New Algorithms
for
Macromolecular Structure Determination

Dissertation
zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades
“Dr. rerum naturalium”
an der Georg-August-Universität Göttingen

vorgelegt von

Burkhard C. Heisen

aus Wolfhagen

Göttingen 2009

Thesis Committee:
Prof. Dr. Holger Stark
Prof. Dr. George M. Sheldrick
Prof. Dr. Markus Wahl

Affidavit

I hereby declare that this PhD thesis 'New Algorithms for Macromolecular Structure Determination' has been written independently with no other aids or sources than quoted. This thesis (wholly or in part) has not been submitted elsewhere for any academic award or qualification.

Burkhard C. Heisen

July, 2009

Göttingen, Germany

Related Publications

Parts of this thesis have been published, or presented at international meetings or symposia:

- B. C. Heisen, M. Schmeisser, H. Stark, D. Moser and A. S. Frangakis (2009). Parallel processing with graphics cards. Workshop. *New Algorithms in Macromolecular Crystallography and Electron Microscopy.* J.P. Abrahams. Leiden, NL. 2009

- M. Schmeisser, B. C. Heisen, M. Luettich, B. Busche, F. Hauer, T. Koske, K.-H. Knauber and H. Stark (2009) Parallel, distributed and GPU computing technologies in single-particle electron microscopy. *Acta Crystallogr D Biol Crystallogr* 65(Pt 7):659–671

- Luettich, M., B. Sander, B. C. Heisen, M. Schmeisser, B. Busche and H. Stark (2008). 3D Structure Determination of Dynamic Macromolecular Complexes by single particle cryo-EM. *New Algorithms in Macromolecular Crystallography and Electron Microscopy.* J. P. Abrahams. Leiden, NL.

In addition, I was involved in cooperations leading to the following publications or manuscripts. I wish to express my kindest regards to all people involved in these studies for the productive collaborations.

- A. Ranganathan, B.C. Heisen, I. Dix and F. A. Meyer (2007) A triazine-based three-directional rigid-rod tecton forms a novel 1D channel structure. *Chem Commun (Camb)*, 35:3637–3639

- G. B. Nikiforov, H. W. Roesky, B. C. Heisen, C. Grosse and Rainer B. Oswald (2008) Formation of a Titanium Complex with a Ti=$CHAl_2$ Structural Unit from $LTiMe_3$ and Trimethylaluminium *Organometallics* 27:2544–2548

Contents

Contents **vii**

List of Figures xi

List of Tables xiii

Acknowledgements **xv**

Abstract **xvii**

1 Introduction **1**

1.1 Three dimensional structure determination of macromolecules 1

1.2 Single particle cryo-electron microscopy 3

1.2.1 Overview 3

1.2.2 Instrumentation 5

1.2.3 Image formation 6

1.2.3.1 Phase contrast 6

1.2.3.2 Amplitude contrast 11

1.2.4 Computational image processing 12

1.2.4.1 Signal-to-Noise Ratio 12

1.2.4.2 Particle picking 12

1.2.4.3 Correction of the PhCTF and image filtering 12

1.2.4.4 Alignment 13

1.2.4.5 Dimension reduction and classification 15

1.2.4.6 3D Reconstruction 20

1.2.4.7 Validation 23

1.3 Scientific software development 23

1.3.1 Management and storage of large datasets 23

1.3.2 Parallel programming 24

1.3.2.1 Farming 26

1.3.2.2 GPU programming 27
1.4 Aim of the work 31

2 Materials and Methods **33**
2.1 Code generation – hard- and software used 33
2.2 Image processing framework 34
2.2.1 Back-end 34
2.2.2 Front-end 40
2.3 Reference-free image classification 42
2.3.1 Motivation 42
2.3.1.1 The objective function 43
2.3.2 Implementation 45
2.3.2.1 Unit cell preparation 46
2.3.2.2 Crystallization 48
2.3.2.3 Crystal Improvement 50

3 Results **53**
3.1 Preparation of synthetic test data 53
3.2 Proof of concept using synthetic data 56
3.2.1 Discrimination of individual images 56
3.2.1.1 Discrimination on invariant representations 61
3.2.2 Classification of synthetic data 64
3.3 Classification of real data 66
3.3.1 Test scenario 1 - 70S ribosome 66
3.3.2 Test scenario 2 - anaphase-promoting complex (APC) 70
3.4 Refinement of already classified data 72
3.5 Performance issues in the light of parallel programming 75

4 Discussion **79**
4.1 Classification - Iterative *vs.* multivariate data analysis 79
4.1.1 Modified scoring function and unit cell setup 81
4.1.2 Taking the best from two worlds - Ideas for a hybrid approach . . . 83
4.2 The *align* in Crystalign 84
4.3 Streamlining the process from raw data to 3D structure 84

A Mathematical Fundamentals **87**

A.1 Fourier theory 87
A.1.1 Convolution 88
A.1.2 Correlation 88
A.1.3 Power spectrum 89
A.2 Fourier based Gaussian image filtering 89
A.3 Fourier-slice theorem 90
A.4 Radon transform 91

List of Symbols and Abbreviations **93**

Bibliography **95**

Curriculum Vitae **103**

List of Figures

1.1 Overview - From sample to structure 4
1.2 The TEM Microscope 5
1.3 Scattering by a two point system 7
1.4 Electron microscopic phase contrast 9
1.5 Wave aberration 10
1.6 Phase contrast transfer function 11
1.7 Introducing bias by aligning random content 14
1.8 3D Reconstruction 21
1.9 Parallel programming on different levels 25
1.10 Differences in hardware between CPU and GPU 27
1.11 CUDA scatter and gather operations 29

2.1 Plug-and-Play architecture of the image processing pipeline 35
2.2 Internal structure of the HDF file 38
2.3 Screenshot of the user front-end 40
2.4 Motivation to classification algorithm 42
2.5 Image pre-processing 47
2.6 Flow-chart of the crystallization algorithm 49

3.1 Synthetic test dataset 55
3.2 Test setup for inidvidual image scoring 57
3.3 Single image scoring - data heterogeneity 58
3.4 Single image scoring - projection angle differences 59
3.5 Single image scoring - contrast enhancement 60
3.6 Single image scoring - growing dissimilarity 61
3.7 Proof of concept - Invariance 62
3.8 Test setup for invariant image scoring 63
3.9 Single image scoring - discrimination of invariants 63

3.10 Complete classification of a synthetic dataset 64
3.11 Classification of a 70S ribosome cryo-EM dataset 67
3.12 Test scenario 1 (70S ribosome) - FRCs for 15 and 25 class sums 68
3.13 Test scenario 1 (70S ribosome) - FRCs for 50 and 80 class sums 69
3.14 Distribution of class sizes 70
3.15 Classification of a cryo-EM dataset of the anaphase-promoting Complex (APC) 71
3.16 Test scenario 2 (APC) - FRCs for 15 and 25 class sums 72
3.17 Test scenario 2 (APC) - FRCs for 50 and 80 class sums 73
3.18 Effect of sorting class sums by quality 74
3.19 Refinement of classified data - Removal of low quality images 75
3.20 Speed measurements for the discrete Radon transformation 76

A.1 2D-Gaussian band-pass filter profile 89

List of Tables

1.1 Comparison of methods used for three dimensional structure determination. 3

2.1 Listing of all external libraries used 34

A.1 Correspondence between symmetries in the two Fourier related domains . 88

Acknowledgements

First of all I would like to thank Prof. Dr. George M. Sheldrick. With his outstanding scientific knowledge, his open mind, and his excellent competence in teaching he provided me with the theoretical foundations this work is built on. Prof. Dr. Holger Stark not only did a great job in supervision, but also in envision the future developments of an interesting scientific field, which I was allowed to participate in. With his always optimistic, motivating attitude and his excellent research environment he contributed to the success of this work.

I am also very grateful to Prof. Dr. Markus Wahl for serving on my thesis committee, and supporting my work. Furthermore, I would especially like to thank my office-colleagues Dr. Martin Schmeißer and Boris Busche for valuable discussions, and their contribution to a very enjoyable working environment. For assistance and advice, I would like to thank Florian Hauer, Niels Fischer, Tobias Koske, Dr. Mario Lüttich, Jan Kirves, Dr. Prakash Dube, Ilonka Bartoszek and Andrius Krasauskas from the cryo-EM group.

I am very grateful and indebted to Dr. Steffen Burkhardt from the Coordination Office. With untiring effort and always personal atmosphere he and his assistants (present and past), Kerstin Grüniger, Ivana Bacakova and Nina McGuiness deliver an excellent framework for scientific research and education within the Molecular Biology MSc/PhD Program.

Another special thanks to my friend Ben Frank, who had the heart to found a company with me and accompany me at every time (even those on motorbike). My friends Florian Hauer, Christian Stegmann, Marc Schneider contributed to an enjoyable time in Göttingen ever since. Finally, for unconditional support and motivational upkeep I wish to thank my parents and my lovely girlfriend Annika Osteroth.

Abstract

Detailed three dimensional information of macromolecules is often crucial to the study of biological systems. Structural data have for example been used to elucidate the basis of diseases resulting from variant forms of proteins and design drugs to inhibit molecules involved in diseases.

However, given a purified macromolecule under investigation there still is no trivial way to experimentally "image" the three dimensional structure directly. All currently available methods (such as X-ray crystallography, single particle cryo-electron microscopy or nuclear magnetic resonance) deliver experimental data which first have to computationally processed prior to obtaining a meaningful three dimensional structure. Thus, the quality and efficiency of computational methods in structure biology have a major impact on the whole field. Recent advantages in technical setup and computational power allow for the design of new methods which were virtually infeasible to use before. This especially holds true in the field of cryo-EM which computationally is most demanding compared to all other methods mentioned above.

In the work presented here new methods for managing and processing the huge amounts of image data produced by the cryo-EM technique are introduced and discussed. Algorithms aiming for accuracy improvements during image processing were implemented using state-of-the-art technology (such as parallel programming on graphic processing devices) and were embedded in a flexible, object-oriented image-processing suite delivering a high degree of operational flexibility and simplicity. Most importantly an algorithm was designed which is able to automatically identify and remove individual images which, upon inclusion, would reduce the overall quality of the final 3D structure.

Chapter 1

Introduction

Mit jedem Fortschritt der Wissenschaft wird die Schwierigkeit der Aufgabe des Forschers immer größer, die Anforderungen an seine Leistungen immer stärker und es stellt sich immer dringender die Notwendigkeit einer zweckmäßigen Arbeitsteilung ein. Vor allem hat sich seit etwa einem Jahrhundert die Teilung in Experiment und Theorie vollzogen.

—Max Planck

1.1 Three dimensional structure determination of macromolecules

Given a purified, intact sample of a biological macromolecule and the interest in knowing about the three dimensional (3D) structure of the latter, at least three different methods are availabe to date. Those are single particle (cryo-)electron microscopy (cryo-EM), X-ray crystallography, and nuclear magnetic resonance spectroscopy (NMR).

X-ray crystallography may safely be termed the oldest, most commonly used and most mature method. The first protein structure to be solved by this method was that of myoglobin at 6 Å resolution already in 1957 (Kendrew et al., 1958). Crystallographic analysis can be applied to a wide range of compounds, starting from the smallest inorganic minerals which may consist of less than 10 atoms up to huge macromolecular complexes like the ribosome which is build up from approximately 14000 atoms (Rossmann, 2006). Unfortunately, this method exhibits a severe bottleneck which prevents its completely routine usage - the preparation of the biological sample, i.e. the growth of a well diffracting crystal. The process of crystallization is still not understood to an extend that would allow the controlled growth of crystals from any biological macromolecule (Mandelkern, 2001a,b). Thus, the structural investigation may be impeded in the very beginning. Having available

crystals, however, potentially very high resolution structures (in extreme cases < 1 Å, for example see (Biadene et al., 2007)) can be achieved. Another limitation, which results from the very concept of analyzing crystals, is the relative difficulty in inspection molecular dynamic effects other than atomic thermal motion (Schneider, 1996; Kidera et al., 1992; Benoit and Doucet, 1995; Thüne and Badger, 1995).

In contrast to crystallography, the analysis of molecular dynamics is a strength of NMR-based structure investigation (for example see (Salmon et al., 2009)). This method utilizes the biophysical properties of nuclear spins to retrieve indirect structural information (such as specific atom linkages and distances) from which the final structure is computed. The biological sample is commonly analyzed in solution (rarely in a solid state), however biomolecules have to be isotope-labeled and quite a big amount of mostly hard to obtain biological starting material is needed. Furthermore NMR is limited to molecules of moderate size ($\sim$ 40 kDa), which severely impairs the investigation of large macromolecular complexes.

Regarding the investigation of asymmetric macromolecules, single particle cryo-electron microscopy historically is the youngest method available. Although the first three dimensional reconstruction from electron micrographs (tail of bacteriophage T4) succeeded in 1968 (de Rosier and Klug, 1968), only 10 years after solution of the first crystal structure, this structure was a mere methodological proof of concept. The first asymmetric structure of sub-nanometer resolution was that of the ribosome obtained by Valle et al. in 2002. Only recent improvements in instrumentation and especially in computing performance empowered this method to be used more or less routinely. Cryo-EM is very well suited for the analysis of huge macromolecules or even complexes of those (Frank, 2006, 2002; Stark and Lührmann, 2006). It has the big advantage of needing only tiny amounts (in the order of three magnitudes less than needed for crystallography or NMR) of the biological sample under investigation. If the dataset (i.e. the collection of single molecule projections) and the computing power is large enough, dynamic effects may also be studied via separating different conformers and performing ensemble refinements (Leschziner and Nogales, 2007). Structural studies additionally exploiting dynamical effects were for example performed for the U4/U6.U5 Tri-snRNP (Sander et al., 2006), the RNA editing machine in trypanosomes (Golas et al., 2009) or the anaphase-promoting complex (Herzog et al., 2009). Conversely, smaller molecules (< 200 kDa) are hard to investigate as they

are lacking contrast in transmission electron-microscopic images.

Common to all methods is their need of extensive computational post-processing of the experimentally obtained data. This is especially true for cryo-EM where the reliability of the final model is directly linked to the accuracy of the engaged algorithms and limited by the amount of computational power available. Given their specific prerequisites and properties, these methods complement each other in determining the 3D structure of a given biological macromolecule. Table 1.1 summarizes the main features of X-ray crystallography, NMR, cryo-EM outlined so far.

Table 1.1: Comparison of methods used for three dimensional structure determination.

Property/Method	**X-Ray**	**NMR**	**Cryo-EM**
Main physical phenomenon	Elastic X-ray scattering on electrons	Nuclear spin transitions and interactions	Elastic electron scattering on nuclei
Size range	$<$ 1.5 MDa	$<$ 40 kDa	$>$ 200 kDa
Typical resolution	1 - 6 Å	n.a.	5 - 30 Å
Analysis of dynamics	$-$	$++$	$+$
Current number of structures solved*	49485	7846	240

*As obtained from the RCSB Protein Data Bank (Berman et al., 2000), May 2009.

1.2 Single particle cryo-electron microscopy

1.2.1 Overview

Briefly, the purified biological sample is applied to a sample grid and either stained at room temperature or shock frozen in its native, hydrated state in a thin layer of vitrified ice (Lepault and Dubochet, 1986). Subsequently, transmission electron microscopic images are recorded, in which various two dimensional projections of identical but randomly oriented 3D biomolecules on the sample grid are represented. As the biological samples are sensitive to radiation damage, image acquisition is done under low dose conditions ($\sim$20 e/Å^2). This, however, results in reduced contrast and increased noise in the recorded

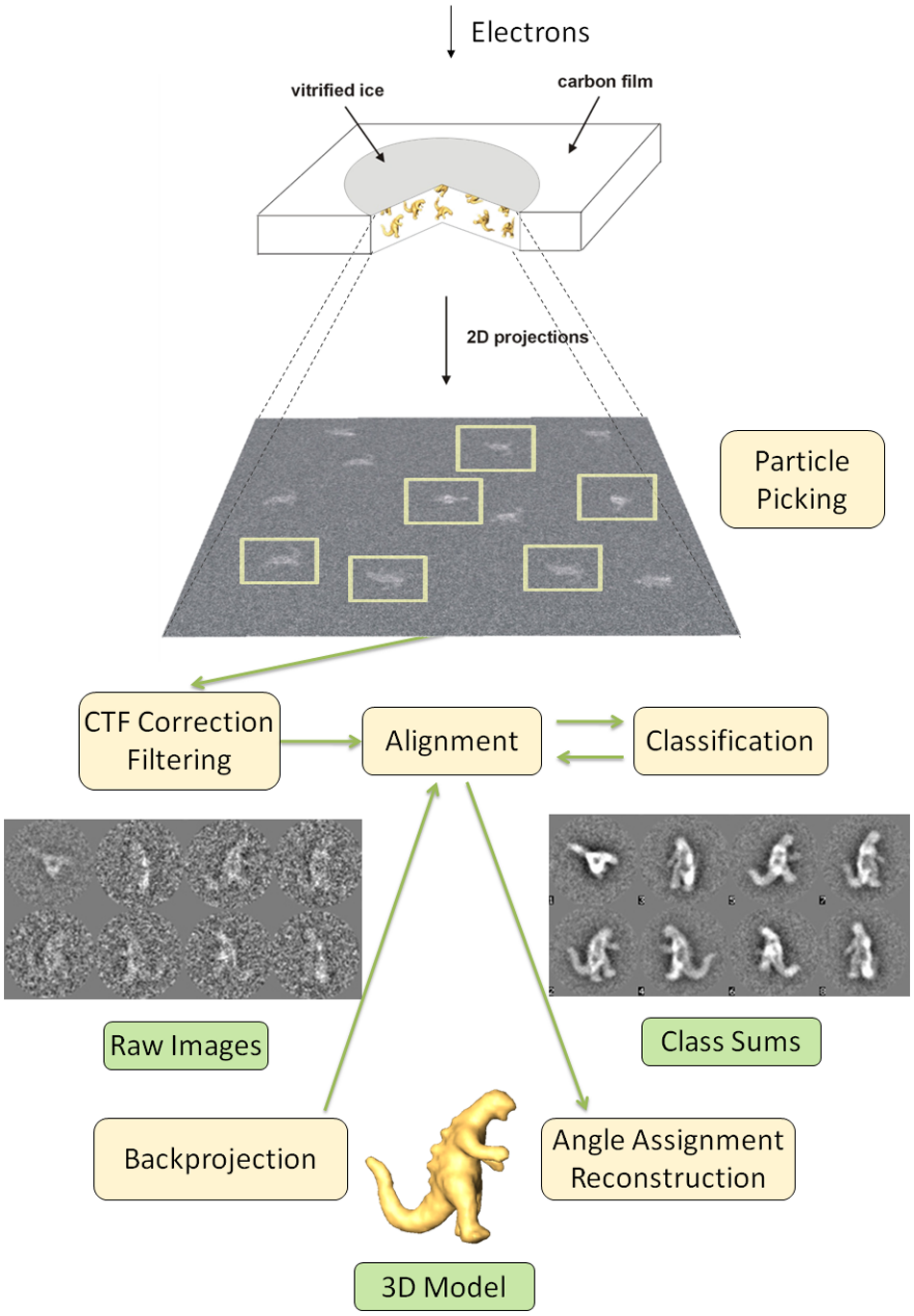

Figure 1.1: Overview - From sample to structure in single particle cryo-electron microscopy.

images. In order to overcome those experimentally imposed limitations, single particle images representing the same orientation of the 3D biomolecule can be averaged. Prior to averaging, images representing the same orientation have to be found and aligned onto

Figure 1.2: **A**: The latest generation of a TEM, the FEI "Titan Krios". The overlay (indicated by the yellow frame) shows the microscope as it would look like with front doors open. **B**: Schematic illustration of the main components a TEM is built of.

each other, such that the averaging is taking place over information related to the same structural content. Averaging can become very difficult in the presence of high levels of noise and extensive sample heterogeneity (i.e. the presence of several, slightly different 3D biomolecules from which the 2D projections are obtained from) and hence is aided by intensive computational image processing (see later sections for more detail).

1.2.2 Instrumentation

The experimental data obtained from a cryo-EM experiment is 2D projection images of single molecules as imaged by a transmission electron microscope (TEM) in bright-field imaging mode. Figure 1.2A shows a *state-of-the-art* TEM (FEI - "Titan Krios") as for example is used in the group of Prof. H. Stark. The *field emission gun* (FEG) is the source of a spatially and temporally highly coherent electron beam. Electrons are accelerated through a selected potential difference of typically 100 – 400 kV resulting in a $\sim 100,000$ fold shorter wavelength than that of visible light (380 – 780 nm). The *condenser system* (consisting of two or more lenses) demagnifies the initial beam and enables the adjustment of the spot size (i.e. the beam diameter on the specimen). Below the condenser lies the *specimen chamber*, one of the most crucial parts of a modern TEM. A special holder must be positioned accurately and under liquid nitrogen temperature inside the objective lens.

It furthermore has to be capable of being moved several millimeters and tilted by large angles, still being operated in Ångstrom precision and stability. The strong *objective lens* forms the first intermediate image and the first diffraction pattern in the back focal plane. By limiting the angular range of the scattered rays the *objective lens aperture* ultimately sets the upper resolution limit (and thus no aperture or a large aperture setting is used for high resolution images). The *intermediate and projector lenses* further magnify the first image and finally project it onto a fluorescent screen or a CCD device, respectively. Figure 1.2B illustrates the most important components schematically.

1.2.3 Image formation

The underlying physical principle of image formation in a TEM is the interaction of beam electrons with the specimen. For this interaction two mutually exclusive effects are distinguished: i) *elastic scattering* and ii) *inelastic scattering*. In the case of elastic scattering the electrons are diffracted by the Coulomb field of the specimens' nuclei. No loss of energy is involved in this interaction. In contrast, inelastic scattering involves energy transfer, i.e. the electrons loose energy which is deposited on the specimen, leading to radiation damage and unwanted background scattering effects. It is the elastic scattering effect which is used for imaging whilst the influence of inelasting scattering is tried to be reduced, which can partly be accomplished by energy filtration (if instrumentally available), i.e. masking out electrons that have lost marginal amounts (0 – 15 eV) of energy. The amount of radiation damage is reduced by cooling the specimen to cryogenic temperatures.

Having scattered electrons, still an image has to be formed. Using electron lenses, Ernst Ruska succeeded in this task with the first electron microscope built in 1931 (for a historical review see (Ruska, 1979)). A meaningful image exhibits image contrast, i.e. a 2D distribution of different intensities. Independent of the scattering type, contrast may physically be generated by means of *amplitude-contrast* (particle-optical effect) and/or *phase-contrast* (wave-optical effect). In cryo-EM a mixture of both effects take part in the image formation process but with $/sim95$ % contribution the effect of phase-contrast is of most importance and is introduced briefly in the following section.

1.2.3.1 Phase contrast

The elastic scattering by the specimen may first be simplified by inspecting the scattering by two points P and Q separated at a distance $\boldsymbol{r}$. Figure 1.3 shows an incident wave along

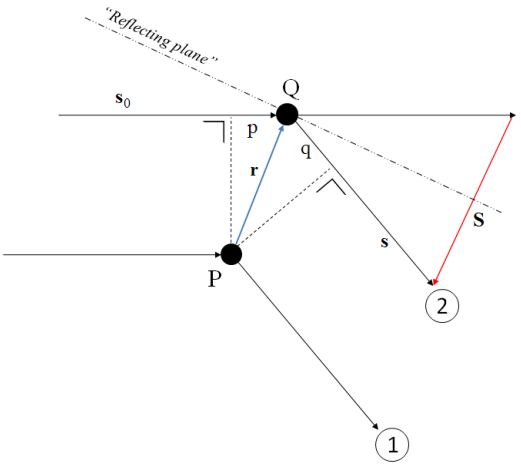

Figure 1.3: The black dots (P and Q) are sources of scattering. The origin of the system is at P; Q is at position $\boldsymbol{r}$. The system is irradiated by an incoming wave in direction $\boldsymbol{s_0}$. The scattered wave is observed in the direction of vector $\boldsymbol{s}$. Because of the path difference $p + q$, the scattered wave 2 will lag behind scattered wave 1 in phase. The total scattering can be described by the vector $\boldsymbol{S}$, which is perpendicular to an imaginary reflecting plane.

$\boldsymbol{s_0}$ scattered by P and Q resulting in the scattered vector $\boldsymbol{s}$. We assume the points to scatter completely independent of each other. Therefore, the amplitudes of the scattered waves 1 and 2 are equal, but have a phase difference resulting from the geometric path difference between the wave passing through point P and the wave passing through point Q. The path difference is $p + q = \lambda[\boldsymbol{r} \cdot (\boldsymbol{s_0} - \boldsymbol{s})]$. The phase angle of wave 2 in respect to wave 1 is

$$\varphi_g = \frac{-2\pi\lambda[\boldsymbol{r} \cdot (\boldsymbol{s_0} - \boldsymbol{s})]}{\lambda} = 2\pi \boldsymbol{r} \cdot \boldsymbol{S} \tag{1.1}$$

where $\boldsymbol{S} = \boldsymbol{s} - \boldsymbol{s_0}$ and λ is the wavelength.

