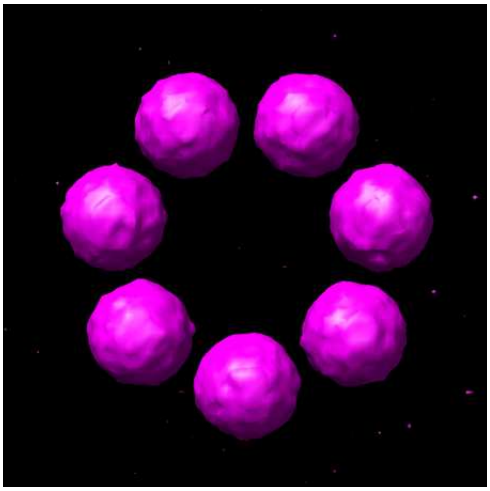


(b) Reconstruction from projections of object S6x.

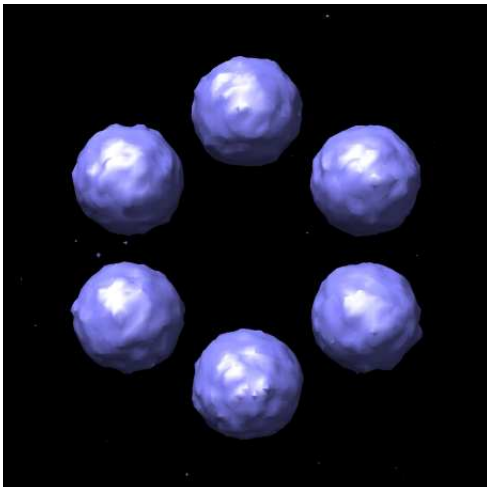


(c) Reconstruction from projections of object S7.

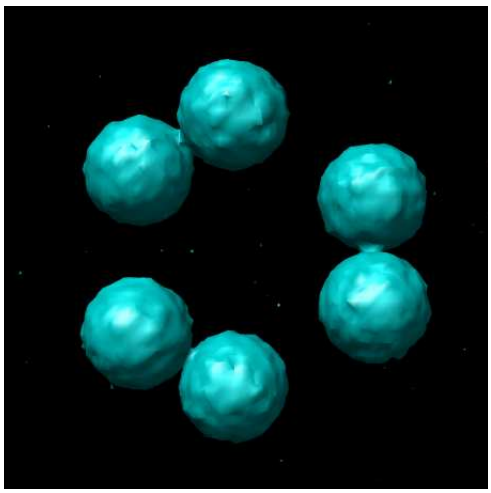
Figure E.2: 3D models obtained by reconstructing from perfectly classified aligned projection images of objects S6, S6x and S7.



(a) Reconstruction from projections in Class 1.

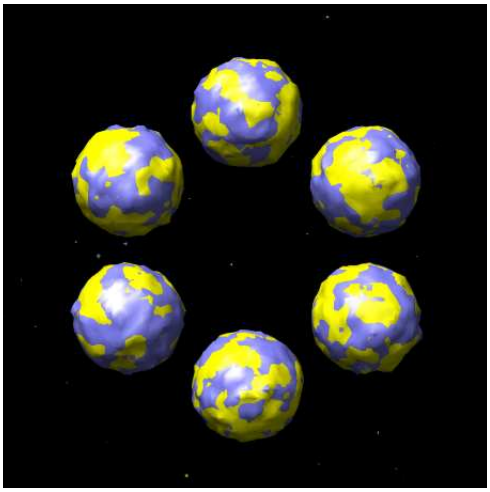


(b) Reconstruction from projections in Class 2.

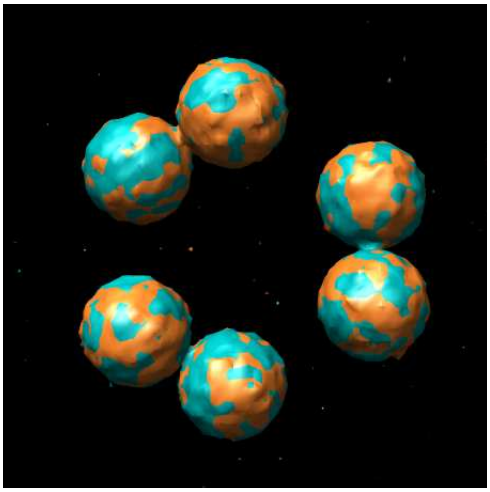


(c) Reconstruction from projections in Class 3.

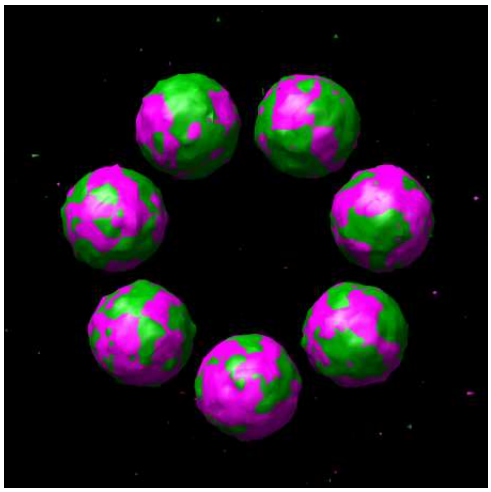
Figure E.3: 3D models obtained by reconstructing from the aligned projection images of objects S6, S6x, S7 classified by the proposed method.



(a) Reconstruction from projections of object S6 and reconstruction from projections in Class 1.



(b) Reconstruction from projections of object S6x and reconstruction from projections in Class 3.



(c) Reconstruction from projections of object S7 and reconstruction from projections in Class 2.

Figure E.4: Differences between 3D models obtained by reconstructing from perfectly classified aligned projection images of objects S6, S6x, S7 and corresponding 3D models obtained by reconstructing from these images classified by the proposed method.
The yellow, orange and light green models are 3D reconstructions from perfectly classified projection images. The violet, blue and green models are 3D reconstructions obtained from images classified by the proposed method.

Appendix F

Example of the Results from Experiment with Misaligned Projection Images I

The dataset used in this appendix comes from the same dataset group as the one used in Appendix G. The results presented here are typical for the datasets in this group.

Note: All 3D models presented in this appendix were reconstructed using the implementation of ART [28] available in Xmipp [42]. These model were produced under the (unrealistic) assumption that the projection angles for all the images are perfectly assigned. The quality of the models reconstructed form these images with more realistic angular assignment may be significantly lower. All images of the 3D volumes were rendered with UCSF Chimera [37].

Dataset

Dataset group: S6-S7_Sh_50:50

Objects represented in the set: S6, S7

Number of projections of object S6: 2500

Number of projections of object S7: 2500

SNR: 0.1

Alignment: misaligned (see 7.3.1 for details)

Reconstructions

No.	Description	Figure
1.	Reconstruction from all images in heterogeneous set.	F.1
2.	Reconstruction from perfectly classified but misaligned projection images. Number of classes: 2 Classification Purity: 100%	F.2
3.	Reconstruction from the images classified by our method. Number of classes: 2 Classification Purity: 97.24%	F.3

Note: The differences between reconstructions 2. and 3. are shown in Figure F.4.

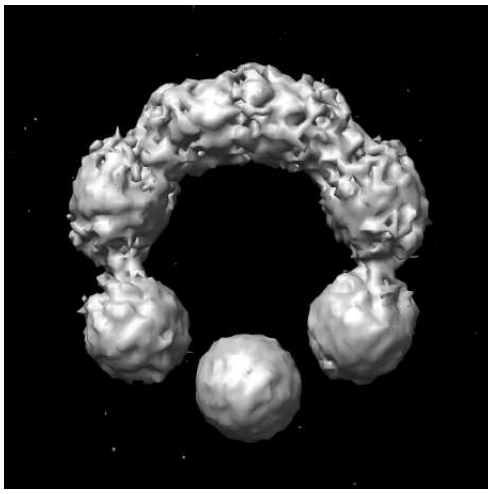
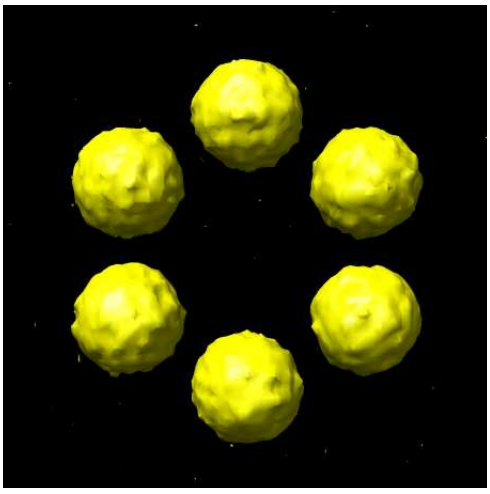
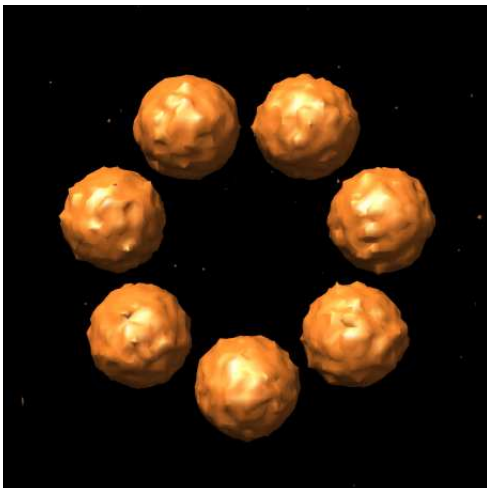


Figure F.1: 3D model obtained by reconstructing from heterogeneous projection set that contains misaligned projection images of objects S6 and S7. (Representation ratio 50:50.)