As depicted in Figure 1.3, $\boldsymbol{S}$ is perpendicular to an imaginary diffraction plane reflecting the incident and the exit beam at equal angles θ, and the length of $\boldsymbol{S}$ is given by

$$|\boldsymbol{S}| = 2sin\frac{\theta}{\lambda} \tag{1.2}$$

Thus the total scattering for the two-point system is

$$\psi(\boldsymbol{r}) = f_p + f_q e^{2\pi i \boldsymbol{r} \cdot \boldsymbol{S}} \tag{1.3}$$

where f_p and f_q are the resultant wave amplitudes for P and Q, respectively. Stepping gradually back to real world, the term "point" may as well be replaced by "atom", such

that f becomes the scattering power of atoms P and Q of the specimen. Unfortunately, f does not linearly change with the atomic number (Z). Especially for high values of Z, absorption effects have to be taken into account. This is practically done by making the atomic scattering amplitude complex and separating it into three parts: $f_{corr} = f + f' + if''$ where f is the contribution of the "original" (uncorrected) scattering amplitude and f' and f'' are the real and imaginary contributions of the absorption effect. In TEM this effect is attributed to those contributing to *amplitude contrast* [1].

The diffraction of the whole specimen may now be regarded as the diffraction of a plane of atoms, which itself leads to a phase retardation of $\varphi_s = \frac{\pi}{2}$ with respect to the scattering by a single atom. This phase retardation is caused by means of Fresnel diffraction which exact derivation is out of the scope of this introduction and is referred to Kauzmann (1957). Thus, the exit wave after passing the specimen can be described by

$$\psi_s(\boldsymbol{r}) = \psi_0 e^{i\varphi_s(\boldsymbol{r})} \tag{1.4}$$

with ψ_0 being the incident wave prior to specimen diffraction. Finally, the amplitude $F(\boldsymbol{S})$ in the diffraction plane can be obtained by integration over all the surface elements $d^2\boldsymbol{r}$ of the specimen plane,

$$F(\boldsymbol{S}) = \iint \psi_s(\boldsymbol{r}) e^{(i\varphi_g)}\, d^2\boldsymbol{r} = \iint \psi_s(\boldsymbol{r}) e^{(2\pi i \boldsymbol{r}\cdot\boldsymbol{S})}\, d^2\boldsymbol{r} \tag{1.5}$$

This shows that $F(\boldsymbol{S})$ mathematically is the (two dimensional) Fourier transform of $\psi_s(\boldsymbol{r})$. The intensities in the final image (which are proportional to the squared amplitudes) may be exemplified by decomposing $F(\boldsymbol{S})$ into an unscattered incident wave amplitude ψ_i and a scattered ($\frac{\pi}{2}$ phase-shifted) amplitude ψ_{sc}. Figure 1.4 shows that, for $\psi_{sc} \ll \psi_i$, the resultant amplitude has approximately the same absolute value as ψ_i, so that $I = |\psi_i + i\psi_{sc}|^2$ does not significantly differ from $I_0 = |\psi_i|^2$. This simply means that no contrast is generated and the phase object (similar to light-microscopy) is invisible. If however, the phase of the scattered wave could be shifted by a further $\pm\frac{\pi}{2}$ the superposition would become $\psi_i \pm \psi_{sc}$ and hence $I = |\psi_i \pm \psi_{sc}|^2 \neq I_0$ resulting in *negative* or *positive phase contrast*, respectively (see Figure 1.4).

In light-microscopy this phase shift is introduced by a so-called *Zernike* (Zernike, 1942) phase plate, which - for technical reasons (Majorovits et al., 2007; Cambie et al., 2007;

[1] In analogy, this phenomenon is known as *anomalous dispersion* (see REF) in X-ray crystallography.

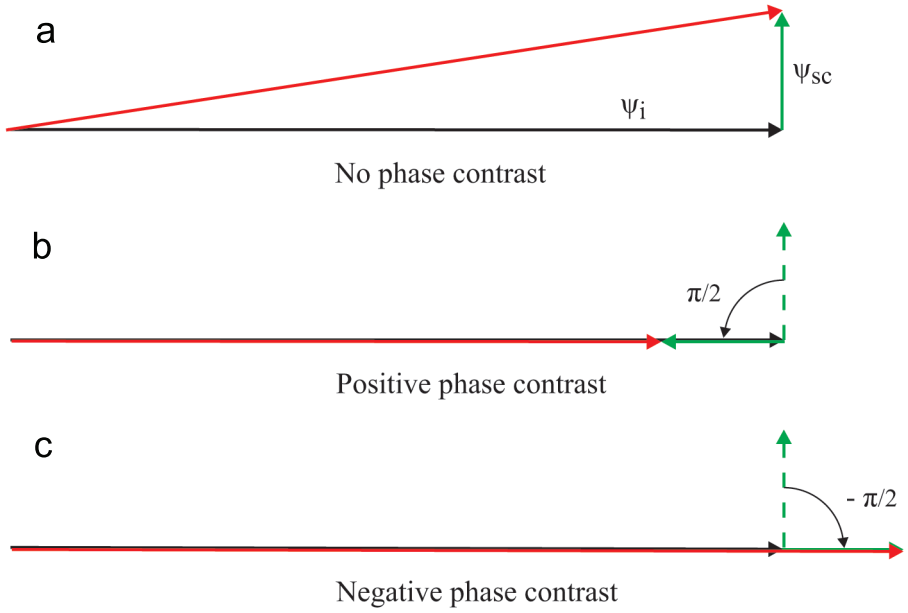

Figure 1.4: Electron microscopic phase contrast. **a**: Addition of the vectors of incoming and diffracted wave under the assumption that no additional phase shift has occurred in the objective lens. **b**: A phase shift of $\frac{\pi}{2}$ causes positive phase contrast. **c**: A phase shift of $-\frac{\pi}{2}$ results in negative phase contrast.

Schroder et al., 2007) - can not easily be used in TEM. Instead, phase contrast is generated by defocussing.

Phase contrast transfer function The phase shift φ_d induced through defocusing is a function of the scattering angle θ or, in other words, varies with the spatial frequency (real-space: resolution-shell). To be accurate, another source of phase shift has to be taken into account which is caused by the so-called *spherical aberration* C_s, a property inherent to all real lenses including light-optical as well as electro-magnetic ones. Together these effects are termed *wave aberration* and can concisely be written using the Scherzer formula (Scherzer, 1949)

$$W(\theta) = \frac{\pi}{2\lambda}(C_s\theta^4 - 2\Delta z\lambda\theta^2), \tag{1.6}$$

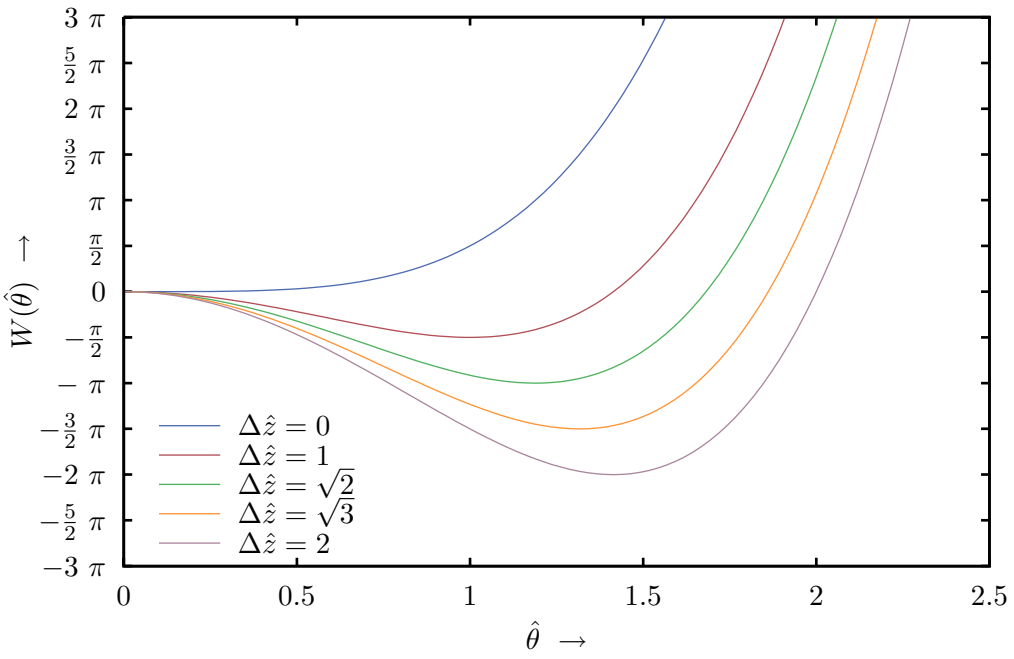

Figure 1.5: Wave aberration $W(\hat{\theta})$ as a function of the generalized scattering angle $\hat{\theta}$ for various reduced focusing distances $\Delta\hat{z}$. Figure modified from Reimer and Kohl (2007).

with θ describing the scattering angle and Δz the current defocus. As the wave aberration depends on two parameters C_s and λ which are typically different among different instruments/experiments it is convenient to discuss 1.6 in terms of generalized coordinates

$$\hat{\theta} = \theta(\frac{C_s}{\lambda})^{\frac{1}{4}} \quad \text{and} \quad \Delta\hat{z} = \Delta z(C_s\lambda)^{-\frac{1}{2}}. \tag{1.7}$$

This results in the general wave aberration

$$W(\hat{\theta}) = 2\pi(\frac{\hat{\theta}^4}{4} - \frac{\Delta\hat{z}\hat{\theta}^2}{2}) \tag{1.8}$$

which is plotted in Figure 1.5. Inspection of this function reveals that a generalized defocus of $\Delta\hat{z} = 1$ (also called Scherzer focus) in terms of creating contrast is most advantageous because $W(\theta)$ has the value $-\frac{\pi}{2}$ over a relatively broad range of scattering angles or spatial frequencies, respectively. However, in practice a disadvantage of images generated in Scherzer focus is that they are hard to detect because lower resolution features (low spatial frequencies) are poorly transmitted thus having the same effect on the image as a drastic high-pass filter (see A.2) would have.

The effect of (1.6) as an additional phase shift on (1.5) can be imagined as a, with increasing spatial frequency accelerating, rotation of the scattered amplitude ψ_{sc} in the complex plane. Thus the final intensity I at scattering angle θ can be described as

$$I(\theta) = |F(\boldsymbol{S})|^2 \sin(W(\theta)) \tag{1.9}$$

where $\sin(W(\theta))$ often is referred to as *phase contrast transfer function* (PhCTF) (Figure 1.6). Equation (1.9) only holds true under the assumption of an infinite aperture and

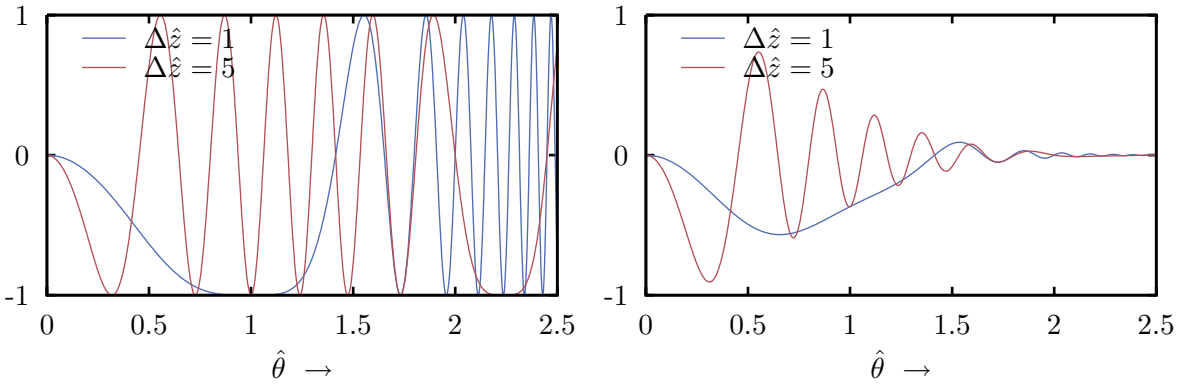

Figure 1.6: Left: Undampened PhCTF. Right: Exponentially dampened PhCTF. Both functions are shown for different generalized defoci $\Delta\hat{z}$.

perfect beam coherence. In practise neither of both assumptions holds true, and the effect of a finite aperture and partial beam coherence dampens the amplitude exponentially. This dampening may be expressed through the convolution with an envelope function for which different representations have been developed (Wade and Frank, 1977) and is termed (experimental) B-factor.

1.2.3.2 Amplitude contrast

Amplitude contrast is mainly generated by two effects, i) strongly scattered electrons that do not hit the recording device anymore ii) electrons that get absorpt by the specimen. Both effects decrease the total electron beam power, hence resulting in amplitude contrast. However, in cryo-EM very thin specimens with maximum aperture settings are investigated hence the effect of amplitude contrast - to a first approximation - can be neglected.

1.2.4 Computational image processing

"Computers are useless. They can only give you answers."
—Pablo Picasso

Having collected digital images of the specimen, a lot of computer aided processing has to be performed to extract the inherent 3D structure(s) out of the 2D projections. To date, the methods available are too numerous (for an overview see Frank (2006)) as to be exhaustively discussed in the context of this thesis. Hence, only a brief introduction of the most important concepts is given in the following sections.

1.2.4.1 Signal-to-Noise Ratio

The signal-to-noise ratio (SNR) is the ratio between the variance of the signal and the variance of the noise in a given image. This measure is extremely useful in assessing the quality of experimental data and in designing appropriate synthetic data of different quality.

1.2.4.2 Particle picking

The raw data produced by any modern TEM are large (typically 4k by 4k pixels - depending on the detector), noisy CCD images of single molecules (particles). Prior to further processing those particles have to be cut out into typically squared windows with an 30 % increased width compared to the particles diameter. This task may already be challenging as particle contrast may be very low depending on the defocus used (see Section 1.2.3.1), the particle size and the preparation quality. Some semi-automated routines exist (e.g. *Boxer* (Ludtke et al., 1999), *Signature* (Chen and Grigorieff, 2007), *Pika* (Busche, 2009)) to perform this otherwise tedious task of boxing out up to 10^6 individual images. Most of them at some point use local variance detection and Fourier-based cross-correlation functions (for a review see Nicholson and Glaeser (2001) and for a comparison see Zhu et al. (2004)).

1.2.4.3 Correction of the PhCTF and image filtering

As outlined in Section 1.2.3.1 the image signal is convoluted with the PhCTF. Hence, the image intensities will flip in sign after every zero crossing (in the frequency domain) of the PhCTF. In an ideal case, these intensity flips are regular over a whole micrograph (i.e. all

images are convolved with the same PhCTF), however as the PhCTF varies with defocus, instrumental instabilities (especially regarding tilt experiments) or different thicknesses of the vitrified ice may locally change the defocus and hence the CTF. To this end automatic procedures have been developed to locally fit the PhCTF and correct for it by appropriate "phase flipping" (Huang et al., 2003; Mindell and Grigorieff, 2003; Sander et al., 2003). Amplitudes are commonly not adjusted for their relative more difficult fitting (envelope function, amplitude contrast effects etc.) and the danger of (unwanted) raw data manipulation.

Typically, images are filtered prior to further processing. Very low frequencies in the Fourier domain correspond to slowly varying features in the real domain rwhich are unrelated to the particle structure (e.g. variations in the carbon film, ice or stain thickness). High frequencies are relatively more deteriorated by noise, leading to a poor spectral signal to noise ratio (SSNR) which is unwanted for the initial alignment and classification routines. Thus, images are band-pass filtered (A.2) to remove those parts of image information that would otherwise compromise further processing.

1.2.4.4 Alignment

In cryo-EM, alignment is understood as the mathematical operation which minimizes the distance between two images. The mathematical operation commonly is a transformation matrix $\boldsymbol{T}$ given by

$$\boldsymbol{T} = \begin{bmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{bmatrix} \tag{1.10}$$

describing three degrees of freedom: a rotation θ and two translations along x and y, respectively. The distance criterion typically used is the total absolute difference between corresponding pixels in a defined region D of the images. Thus, in a least squares sense, the problem of aligning image f (*reference*) and g (*destination*) can be reformulated as the minimization of

$$\int_{\boldsymbol{u}\in D} |f(\boldsymbol{u}) - g(\boldsymbol{T}\boldsymbol{u})|^2 \mathrm{d}\boldsymbol{u}, \tag{1.11}$$

where D is the region of interest (for example, a disk with diameter d_0), $\boldsymbol{u}$ is the pixel-coordinate vector $[u_x \quad u_y \quad 1]^{\mathrm{T}}$ and $\boldsymbol{T}$ is the transformation matrix defining the rotation and the translations of the image (1.10). It is the optimization algorithm and the type

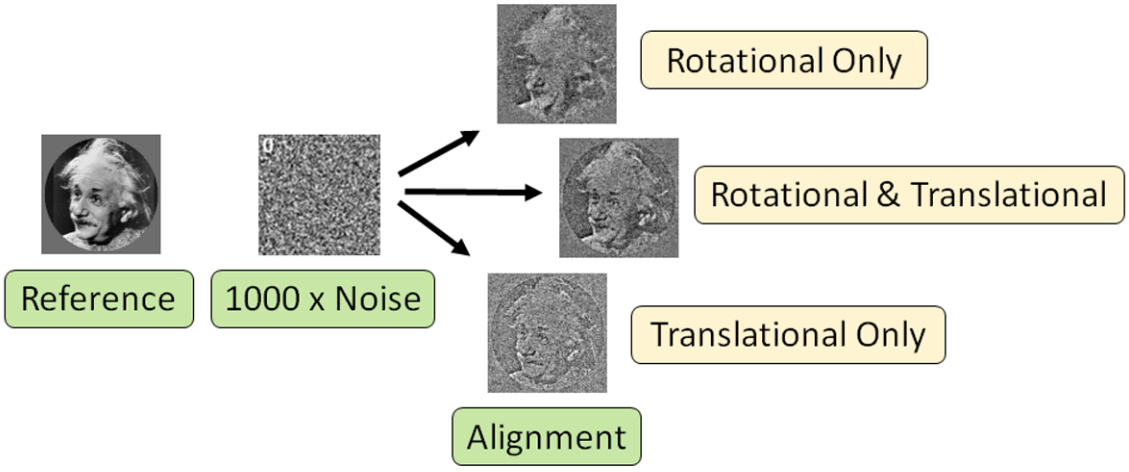

Figure 1.7: The figure shows how strong reference bias can be if non-meaningful content is aligned. In this extreme case 1000 samples of pure noise were aligned using standard algorithms to the Einstein reference. The resultant images to the right show the clearly biased 2D averages of the 1000 images for each alignment protocol.

of interpolation in what currently available methods differ. Conceptually, 2D alignment methods can be divided into two classes: those that exhaustively sample all possible combinations of the three orientation parameters, and those that use either simplifications (by separating the search problem into a translational and a rotational part) (Penczek et al., 1992), or take advantage of invariant image representations (for example see (Schatz, 1992; Frank et al., 1992)). Technically, five main methods can be distinguished:

Direct alignment in real space The two images are directly compared in real space, sampling all possible orientations of the particle view.

Direct alignment using 2D FFT The reference image is rotated and for each rotated version the 2D FFT is computed and stored. In a second step the 2D FFT of the destination image is compared to all references making use of the rapid Fourier-based cross-correlation function (FT-CCF, see A.1.2).

Alignment using the Radon transform The 2D discrete Radon transform (also known as *sinogram*) (A.4) of the reference and the destination image is computed. Briefly, the properties of the generated sinograms combined with appropriate additional 1D and 2D FFTs allow for the determination of the three parameters (θ, x and y) at once. For a detailed description see Lanzavecchia et al. (1996).

Alignment using re-sampling to polar coordinates In this method, the destination image is re-sampled to polar coordinates with respect to selected origin locations within the image frame. The different origin locations reflect the translational degrees of freedom and have to be searched for exhaustively. The reference image is also re-sampled to polar coordinates but is not being shifted. As the polar coordinate re-sampling transforms rotational relationships into translational ones the rotation angle θ can rapidly be found using the FT-CCF.

Autocorrelation based (non-exhaustive) alignment This method is based on the idea of inspecting invariants which are obtained by computing the auto-correlation-function (ACF) (A.1.2). In essence, the relative rotation between reference and destination is obtained by comparing their translation-invariant representations, whereas the relative translation is revealed by comparing their rotational invariant representations. The comparisons are always performed using the fast FT-CCF.

All described methods are proven to deliver good results, however they vary in performance and in accuracy given different input data. It is important to choose the best performing method on the given problem, because even smallest differences both in accuracy and performance (speed) have a huge impact on the global 3D reconstruction process. Inaccuracies ultimately lead to blurring or biasing effects and thus hinder high-resolution determination as do slow performing algorithms which easily scale up to several years of computation on huge ($> 10^6$ images) datasets. For a comparison and evaluation between the outlined methods see for example Joyeux and Penczek (2002).

1.2.4.5 Dimension reduction and classification

Whilst the alignment algorithm may find the optimal transformation to make a pair of images as similar as possible to each other, this by no means implies that also the *content* of the two images has to be similar. Formulated more drastically, even pure noise can be aligned to a reference in an optimal way. And exactly this behavior leads to one of the currently biggest problems in the field known as *reference-bias* or more generally: *model-bias* (see Figure 1.7). To prevent this from happening, it has to be ensured that the images subjected to alignment and subsequent averaging indeed represent the same (or at least very similar) content. Hence, the dataset has to be sorted into subsets prior to alignment or averaging. Obviously, if the classification is *not* invariant against translation and rotation of the individual images, alignment and classification are closely intertwined.

Although methods for invariant classification exist (Schatz, 1992; Tang et al., 2007) they are not commonly used for their relative worse performance regarding lower SNRs. Instead, alignment and classification are performed separately in an iterative manner (see overview Figure 1.1). The classification problem mentioned above can be formulated in a general way:

Given a non-empty set B of individual images $x_i \in B$ the task is to group elements x_i into n subsets $A_i \subseteq B$ with $\prod_{i=1}^{n} A_i = \emptyset$ such that for all subsets the intra-subset similarity

$$S_{xy} := \{x, y | x \in A_i \wedge y \in A_i\} \tag{1.12}$$

gets maximized and the inter-subset similarity

$$\bar{S}_{xy} := \{x, y | x \in A_i \wedge y \in A_j, i \neq j\} \tag{1.13}$$

gets minimized. As the stated problem is inherent to many other disciplines, standard tools like K-means, hierarchical-, spectral-, fuzzy clustering etc. exist and are used in cryo-EM. The *Imagic* (van Heel et al., 1996) software suite for example uses an ascending hierarchical clustering algorithm. However, given the size of standard datasets, direct classification using the full pixel information of each image (which can easily total numbers $> 10^9$) is - even on modern workstations - infeasible to compute. Therefore, a dimensionality reduction step prior to classification is routinely performed. The *Imagic* suite for example uses multivariate statistics based on the *principle-component-analysis* (PCA) (Frank, 2002).

The PCA (or synonymously *Eigenanalysis*) is a statistical method to describe a multidimensional dataset. Orthogonal, principle extensions (also termed *Eigenvectors*, or *Eigenimages*) of the data cloud are found and weighted by their total interimage variance. By means of a coordinate transformation, the original data-points can be projected into a new coordinate system with the Eigenimages as basis vectors. A dimensionality reduction can now be achieved by taking only a limited number of Eigenimages (sorted by decreasing associated variances) into account. The first coordinate axis will thus point in the direction of highest variance of the data set, the second axis into the second highest variance and so on. As a consequence of the dimensionality reduction, the classification can now be performed much faster, but still on data containing information about the main differences within the dataset. Mathematically the Eigenvectors of an image dataset

can be found by populating a matrix $\boldsymbol{X}$, such that each row contains the linearized pixels of each image.

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & . & . & x_{1J} \\ x_{21} & x_{22} & . & . & x_{2J} \\ . & & & & . \\ . & & & & . \\ x_{I1} & x_{I2} & . & . & x_{IJ} \end{pmatrix} \tag{1.14}$$

where x_{ij} represents the j'th pixel of the i'th image. Eigenvectors and Eigenvalues are calculated using the Eigenvector-Eigenvalue equation:

$$\boldsymbol{D}\boldsymbol{u} = \lambda \boldsymbol{u} \tag{1.15}$$

where matrix $\boldsymbol{D}$ is defined as

$$\boldsymbol{D} = (\boldsymbol{X} - \bar{\boldsymbol{X}})'(\boldsymbol{X} - \bar{\boldsymbol{X}}) \tag{1.16}$$

where $\bar{\boldsymbol{X}}$ represents a matrix containing the average image in each row, and λ is a multiplier and $\boldsymbol{D}$ is termed the *covariance matrix*. Equation (1.15), solved by diagonalizing the matrix $\boldsymbol{D}$, has at most p solutions $\{\boldsymbol{u_1}, \boldsymbol{u_1}, ..., \boldsymbol{u_p}$ where $p = \min(I, J)$. The vectors $\boldsymbol{u_i}$ describe the basis vectors of an orthogonal coordinate system in $\mathbb{R}^J$ and are termed *Eigenvectors*, with their associated *Eigenvalues* λ_i (see (1.15)).

Irrespective the preprocessing procedure, a degree of similarity (1.12) between images (or reduced versions of those) has to be determined in the end. The most important similarity measures published and proposed for use in cryo-EM are listed in the following.

Cross Correlation Coefficient Let f_1 and f_2 denote two images each with J discretely sampled pixels (represented as J-dimensional vector $\boldsymbol{r}_j; j = 1, ..., J$) the *cross-correlation coefficient* (CCC) is defined as:

$$\mathrm{CCC} = \frac{\sum_{\mathrm{j}=1}^{\mathrm{J}} (\mathrm{f}_1(\boldsymbol{r}_\mathrm{j}) - \bar{\mathrm{f}}_1)(\mathrm{f}_2(\boldsymbol{r}_\mathrm{j}) - \bar{\mathrm{f}}_2)}{\sqrt{\sum_{\mathrm{j}=1}^{\mathrm{J}} (\mathrm{f}_1(\boldsymbol{r}_\mathrm{j}) - \bar{\mathrm{f}}_1)^2 \sum_{\mathrm{j}=1}^{\mathrm{J}} (\mathrm{f}_2(\boldsymbol{r}_\mathrm{j}) - \bar{\mathrm{f}}_2)^2}} \tag{1.17}$$

where

$$\bar{f}_i = \frac{1}{J} \sum_{j=1}^{J} f_i(\boldsymbol{r}_j); \qquad i = 1, 2 \tag{1.18}$$

The CCC can be regarded as the "cross-variance" between two images.

Differential Phase Residual Let $F_1(\boldsymbol{k})$ and $F_2(\boldsymbol{k})$ denote the 2D Fourier transforms of images f_1 and f_2 respectively, the *differential phase residual* (DPR) (Frank et al., 1981) is defined as:

$$\mathrm{DPR(k)} = \sqrt{\frac{\sum_{\mathrm{k}}[\Delta\varphi_{1,2}(\boldsymbol{k})]^2 \cdot (|\mathrm{F}_1(\boldsymbol{k})| + |\mathrm{F}_2(\boldsymbol{k})|)}{\sum_{\mathrm{k}}(|\mathrm{F}_1(\boldsymbol{k})| + |\mathrm{F}_2(\boldsymbol{k})|)}} \tag{1.19}$$

where

$$\Delta\varphi_{1,2}(\boldsymbol{k}) = \arg(F_2(\boldsymbol{k})) - \arg(F_1(\boldsymbol{k})) \tag{1.20}$$

and $\boldsymbol{k} = [k_x \quad k_y]^{\mathrm{T}}$ is the discretely sampled spatial frequency. The sums are computed over Fourier components falling within concentric rings of spatial frequency radii $k = |\boldsymbol{k}|$. Hence, unlike the CCC which gives a single scalar value, the DPR evaluates to a function of k. In order to obtain a single figure of quality, the spatial frequency where the DPR equals $\frac{\pi}{4}$ (often also termed k_{45}) is used. The DPR can be understood as the root mean square (r.m.s) deviation of the phase difference between two Fourier transforms, weighted by the average Fourier amplitude.

Fourier ring correlation The *Fourier ring correlation* (FRC) (Saxton and Baumeister, 1982; van Heel and Stöffler-Meilicke, 1985) is defined as:

$$\mathrm{FRC(k)} = \frac{\mathrm{Re}\{\sum_{\mathrm{k}} \mathrm{F}_1(\boldsymbol{k})\mathrm{F}_2^*(\boldsymbol{k})\}}{\sqrt{\sum_{\mathrm{k}} |\mathrm{F}_1(\boldsymbol{k})|^2 \sum_{\mathrm{k}} |\mathrm{F}_2(\boldsymbol{k})|^2}} \tag{1.21}$$

The meanings of k, $F_1(\boldsymbol{k})$, and $F_2(\boldsymbol{k})$ are the same as in equation 1.19 and again all summations are over specific rings in Fourier space. If the images are real, the corresponding Fourier transforms exhibit hermitian-symmetry (also called *Friedel symmetry*), hence the phase of the complex conjugated product in the numerator will add up to zero whilst scanning one concentric ring. The FRC is thus a real cross-correlation coefficient, normalized by the square root of the power in the rings in each of the transforms. Like the DPR the FRC evaluates to a function of spatial frequency k. Different criteria have been described to retrieve a single quality figure from the FRC, such as the 2σ (van Heel and Stöffler-Meilicke, 1985), the 3σ (Orlova et al., 1997), 5σ (Radermacher, 1988; Radermacher et al., 2001)or simply the 0.5 value (Böttcher et al., 1997) which is most frequently used.

Q-Factor In contrast to all similarity measures mentioned so far, the *Q-Factor* (van Heel and Hollenberg, 1980; Kessel et al., 1985) can be used to evaluate more than two images at once and is defined as:

$$\mathrm{Q}(\boldsymbol{k}) = \frac{|\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{F}_{\mathrm{i}}(\boldsymbol{k})|}{\sum_{\mathrm{i}=1}^{\mathrm{N}} |\mathrm{F}_{\mathrm{i}}(\boldsymbol{k})|} \tag{1.22}$$

where $F_i(\boldsymbol{k})$ denotes the 2D Fourier transform of image i with $\boldsymbol{k}$ being the spatial frequency. If the complex amplitudes of all images at a fixed spatial frequency are drawn into the diagram, the Q-Factor describes the ratio between the length of the sum vector and the length of the total pathway of the vectors contributing to it. Hence, the Q-Factor evaluates to a number between zero (worst similarity) and one (identical images) for each specific spatial frequency. This furthermore implies that the Q-Factor is *not* weighted by amplitude powers. The expectation value for pure noise is $Q(\boldsymbol{k}) = \frac{1}{\sqrt{N}}$, which can be derived in equivalence to the random wandering of a particle under Brownian motion (Einstein equation). Images are commonly regarded as having significant similarity if their score is three times higher than the corresponding pure noise expectation (i.e. $\frac{3}{\sqrt{N}}$).

Spectral signal-to-noise ratio Like the Q-Factor the *spectral signal-to-noise ratio* (SSNR) (Unser et al., 1987) can be used to evaluate any number of images at once. The SSNR is defined as the ratio of the estimated normalized signal energy $\hat{\sigma}^2_{k_s}$ and the estimated noise variance $\hat{\sigma}^2_{k_n}$ in a local region of Fourier space (typically annuli with distinct spatial frequency radii $k = |\boldsymbol{k}|$):

$$\mathrm{SSNR(k)} = \frac{\mathrm{N}\hat{\sigma}^2_{\mathrm{k_s}}}{\hat{\sigma}^2_{\mathrm{k_n}}} \tag{1.23}$$

where

$$\hat{\sigma}^2_{k_s} = \frac{1}{K}\sum_{k}^{K} |\bar{F}(\boldsymbol{k})|^2 \tag{1.24}$$

$$\hat{\sigma}^2_{k_n} = \frac{\sum_k^K \sum_{i=1}^N |F_i(\boldsymbol{k}) - \bar{F}(\boldsymbol{k})|^2}{K(N-1)} \tag{1.25}$$

and

$$\bar{F}(\boldsymbol{k}) = \frac{1}{N}\sum_{i=1}^{N} F_i(\boldsymbol{k}) \tag{1.26}$$

with N denoting the total number of images evaluated and K denoting the number of Fourier components per ring. Hence, the estimate for the normalized signal energy $\hat{\sigma}^2_{\boldsymbol{k}_s}$ is computed from the Fourier transform of the averaged image.

Whilst similarity measures for comparing more than two images at a time are proposed (see (1.22) and (1.23)), they are almost never used during optimization (a notable exception being the program *Frealign* (Grigorieff, 2007)). DPR, FRC, Q-Factor and also SSNR are rather used for final quality assessment and not as objective function. It is the CCC

distance criterion which is most frequently used during alignment and classification. Albeit being able to be rapidly computed, the CCC is a relatively rough similarity measure, strongly varying with content independent features of the data. Within the presented work the effect of using similarity measures other than the CCC is exploited and detailed in later sections.

1.2.4.6 3D Reconstruction

The problem of reconstructing a 3D structure from the measured 2D projections finds its analogy in the phase-problem encountered in X-ray crystallography. Both experimental methods deliver only indirect structural information such that the experimental data can not be used to simply compute the corresponding 3D object. Albeit not missing phases, the information lost during cryo-EM is the 3D orientation (commonly described in Euler angles) of each recorded 2D projection, which will be termed "orientation-problem" throughout this thesis.

Having once assigned correct angles to each projection, computing the 3D structure is straight forward (see Figure 1.8), as is for X-ray crystallography if phases are assigned to the structure factors. Like in crystallography also in cryo-EM several methods exist to solve the orientation-problem and the best choice is made dependent on the properties of the current experiment. A short overview is given below:

1. If a similar 3D model to the structure under investigation is already available, this model can be used as a *molecular-replacement*. Projections with known orientations are generated from the molecular-replacement structure *in-silico*. Those projections are then used as references for the experimental projections during alignment (*projection-matching*). As already discussed this procedure can lead to substantial model-bias if not used extremely cautiously (see Figure 1.7).

2. If no startup 3D model is available, the orientation-problem can theoretical be solved by utilizing the *Fourier slice-theorem* which states that the Fourier transforms of different pairs of projections resulting from the same 3D object should share at least one common line. Thus, by finding those common lines the relative orientations of the projections can be determined (Crowther et al., 1970). However, in practice several factors like uneven distribution of the projection directions, the amount of noise in the data, and sample heterogeneity limit the success of this method.

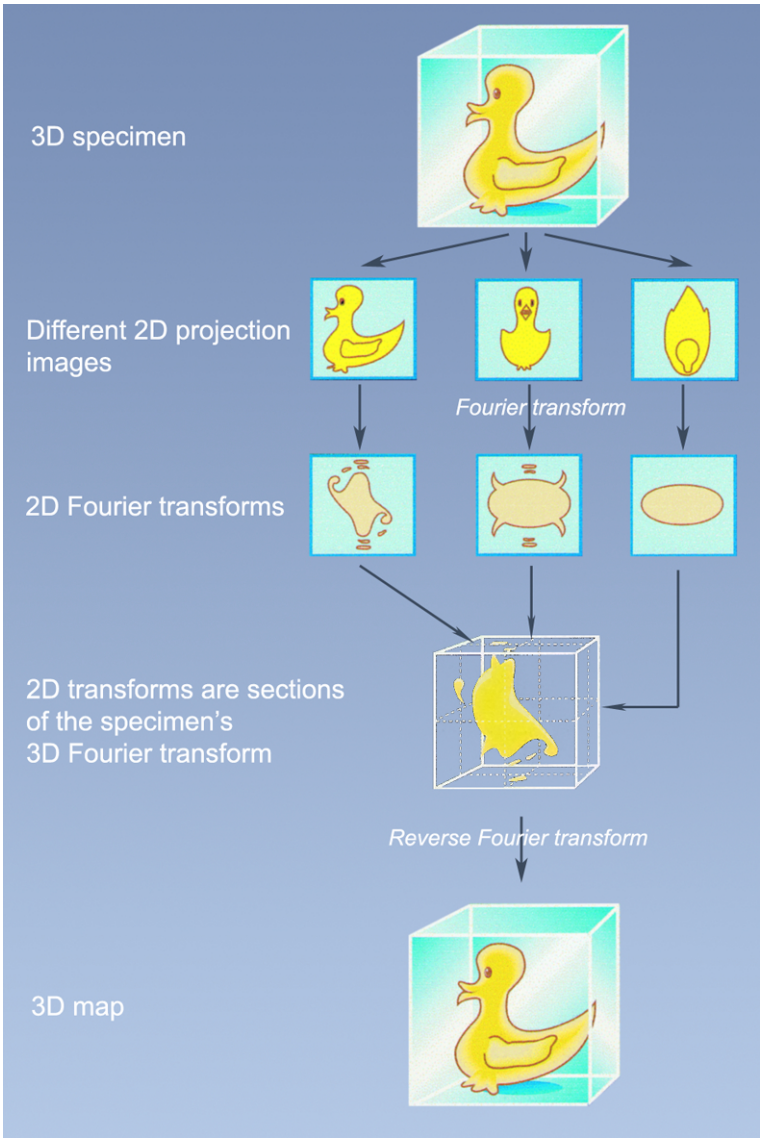

Figure 1.8: Schematic diagram to illustrate the principle of 3D reconstruction. Each projected 2D image, as obtained from the micrograph and after CTF correction and averaging (through classification and alignment) is subjected to a 2D Fourier transform. Following the so-called *projection-theorem* (see A.3) those transforms represent central sections in the 3D Fourier transform of the underlying 3D object. Hence, after accumulation enough sections from different views a 3D map of the structure can be calculated by a 3D inverse Fourier transform.

3. Another method for de-novo structure determination is the so-called *random-conical tilt* method (Radermacher et al., 1987). The basic idea is to take at least two images

of the same specimen detail under different tilt angles. Commonly an untilted (i.e. specimen plane perpendicular to electron beam) and another strongly tilted (by rotating the whole specimen-holder in the beam up to $\pm 60^{\circ}$) image is collected. This additional information can be used to solve the orientation problem of an unknown structure. Drawbacks of this method are the increased radiation damage (due to multiple image exposure) and the complicated image processing caused by merging of the datasets (e.g. pairwise particle selection, precise tilt-geometry determination, relative scaling etc.).

After assignment of initial projection directions by either of the above mentioned methods, the resultant 3D (Coulomb-)density function of the reconstructed object is refined in an iterative procedure in which reconstruction steps are alternated with estimation and re-evaluation of the projection directions (Penczek et al., 1994; Grigorieff, 2007). Hence, the reconstruction algorithm has to be fast and accurate in order to prevent propagation of errors.

The reconstruction can be achieved by using several general strategies. One strategy is to algebraically solve a linear system of equations built up from the individual rays (i.e. the discrete *3D ray transform*) of the 3D object. This problem commonly is too huge for direct matrix inversion, hence iterative techniques such as the *simultaneous iterative reconstruction technique* (SIRT) (Gilbert, 1972; Penczek et al., 1992) or the *algebraic reconstruction technique* (ART) (Marabini et al., 1998) are used. Methods of this class are very accurate but also very slow. Another strategy is to use the inversion of the 3D ray transform. The widely used *(weighted) filtered back-projection methods* (Radermacher, 1992) belong to this class. Other methods of the same strategy are the so-called *direct Fourier methods*, which exploit the projection theorem by directly reconstructing in Fourier space and finally reversing the 3D reconstruction to result in a real space density map. Those methods require sophisticated Fourier interpolation schemes as non-uniformly sampled Fourier grids are necessary to be computed during this technique. A recent approach to this problem was achieved by Penczek et al. (2004) making use of a *gridding-based direct Fourier reconstruction* (GDFR), which seems to be the superior method (both in speed and accuracy) in comparison to all other existing ones.

1.2.4.7 Validation

Literature focussing on quality control of cryo-EM structures is hard to find. Admittedly, individual methods, such as the alignment or the reconstruction process (see Joyeux and Penczek (2002) or Penczek et al. (2004)) are individually validated for consistency and reproducibility. However, assessment of the global influences and the propagation of errors as for example introduced through the iterative reference-based alignment and the separation of the latter from the classification process, is - at least to the authors knowledge - never done systematically. The obvious reason for the shortage in validation tools roots in the very poor quality of the experimental raw data. Well proven measures, such as the R_{free} (Brünger, 1992) known from crystallography can not be applied to cryo-EM as the statistical significance per raw data is much to low. In other words, performing a cross-validation against noisy raw data results in noisy (meaningless) accuracy figures.

1.3 Scientific software development

1.3.1 Management and storage of large datasets

Scientists working in a laboratory know about the importance to keep track about performed experiments including all related parameters (e.g. chemicals, concentrations, temperature etc.). Frequently, this information is archived in a lab-book which may later be consulted for repetition of experiments or as basis for the design of new experimental strategies. The exact and complete storage of scientific data forms a major part of a set of general guidelines termed “good laboratory practice” (GLP). Similarly, computer-based experiments like the image-processing part of cryo-EM have to be documented. This however is non-trivial thanks to the huge amount of data involved. Furthermore some requirements special to digital data storage should be fulfilled: i) Data should be query able and saved consistently ii) Data should be portable (different operating systems) iii) Data should be easy to archive (e.g. to be concisely stored on tape-based backup systems) iv) Data reading and writing should be fast.

The type of data to be stored in cryo-EM are images, corresponding meta data and the history of manipulation routines performed on those images. Technically, only two different systems are capable of fulfilling most of the mentioned requirements. This is at the one hand a relational database (DB) and on the other hand a sophisticated, in-

ternally structured file. An example of a DB based solution to cryo-EM can be found by Liang et al. (2002). A file based approach to multi-data storage is not easily found in cryo-EM but is very popular and frequently used in protein X-ray crystallography. Crystallographic data often is collected in a so-called MTZ-file, which can be regarded as an binary, hierarchical, fixed-format data and metadata storage system[2]. However, most software packages in cryo-EM use more or less flat-file based approaches, featuring no, or only limited amount of history tracking and enforced naming and storing consistency. Consequently, each user stores files in different folders among different operating systems on different physical places (local or server site) with individual naming and sorting convention. This most often wreaks havoc with increased time and number of projects.

Addressing the above mentioned shortcomings a new file format for project based data management was developed as part of this thesis. The file stores image data and all relevant meta data (such as image headers, manipulation parameters, internal linkage and history) in a binary, hierarchical format and is based on the HDF (hierarchical data format) framework[3]. HDF was chosen for its long history, proven stability and support. A prominent user of this framework for example is the NASA with its Earth Observing System (EOS), the primary data repository for understanding global climate change. Over the 15 year lifetime of this project NASA will store 15 petabytes of data in HDF, demonstrating the data management capabilities of this framework.

1.3.2 Parallel programming

It is a general trend in scientific and engineering disciplines that with the increased sophistication in instrumentation also more data per time unit are produced. Hence, to keep up with computational downstream processing, either the algorithms have to perform more efficiently or the underlying hardware has to boost the execution time accordingly. Whilst the former approach most often is infeasible, hardware manufactures where quite successful in doing the job over the past decades. Throughout the entire 1990s the CPU (central processing unit) processing power doubled almost every 18 month. However, recently this pace can not be kept up anymore. Physical limitations such as heat generation and the very small sizes (lithographic scattering limits) prevent the CPU from getting even faster. Consequently, if a single unit can not be made faster, the idea is to use more of them in

[2] See http://www.ccp4.ac.uk/html/mtzformat.html#fileformat for more details.

[3] http://www.hdfgroup.org/

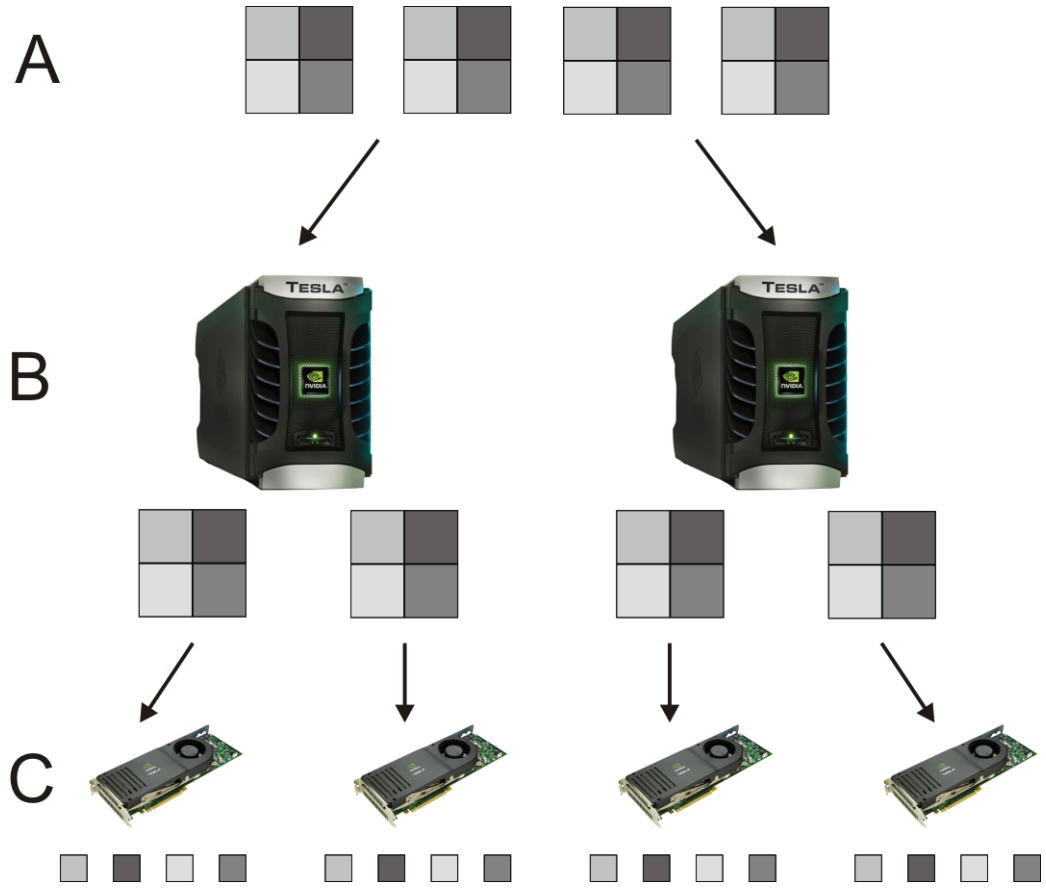

Figure 1.9: This figure schematically illustrates different levels on which parallel programming may be applied. **A** Shows the input data to be computed (here 2D images with four pixels each). **B** On a first level of parallelization the data may be split amongst physically different computing units. **C** The data is furthermore split (still as whole image entities) and distributed among available GPUs. The highly optimized GPUs are able to break the parallelization down to the individual pixels of the images. Thus if parallelization overhead is neglected, the processing time of the illustrated example that execution time of a single pixel-operation.

parallel. At the time of writing even a moderate-priced desktop PC features at least two, but sometimes also up to eight CPUs. Furthermore, since very recently a new technology which allows for massively parallel computing on the GPU (graphics computing unit) got available (see Section 1.3.2.2). Hence, a single desktop PC (possessing a descent GPU in the optimal case) is already able to theoretically massively speed up computations through parallelization (Schmeisser et al., 2009).

Unfortunately, algorithms that are not explicitly designed to make use of more than one computing unit will be unaffected in speed, irrespective the potential power of the underlying hardware. Admittedly, it is a tedious job to rewrite already existent algorithms to make use of this new kind of power. But when designing new algorithms with potentially very long execution times or rapidly increasing input data sizes, it is a must to think about a design which scales with the number of parallel computing units available. Method development for the single particle cryo-EM technique is an excellent example of such a situation. The data load is very heavy (routinely several GB and increasing) and algorithms get increasingly complex with the improvement of overall accuracy. Thus, new methods not only have to be scientifically accurate, but also - and with the same importance - technically feasible to be computed in a descent time. New developments regarding both aspects are subject to this thesis and will be presented in later sections.

Parallelism may be applied at different levels of granularity as is illustrated in Figure 1.9. It is thus important to divide the computing problem in a way such that the overhead introduced by the additional mechanisms needed for the very act of parallelization gets reduced.

1.3.2.1 Farming

The uppermost level of parallelism is achieved by scaling out the computational problem to many physical computers, also called nodes (compare Figure 1.9B). This technique frequently referred to as farming can be further subdivided according to the properties of the individual nodes.

If a fixed number of nodes with exactly the same hardware and thus the same computing power are available, they can be connected to a dedicated, homogeneous cluster. Internode communication is established via message passing through high speed point-to-point network connections. A de facto standard for such a communication is the language independent protocol MPI (message passing interface) (Park and Hariri, 1997), which is at the foundation of many software solutions aimed to facilitate dedicated parallel programming.

On the other extreme, if the nodes participating in the parallel computation are of different hardware, they can not be trusted to be available and are only connected to single master node but not between each other. This arrangement consequently is termed a non-

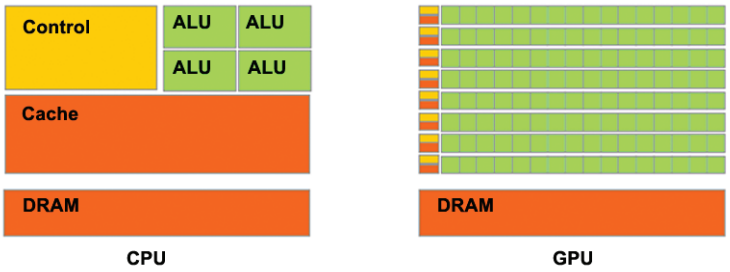

Figure 1.10: The GPU consists of more artihemtic logic units (ALUs) than a CPU, but has relatively less control and cache capabilities.

dedicated, heterogeneous environment. Being much less demanding in hardware infrastructure, even the Internet can be used as network and the nodes are provided voluntary by any Internet user. Such frameworks exist and are also used for scientific purposes, the most prominent examples being SETI@HOME or Folding@HOME using the BOINC framework (Anderson, 2004). A similar system (albeit using a smaller network) especially designed for the needs of cryo-EM was designed by Schmeisser (2009), which may also serve as reference for more detailed information on farming.