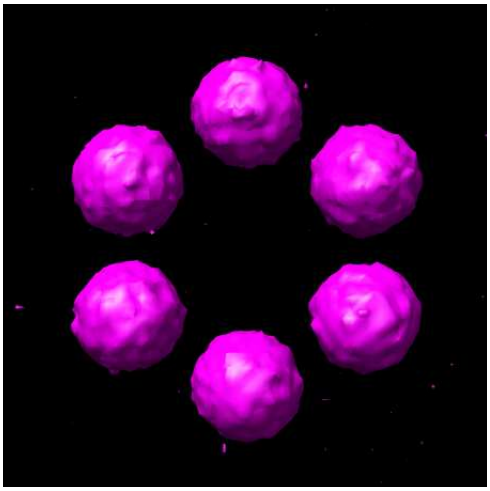


(a) Reconstruction from projections of object S6.

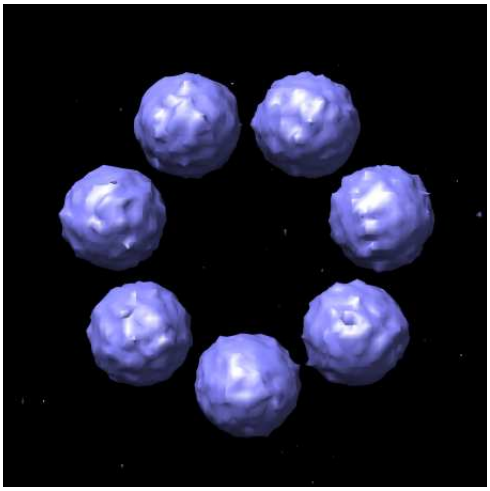


(b) Reconstruction from projections of object S7.

Figure F.2: 3D models obtained by reconstructing from perfectly classified misaligned projection images of objects S6 and S7. (Representation ratio 50:50.)

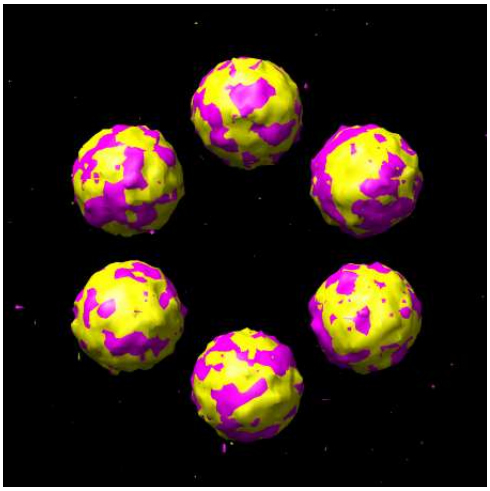


(a) Reconstruction from projections in Class 1.

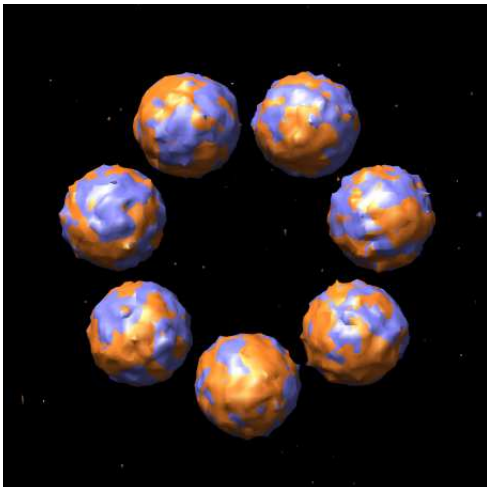


(b) Reconstruction from projections in Class 2.

Figure F.3: 3D models obtained by reconstructing from the misaligned projection images of objects S6, S7 classified by the proposed method. (Representation ratio 50:50.)



(a) Reconstruction from projections of object S6 and reconstruction from projections in Class 1.



(b) Reconstruction from projections of object S7 and Reconstruction from projections in Class 2.

Figure F.4: Differences between 3D models obtained by reconstructing from perfectly classified misaligned projection images of objects S6, S7 and corresponding 3D models obtained by reconstructing from these images classified by the proposed method. (Representation ratio 50:50.)
The yellow and orange models are 3D reconstructions from perfectly classified projection images. The violet and blue models are 3D reconstructions obtained from images classified by the proposed method.

Appendix G

Example of the Results from Experiment with Misaligned Projection Images II

The dataset used in this appendix comes from the same dataset group as the one used in Appendix F. The results presented here correspond to the only dataset for which our method failed.

Note: All 3D models presented in this appendix were reconstructed using the implementation of ART [28] available in Xmipp [42]. These model were produced under the (unrealistic) assumption that the projection angles for all the images are perfectly assigned. The quality of the models reconstructed form these images with more realistic angular assignment may be significantly lower. All images of the 3D volumes were rendered with UCSF Chimera [37].

Dataset

Dataset group: S6-S7_Sh_50:50

Objects represented in the set: S6, S7

Number of projections of object S6: 2500

Number of projections of object S7: 2500

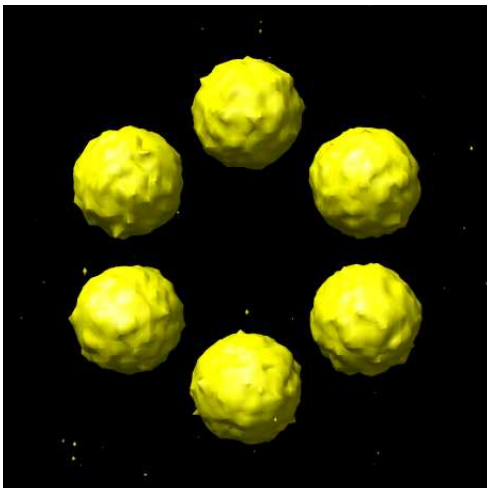
SNR: 0.1

Alignment: misaligned (see 7.3.1 for details)

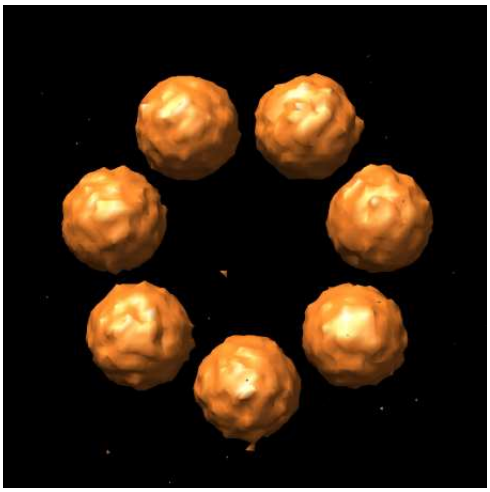
Reconstructions

No.	Description	Figure
1.	Reconstruction from all images in heterogeneous set.	G.1
2.	Reconstruction from perfectly classified but misaligned projection images. Number of classes: 2 Classification Purity: 100%	G.2
3.	Reconstruction from the images classified by our method. Number of classes: 2 Classification Purity: 53.66%	G.3

Figure G.1: 3D model obtained by reconstructing from heterogeneous projection set that contains misaligned projection images of objects S6 and S7 (case 2). (Representation ratio 50:50.)

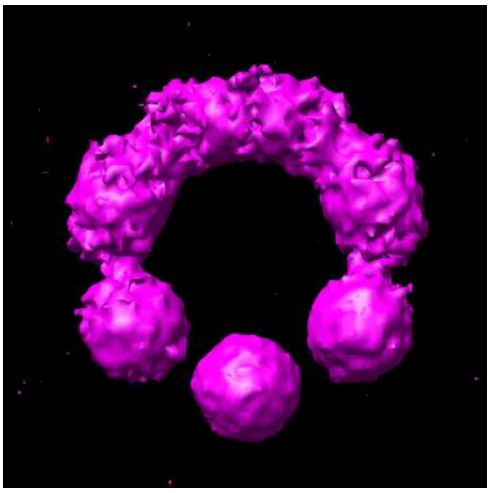


(a) Reconstruction from projections of object S6.

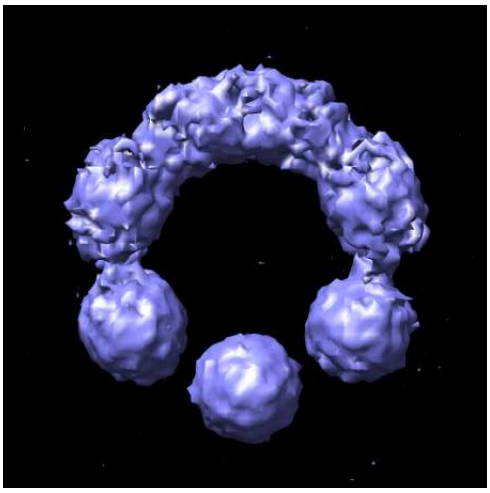


(b) Reconstruction from projections of object S7.