1.3.2.2 GPU programming

GPU programming is at the finest level of parallel programming, allowing for data-element wise multi-threading (compare Figure 1.9 C). GPUs are mainly build into specialized devices (e.g. graphic boards) but can sometimes also be found directly as part of the motherboard. Originally designed for the purpose of rendering pixels onto a screen, GPUs exhibit an inherent parallel hardware design. Many more transistors (computer lingo: *arithmetic logic unit*, ALU) are devoted for data processing rather than data caching and flow control, as schematically illustrated in Figure 1.10. Hence, the GPU is especially well suited for data parallel programming (i.e. the same program is executed on many data elements in parallel) with high arithmetic density[4]. In the context of cryo-EM, a typical data element to be processed in parallel (in a so-called *thread*) is a pixel or voxel of a 2D or 3D image, respectively. Relating back to the concept of granularity, executing an individual thread for each data-element reflects the finest level of concurrency theoretically possible and allows for a maximum degree of scaling under varying numbers of parallel processing units.

[4] Arithmetic density describes the ratio of arithmetic operations to memory operations

Prior to GPU computing, the data has to be transferred from CPU RAM (random access memory) to GPU RAM. This is done via direct memory access (DMA) which is controlled by a so-called DMA-Controller featuring an individual BUS system and thus is decoupled from the CPU. General purpose GPU programming became more interesting only very recently, the reasons being manifold:

i) Memory transfer still is the relatively slowest operation but an unavoidable overhead for GPU programming. Former graphic boards were very limited in dedicated memory (GPU-RAM) such that arithmetically dense programming was virtually not possible. Recent boards feature up to 4 GB dedicated memory (e.g. NVidias Tesla 10 Series), hence reducing the need of memory transfer.

ii) For a long time, writing GPU code was merely like "hacking" the graphic board to compute custom problems. Only with the advent of completely new designed and well documented specific programming languages for GPUs, it is possible to write code in a similarly convenient way as is possible for "normal" CPU code.

iii) Recent graphic boards feature a much for flexible design for on-board memory access. All ALUs are allowed to perform reading from (*gather*) and writing to (*scatter*) any memory location (Figure 1.11). This was not possible on older graphic boards, but is needed for a CPU-like programming flexibility.

Unfortunately, the two main hardware vendors for graphic boards (ATI and NVidia) are developing mutually incompatible product solutions up to now. However, a new open, royalty-free standard for cross-platform parallel programming of modern processors (as found in personal computers, servers or even mobile devices) called "OpenCL" (Khronos-Group, 2008) promises to overcome these limitations in future. For the GPU based algorithms described in this thesis the solution offered by NVidia, the so-called CUDA (compute unified device architecture) language was used. A very brief introduction to CUDA will be given withing the next section.

The CUDA programming language In CUDA lingo, anything related to the main CPU is termed "host" whereas the GPU is regarded to as "device", respectively. These two keywords will be used throughout this thesis from now on.

The CUDA language can be regarded as an extension to the well known "C" language.

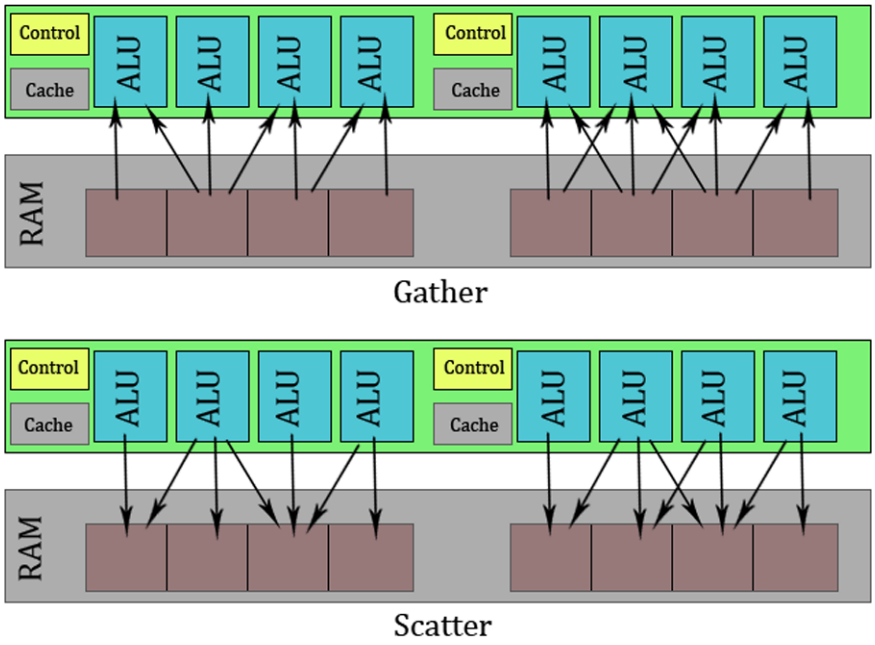

Figure 1.11: CUDA scatter and gather operations. Reproduced with permission from Busche (2009).

Hence, only minor syntactical extensions have to be learned by a C-programmer to be able to write device code. Additionally, CUDA code is easily integrated into an object-oriented C++ design, which is a great advantage for the work presented here, as all host code is written in C++. The integration is facilitated by a specialized compiler (NVCC - NVidia C-Compiler) that is part of the CUDA package available as a free download from the Internet. The NVCC generates C conform object files, which can be linked to any C/C++ code via standard C/C++ compilers. The most important concept specific to CUDA programming is the - in comparison to "normal" CPU threads - much extended thread management system. Threads are batched into blocks, and blocks are batched into grids. Each thread possess a unique ID (thread-ID) within a block also identified by a unique ID (block-ID). All threads within a block can cooperate together by efficiently sharing data through some fast shared memory. All threads within a block have to be synchronized by the programmer to assure coordinated data access. For memory reasons, each block can only have a limited number of threads, such that multiple blocks are created and arranged in a grid. Communication between threads of different blocks is not possible. A parallel

computation has to be launched through calling a specialized function termed kernel with a defined number of same-sized blocks and threads. Thus, the total number of parallel operations (threads) equals the number of blocks multiplied with the number of threads per block.

CUDA example - inverting image contrast Listing 1.1 shows a small example program, which inverts the contrast of an image. The function to be called from outside is a pure C-function (line 14), which may be easily embedded into a C++ framework. The first thing to do is to allocate memory on the device (lines 16-17). Next, the host data (here the array of an image) has to be copied to the device (line 19). Line 21 actually launches the GPU part of the computation with 64 blocks and 128 threads as indicated by the special $<<< 64, 128 >>>$ syntax.

The code to be executed on the device is implemented by the function starting with the CUDA directive `__global__` (line 5) which just means that this function can be called by the host, i.e. represents a kernel (`__device__` in contrast would indicated a function only callable by the device). The current thread index is calculated utilizing in-build variables, and the total number of threads is stored in `gridSize` (line 6,7). The while loop ensures processing of all array elements (pixels) even if less threads than pixels are available. This strategy ensures full usage of the device capability and renders the code scalable with respect to different GPU boards. The actual contrast inversion happens in line 9 by negating the current pixel.

Information about more complex issues of CUDA programming may be found in NVidia (2009).

Listing 1.1: CUDA example code. The program shown inverts the contrast of an image.

```
1 #include <cuda_runtime_api.h>
2
3 extern "C" void invertContrast(float* arrayH, int size);
4
5 __global__ kernel(float* arrayH, int size) {
6   int idx = threadIdx.x + blockDim.x * blockIdx.x;
7   int gridSize = blockDim.x * gridDim.x;
8   while (idx < size ) {
9     arrayH[idx] *= -1.0;
10    idx += gridSize;
11  }
```

```
12 }
13
14 extern "C" void invertContrast(float* arrayH, int size) {
15   // Allocate device memory
16   float* arrayD = 0;
17   cudaMalloc((void**)&arrayD, sizeof(float)*size);
18   // Copy image data from host to device
19   cudaMemcpy(arrayD, arrayH, sizeof(float)*size, cudaMemcpyHostToDvice);
20   // Launch the kernel (64 blocks with 128 threads each)
21   kernel<<<64,128>>>(arrayD, size);
22   // Copy data back from device to host
23   cudaMemcpy(arrayH, arrayD, sizeof(float)*size, cudaMemcpyDeviceToHost);
24 }
```

1.4 Aim of the work

To date, the bottleneck on the way to high resolution cryo-EM structures is not the amount of raw experimental data that has to be produced. It is rather the image manipulation process, which has to be capable of retrieving the statistically available high resolution information from the huge dataset. For such a process to be successful, it has to run fully automatically, user-interaction and bias-free and it has to robustly increase the classification and alignment quality under increasing amounts of available raw data. That implies the identification and removal of individual raw images that are not of sufficient quality to help improving the overall accuracy.

Currently available tools are already very powerful, but are believed to still not facilitate the ultimate high resolution information available in state-of-the-art cryo-EM datasets. Reasons for that may be manifold, but it is fair enough to formulate them as hypotheses underlying the presented work:

- Most of the currently available methods use a pair-wise distance function to judge image content similarity in the end. Under the present noise in the data, those measures are believed to have a huge error associated to their result, limiting the overall performance of any sophisticated manipulation based on those measures from the very beginning on. It is the hypothesis that similarity should always be measured for several images at once, thus statistically reducing the associated error to this measurement.

- Almost all current methods drive a supervised optimization strategy. Critical parameters (such as number and type of references for alignment, number of classes and Eigen-images to be used for classification etc.) are determined upon subjective human decisions. This is believed to introduce bias and to reduce reproducibility of the whole process. It is another hypothesis of this work, that subjective human interaction should be reduced to an absolute minimum and hence processes be designed to run in an unsupervised fashion.

- As outlined in Section 1.2.4.5 the *de-facto* standard for image pre-processing is to separate the alignment from the classification process and to overcome their interdependency though an iterative approach. This process is believed to introduce uncontrollable model-bias and may lead to albeit high in resolution but wrong in geometry 3D structures. It is the third hypothesis of this work, that in order to achieve accurate high-resolution 3D structures all processing routines have to be absolutely reference/bias-free and that the alignment and classification problem should be treated as a combined problem and solved at once.

In order to accept or reject any of the hypotheses mentioned above a new similarity measure was developed that is able to evaluate an arbitrary number of images at once. Furthermore, a new image processing strategy (named *Crystalign*) is introduced within this thesis which will run in an unsupervised fashion and is aimed to combine classification and alignment to a single reference-free optimization process. The efforts underlying this thesis thus are focussing on an improvement of the overall accuracy and resolution of cryo-EM based 3D structural investigation of biological macromolecules.

Chapter 2

Materials and Methods

2.1 Code generation – hard- and software used

Software development and all described local computations were mainly performed on a desktop PC equipped with an AMD Phenom™ 9650 Quad-Core Processor with 8 GB RAM, a NVidia GeForce 9800 GTX+ (500 MB RAM) graphics board and a NVidia Tesla C870 (1.5 GB RAM) compute board.

Coding was mainly done using *Emacs* or (more rarely) *Microsoft Visual Studio*. Software versioning was performed using *Subversion* running on a web-server and allowing for team-oriented programming. For quality control and unit testing the *CppUnit* framework was used. Automatic generation of documentation pages (preferably in html format) was performed using the *Doxygen* software. The programming language used throughout all described algorithms was *C++* or *CUDA* for code running on the host or device, respectively. Linux host-code compilation was aided by the *CMake* (Cross-Platform-Make) meta-language and using the *GNU-C++* compiler and linker. Windows builds were aided by the tools offered within the *Microsoft Visual Studio* software. Building and linking under windows was performed using the *Microsoft C compiler/linker*. Device code was compiled separately using the *NVidia C Compiler* (NVCC). For tasks such as multi-threading, reading/writing of various image formats, fast Fourier transform computation, external freely available libraries were used, which are summerized in Table 2.1.

Table 2.1: Listing of all external libraries used

Library	Reference	Purpose
boost	www.boost.org	Extensions to the C++ STL
hdf	www.hdfgroup.org	File-based data management
freeImage	freeimage.sourceforge.net	Image reading/writing of standard file formats
cppUnit	cppunit.sourceforge.net	Unit-testing framework
fftw, *cufft*	www.fftw.org	Fast Fourier transform (CPU, GPU)
openMpi	www.open-mpi.org	Parallel message passing interface
qt	www.qtsoftware.com	Cross-platform GUI development

2.2 Image processing framework

The post-processing of raw images as obtained from a cryo-EM experiment is a complex process, involving various different manipulations such as normalization, filtering, alignment, classification and reconstruction operations. In order to streamline this computationally expensive process and to avoid unnecessary and error-prone file conversions needed for the various expert software modules available, an in-house software pipeline (termed *Cow-Framework*) is being developed under the administration of Prof. H. Stark. The core of this pipeline (referred to as *back-end* from now on) is formed by a modular, object-oriented C++ library, which has been already available at the beginning of this work, but heavily extended and redesigned throughout. To be useful, the flexibility and modularity of the back-end has to be reflected in an user-friendly interface (referred to as *front-end* from now on). Consequently, next to a standard console application a graphical user interface was designed, featuring "visual programming" and intuitive project management for maximum usability.

2.2.1 Back-end

The design goals of the back-end may be summarized into six points:

1. Maximum possible performance for image manipulation algorithms

2. Uniform interfaces for core functionality (such as parameter handling, input/output, image operations) allowing a "Plug-and-Play" architecture

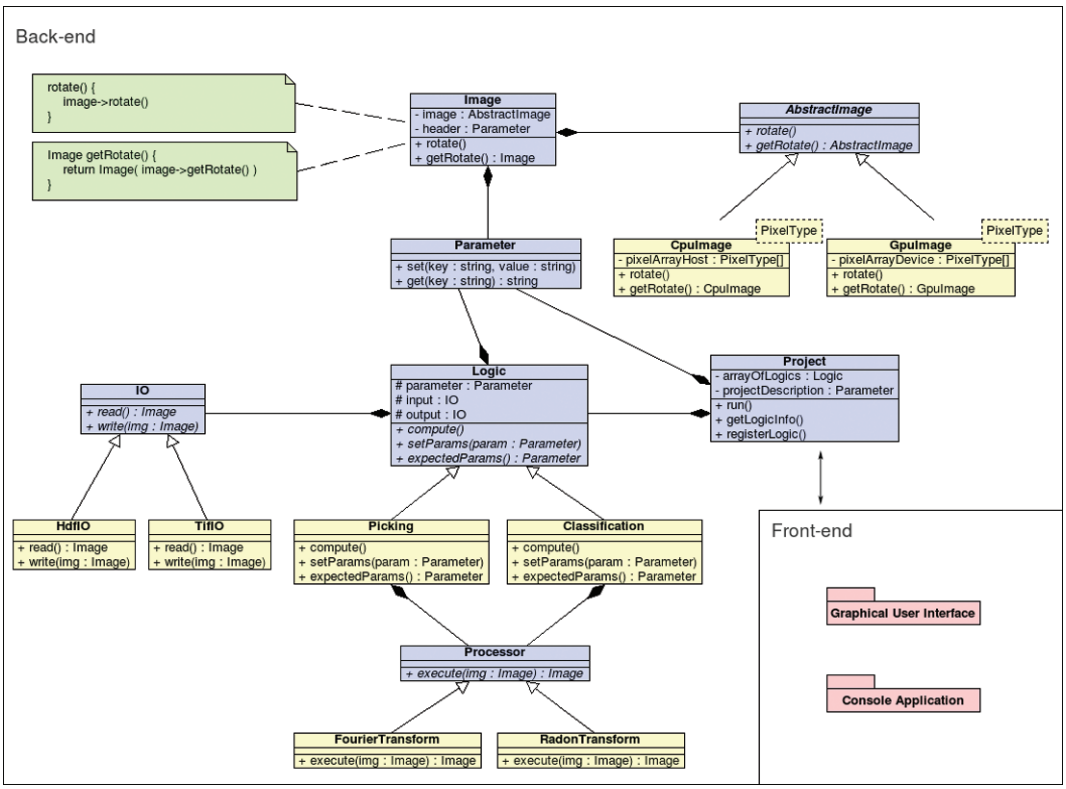

Figure 2.1: Simplified UML (Alhir, 1998) diagram illustrating the relationships of the main program modules. Connections starting with a diamond represent "has a" relationships (aggregation), arrows indicate "is a" relationships (inheritance). Classes in blue represent fixed core modules, whereas yellow classes are specialized implementations of a given interface and may be added in a Plug-and-Play manner. Instances of the *Image* class typically represent individual cryo-EM images, and supply some basic functionality. The image functionality is fully abstracted from the underlying type (i.e. could be GPU located) by making use of the so-called "State" design pattern (Gamma et al., 2005). Instances derived from the *Logic* class describe manipulations which are performed on multiple images. *Logics* may have several input and output channels. *Processors* in contrast describe single image manipulations, hence read one image in and write the processed version out. Typically, a logic makes use of one or more processors (aggregation). Derived instances of the *IO* base class implement image reading and writing functionality for various formats. Newly added back-end modules (yellow) will immediately be available in the front-end application, as the communication interface (the *Project* class) abstracts all back-end functionality.

3. Simple set up of individual tests for new functionality, which integrate into a unified testing framework

4. Consistent coding and documentation conventions
5. Project based data management, storage and archival
6. Cross-platform portability

As outlined above, the computational load for cryo-EM image processing is very high. Specific operations such as image-image alignment may be necessary to compute hundreds of thousands times. Consequently, a maximum performance of highly repetitive functions has to be strived for. However, speed optimized code typically exhibits harder readability, especially for non-authors. It is the very concept of object-oriented programming to hide those pieces of code behind clean, well defined interfaces. Other programmers just need to know about the interfaces in order to make use of the existing highly optimized code.

In case of the Cow-Framework this clean interface is mainly guaranteed by the *Parameter* object. The *Parameter* object can be regarded as a dictionary of key/value pairs where the key is represented as a string (e.g. "numberOfIterations") and the value can be of any type[1]. Each dictionary entry must have a conventionalized description connected to it, again expressed as a list of key/value pairs, including default values, parameter description, valid ranges etc. . The correct setup of the interfaces is ensured by the technique of inheriting (in other words: detailing) base classes (see the blue boxes in Figure 2.1) defining all important function signatures in an abstracted way. Thus, once a new functionality is added to the back-end (be it a new IO channel for jpg reading, a new processor for filtering, or a new logic for alignment) it will immediately be available throughout the whole software system and can be directly used by back-end programmers or front-end users (e.g. via the visual programing interface). This technique is frequently known as *Plug-and-Play* architecture and has the big advantage of enabling new programmers to be rapidly productive in developing new functionality as almost no knowledge or learning of the tiring details of the underlying library is needed. Such a strategy ensures continual improvement and completion of the software pipeline even under frequently varying personal conditions as is the common case in academic institutions.

Even more important than adding new functionality is testing it and ensuring proper behavior under all possible conditions. This is especially true as biological conclusions

[1] This is convenient as in C++ all data types such as integral and floating point numbers, characters etc. need to be differentiated

are drawn and eventually published *not* based on the original experimental data but on heavily computationally manipulated versions of the latter. To this end a unit-test framework was developed which exploits the same convenience in usability as described above. At any time the test can be run and the complete functionality of the whole pipeline be tested and ensured for proper operation. Especially in team driven projects, an individual programmer can not easily assure that a modification to a shared, widely used functionality will not affect the code generated by others. Running the unit test framework with all tests passed however will.

As outlined in the introduction (see Section 1.3.1) a novel data file for storing project-based information was developed within the HDF framework. Figure 2.2 illustrates the internal file organization, which is not presented to the end-user in that detail. As can be seen from the figure, the internal structure is quite similar to that of a linux file system. The blue rectangles can be regarded as directory analogs, the red data symbols would consequently correspond to ordinary files. Staying with the original HDF terminology, folders will be referred to as *groups* and files will be termed *datasets* from now on. The green dog-eared boxes (termed *attribute-lists*) are a special feature of the HDF technology and have no analog in the linux file system. Attribute lists can be associated to both, groups and datasets and are intended as commentary fields organized as a list of key/value entries, much like the above described *Parameter* object as is used in the Cow-Framework.

In fact, a function was designed that takes a *Parameter* object as an argument and translates it one-to-one into a HDF attribute-list. Each image header, for example, is represented by a *Parameter* object within the Cow-Framework (every *Image* has a *Parameter* to store header information, see Figure 2.1), which after writing to the HDF file will be converted to an attribute-list associated to the corresponding image meta dataset (compare M1-3 from Figure 2.2 - only one attribute-list is shown for the sake of clarity).

Starting at the top, the HDF file itself contains some global project information such as authorship, experimental setup, project date/time etc. stored as an attribute-list (top green box in Figure 2.2). On the next hierarchy level, the file is separated into two groups: i) a *Data* group which sequentially stores all data produced through execution of a *Logic* (i.e. a multi-image manipulation routine) and ii) a *Collection* group which only stores lightweight internal links (illustrated as black arrows) to all images produced. The *Keep*

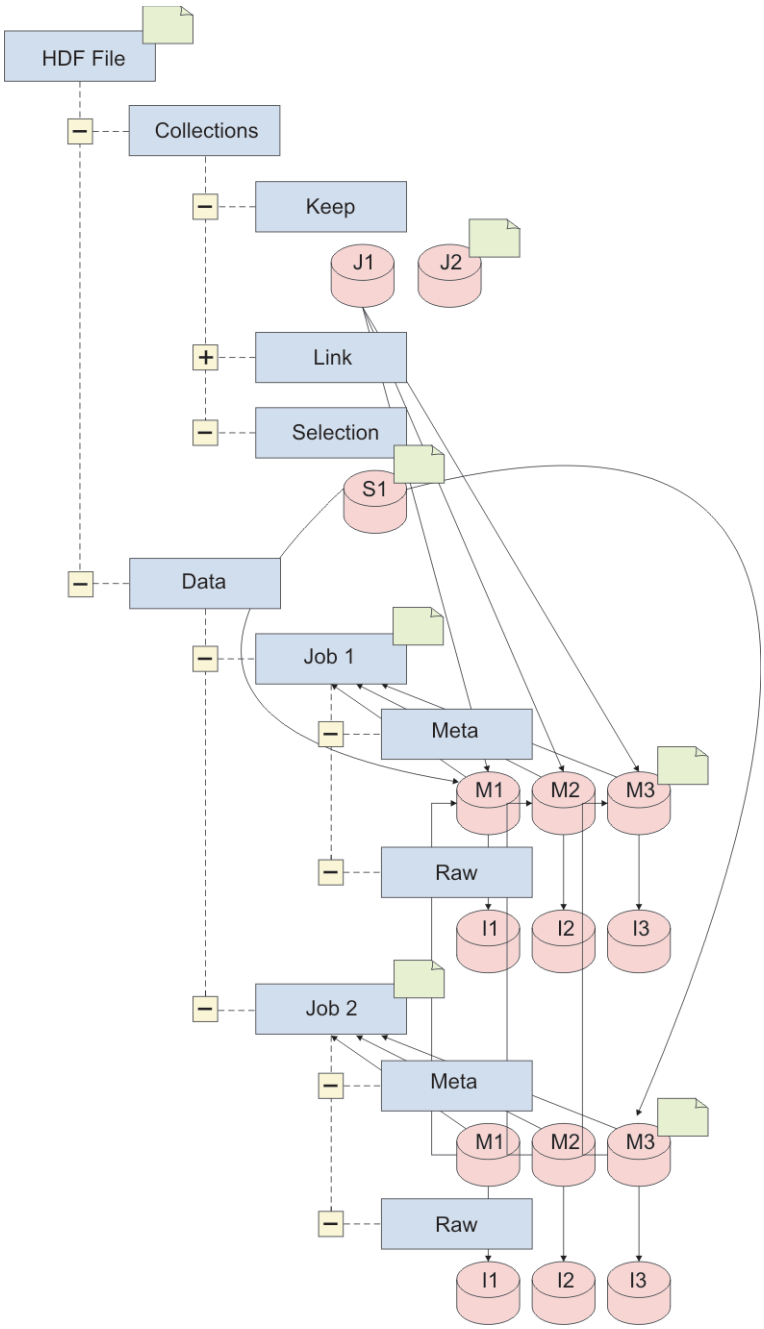

Figure 2.2: Internal structure of the HDF file. Blue rectangles represent groups (directory-like), red boxes represent datasets (file-like) and green boxes represent lists of key/value data. Black arrows indicate internal linkage (shortcut-like). See the text for more details.

group is updated automatically after each computation and items are only allowed to be removed from the end on, ensuring full project history tracking and thus good scientific practice. Custom selections of individual images (compare S1 in Figure 2.2) are stored in the *Selection* group, which allows random access and removal. The *Link* group is of technical importance only and is used to store complex input/output connections that are needed for the visual programming feature (see Section 2.2.2).