Figure G.2: 3D models obtained by reconstructing from perfectly classified misaligned projection images of objects S6 and S7 (case 2). (Representation ratio 50:50.)



(a) Reconstruction from projections in Class 1.



(b) Reconstruction from projections in Class 2.

Figure G.3: 3D models obtained by reconstructing from the misaligned projection images of objects S6, S7 classified by the proposed method (case 2). (Representation ratio 50:50.)

Appendix H

Example of the Results from Experiment with Misaligned Projection Images III

Note: All 3D models presented in this appendix were reconstructed using implementation of ART [18] available in Xmipp [42]. These model were produced under unrealistic assumption that the projection angles for all the images are perfectly assigned. The quality of the models reconstructed form these images with more realistic angular assignment may be significantly lower. All images of the 3D volumes were rendered with UCSF Chimera [37].

Dataset

Dataset group: S6-S7_Sh_50:50

Objects represented in the set: S6, S7

Number of projections of object S6: 2500

Number of projections of object S7: 2500

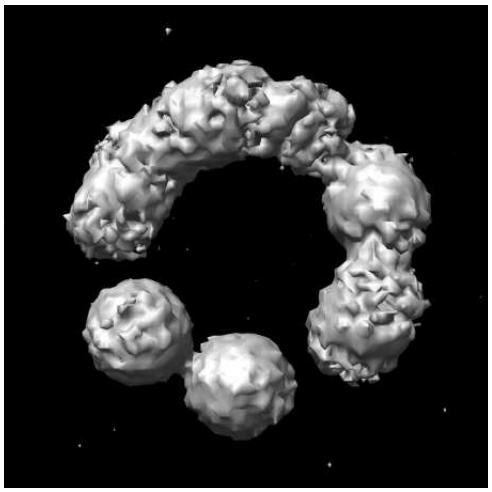
SNR: 0.1

Alignment: misaligned (see 7.3.1 for details)

Reconstructions

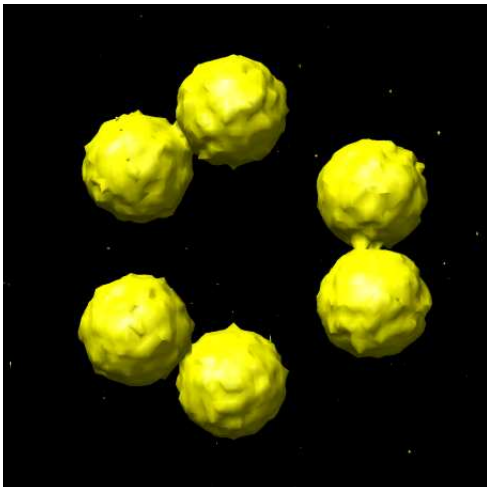
No.	Description	Figure
1.	Reconstruction from all images in heterogeneous set.	H.1
2.	Reconstruction from perfectly classified but misaligned projection images. Number of classes: 2 Classification Purity: 100%	H.2
3.	Reconstruction from the images classified by our method. Number of classes: 2 Classification Purity: 97.24%	H.3

Note: The differences between reconstructions 2. and 3. are shown in Figure H.4.

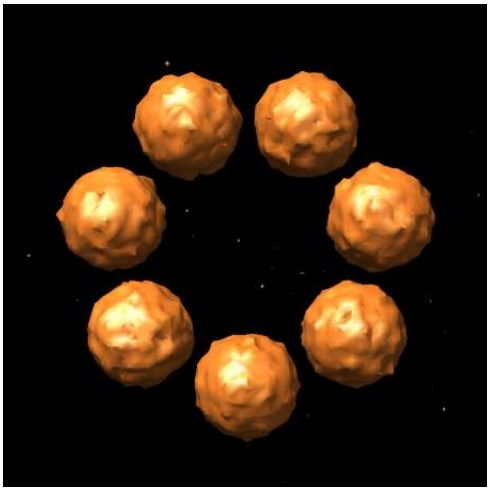


(a) Reconstruction from projections of object S6x.

Figure H.1: 3D model obtained by reconstructing from heterogeneous projection set that contains misaligned projection images of objects S6x and S7. (Representation ratio 50:50.)

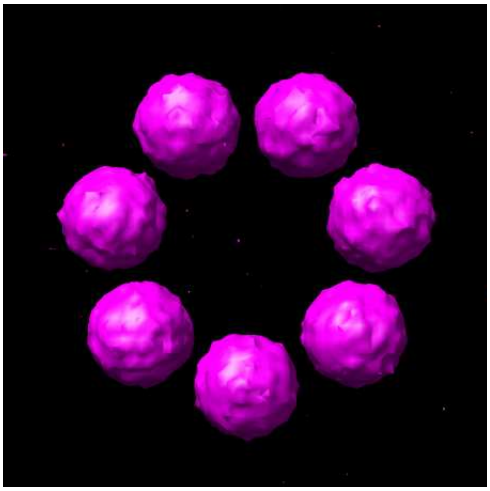


(a) Reconstruction from projections of object S6x.

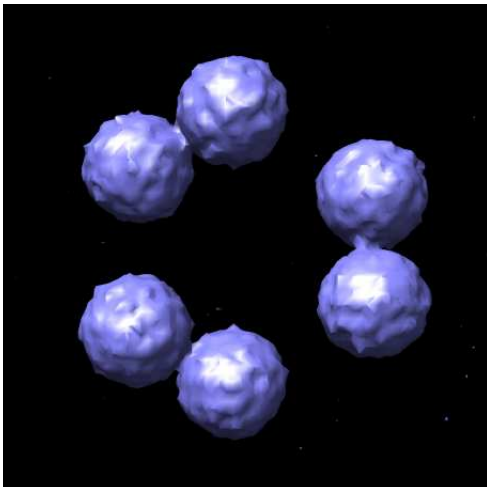


(b) Reconstruction from projections of object S7.

Figure H.2: 3D models obtained by reconstructing from perfectly classified misaligned projection images of objects S6x and S7. (Representation ratio 50:50.)

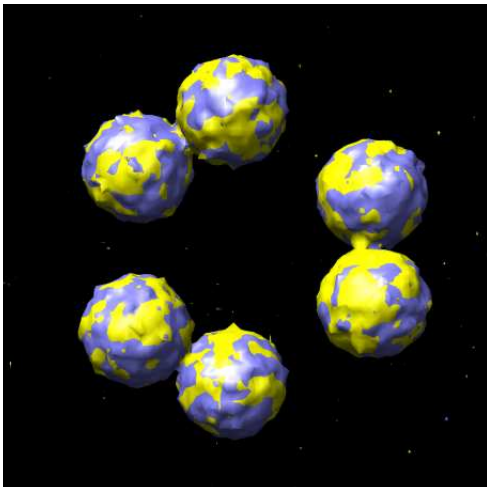


(a) Reconstruction from projections in Class 1.

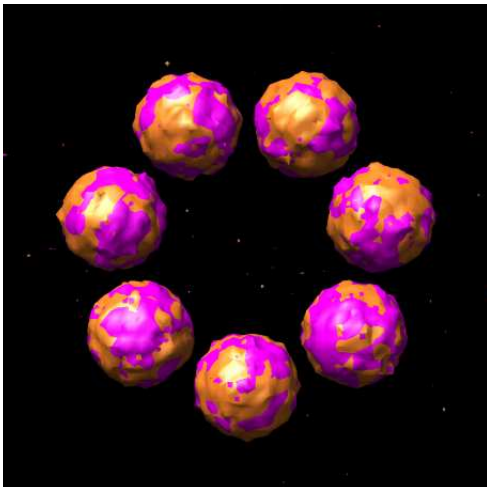


(b) Reconstruction from projections in Class 2.

Figure H.3: 3D models obtained by reconstructing from the misaligned projection images of objects S6x, S7 classified by the proposed method. (Representation ratio 50:50.)



(a) Reconstruction from projections of object S6x and reconstruction from projections in Class 2.



(b) Reconstruction from projections of object S7 and Reconstruction from projections in Class 1.

Figure H.4: Differences between 3D models obtained by reconstructing from perfectly classified misaligned projection images of objects S6x, S7 and corresponding 3D models obtained by reconstructing from these images classified by the proposed method. (Representation ratio 50:50.)
The yellow and the orange models are 3D reconstructions from perfectly classified projection images. The violet and the blue models are 3D reconstructions obtained from images classified by the proposed method.

Appendix I

Example of the Results from Experiment with Misaligned Projection Images IV

Note: All 3D models presented in this appendix were reconstructed using the implementation of ART [28] available in Xmipp [42]. These model were produced under the (unrealistic) assumption that the projection angles for all the images are perfectly assigned. The quality of the models reconstructed form these images with more realistic angular assignment may be significantly lower. All images of the 3D volumes were rendered with UCSF Chimera [37].