For each image, meta information (M1-3) and raw pixel data (I1-3) are stored separately, allowing to write header data only and linking it to previously stored raw pixel data. This is very useful for all algorithms that do not manipulate the actual pixel values, but merely do a reordering or grouping of images (as for example all classification algorithms do). Hence, redundant copies of pixel data are avoided which has a great impact on the total memory needed to store and archive project related data (a single set of $10,000$ images with 128x128 pixels already totals approximately 655 MB in size!). Furthermore, each image meta dataset is linked to its job group containing all information (in form of an attribute-list) necessary to be re-computed. Finally image-image linkages are established, enabling history tracking of each individual image from the very raw data to the final 3D structure. Together the tight internal linkage allows for a deallocation of raw pixel data at intermediate steps (which will shrink the file size drastically and is useful for long term archival). A restoration of previously deallocated data can be achieved by simply recomputing the corresponding jobs.

All software was developed with portability issues in mind from the very beginning on. Theoretically, ISO/IEC conform C++ code should be platform independent, practically this however is simply not true due to limitations and bugs in compilers and libraries. The Linux GNU compiler and the Windows C compiler differ substantially in various specific aspects of the language, such as Real Time Type Identification (RTTI), handling of static members and template management/linking. Differences also exist in very basic functionality provided by the standard template library, termed STL. Only a parallel development (in this case Linux and Windows) from the beginning on will result in reliable, fully platform independent code. All described algorithms in this thesis can, without exception, be equally well executed on a Linux or a Windows operating system.

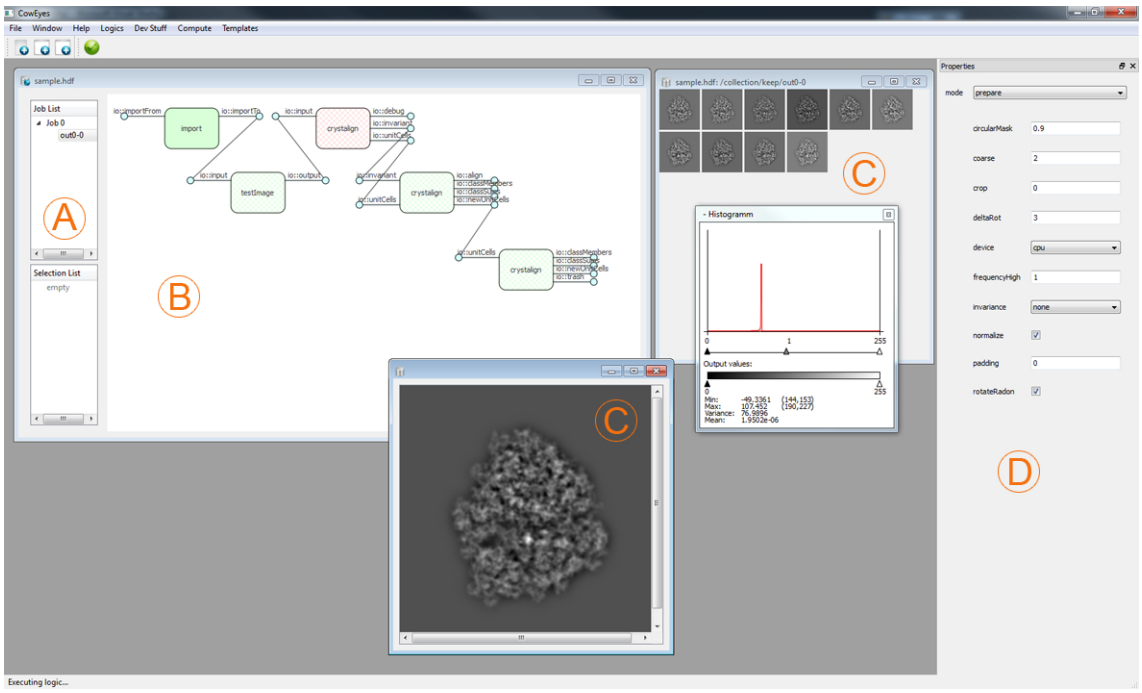

Figure 2.3: Screenshot of the user front-end *CowEyes*. The main modules are indicated by the circled characters. **A**: Project management module **B**: Visual programming module **C**: Display module **D** Property module.

2.2.2 Front-end

Having encapsulated all functionality in the back-end, front-end design only needs to implement user-interaction and viewing possibilities. Hence, a console application can be written very concisely whilst retaining full back-end functionality. A console application allowing to use the new algorithms described in this thesis was developed. It can be executed either command-line based or interactively. In order to further improve overall usability and to simplify project managing tasks a graphical user interface (GUI) was implemented in collaboration with B. Busche. The GUI currently features four main modules (see Figure 2.3 for a corresponding screenshot):

Project management module The project management module gives the user access to the HDF file, but abstracts the complex internal structure. The module is divided into two parts: a history tracking and selection. The history part corresponds to the *Keep* group (compare Figure 2.2) and lists jobs with their corresponding output collections. Only the most recent job may be deleted from the history. The selection

part corresponds to the *Selection* group (compare Figure 2.2) and can be used to store any number of custom selections (i.e. collections of individual images). Random deletion is allowed. Content information is retrieved through the Property and Display modules (see below).

Property module The property module is a generic information window of key/value type. The module is represented as an individual window, and shows information about any object currently selected. Selecting a job item in the project management module for instance will display all job related parameters, whilst selecting an image thumbnail in contrast will display the corresponding header information.

Display module The display module is designed for image inspection. Images may be displayed as a collection (thumbnail representation) or individually. Common manipulation routines such as scaling, grey-value and mid-tone adjustment are available.

Visual programming module In the visual programming module all available back-end modules (*Logics*) can be graphically represented and connected via their individual input an output channels. Assemblies of several logic elements can thus be regarded as small programs which we call *templates*. Templates can be saved for later reuse or exchange among different users. Figure 2.3 shows a sample template which was frequently used for testing the performance of the classification algorithms developed within this thesis.

Currently the GUI operates as a stand-alone program on the computer it was started. Although the back-end will detect the number of processors available (CPU and GPU) and automatically makes use of these, the computing power available may still not suffice to compute large jobs in a reasonable time. For this reason a server interface is in preparation, which runs the GUI as a client for submitting jobs to computer-clusters. The user will be able to simply connect to a master server, hence switching the background computation from local- to server-side without changing the overall appearance and behavior of the GUI.

If not explicitly specified otherwise, all image manipulation routines and all figures (showing cryo-EM related image representations) of this thesis were generated using the CowFramework software including the described back-end and the GUI which will be referred to as *CowEyes* from now on.

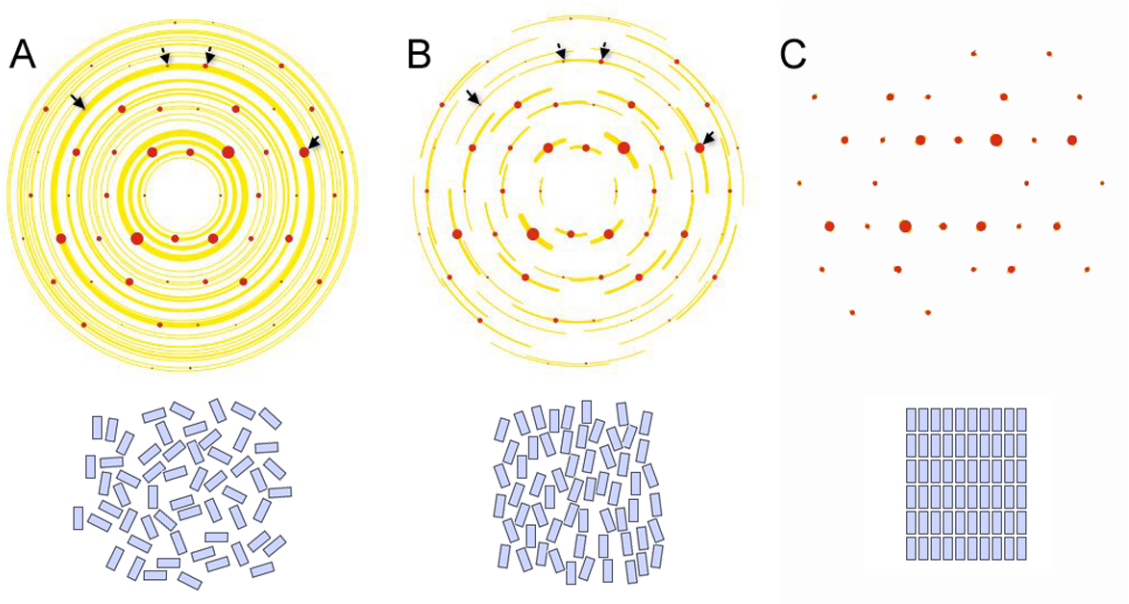

Figure 2.4: The figure illustrates the relation of the crystal's quality to its diffraction pattern. Crystals are depicted by the collection of blue rectangles (representing nano-crystals). The red spots indicate the diffraction pattern for a perfect crystal, which is overlaid in **A** and **B** for comparison. With increased ordering of the crystal the diffraction spots get more and more defined.

2.3 Reference-free image classification

In this thesis a novel algorithm for reference-free classification of noisy images was developed (named *Crystalign*). New concepts for measuring image similarity and clustering of images are introduced. All algorithms fit in the software framework described above and are thus available to the user in various front-end realizations.

2.3.1 Motivation

The basic idea underlying all described algorithms is best understood by drawing an analogy to X-ray crystallography. Disregarding all systematic and random errors during a diffraction experiment, the quality of the diffraction pattern is directly linked (via the linear Fourier transform) to the quality (i.e. the regularity of the unit cells) of the crystal. Thus, if one were to reverse the experiment, i.e. improve the diffraction pattern, the unit cells would rearrange to be more regular and more similar to each other. The analysis of the diffraction pattern of simulated crystals formed by individual images as unit cells is the

key idea of all classification strategies described in this thesis. The principle idea however is not new, an experimental version termed quasi-optical Fourier filtration was described by Ottensmeyer et al. already in 1977. At this time no digital image processing was available, but the problem of aligning and classifying images taken on photographic film (micrographs) was still present. To this end Ottensmeyer developed an optical arrangement that would visualize the diffraction pattern of stacked micrographs and selectively display only the periodic information represented by the Bragg lattice. Alignment of images was achieved by manually moving and rotating individual micrographs of the stack in a way that would lead to an improved diffraction pattern. With the advent of modern digital image processing, this technique was superseded by other similarity measures (see Section 1.2.4.5) such as the most frequently used CCC.

2.3.1.1 The objective function

Following the idea of measuring the quality of the diffraction pattern as an indicator of image similarity the way how to assess this quality has mathematically to be specified. In X-ray crystallography an important factor to address the quality of the diffraction pattern is its resolution. A diffraction pattern in X-ray crystallography can be understood as a 2D section of the power spectrum (A.1.3) of the 3D Fourier transform of the crystal's electron density distribution under a given beam-crystal orientation. The diffraction pattern of a crystal exhibits so-called diffraction peaks, which can be regarded as the discrete sampling of the continuous diffraction pattern that would result from a single unit cell. In other words, due to the periodic arrangement of many copies of the unit cell, the amplitudes for frequencies that are not also periodic with the unit cell dimensions will average to zero (those frequencies correspond to sine waves sampling over statistically random content and hence $\int_{-\infty}^{+\infty} \sin(x)\, dx = 0$ with $x_i = rand.$). Diffraction peaks on concentric rings refer to the same spatial frequency with increasingly higher frequency for larger ring radii. The outermost diffraction peaks with average intensity above some certain threshold (for example two standard deviations over background noise) are used to define the real-space resolution (which is the reciprocal of the current spatial frequency). This concept is very similar to the former described Q-factor (1.22), with the difference that the Q-factor also exploits individual unit cell phase information unavailable in crystallography. The squared numerator of the Q-Factor equation is analogous to the intensity of a crystallographic diffraction peak (also termed structure factor). But instead of referencing the total intensity to background noise the Q-Factor inspects the individual complex

contributions of each unit cell from which (by complex summation) the structure factor is built of. The resolution of a synthetic crystal built up by cryo-EM images as unit cells could thus be estimated by inspecting the Q-factor and choosing the highest frequency for which the Q-Factor is above some certain threshold. Resonable threshold values can be estimated from the expectation value of the Q-Factor for random noise which is $Q(\boldsymbol{k}) = \frac{1}{\sqrt{N}}$.

The maximum resolution is a good estimation for the total crystal quality as high resolution enforces accurate periodicity of the unit cell's content on very fine detail (i.e. small real-space distances) which commonly will only emerge if grainer detail (larger real-space distances) is already perfectly repeated. The objective function used for all described experiments and results in this thesis reads:

$$\mathrm{S} = \frac{1}{K} \sum_{k=1}^{K} w_k \begin{cases} q_k & \text{if } q_k > t_N \\ 0 & \text{else} \end{cases} \tag{2.1}$$

with

$$w_k = \frac{k}{K} \,, \tag{2.2}$$

where

$$q_k = \frac{|\sum_{i=1}^{N} F_i(\boldsymbol{k})|}{\sum_{i=1}^{N} |F_i(\boldsymbol{k})|} \,, \tag{2.3}$$

and

$$t_N = \frac{1.7}{\sqrt{N}} \,, \tag{2.4}$$

where $F_i(\boldsymbol{k})$ is the Fourier transform of the i'th unit cell for frequencies $|\boldsymbol{k}| = k$, K is the maximum frequency (Nyquist), and N is the total number of unit cells in the crystal. The scalar score S thus results from the integration over Q-Factors of individual spatial frequency up to a threshold value which is determined by a noise estimate t_N (2.4) for the current crystal size N. The integration is linearly weighted by a ramping function w_k (2.2) which enhances the contribution of higher spatial frequencies. In addition to the total crystal score S, individual unit cell scores S^i are computed, describing how well a given unit cell fits into the context of the current crystal. These scores S^i are obtained by a statistical analysis of the weighted complex distances of each unit cell contribution to the current average at frequency $|\boldsymbol{k}| = k$:

$$\mathrm{S}^i = \frac{d_i - \bar{d}}{\sigma_d} \tag{2.5}$$

with

$$d_i = \sum_{k=1}^{K} w_i (F_i(\boldsymbol{k}) - \bar{F}(\boldsymbol{k}))^2 \tag{2.6}$$

where

$$w_i = \frac{|\sum_{i=1}^{N} F_i(\boldsymbol{k})|}{\sum_{i=1}^{N} |F_i(\boldsymbol{k})|} , \tag{2.7}$$

and

$$\bar{d} = \frac{1}{N} \sum_{i}^{N} d_i , \quad \sigma_{d_i} = \sqrt{\frac{(d_i - \bar{d})^2}{N-1}} , \quad \bar{F}(\boldsymbol{k}) = \frac{1}{N} \sum_{i=1}^{N} F_i(\boldsymbol{k}) . \tag{2.8}$$

It should be noted, that the sum over frequencies in (2.6) is also limited in same way as q_k is in (2.1) but not explicitly written for clarity.

2.3.2 Implementation

The optimization process consists of building several crystals and changing their unit cells in order to result in an improved diffraction pattern. Hence, memory and time efficient methods had to be developed to allow for continuous updates of the evolving crystals. To this end three main structures (C++ objects) were developed:

Unit cell In crystallography a unit cell is defined as the smallest entity from which - by applying translational operations only - the whole crystal can be build of. Similarly the individual 2D images (on a fixed pixel frame) form the unit cells of the *in-silico* crystals. However, unlike in crystallography not the original image itself, but a complex representation of the latter is actually used as unit cell for computational reasons (see Section 2.3.2.1).

Crystal Like in crystallography the crystal object represents a collection of unit cells. In contrast to physical 3D crystals, the simulated crystals used here are one dimensional only. They are generated by arranging unit cells in a linear array next to each other. To be memory efficient an algorithm was implemented that minimizes the total memory rearrangement operations upon change of the current unit cell composition. Each crystal object can be individually evaluated for its diffraction quality and its unit cells be sorted according to their contribution to the overall diffraction quality (see Section 2.3.1.1 for details). The crystal object was designed in a way that the details of the objective function can be changed without needing to adjust any other code of the global framework.

Crystal Map The crystal map contains crystal objects and hence represents the result of the classification process. Several algorithms intended for parallel computation act on the crystal map structure.

From now on the term "crystal" or "class" will be used interchangeably describing a set of images found to be similar enough for later averaging. Images of a class will be referred to as "unit cells" or "class members". Having mentioned the main structural modules, the classification process can also be divided into functional modules.

2.3.2.1 Unit cell preparation

Prior to all optimization functions, the input images are pre-processed. In this regard it is of importance to notice that the discretely sampled Fourier diffraction peaks obtained by Fourier transforming an array of periodic elements (i.e. a crystal) can mathematically exactly be reproduced by summing up the complex Fourier coefficients of same frequency as obtained from Fourier transforming each periodic element (i.e. a unit cell) individually. Hence, the Fourier coefficients needed for crystal evaluation (see (2.1) and (2.5)) can already be pre-computed. In contrast to a naive coding scheme in which each crystal that changes its unit cell composition (during optimization) would have to be Fourier transformed in full length, the speedup in computing much smaller Fourier transforms of each unit cell and this only once for the rest of the program run is tremendous. Indeed, without this "trick" the described algorithms would be infeasible to be computed in any reasonable time.

Secondly, some reasoning of how to exploit the 2D information of each unit cell has to be done. In a naive setup, 1D Fourier transforms along image rows could be computed resulting in row-wise crystals. Albeit easy to treat, such an arrangement would completely ignore valuable 2D relations present in the data. A better approach would be to compute the 2D Fourier transform of each image and use the resulting coefficients for crystallization. However, this approach results in difficulties for later quality evaluation as Fourier coefficients of same frequency lie on concentric circles, which are computationally difficult to treat efficiently. The currently implemented solution to this problem makes use of the so called discrete Radon transform (Sinogram) which can be understood as a kind of polar sampling of the 2D Fourier transform (see A.4). The radon transform is rotated and reordered in such a way that rows through the crystals correspond to image features of increasing radial distance to the image center.

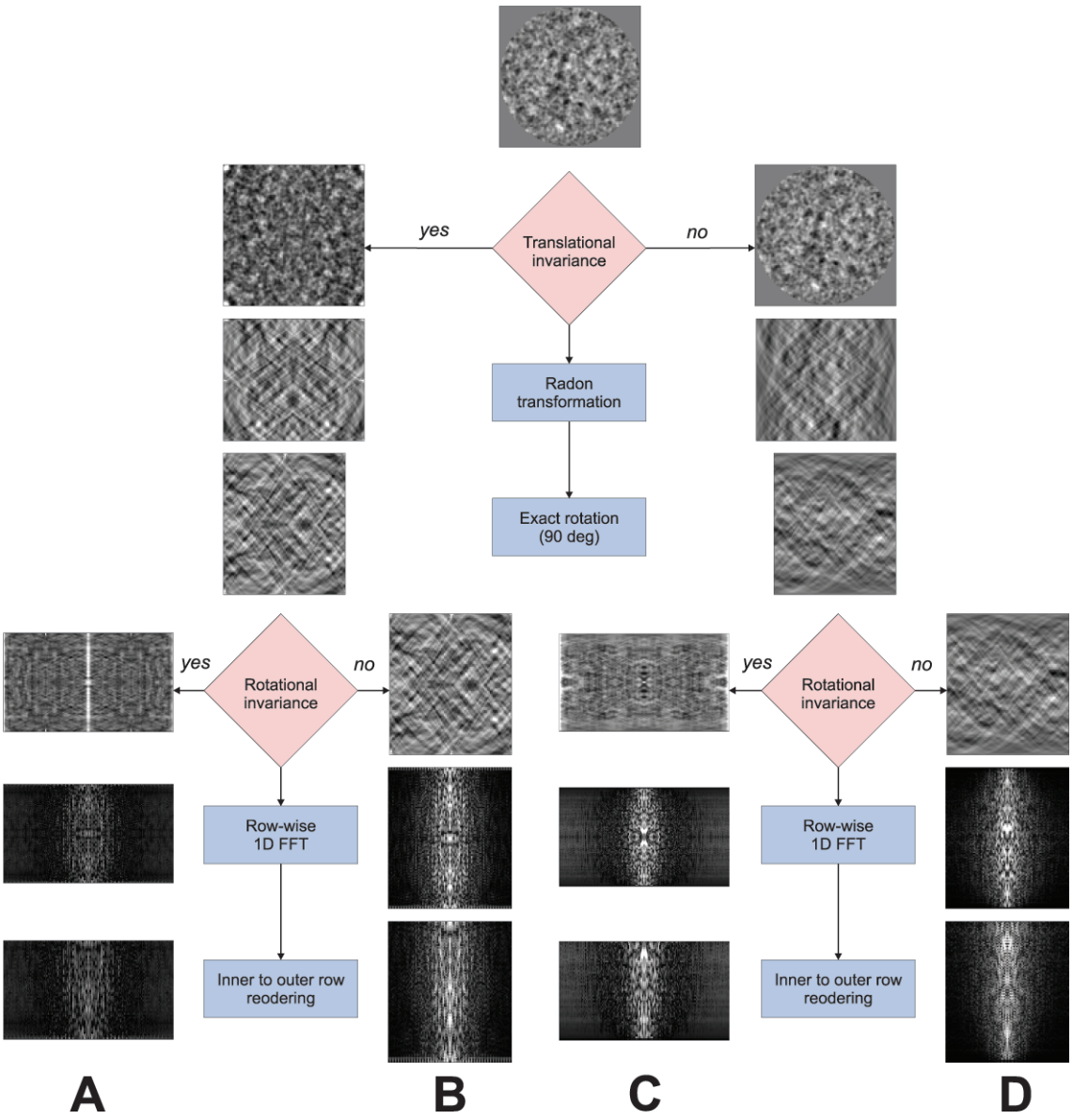

Figure 2.5: Image pre-processing.

Thirdly, regarding the previously mentioned critical separation of alignment and classification (see Section 1.2.4.5), the possibility to pre-compute translational, rotational or translational and rotational invariant representations of the input images was build into the preparation module. Translational invariance is achieved via the computation of the image's auto-correlation function (A.1.2). Rotational invariance is generated in the same

way but on a polar sampling of the image (as polar sampling transforms rotational relations into translational ones). Due to the usage of the Radon transformation during the preparation process, a polar sampling of the image is already available such that the invariance computation just has to occur at the correct point of time during pre-processing. For rotational invariance the so-called self-correlation (SC) is computed which differs from the AC by taking the square root of the complex Fourier amplitudes prior to transforming them back to real space. The reasoning behind this is a numerical one. Each evaluation of an AC will result in squaring the complex Fourier amplitudes. Multiple squaring of the Fourier amplitudes (which even without manipulation show a very high dynamic range) quickly exceeds the maximum dynamic range of the floating point number a computer can represent, hence leading to undefined behavior and artefacts. With regard to numerical and also algorithmic stability another modification to the AC/SC is performed. Origin peaks are removed (set to 0), as they only present the total sum of all pixel values (SC) or the total sum of all squared pixels (AC) respectively, and hence will have no impact on discrimination of structural features.

Prior to all the manipulations mentioned above, image mean normalization, circular masking, and internal coarsening (pixel binning) is performed. Figure 2.5 summarizes the pre-processing steps in a flow diagram.

2.3.2.2 Crystallization

The crystallization module is intended for de-novo classification of unit cells (i.e. prepared images as described in Section 2.3.2.1). Figure 2.6 illustrates the algorithm as a flow chart and the following explanations are ought to be understood in line with this figure.

Briefly, each unit cell at a given point in time may be in one of two possible "pots", either in a linear queue (with no specific ordering imposed) or in a crystal map (see definition above) as part of one specific crystal. Initially, the linear queue contains all unit cells and the crystal map is empty. In the end all unit cells are ordered in crystals that are part of the crystal map and the linear queue is empty. Hence, during the algorithm unit cells are popped from the queue and incorporated into crystals. The decision to which crystal a given unit cell should belong to and how many crystals are built in total emerges completely automatically by applying the objective function criteria as defined in Section 2.3.1.1. A unit cell is added to a given crystal if two scoring criteria are fulfilled,

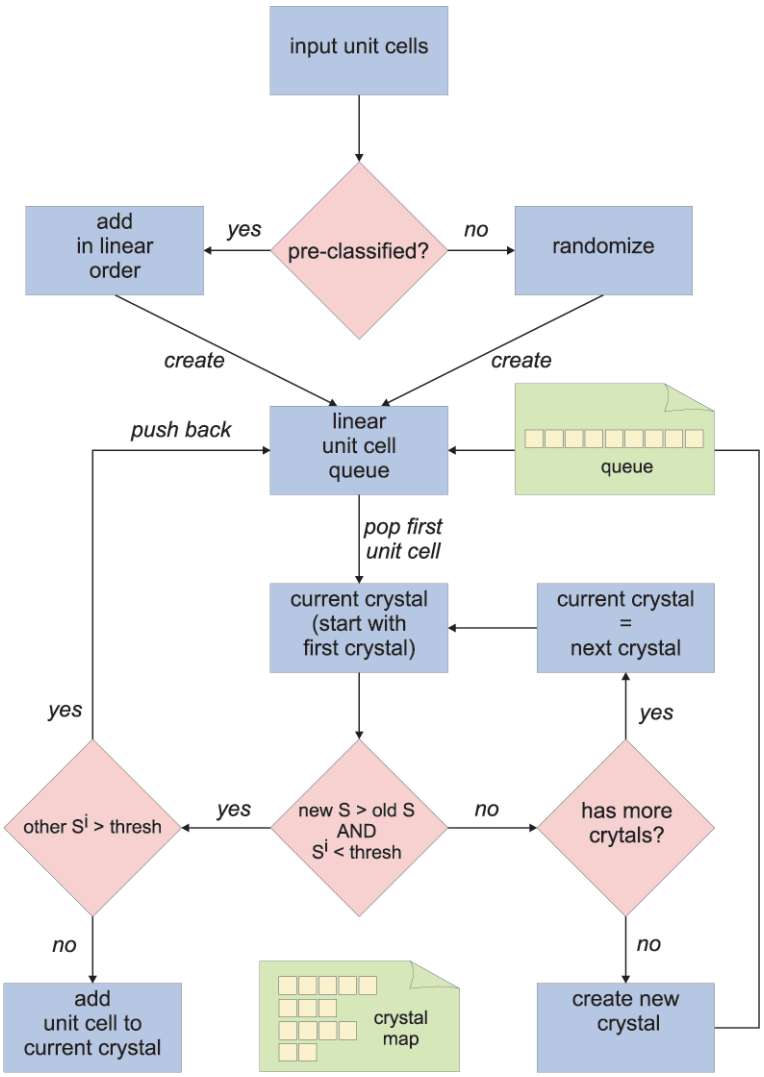

Figure 2.6: Flow-chart of the crystallization algorithm. Blue boxes describe operations, red ones decision points. The two main structures, i.e. the linear queue of unit cells and the crystal map between which the unit cells are shuffled, are illustrated in green. Refer to the text for more details.

i) the total crystal score S (2.1) with the unit cell is higher than the crystal score without the unit cell

ii) the individual score S^i (2.5) of the unit cell is below a given threshold value (user adjustable, default 1.2)

If a new unit cell is built in, subsequently the worst scoring unit cell of the crystal is determined. If its score S^i is above the threshold value (as in ii) this unit cell is removed from the crystal and pushed to the back of the linear queue of unit cells for later re-evaluation. If a unit cell could not be incorporated into a crystal because one or both of the above scoring criteria were not fulfilled, this unit cell is tried on the next crystal available. If all crystals were tried and at no time the unit cell could be build in, this unit cell seeds a new crystal.

Two special cases of the algorithm described so far should be mentioned. In order to avoid possible infinite looping (due to circular pushing and popping the same set of unit cells to and from the linear queue, respectively), individual unit cells are allowed to be pushed back only a finite amount of times (typically once) to the linear queue and are afterwards forced to form new crystals. A second special case is a user adjustable maximum crystal size, which when reached always removes the currently worst performing unit cell upon registration of a new unit cell in the crystal (i.e. ignoring scoring criterion ii) in favor of constant crystal size).

Thus the only factors manually adjustable and possibly influencing the number and size of the crystals (after the preparation procedure) are the threshold value for the individual unit cell scores S^i (2.5) and the maximum class size. This behavior is a remarkable difference to most of the other clustering algorithms (like K-means, hierarchical clustering, etc.) which need a predefined amount of classes or class members, which of course can not be known in advance. On the other hand the ability of automatic detection of both, number of classes and class size comes at the cost of an expensive iterative process as described in this section. Furthermore, the described algorithm scales worse than linear with increased size of input data. For this problem to be handled, initial trials on mixing the current algorithm with standard techniques of dimension reduction and classification (see Section 4.1.2) were performed and seem promising.

2.3.2.3 Crystal Improvement

The crystal improvement module can be regarded as a refinement tool to be used on already classified data (e.g. such as obtained after running the crystallization module, or after any other classification procedure). The procedure can be restricted to perform an intra-class refinement only or may be allowed to also reorder unit cells between different

crystals (inter-class refinement).

The intra-class refinement procedure is very similar to the optimization algorithm described for crystallization (see Section 2.3.2.2). It can be understood as a crystallization process which is performed on the unit cells of every already existent crystal individually. Thus, the algorithm outlined in the flow chart of Figure 2.6 has to be thought of being executed N times with N being the number of pre-built crystals and the input being the unit cells of the current crystal. Depending on the quality of pre-classification, the refinement will remove more or less unit cells from the original crystal and split them into several smaller crystals of higher quality. A user adjustable threshold determines to which total size crystals will be kept, smaller crystals are removed from the dataset. Being a mutually independent operation the intra-class optimization can be heavily sped up by massive parallel computation, and thus does not suffer from the critical performance issues mentioned in Section 2.3.2.2.

Inter-class refinement is achieved by running an intra-class refinement first, collecting all solvated unit cells (i.e. those that are found not to fit the current crystal) and then starting a complete crystallization procedure (see Section 2.3.2.2). However, already existent (refined) crystals are not destroyed and only the former removed unit cells are added to the linear input queue. Hence, the input data size is typically much smaller in comparison to a de-novo crystallization resulting in a heavily improved performance for this procedure.

Chapter 3

Results

3.1 Preparation of synthetic test data

An objective evaluation of the quality of any image manipulating algorithm is only possible on data with exactly defined properties. Those properties (e.g. original signal, type and amount of noise present, random and systematic frequency aberrations etc.) can not at all or only very inaccurately be retrieved from real (cryo-EM) data. Consequently, a defined synthetic test set of images has to be generated for performance evaluation purposes.

Classification algorithms intended for single molecule reconstruction typically face the problem of having to cluster images varying in content for two very different reasons. Firstly, images differ because the underlying 3D object (the molecule) is frozen in random orientation in the electron microscopically investigated sample, leading to different 2D projections in possibly very close angular distances. The second reason is caused by sample heterogeneity, which invalidates the assumption that all projection images stem from the same 3D object. Typically heterogeneity is caused by effects like ligand binding/absence, movement of domains or random fragmentation of the molecule. Thus, images which are identical in their projection direction may still differ because of the aforementioned reason. Clearly, those two phenomena are interrelated as sample heterogeneity may be detected better or worse under different projection angles. To exploit this additional information for classification it is necessary to include 3D relationships to the classification process which is not subject to this thesis. However, research related to this problem is ongoing and combinations of the ideas outlined in Schmeisser (2009) with this thesis' findings are under currnt investigation.

The test dataset was generated from the 3D structure of the 70S ribosome as available from the RCSB Protein Data Bank (Berman et al., 2000). In order to reflect sample heterogeneity, five different modifications of this structure were generated[1]:

1. 50S subunit only
2. 70S ribosome without tRNA and ternary complex
3. 70S ribosome without tRNA and ternary complex, 30S rotated by 7 degree
4. 70S ribosome with ternary complex bound
5. 70S ribosome with ternary complex and P-site tRNA bound

Modification 1 is intended to simulate molecule fragmentation as may occur in reality during sample preparation. Modification 3 simulates domain movement and all other modifications model ligand binding effects with different detail (with the ternary complex having a molecular weight of ∼3% and the P-site tRNA of only ∼1% relative to the molecular weight of the total 70S ribosome).

Projection angle differences were simulated by projecting each of the five mentioned modifications under 10 different solid angles in 4 degree intervals. In the context of cryo-EM reconstruction this spacing would be regarded to already be very fine grained and adequate for high-resolution reconstruction. Consequently, the test data set consists of 50 individual 2D images differing in the features described above. For preparation the image processing software *Imagic* (van Heel et al., 1996) was used, with which the atomic model was converted to a density map of grey-valued pixels. The final projection images were adjusted to fit a 384x384 pixel frame, with each pixel corresponding to 1 Å. Figure 3.1 shows the noise free test dataset.

Further preparation of the data was performed with the test-image module of our software. Typically the noise-free images were first band-pass filtered (Gaussian kernel) with a high-frequency threshold of 0.8, a low-frequency threshold of 0.05, and zero transmission (see A.2). The idea behind high-pass filtering is to lower interpolation artifacts eventually introduced during projection of the 3D model. The low-pass filtering is intended to simulate the decrease in resolution with increased spatial frequency as the latter effect is

[1]PDB-IDs for 50S subunit: 1VOU, 30S subunit: 2UUB, ternary complex: 1OB2, and tRNA: 1TRA

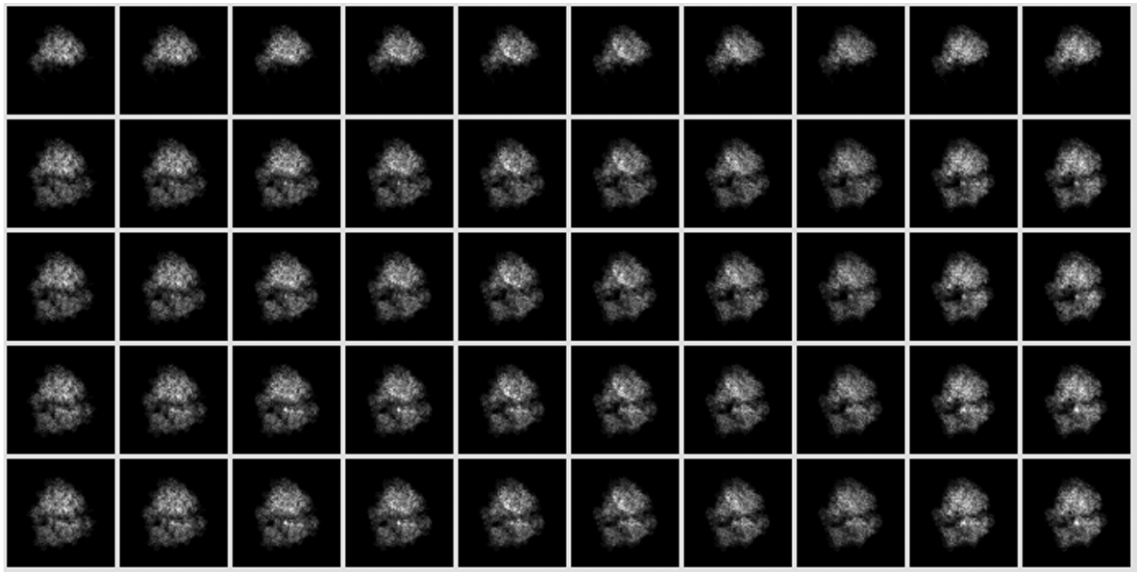

Figure 3.1: Synthetic test dataset of 50 different 2D projections generated from 5 modifications of a 70S ribosome. Images of each row represent projections of solid angles incremented by 4 degrees each.

introduced to a greater or lesser extend by all experimental imaging processes including TEM. Filtered images subsequently were normalized to zero mean pixel-intensity and to a fixed variance (typically 1.0).

Realistic noise generation is a non-trivial task as it needs knowledge of the noise distribution features of real cryo-EM data. Indeed, complicated noise models have been derived (e.g. Scheres et al. (2007)) to most accurately simulate and treat real-life images. However, as the synthetic data generated here only serves the purpose of proving concepts of the new algorithms and are not used for extensive qualitative analyses, a simplified noise model was used throughout. Uniformly distributed noise was generated using a *mersenne twister* (Matsumoto and Nishimura, 1998) random number generator. Random numbers generated this way will have a period of $2^{19937} - 1$, hence suffice for uncorrelated noise generation of the decribed test datasets. Uniform noise generated in this way was subjected to band-pass filtering equal to that applied for signal filtering. Finally, the noise was normalized to also have zero mean intensity and its variance was adjusted according to the currently aimed for SNR. The final noisy images were generated by simple pixel-wise addition of the previously described signal and noise components, respectively.

3.2 Proof of concept using synthetic data

This section is shows that the new scoring functions and optimization algorithms introduced in the last chapter conceptually work on synthetic data. The terms cluster, class or crystal will be used interchangeably describing a set of individual images that are believed to fulfill the classification conditions as described in (1.12) and (1.13) of Section 1.2.4.5. Individual images of a class will be referred to as class members or unit cells.

The most difficult part of the classification process is to determine not only the number of classes but also the number of members for each class automatically (i.e. unsupervised) in an optimal way or at least near optimal way. Not enough though, the process should be robust under a wide range of noise and possibly work with translational and rotational invariant representations of the images to break the interrelation of alignment and classification.

To the author's knowledge no tool is currently available that can fulfill all the above mentioned criteria. All classification methods implementing K-means or any type of hierarchical clustering will by design not be able to cluster in an unsupervised way. Methods that utilize PCA or CA will have trouble with invariant representations as they are very sensitive to the strongly scaled Fourier amplitudes resulting from the auto-correlation processes frequently used to compute the latter (Frank, 2006). In the following the performance of the new algorithms with respect to the mentioned classification criteria is described.

3.2.1 Discrimination of individual images

As described in Section 2.3.1.1, for each class two scores are computed. One score (S) is intended to evaluate the total quality of the corresponding class, which is important for unsupervised assignment of the correct number of class members and cross-class quality comparison (sorting). The second score (S^i) is intended to evaluate how well each individual image fits into its current class context and hence is most important during optimization. Smaller scores indicate a better, and higher scores a worse fit to the current class and can be understood as the distance to the average fitting class member in units of standard deviations. The performance of image scoring will theoretically be assessed by evaluating synthetically constructed test data sets.

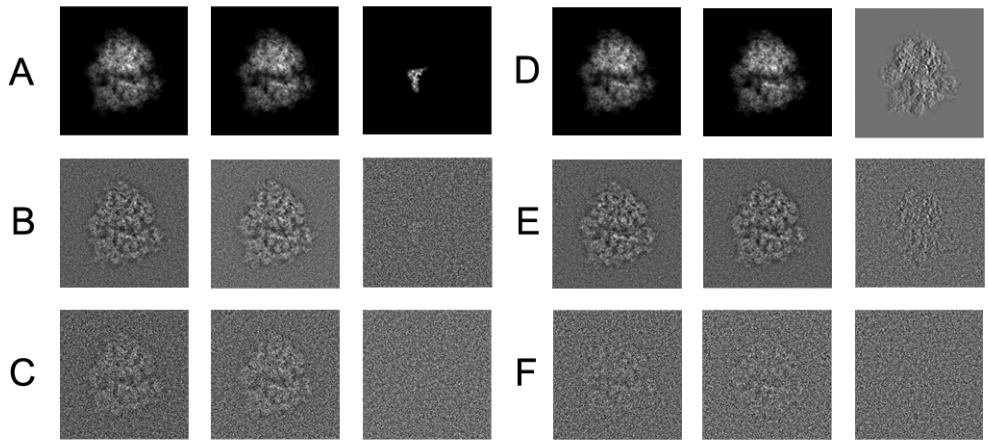

Figure 3.2: Test setup for inidvidual image scoring. Two different experiments are performed to evaluate the theoretical performance in scoring individual images in their current class context. The first experiment is intended to simulate sample heterogeneity by comparing ribosomes with or without p-Site tRNA with each other. A representative of each and the difference density is shown at different SNRs (A: noise-free, B: 0.4, and C: 0.06 respectively) in rows A-C. The second experiment is intended to simulate projection angle differences. D-F illustrate the images used and the corresponding difference density at various SNRs (D: noise-free, E: 0.4, F: 0.02). The difference in projection angle used here is 4 degree.

To a class consisting of already 10 correct images (i.e. difference is introduced only by noise), a single "bad image" is added and subsequently S^i is evaluated for all members. This experiment is done on two different setups, one simulating sample heterogeneity (presence/absence of a part of density) the other one simulating projection angle differences (see Figure 3.2). As can be seen from Figure 3.3 and Figure 3.4 the scoring performs well over a wide range of SNRs. Real cryo-EM data typically do not exceed SNRs far below ~ 0.1, such that in this case the "bad image" would always be correctly detected.

For the computation of the individual S^i scores all images of the current class are investigated at once, such that the accuracy or in other words the contrast of one bad image in a growing number of good images should increase. This behavior can indeed be observed on the test data. Figure 3.5 shows the results for the identification of one image with its projection angle different to a growing number of images resulting

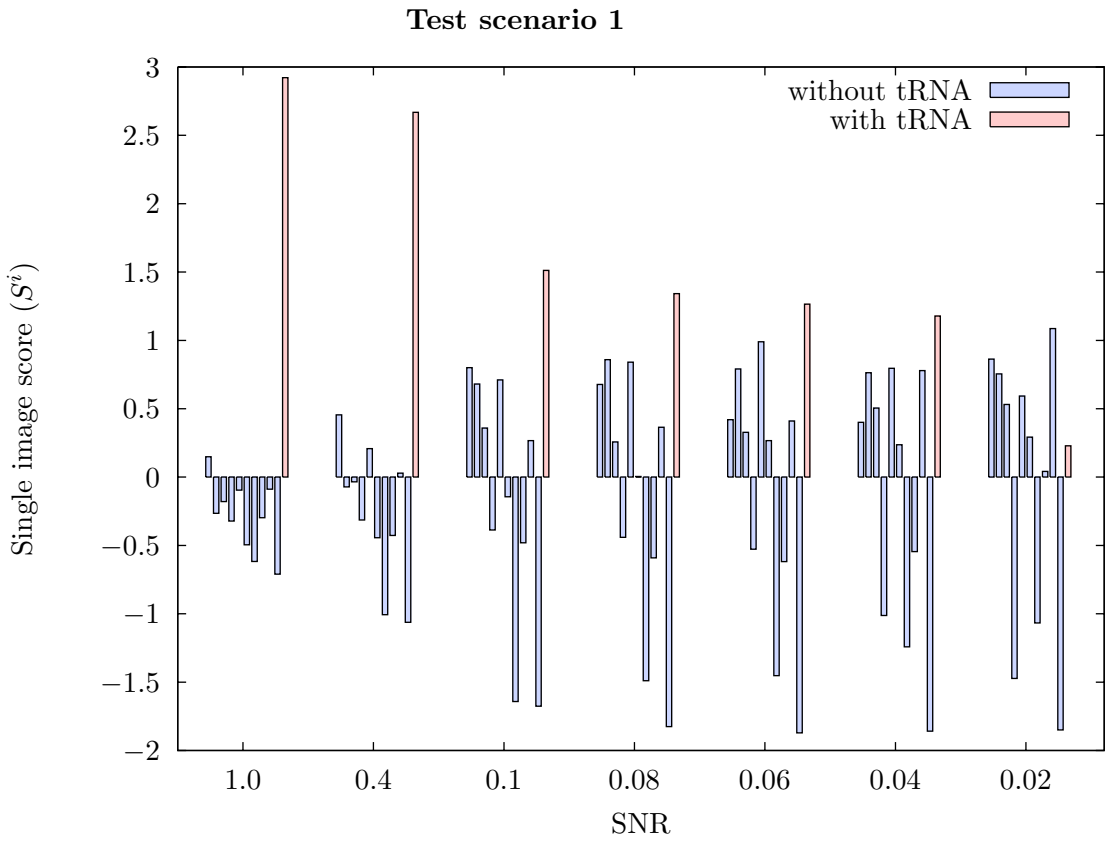

Figure 3.3: Single image scoring - data heterogeneity. Discrimination of a single "bad image" next to 10 "good" ones. The difference of the images being presence/absence of the p-site tRNA (see Figure 3.2 A-C).

from the same projection angle. For this setup the SNR was kept constant at 0.04 and the projection difference was that described in Figure 3.2 (D-F).

This test is intended to show the scoring behavior under a growing number of "bad images" but keeping the "good images" constant in size (here 10). The test data are the same as described in scenario 2. Image discrimination is completely accurate for less than 50 % of wrong images. Higher ratios lead to increasing inaccuracies (see Figure 3.6).

As can be seen from the results shown so far, image discrimination improves with increasing accuracy of the underlying class. This behavior suggests an iterative optimization algorithm which grows classes under permanent control of the individual image scores.

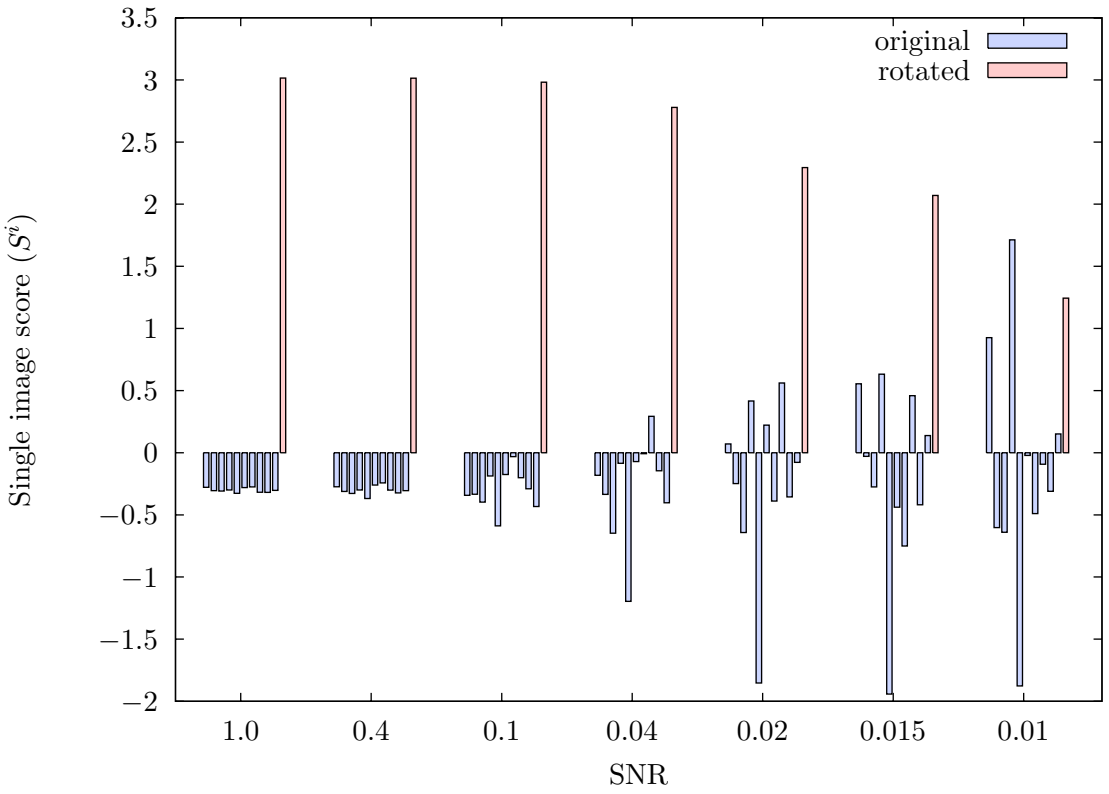

Figure 3.4: Single image scoring - projection angle differences. Discrimination of a single "bad image" next to 10 "good" ones. The difference of the images being the projection angle which differs by 4 degree (see Figure 3.2 D-F).