Dataset

Dataset group: S6-S6x_Sh_50:50

Objects represented in the set: S6, S6x

Number of projections of object S6: 2500

Number of projections of object S6x: 2500

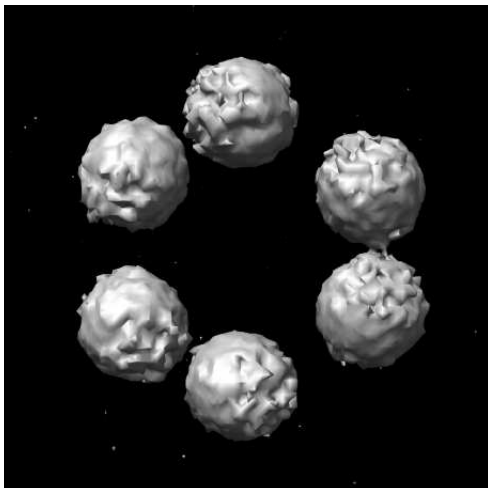
SNR: 0.1

Alignment: misaligned (see 7.3.1 for details)

Reconstructions

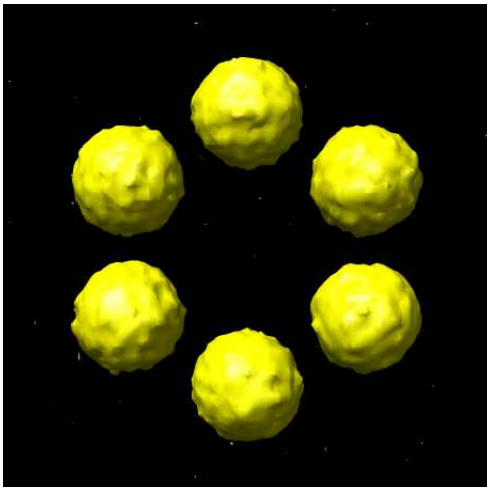
No.	Description	Figure
1.	Reconstruction from all images in heterogeneous set.	I.1
2.	Reconstruction from perfectly classified but misaligned projection images. Number of classes: 2 Classification Purity: 100%	I.2
3.	Reconstruction from the images classified by our method. Number of classes: 2 Classification Purity: 98%	I.3

Note: The differences between reconstructions 2. and 3. are shown in Figure I.4.

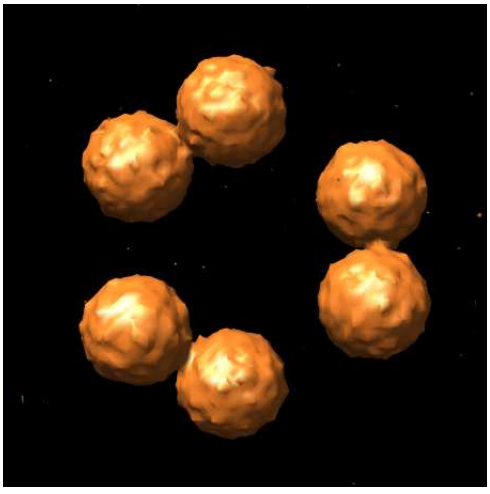


(a) Reconstruction from projections of object S6x.

Figure I.1: 3D model obtained by reconstructing from heterogeneous projection set that contains misaligned projection images of objects S6 and S6x. (Representation ratio 50:50.)

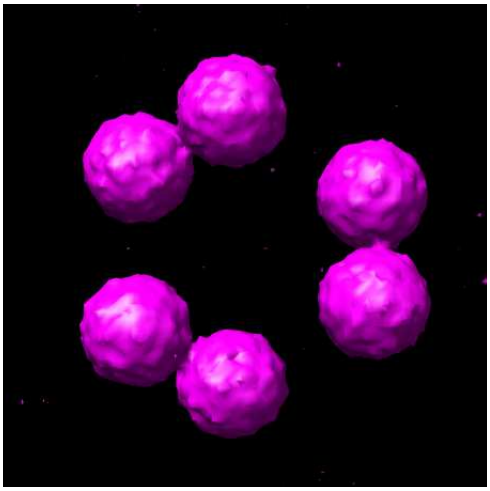


(a) Reconstruction from projections of object S6.

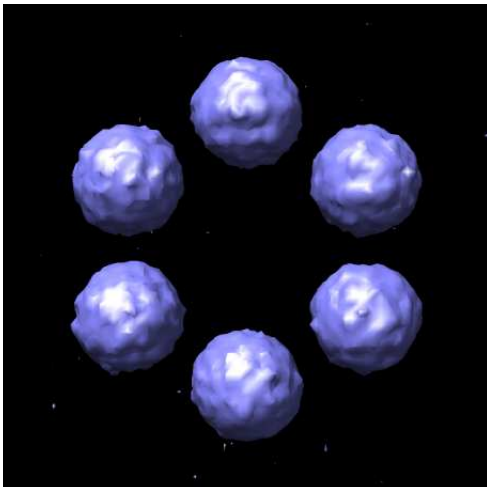


(b) Reconstruction from projections of object S6x.

Figure I.2: 3D models obtained by reconstructing from perfectly classified misaligned projection images of objects S6 and S6x. (Representation ratio 50:50.)

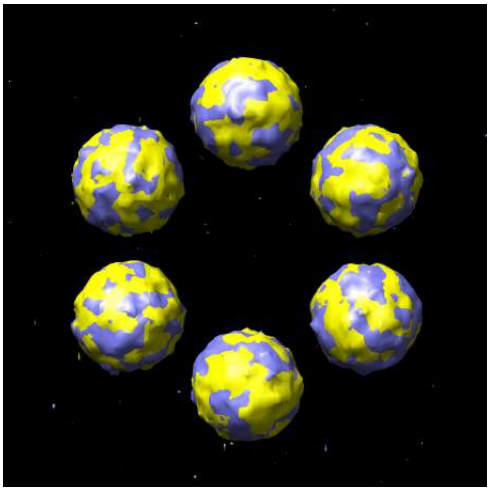


(a) Reconstruction from projections in Class 1.

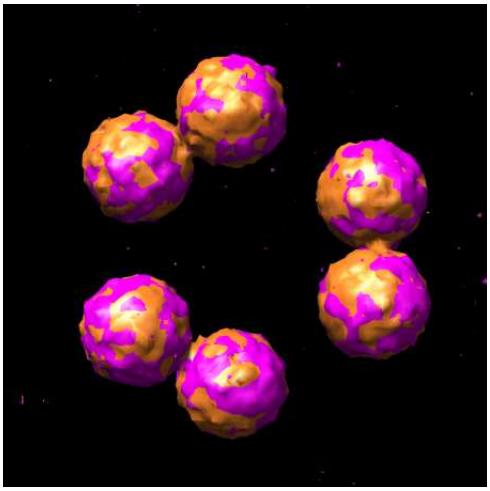


(b) Reconstruction from projections in Class 2.

Figure I.3: 3D models obtained by reconstructing from the misaligned projection images of objects S6, S6x classified by the proposed method. (Representation ratio 50:50.)



(a) Reconstruction from projections of object S6 and reconstruction from projections in Class 2.



(b) Reconstruction from projections of object S6x and Reconstruction from projections in Class 1.

Figure I.4: Differences between 3D models obtained by reconstructing from perfectly classified misaligned projection images of objects S6, S6x and corresponding 3D models obtained by reconstructing from these images classified by the proposed method. (Representation ratio 50:50.)
The yellow and the orange models are 3D reconstructions from perfectly classified projection images. The violet and the blue models are 3D reconstructions obtained from images classified by the proposed method.

Appendix J

Example of the Results from Experiment with Misaligned Projection Images V

Note: All 3D models presented in this appendix were reconstructed using the implementation of ART [28] available in Xmipp [42]. These model were produced under the (unrealistic) assumption that the projection angles for all the images are perfectly assigned. The quality of the models reconstructed form these images with more realistic angular assignment may be significantly lower. All images of the 3D volumes were rendered with UCSF Chimera [37].

Dataset

Dataset group: S6-S6x-S7_Sh_33:33:33

Objects represented in the set: S6, S6x, S7

Number of projections of object S6: 1666

Number of projections of object S6x: 1666

Number of projections of object S7: 1667

SNR: 0.1

Alignment: misaligned (see 7.3.1 for details)

Reconstructions

No.	Description	Figure
1.	Reconstruction from all images in heterogeneous set.	J.1
2.	Reconstruction from perfectly classified but misaligned projection images. Number of classes: 3 Classification Purity: 100%	J.2
3.	Reconstruction from the images classified by our method. Number of classes: 3 Classification Purity: 96.4	J.3

Note: The differences between reconstructions 2. and 3. are shown in Figure J.4.