This idea is similar to the physicochemical process of crystal growth. In fact, in chemistry one way to purify a compound is through crystallization. For that a saturated solution of the impure substance is prepared and subsequently crystals are grown by shifting the solution to an even higher saturation level (by means of cooling, vaporization etc.). In this way even mixtures of different compounds may be purified by pooling similar crystals and repeated crystallization approaches on those ("fractional or selective crystallization"). It has been experimentally observed that the process of crystal growth has to be slow in order to obtain highly homogeneous crystals. Too fast crystal growth results in crystals of inferior quality i.e. in crystals exploiting irregularities in their unit cell composition. Another similar process to crystallization is that of metal annealing (as for example used during alloying) for which the metals are first heated up and then slowly cooled down re-

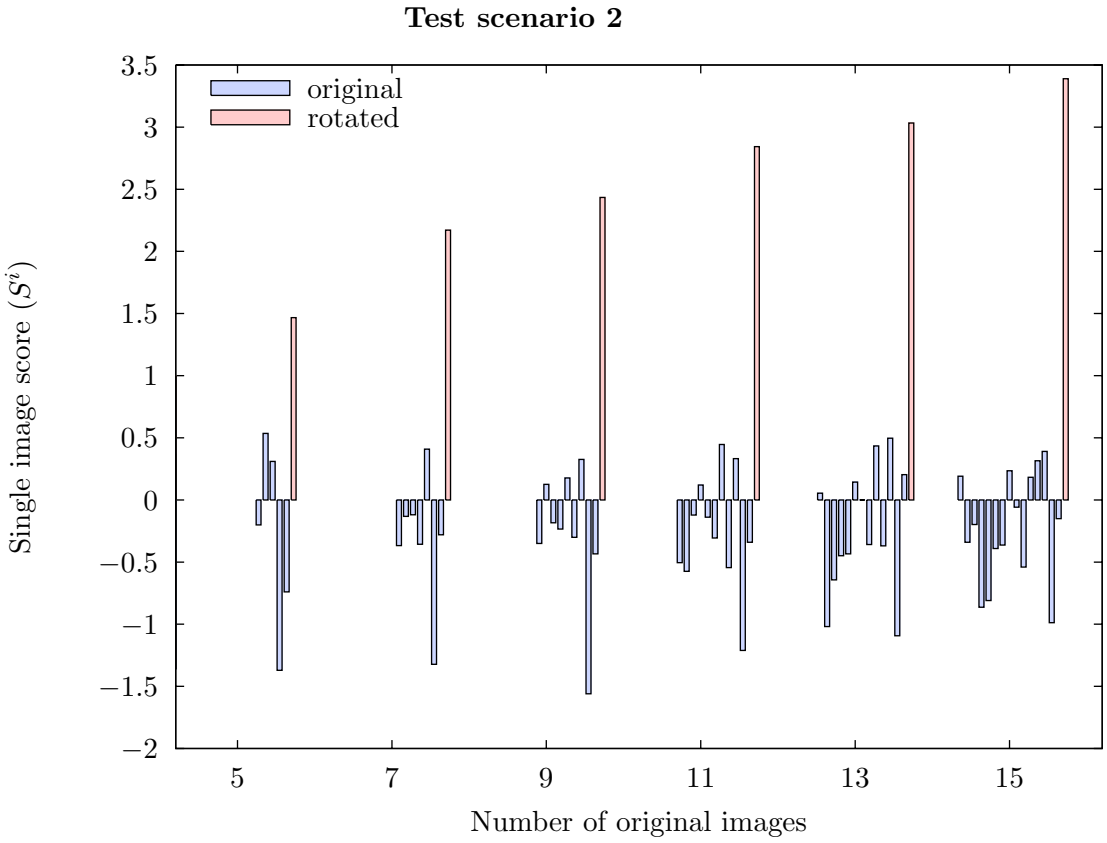

Figure 3.5: Single image scoring - contrast enhancement. Identification of the "bad image" is dependent on the current class context and will improve if more images of good similarity are available. All results are generated at a constant SNR of 0.04.

sulting in a very ordered (atomic) fine structure. This process has even found its way into numerical computing and has become a widely used tool for maximization/minimization of functions and is known as "simulated annealing" (Salamon et al., 2002; van Laarhoven and Aarts, 1987).

The optimization algorithm described in Section 2.3.2.2 can be understood as an *in-silico* crystallization method, similar to a simulated annealing procedure. Crystals are grown slowly and are repeatedly updated and cleaned using the individual image scoring S^i ensuring high quality crystals and thus good classification results.

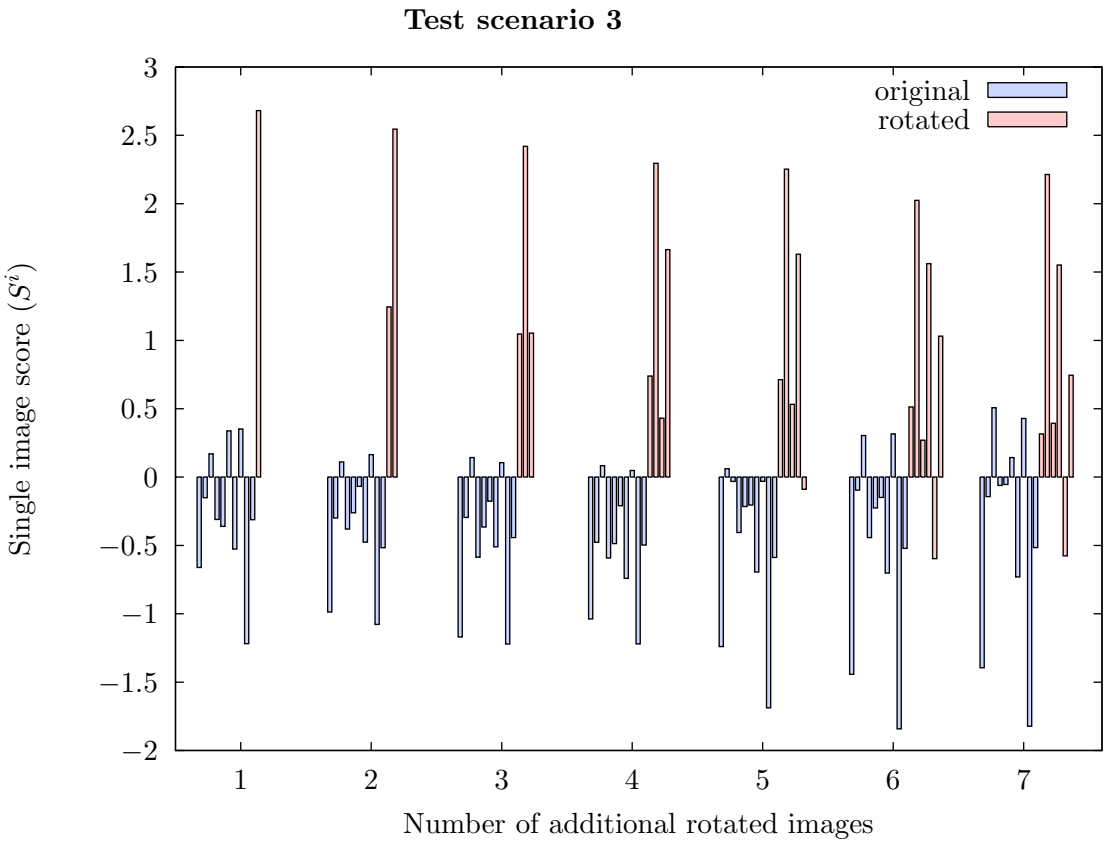

Figure 3.6: Single image scoring - growing dissimilarity. To a class of 10 correct images, a constantly growing number of images being wrong in the projection angle by 4 degree are added. All results are generated at a constant SNR of 0.04.

3.2.1.1 Discrimination on invariant representations

This section is intended to prove the concept generating invariant representations of images by the method described in Section 2.3.2.1. Furthermore the performance of image scoring on these invariant representations is assessed.

Computing translational and rotational invariance For test purposes a synthetic image with three circles of different intensities were generated in a 128x128 pixel frame. Using the test-image module of *CowEyes*, 10 copies of the original image were randomly translated and rotated (see first row of Figure 3.7). The second row of Figure 3.7 shows the images after computing translational invariance through Fourier based auto-correlation. As can be seen, the images are indeed identical in their relative translation, but still differ in rotation. For rotational invariance the full discrete Radon transform (0–2π) is com-

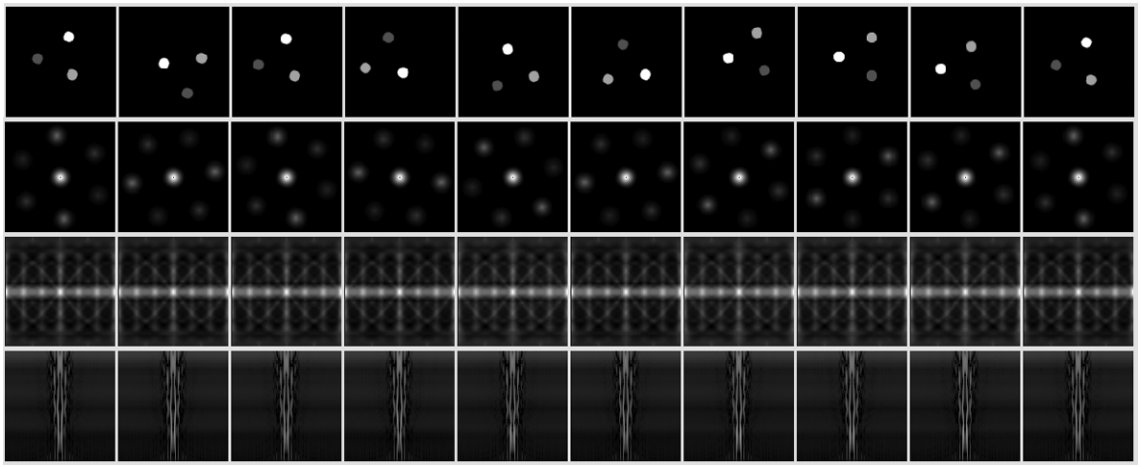

Figure 3.7: Proof of concept for generating translational and rotational invariance. The first row shows the original image, rotated and translated at random. The second row shows the images after auto-correlation and reodering such that the origin comes to lie in the center of the image. The third row shows the images after 1D self-correlation of each row of the rotated Radon transform (i.e. horizontal lines describe polar samplings of constant radius). The fourth row finally shows complex representations of the images (unit cells), as are used for further processing.

puted and rotated such that horizontal lines describe features of constant polar radius and are subsequently row-wise self-correlated. After this manipulation the resulting images are both translationally and rotationally invariant (compare third row of Figure 3.7). The last row of Figure 3.7 shows the complex representation of the images (unit cells) as used for further processing. Reducing the image features to those which are invariant under translation and rotation of course results in a loss of data. More precisely, Fourier phases are removed (set to zero) twice along the process (due to the two auto/self-correlation operations). Moreover, this process has a strong influence on the image's Fourier amplitudes which - as multiplied (and in case of auto-correlation even squared and multiplied) with each other - are relatively overestimated for low spatial frequencies (as those commonly having large amplitudes) and underestimated for higher spatial frequencies, respectively. Together, the effect of Fourier rescaling and the loss of information renders the discrimination of invariant images (especially for low SNRs) extremely difficult.

Individual scoring of invariant images Discrimination of images differing by the presence/absence of tRNA only (as used previously) is currently beyond the sensitivity

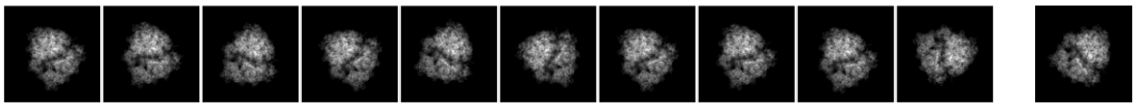

Figure 3.8: Test setup for invariant image scoring. 10 copies of a ribosome are rotated in the range of -90 to $+90$ degree and translated (± 10 pixels in each direction) at random. The separated image furthermore differs in its projection angle which is altered by 4 degree. All images shown here are unfiltered and noise-free.

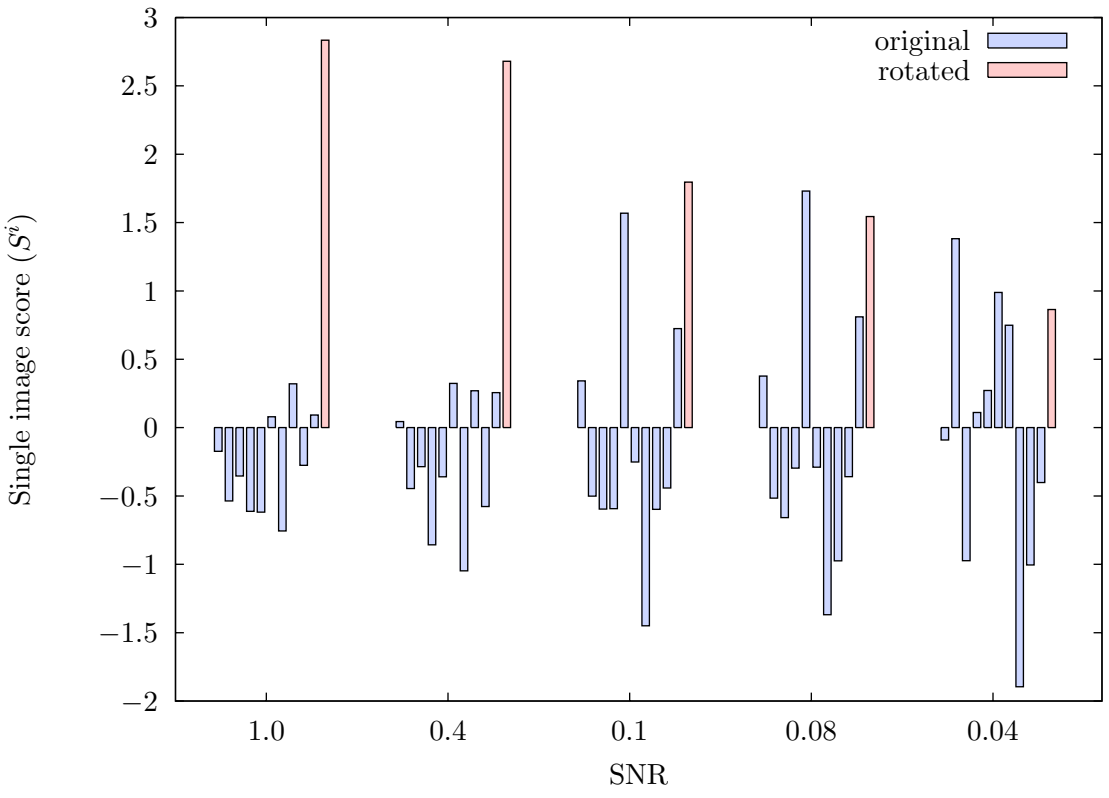

Figure 3.9: Single image scoring - discrimination of invariants.

at SNRs relevant to real cryo-EM data. However, separation of images differing in their projection angles is possible, even at relatively low SNRs (see Figure 3.9). In the context of a processing strategy in which images after invariant classification and subsequent alignment are subjected to a second but *non-invariant* classification, the shown discrimination behavior may already be sufficient. This is because sample heterogeneity can - as

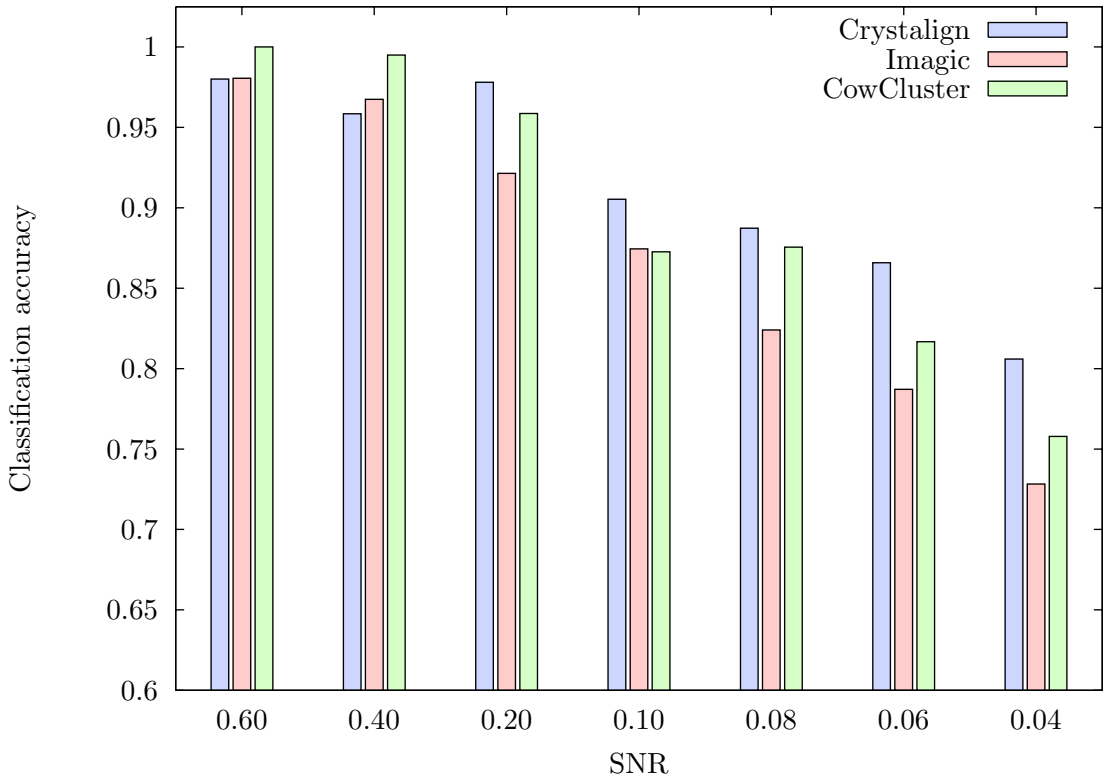

Figure 3.10: Complete classification of a synthetic dataset. The data were prepared as described in Section 3.1 and adjusted to consist of 10 images per class, resulting in a total size of 500 images. A classification was performed using three different programs (Crystalign, Imagic, and CowCluster). The plot is showing the different scoring accuracies as assessed by (3.1) for various SNRs and analysing the first 80 classes ($N = 80$) of each program's output.

shown previously for the aligned datasets - be detected and separated for in the second classification process.

3.2.2 Classification of synthetic data

Having demonstrated the performance of individual image scoring, this section is intended to show the capabilities of the new algorithms for a complete classification of a synthetic dataset. The test data were designed as described in Section 3.1, hence 50 different classes were produced. Class sizes were adjusted such that each class is represented by 10 copies (with different noise) of the same image, i.e. resulting in a dataset totalling 500 images.

The aim of this test is to determine the maximum classification accuracy possible under decreasing SNRs. To that end a very low S^i threshold of 1.0 was chosen, which has the side-effect of producing relatively small classes compared to those resulting from a larger threshold values (as statistically fewer very similar images under a given noise level will be available). The new method Crystalign on average splits each class, resulting in approximately 100 classes per classification run. In order to judge the overall classification accuracy the following scoring system was used:

$$Accuracy = 1 - \frac{1}{N}\sum_{i}^{N} c_i \tag{3.1}$$

with

$$c_i = \frac{\sum_j^M \sum_{k=j+1}^M \delta_{jk} \cdot p_{jk}}{0.5 \cdot M(M-1)} \ , \tag{3.2}$$

where N is the total number of classes to be scored, c_i is the per-class score, M the number of members of the ith class, δ_{jk} the Kronecker delta and p_{jk} the penalty for a mismatch of images j and k. Thus all unordered pairs of images within one class are inspected and a penalty value for all mismatches is summed up. Subsequently this penalty value us divided by the number of total pairs per class, before computing the score for several classes from the average of the individual class scores.

Consequently, if all images in a class are correctly classified c_i will be 0 conversly, c_i will be 1 if all images are incorrect (i.e. all pairs mismatch). This relatively stringent scoring is relaxed by the fact that images under the same projection angle differing only in the presence/absence of the tRNA (compare Section 3.1) are penalized by $p = 0.5$ in contrary to all other mismatches which are penalized by $p = 1.0$.

In order to evaluate the performance of Crystalign in the context of other existing classification routines, the accuracy evaluations were performed on the results as obtained by running Imagic (van Heel et al., 1996) and CowCluster (Lüttich, 2007). As both programs can not run in an unsupervised fashion the number of classes to be generated was set to 100 and the number of Eigenimages to use for classification was set to 69 (the upper limit of the Imagic software). For all three programs the 384x384 input images were coarsened by a factor of 4 (i.e. binning of square blocks containing 16 pixels each) thus resulting in 96x96 pixel sized input images. Figure 3.10 show the results over a range of SNRs for the first 80 classes $N = 80$. As is evident from the figure all three programs perform

very well on the synthetic dataset, the differences between them being only marginal. For the two highest SNR values of 0.6 and 0.4 respectively, CowCluster does the best job in classifying almost without any mistake. However, over the full remaining range from SNR 0.2 down to a SNR of 0.04, Crystalign is the most accurate routine in classifying this synthetic dataset. Notably, Crystalign seems to be less sensitive to increased noise ratios as the relative difference to the other routines increases with decreasing SNR. The improved accuracy at SNR 0.2 compared to SNR 0.4 of Crystalign is contra-intuitive and may be a sign for an incomplete convergence of the optimization algorithm as better results should theoretically be achieved for higher SNRs (see Section 4.1 for discussion).

3.3 Classification of real data

The evaluation of any classification performed on real data is - as the correct result is unknown - much more difficult. Commonly, for a final 3D structure the quality and resolution is assessed by computing the Fourier Shell Correlation (FSC) (Harauz and van Heel, 1986) which is the 3D equivalent of the 2D Fourier Ring Correlation (FRC) already described in Section 1.2.4.5. For this reason it was decided to use the FRC as a quality criterion for the 2D classification results presented in this work.

As the FRC is a cross-correlation measure, only two images at a time can be compared to each other. To overcome this obstacle each class resulting from a classification experiment was randomly split into two and individually averaged. Subsequently, the FRC between the two averages of each class was computed and finally all FRCs were averaged to result in a Fourier based correlation measure for the whole dataset over the entire spatial frequency range. A drawback of the described comparison strategy however, is its strong dependence on the number of images subjected to averaging (i.e. the class size). Larger classes will always tend to have higher correlations than smaller classes as long as not only pure noise is being averaged. To this end, all classification experiments were designed to produce classes of possibly similar size.

3.3.1 Test scenario 1 - 70S ribosome

For this test, a dataset of 1000 already filtered and aligned cryo-EM images of a 70S ribosome sample was used. The classification was first done using Crystalign which resulted in 83 classes with approximately 10 images each. 165 images were automatically discarded

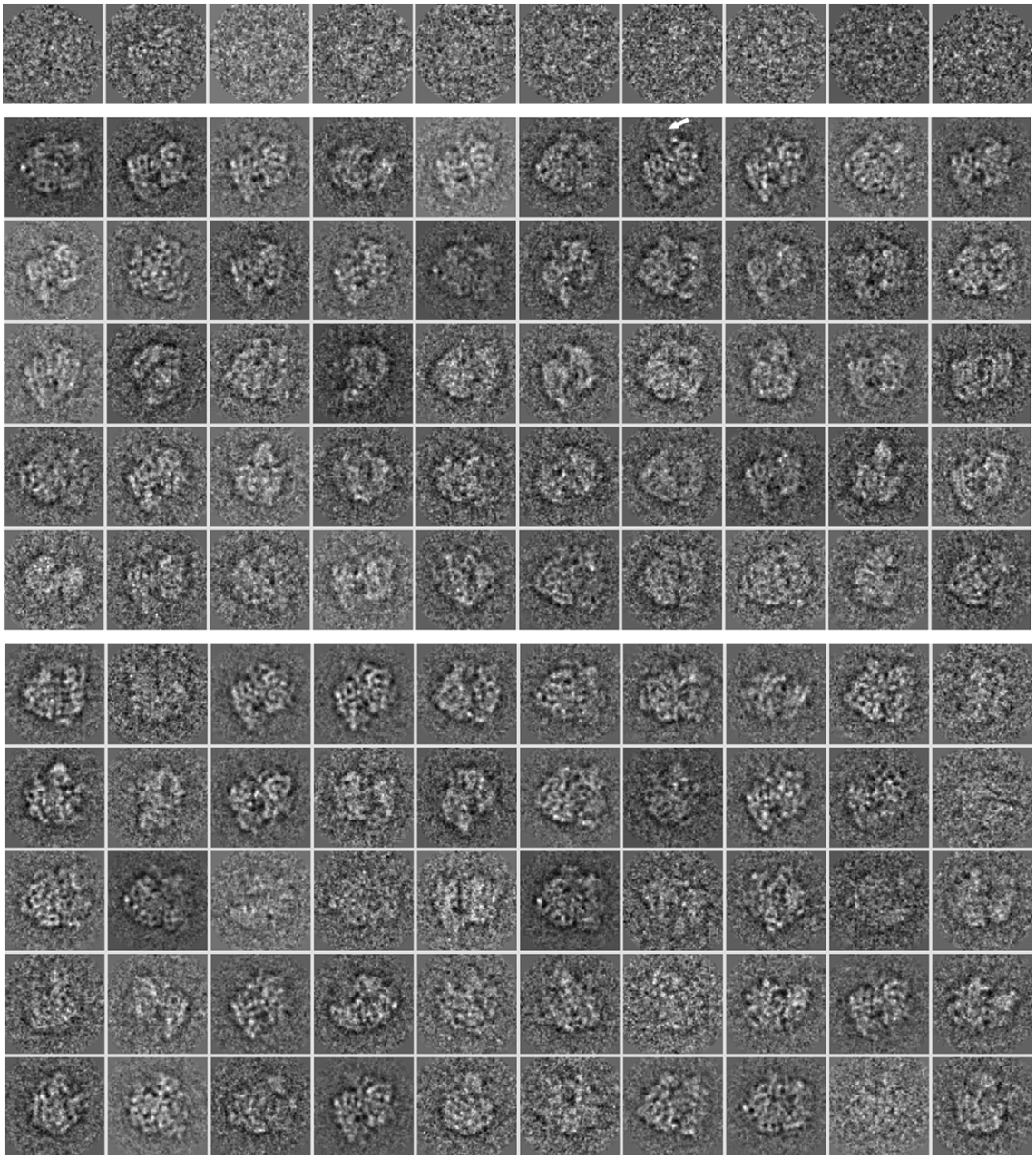

Figure 3.11: The first row shows a sample of filtered and aligned images of the 70S ribosome used as input for classification. The next block shows the class sums generated by Crystalign for the first 50 images. The last block corresponds to the first 50 class sums of an Imagic classification. Marked with a white arrow (in the first block) is the L7/L12 stalk of the ribosome (Diaconu et al., 2005), which is believed to be very flexible and thus only visible if accurately averaged.