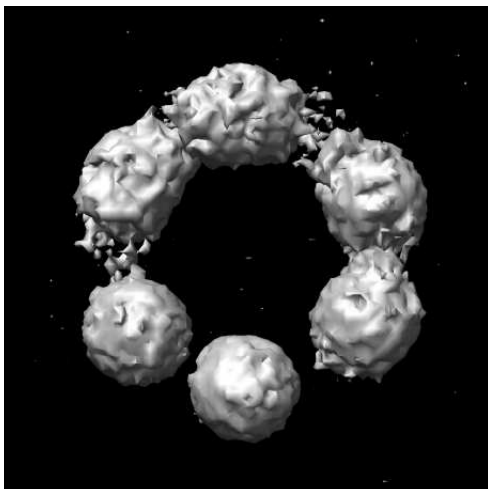
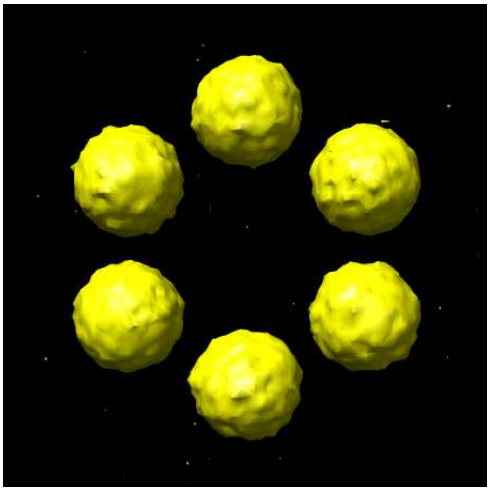
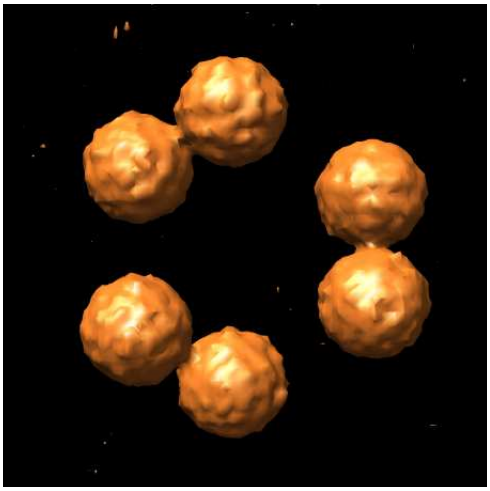


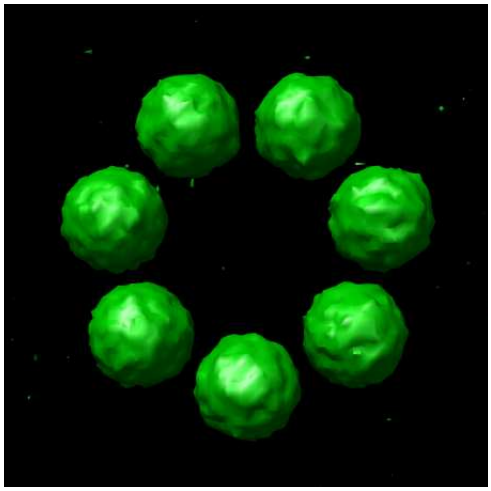
Figure J.1: 3D model obtained by reconstructing from heterogeneous projection set that contains misaligned projection images of objects S6, S6x and S7.



(a) Reconstruction from projections of object S6.

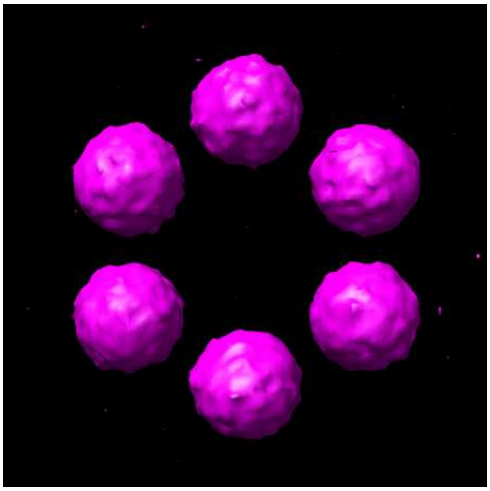


(b) Reconstruction from projections of object S6x.

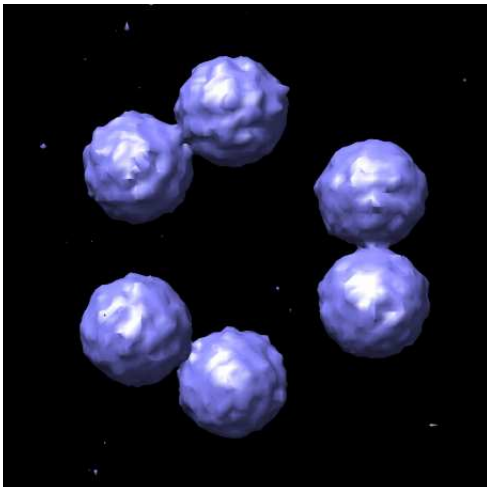


(c) Reconstruction from projections of object S7.

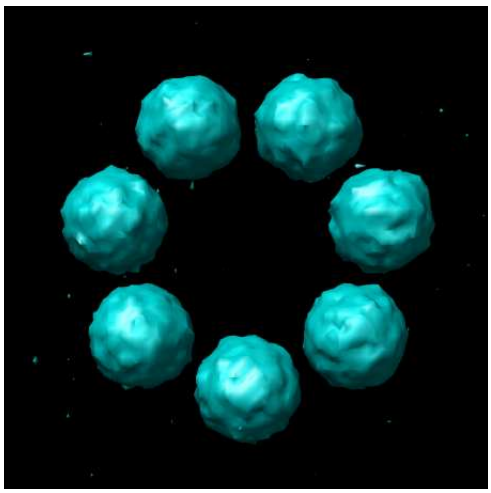
Figure J.2: 3D models obtained by reconstructing from perfectly classified misaligned projection images of objects S6, S6x and S7.



(a) Reconstruction from projections in Class 1.

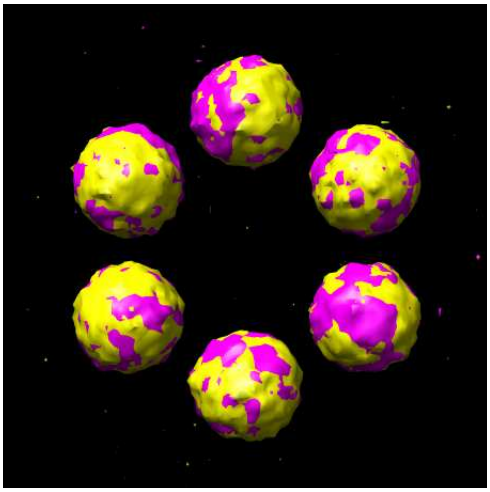


(b) Reconstruction from projections in Class 2.

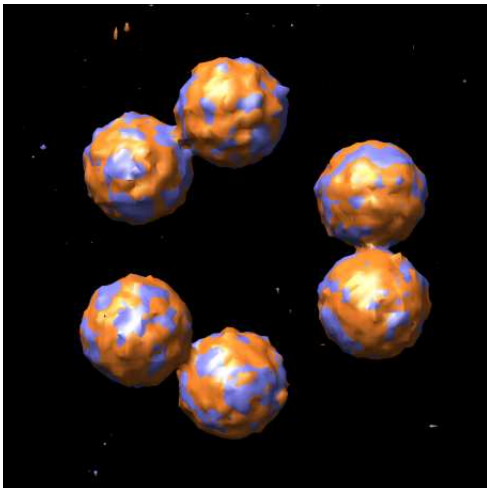


(c) Reconstruction from projections in Class 3.

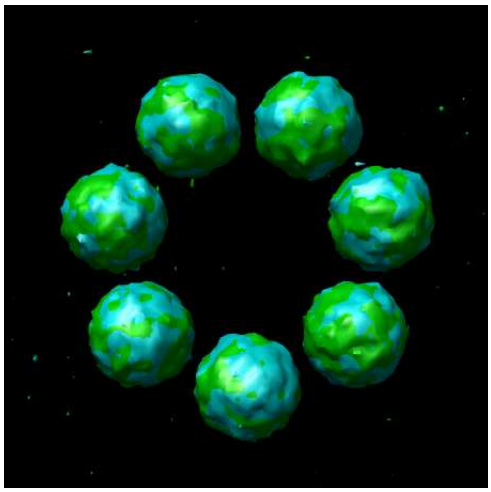
Figure J.3: 3D models obtained by reconstructing from the misaligned projection images of objects S6, S6x, S7 classified by the proposed method.



(a) Reconstruction from projections of object S6 and reconstruction from projections in Class 1.



(b) Reconstruction from projections of object S6x and reconstruction from projections in Class 2.



(c) Reconstruction from projections of object S7 and reconstruction from projections in Class 3.

Figure J.4: Differences between 3D models obtained by reconstructing from perfectly classified misaligned projection images of objects S6, S6x, S7 and corresponding 3D models obtained by reconstructing from these images classified by the proposed method.
The yellow, the orange and the light green models are 3D reconstructions from perfectly classified projection images. The violet, the blue and the green models are 3D reconstructions obtained from images classified by the proposed method.

Appendix K

Objects Used in Evaluation of the Impact of Differences Between Conformations and Noise on Classification Quality

The objects O1, ..., O11 are eleven different configurations of six identical spheres. The coordinates of sphere centers (measured using the size of the pixel edge in the projection image as a unit and assuming the origin to be at the geometrical center of the configuration) are:

O1: A (0, 19.00, 0), B (16.45, 9.50, 0), C (16.45, -9.50, 0), D (0, -19.00, 0), E (-16.45, -9.50, 0), F (-16.45, 9.50, 0).

O2: A (0, 18.05, 0), B (16.97, 9.50, 0), C (16.97, -9.50, 0), D (0, -18.05, 0), E (-16.97, -9.50, 0), F (-16.97, 9.50, 0).

O3: A (0, 17.10, 0), B (17.43, 9.50, 0), C (17.43, -9.50, 0), D (0, -17.10, 0), E (-17.43, -9.50, 0), F (-17.43, 9.50, 0).

O4: A (0, 16.15, 0), B (17.80, 9.50, 0), C (17.80, -9.50, 0), D (0, -17.80, 0), E (-17.80, -9.50, 0), F (-17.80, 9.50, 0).

O5: A (0, 15.20, 0), B (18.12, 9.50, 0), C (18.12, -9.50, 0), D (0, -15.20, 0), E (-18.12, -9.50, 0), F (-18.12, 9.50, 0).

O6: A (0, 14.25, 0), B (18.40, 9.50, 0), C (18.40, -9.50, 0), D (0, -14.25, 0), E (-18.40, -9.50, 0), F (-18.40, 9.50, 0).

O7: A (0, 13.30, 0), B (18.62, 9.50, 0), C (18.62, -9.50, 0), D (0, -13.30, 0), E (-18.62, -9.50, 0), F (-18.62, 9.50, 0).

O8: A (0, 12.35, 0), B (18.79, 9.50, 0), C (18.79, -9.50, 0), D (0, -12.35, 0), E (-18.79, -9.50, 0), F (-18.79, 9.50, 0).

O9: A (0, 11.40, 0), B (18.90, 9.50, 0), C (18.90, -9.50, 0), D (0, -11.40, 0), E (-18.90, -9.50, 0), F (-18.90, 9.50, 0).

O10: A (0, 10.45, 0), B (18.98, 9.50, 0), C (18.98, -9.50, 0), D (0, -10.45, 0), E (-18.98, -9.50, 0), F (-18.98, 9.50, 0).

O11: A (0, 9.50, 0), B (19.00, 9.50, 0), C (19.00, -9.50, 0), D (0, -9.50, 0), E (-19.00, -9.50, 0), F (-19.00, 9.50, 0).

The radius of all spheres (in the same unit) is 8. The density of all the spheres is 1.0.

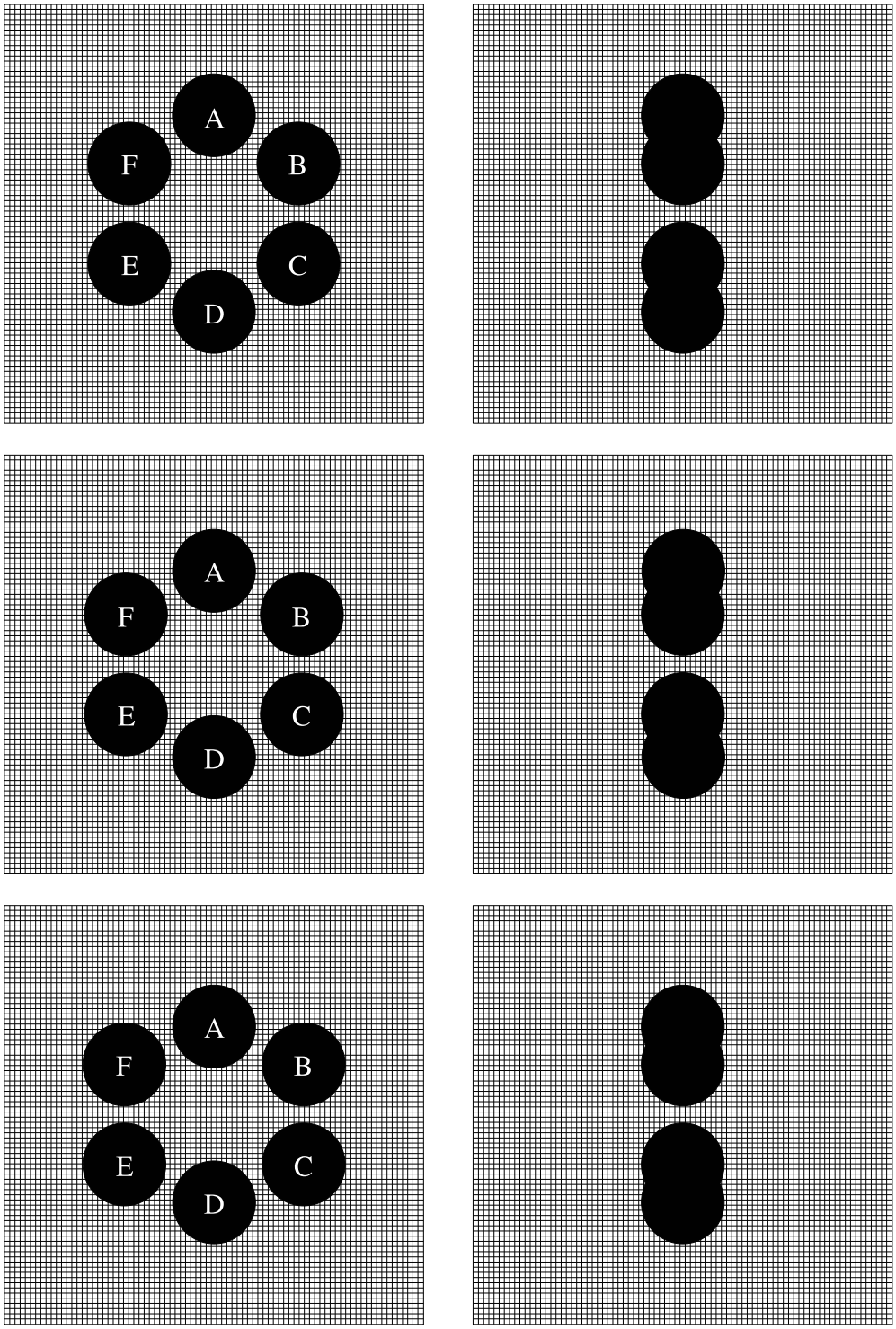


Figure K.1: Geometries of the objects O1, O2, O3 and pixel grid of the projection images.
Top (left column) and side (right column) views of O1 (top row), O2 (center row), O3 (bottom row).

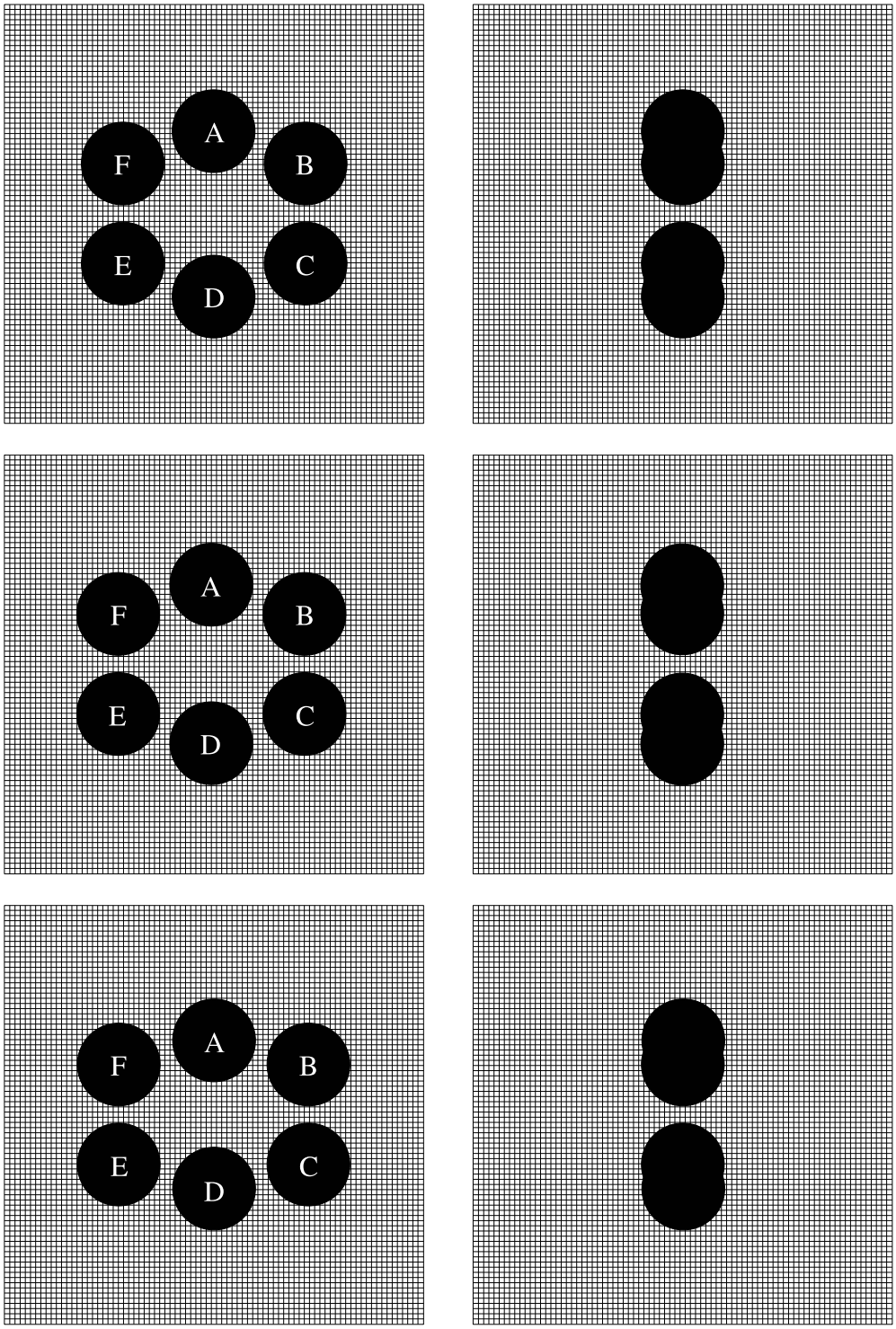


Figure K.2: Geometry of the objects O4, O5, O6 and pixel grid of the projection images.
Top (left column) and side (right column) views of O4 (top row), O5 (center row), O6 (bottom row).

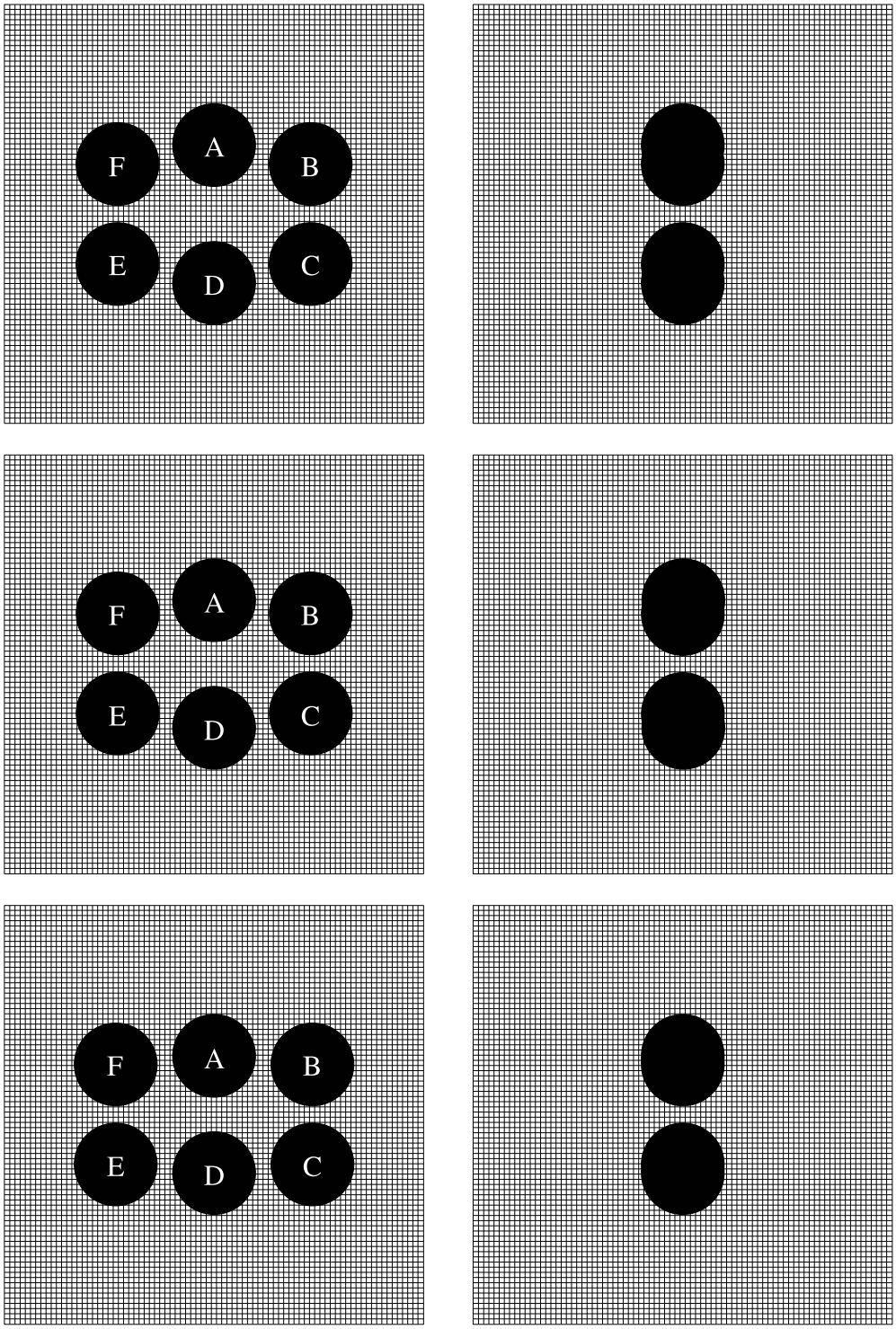


Figure K.3: Geometries of the objects: O7, O8, O9 and pixel grid of the projection images.
Top (left column) and side (right column) views of O7 (top row), O8 (center row), O9 (bottom row).

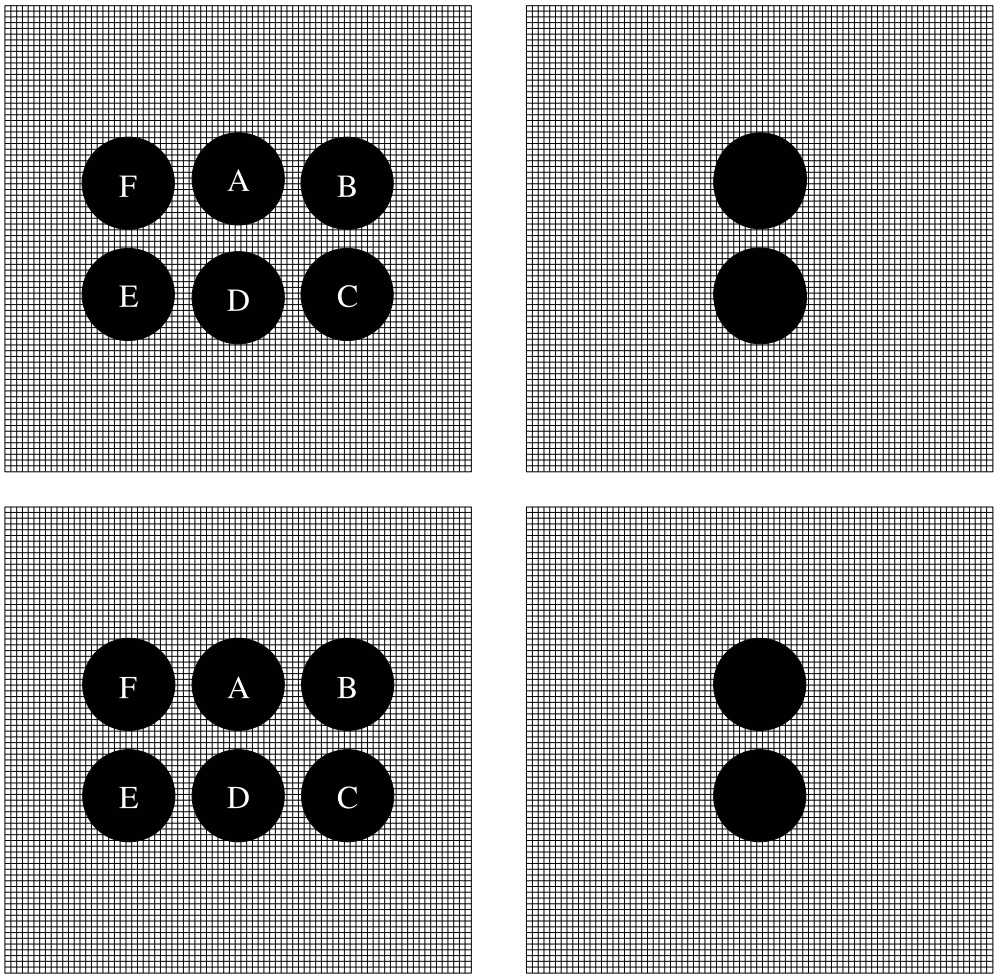


Figure K.4: Geometries of the objects: O10, O11 and pixel grid of the projection images.
Top (left column) and side (right column) views of O10 (top row), O11 (bottom row).

Bibliography

[1] S.J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM J. Opt.*, 10:443–461, 2000.

[2] C.-D. Bey and R.M. Gray. An improvement of the minimum distortion encoding algorithm for vector quantization. *IEEE Trans. Commun.*, 33:1132–1133, 1985.

[3] B. Böttcher, I. Bertsche, R. Reuter, and P. Grabe. Direct visualization of conformational changes in EF0F1 by electron microscopy. *J. Mol. Biol.*, 296:449–457, 2000.

[4] J. Brink, S.J. Ludtke, Y. Kong, S.J. Wakil, J. Ma, and W. Chiu. Experimental verification of conformational variation of human fatty acid synthase as predicted by normal mode analysis. *Structure*, 12:185–191, 2004.

[5] B. M. Carvalho, W. Chen, J. Dubowy, G.T. Herman, M. Kalinowski, H.Y. Liao, L. Rodek, L. Ruskó, S.W. Rowland, and E. Vardi-Gonen. *SNARK05: A Programming System for the Reconstruction of 2D Images from 1D Projections*. CUNY Institute for Software Design and Development, New York, 2006, http://www.cisdd.org/snark05/SNARK05.pdf.

[6] E. Chávez, G. Navarro, R. Baeza-Yates, and J.L. Marroquín. Searching in metric spaces. *ACM Comput. Surv.*, 33:273–321, 2001.

[7] B.H. Cohen. *Explaining Psychological Statistics*. John Wiley & Sons, New York, 2007.

[8] R.A. Crowther. Procedures for three-dimensional reconstruction of spherical viruses by Fourier synthesis from electron micrographs. *Philos. Trans. R. Soc. Lond., B*, 261:221–230, 1971.

[9] R.A. Crowther, D.J. DeRosier, and A. Klug. The reconstruction of a three-dimensional structure from projections and its application to electron microscopy. *Philos. Trans. R. Soc. Lond., B*, 317:319–340, 1970.

[10] R. O. Duda and P.E. Hart. *Pattern Recognition and Scene Analysis*. John Wiley & Sons, New York, 1973.

[11] J. Frank. Single-particle imaging of macromolecules by cryo-electron microscopy. *Annu. Rev. Biophys. Biomol. Struct.*, 31:303–319, 2002.

[12] J. Frank. *Three-Dimensional Electron Microscopy Of Macromolecular Assemblies: Visualization of Biological Molecules in Their Native State*. Oxford University Press, USA, 2006.

[13] J. Frank and P. Penczek. On the correction of the contrast transfer function in biological electron microscopy. *Optik*, 98:125–129, 1995.

[14] J. Frank, M. Radermacher, P. Penczek, J. Zhu, Y. Li, M. Ladjadj, and A. Leith. SPIDER and WEB: Processing and visualization of images in 3D electron microscopy and related fields. *J. Struct. Biol.*, 116:190–199, 1996.

[15] J. Frank, B. Shimkin, and H. Dowse. SPIDER - a modular software system for electron image processing. *Ultramicroscopy*, 6:343–358, 1981.

[16] J. Fu, H. Gao, and J. Frank. Unsupervised classification of single particles by cluster tracking in multi-dimensional space. *J. Struct. Biol.*, 157:226–239, 2007.

[17] H. Gao, M. Valle, M. Ehrenberg, and J. Frank. Dynamics of EF-G interaction with the ribosome explored by classification of a heterogeneous cryo-em dataset. *J. Struct. Biol.*, 147:283–290, 2004.

[18] R. Gordon, R. Bender, and G.T. Herman. Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. *J. Theoret. Biol.*, 29:471–482, 1970.

[19] F. Harary. *Graph Theory*. Addison-Wesley, New York, 1969.

[20] P. Heckbert. Color image quantization for frame buffer display. In *Proc. of SIGGRAPH '82*, pages 297–307, 1982.

[21] G.T. Herman. *Image Reconstruction from Projections: The Fundamentals of Computerized Tomography*. Academic Press, New York, 1980.

[22] G.T. Herman and M. Kalinowski. Classification of heterogeneous electron microscopic projections into homogeneous subsets. *Ultramicroscopy*, in press, 2007.

[23] J.B. Heymann, J.F. Conway, and A.C. Steven. Molecular dynamics of protein complexes from four-dimensional cryo-electron microscopy. *J. Struct. Biol.*, 147:291–301, 2004.

[24] M. Kalinowski, A. Daurat, and G.T. Herman. A fast construction of the distance graph used for the classification of heterogeneous electron microscopic projections. In *Proc. of the 6th International Workshop on Graph-Based Representations in Pattern Recognition, LNCS 4538*, pages 263–272, Berlin Heidelberg, 2007. Springer-Verlag.

[25] V. Kann, S. Khanna, J. Lagergren, and A. Panconesi. On the hardness of approximating Max k-Cut and its dual. *Chicago J. Theoret. Comp. Sci.*, http://cjtcs.cs.uchicago.edu/articles/1997/2/contents.html, 1997.

[26] J.Z.C. Lai, Y.-C. Liaw, and J. Liu. Fast k-nearest-neighbor search based on projection and triangular inequality. *Pattern Recogn.*, 40:351–359, 2007.

[27] R. Marabini, , and J.M. Carazo. Patern recognition and classification of biological macromolecules using artificial neural networks. *Biophys. J.*, 66:1804–1814, 1994.

[28] R. Marabini, G.T. Herman, and J.-M. Carazo. 3D reconstruction in electron microscopy using art with smooth spherically symmetric volume elements (blobs). *Ultramicroscopy*, 72:53–65, 1998.

[29] M.Radermacher, T.Wagenknecht, A.Verschoor, and J.Frank. Three-dimensional reconstruction from a single-exposure random conical tilt series applied to the 50S ribosomal subunit of escherichia coli. *J. Microsc.*, 146:113–136, 1987.

[30] M.Radermacher, T.Wagenknecht, A.Verschoor, and J.Frank. Three-dimensional structure of the large subunit from escherichia coli. *EMBO J.*, 6:1107–1114, 1987.

[31] F. Natterer and F. Wübbeling. *Mathematical Methods in Image Reconstruction*. SIAM, Philadelphia, 2001.

[32] M. Radermacher P. Penczek and J. Frank. Three-dimensional reconstruction of single particles embedded in ice. *Ultramicroscopy*, 40:33–53, 1992.

[33] A.N. Papadopoulos and Y. Manolopoulos. *Nearest Neighbor Search: A Database Perspective*. Springer, New-York, NY, USA, 2005.

[34] P.M. Pardalos and M.G.C. Resende. *Handbook of Applied Optimization*. Oxford University Press, New York, 2002.

[35] P.A. Penczek, J. Frank, and C.M.T. Spahn. A method of focused classification, based on the bootstrap 3D variance analysis, and its application to EF-G-dependent translocation. *J. Struct. Biol.*, 154:184–194, 2006.

[36] P.A. Penczek, J. Zhu, R. SchrÃ¶der, and J. Frank. Three dimensional reconstruction with contrast transfer compensation from defocus series. *Special Issue on Signal and Image Processing, Scanning Microscopy*, 11:147, 1997.

[37] E.F. Pettersen, T.D. Goddard, C.C. Huang, G.S. Couch, D.M. Greenblatt, E.C. Meng, and T.E. Ferrin. UCSF Chimera - a visualization system for exploratory research and analysis. *J. Comput. Chem.*, 25:1605–1612, 2004.

[38] M. Samsó and M.P. Koonce. 25 Å resolution structure of a cytoplasmic dynein motor reveals a seven-member planar ring. *J. Mol. Biol.*, 340:1059–1072, 2004.

[39] S.H.W. Scheres, H. Gao, M. Valle, G.T. Herman, P.P.B. Eggermont, J. Frank, and J.M. Carazo. Disentangling conformational states of macromolecules in 3D-EM through likelihood optimization. *Nature Methods*, 4:27–29, 2007.

[40] S.H.W. Scheres, M. Valle, R. Nunez, C.O.S. Sorzano, R. Marabini, G.T. Herman, and J.M. Carazo. Maximum-likehood multi-reference refinement for electron microscopy images. *J. Mol. Biol.*, 348:139–149, 2005.

[41] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer Verlag, Berlin, 2003.

[42] C.O.S. Sorzano, R. Marabini, J. Velazquez-Muriel, J.R. Bilbao-Castro, S.H.W. Scheres, J.M. Carazo, and A. Pascual-Montano. Xmipp: a new generation of an open-source image processing package for electron microscopy. *J. Struct. Biol.*, 148:194–204, 2004.

[43] F. Tama, O. Miyashita, and C.L. Brooks III. Normal mode based flexible fitting of high-resolution structure into low-resolution experimental data from cryo-em. *J. Struct. Biol.*, 147:315–326, 2004.

[44] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2006.

[45] D.J. Taylor, J. Nilsson, A.R. Merrill, G.R. Andersen, P. Nissen, and J. Frank. Structures of modified eEF2.80S ribosome complexes reveal the role of GTP hydrolysis in translocation. *EMBO J.*, 26:2421–2431, 2007.

[46] M. Valle, J. Sengupta, N.K. Swami, R.A. Grassucci, N. Burkhardt, K.H. Nierhaus, R.K. Agrawal, and J. Frank. Cryo-EM reveals an active role for aminoacyl-trna in the accommodation process. *EMBO J.*, 21:3557–3567, 2002.

[47] M. Van Heel. Angular reconstitution: A posteriori assignment of projection directions for 3D reconstruction. *Ultramicroscopy*, 21:111–124, 1987.

[48] E. Vidal. An algorithm for finding nearest neighbours in (approximately) constant time. *Pattern Recogn. Lett.*, 4:145–157, 1986.

[49] W. Wriggers, R.K. Agrawal, D.L. Drew, A. McCammon, and J. Frank. Domain motions of EF-G bound to the 70s ribosome: insights from a hand-shaking between multi-resolution structures. *Biophysical J.*, 79:1670–1678, 2000.

[50] S. Yang, X. Yu, V.E. Galkin, and E.H. Egelman. Issues of resolution and polymorphism in single-particle reconstruction. *J. Struct. Biol.*, 144:162–171, 2003.

[51] P.N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proc. of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 311–321, 1993.

Index