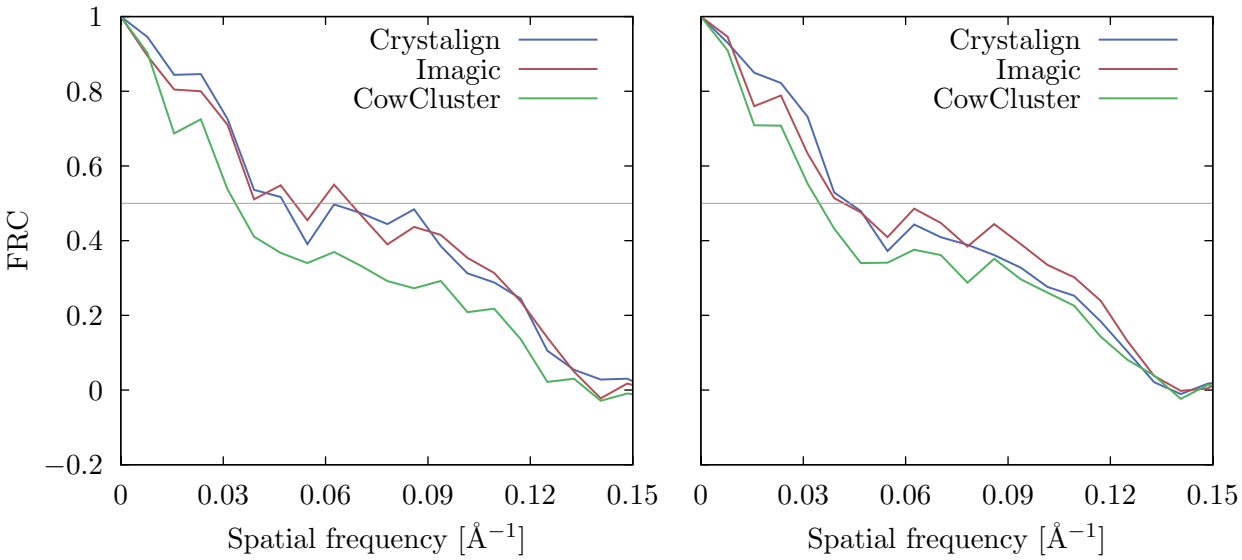

Figure 3.12: Test scenario 1 - Classification of a real cryo-EM dataset (70S ribosome) containing 1000 images. Plotted are the average FRC curves for the first 15 (left plot) and 25 (right plot) class sums. The gray horizontal line at indicates the 0.5 threshold value (refer to text for more details).

as the result of an user-defined minimum class size of 8 members. For comparison, classification of the same data was performed using Imagic and CowCluster with class sizes set to 100 and the number of Eigenimages to 69 (upper limit of the Imagic software).

Figure 3.11 shows a sample of the input images and the first 50 class sums as result of a classification with Crystalign and Imagic, respectively. Unlike in Imagic, class sums in Crystalign are individually scored and are sorted accordingly. This becomes already clear by human inspection of the class sums. In case of Imagic, class sums are seemingly better and worse at random, whereas the Crystalign classes have a more homogeneous (however in quality decreasing) appearance. The same observation is made when inspecting the averaged Fourier Ring Correlations for increasing number of classes. Figure 3.12 shows the FRC as a function of spatial frequency for the first 15 class sums (left) and the first 25 class sums (right). Statistically most significant are the values above a correlation of FRC $= 0.5$ (indicated by the grey line in all FRC plots) which correspond to SNRs > 1. For lower FRC values the noise will be stronger than the signal and therefore analysis of these is regarded risky (Frank, 2006). For spatial frequencies corresponding to FRC values greater 0.5 Crystalign performs very well in classification. The partly better

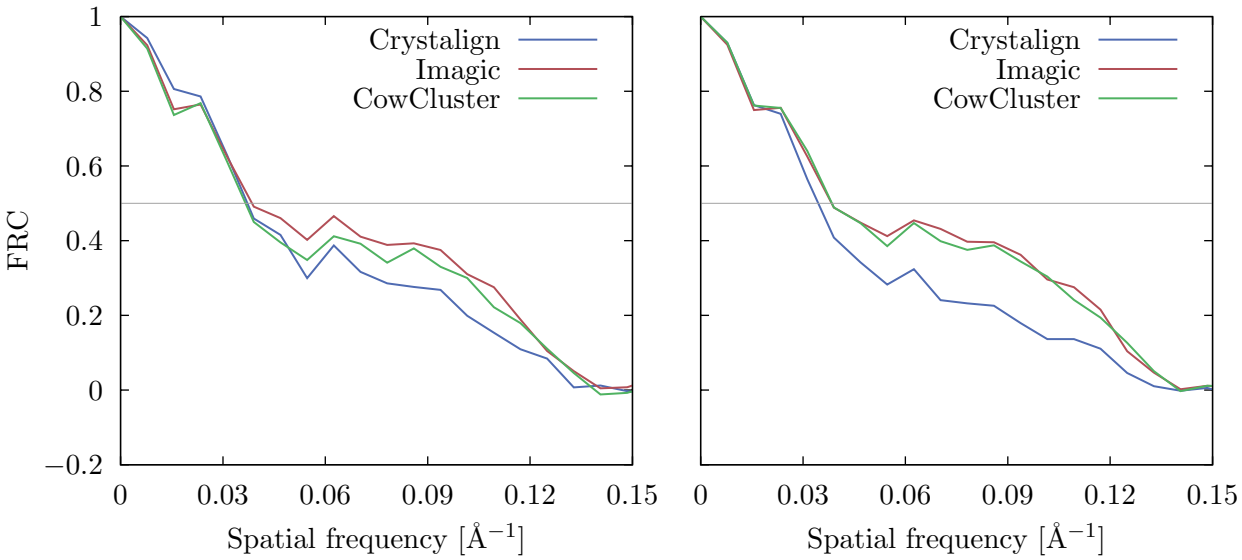

Figure 3.13: Test scenario 1 - Classification of a real cryo-EM dataset (70S ribosome) containing 1000 images. Plotted are the average FRC curves for the first 50 (left plot) and 80 (right plot) class sums. The gray horizontal line at indicates the 0.5 threshold value (refer to text for more details).

performance of Imagic with respect to higher spatial frequencies has to be evaluated with care as class sizes are in average slightly greater than those of Crystalign and thus may reflect an unspecific statistical effect (compare Figure 3.14). The performance of the third program CowCluster can not objectively be discussed if only part of the class sums are taken into account. This is because the distribution of class sizes is monotonically increasing to higher class sum indices as illustrated in Figure 3.14.

With inclusion of more class sums into the quality evaluation the performance of Crystalign gets increasingly weaker (compare Figure 3.13). This behavior is not surprising as class sums are sorted by quality. The FRC curve for the first 80 class sums clearly shows that too many class sums were selected from Crystalign's result set. For routine usage this can be avoided by setting a sensible threshold for the total image score S, which - as an absolute and data independent measure - can be applied among different datasets and prevents inclusion of low quality classes. In summary, Crystalign produces classes of high quality but in a relatively small amount. This phenomenon is believed to root in the optimization rather than the scoring strategy of Crystalign. Ideas for further improvement of the optimization process are discussed in Section 4.1.2.

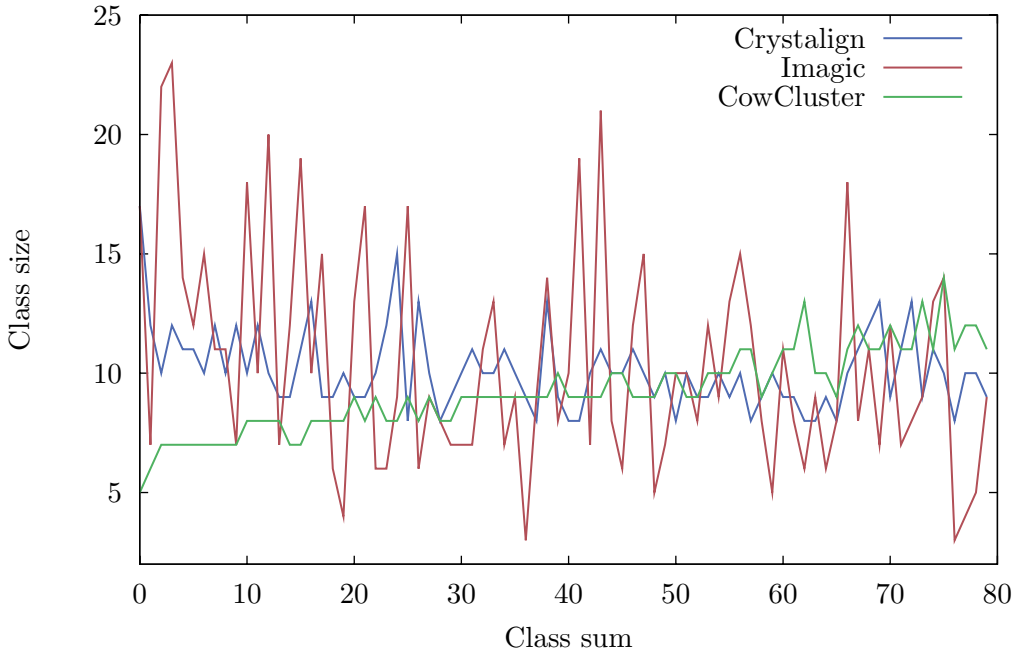

Figure 3.14: Test scenario 1 (70S ribosome) - The figure shows the sizes of the class sums as generated by the programs Crystalign, Imagic and CowCluster.

3.3.2 Test scenario 2 - anaphase-promoting complex (APC)

This second test was performed exactly identical to the first test described above. Out of 1000 input images, 201 images that ended up in classes smaller than eight members in size were discarded by Crystalign. The remaining 799 images were clustered into 80 classes, i.e. again the average class size was 10. Consequently, Imagic and CowCluster were run with the same setup as described for the ribosome case. Although the APC dataset has quite different characteristics compared to the ribosome dataset the overall results are very similar. Again, Crystalign is able to classify and sort the data such that for at least the first 50 class sums its performance is clearly better in comparison to the other routines (see Figure 3.16). The inclusion of 50 class sums into the evaluation results in approximately equal performance of all routines (compare left plot of Figure 3.17). Evaluation of all 80 classes produced by Crystalign, results in a similar effect as was observed for the ribosome case. The lowest quality classes (which under routine usage should be discarded from further analysis) are decreasing the overall quality of Crystalign's results to an amount that results in total worse performance relative to the other routines.

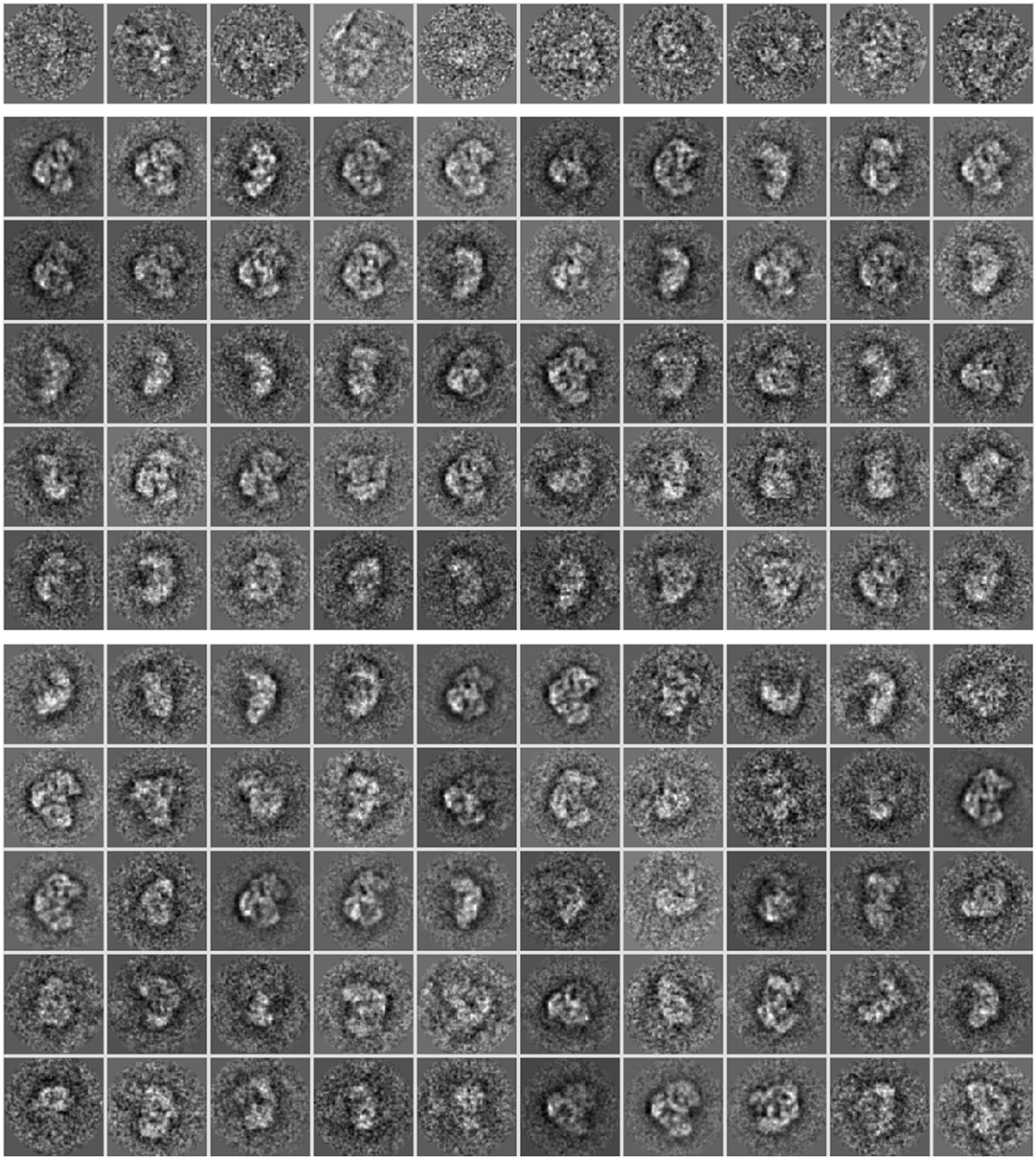

Figure 3.15: The first row shows a sample of already filtered and aligned images of the anaphase-promoting complex used as input for classification. The next block of images shows the classification results (class sums) of Crystalign for the first 50 images. The last block corresponds to the first 50 class sums of an Imagic classification.

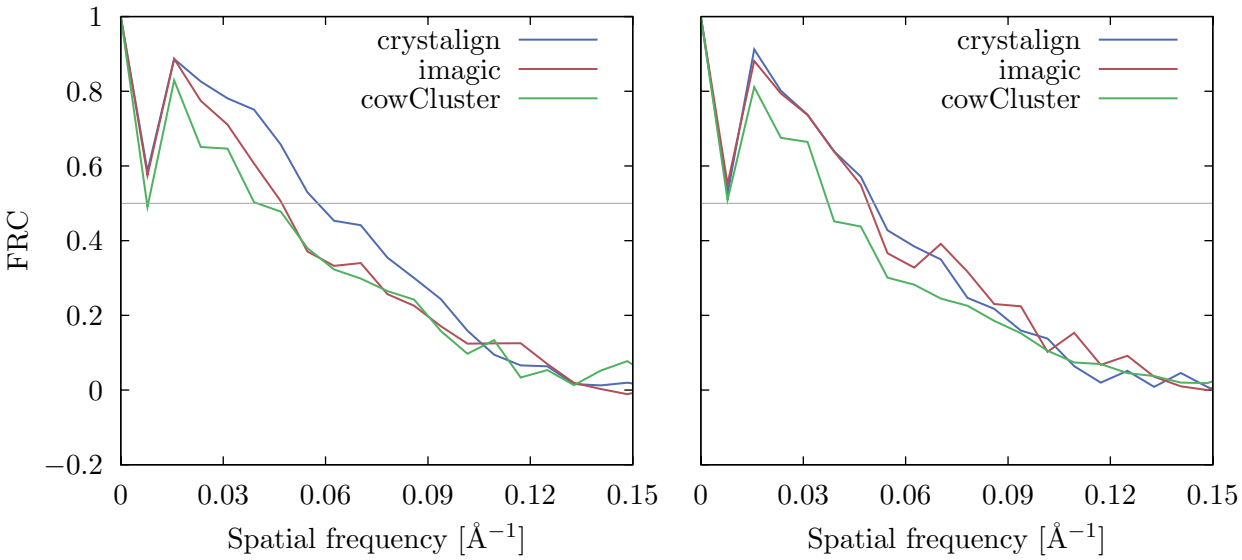

Figure 3.16: Test scenario 2 - Classification of a real cryo-EM dataset (anaphase-promoting complex) containing 1000 images. Plotted are the average FRC curves for the first 15 (left plot) and 25 (right plot) class sums. The gray horizontal line at indicates the 0.5 threshold value (refer to text for more details).

3.4 Refinement of already classified data

...die guten ins Töpfchen, die schlechten ins Kröpfchen!

—Aschenputtel, Gebrüder Grimm

As demonstrated in Section 3.2.1 the new method Crystalign is able to identify images that are not fitting into the current class context. To be able to make use of this property outside the frame of the de-novo classification a separate “clean” module was implemented. Any pre-classified data can be refined using this module, which in the simplest case sorts all classes by quality and removes the worst images (to be defined *via* the S^i threshold) from each class. As cleaning can be performed on each class individually, with respect to computation this problem can be described as “embarrassingly parallel” and is indeed implemented to make use of the maximum number of CPUs or GPUs available. Hence, even for large datasets this kind of refinement is extremely fast.

A robust routine which is able to sort class sums by quality and prevents accumulation of low-quality or misfitting raw images is especially important for automation. To date,

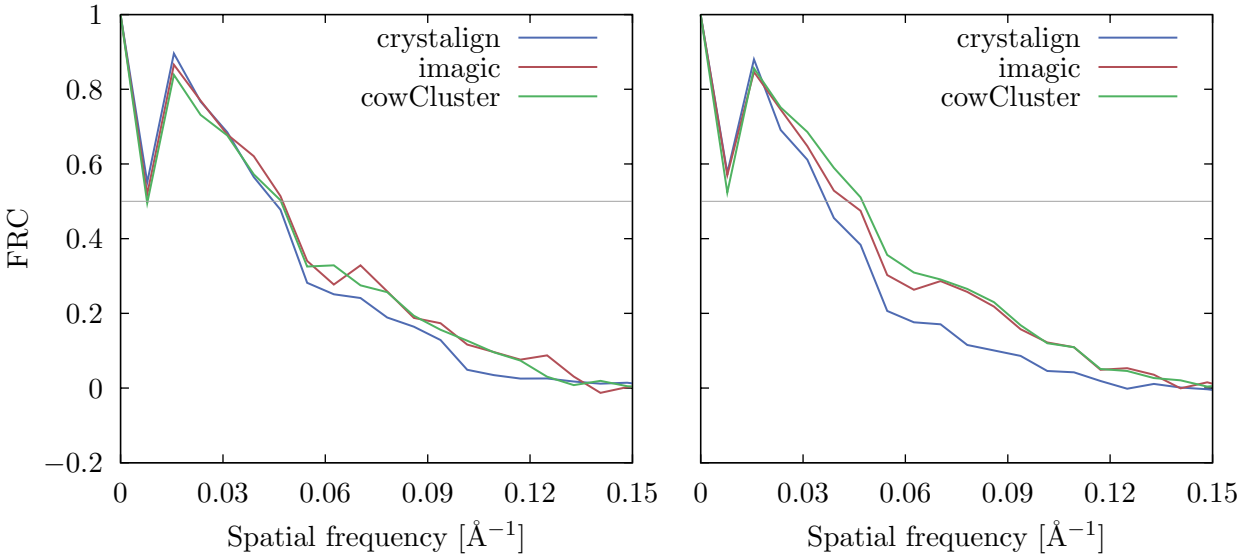

Figure 3.17: Test scenario 2 - Classification of a real cryo-EM dataset (anaphase-promoting complex) containing 1000 images. Plotted are the average FRC curves for the first 50 (left plot) and 80 (right plot) class sums. The gray horizontal line at indicates the 0.5 threshold value (refer to text for more details).

the selection of class sums that are subject to further processing is commonly performed manually, by visual inspection. This firstly is a tedious job to do (especially in respect of the ever increasing amount of data), and secondly disrupts any automation pipeline from image acquisition to 3D structure and finally adds a subjective (possibly biasing) component to the whole process. The routine described herein is intended to abolish the need of manual inspection and is designed to fill another gap on the way to full automation. To demonstrate the capability of this module the previously described cryo-EM data of the 70S ribosome was used. Using Imagic, 1000 images of this dataset were classified into 50 classes. Subsequently, the average FRC (as described in Section 3.3.1) for an increasing number of class sums (intervals of 10) was computed. As expected for unsorted class sums, no trend in the overall FRC quality was observable (compare the left plot of Figure 3.18). In contrast, the right plot of Figure 3.18 shows the results for the same data, but after quality sorting by Crystalign. Notably, the class sums are not simply sorted by their individual class sizes (the average class sizes are shown in round braces for each block in Figure 3.18); class sums 10 – 20 for example are on average smaller in size than those from 20 – 30 still showing a higher quality. Thus, through omitting low

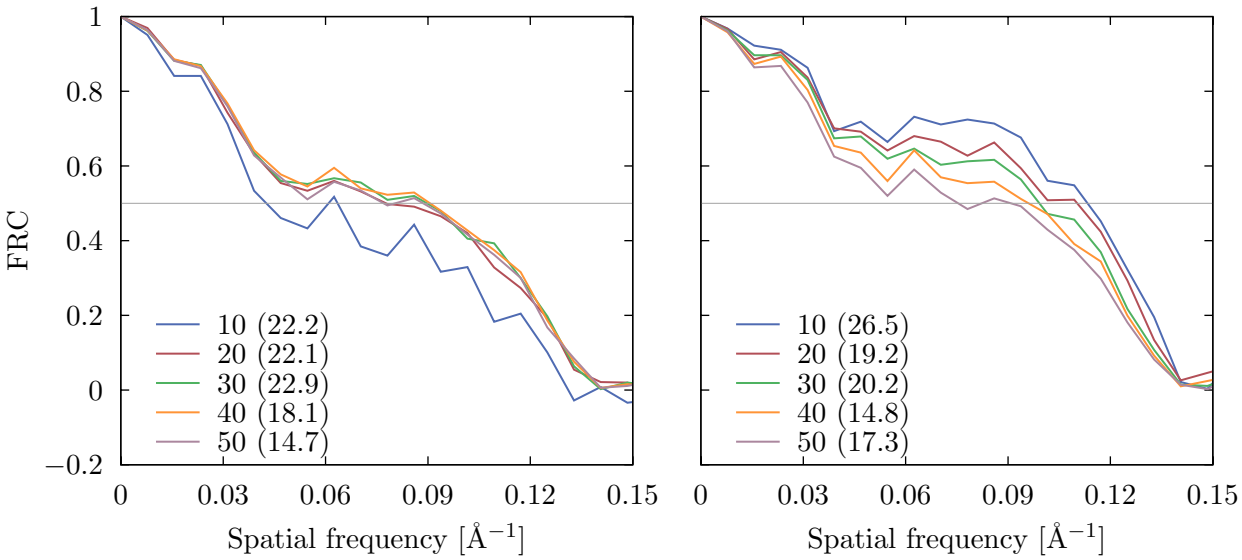

Figure 3.18: Effect of sorting class sums by quality. Left: Original data as obtained from an Imagic classification. Right: The same data, but after sorting using Crystalign. Scores are computed for increasing number of class sums. The number of class sums was increased in intervals of 10, the average class size for each interval is given in round braces.

quality class sums a total higher accuracy for the remaining dataset can be achieved. With regard to automation, a threshold may be determined and images below this threshold may safely be removed as they will be of increasingly lower quality. Due to the Fourier based computation of the class scores, these scores are data independent and thus allow quality comparisons of different image sets. This is generally not possible on results obtained through Principle Component Analysis (PCA) or Correspondence Analysis (CA) as their coordinate system of Eigenvectors is relative and data dependent. In comparison the Eigenvectors of a Fourier transform, which are the complex exponentials, also reside in an multi-dimensional and orthogonal but fixed and data independent coordinate system.

In addition to data sorting and eventually skipping the worst class sums, further improvement may be achieved by removing individual images identified to statistically misfit their current class context. Unfortunately, this effect is difficult to show using the FRC based evaluation, as removing images from the averaging process during the FRC calculation statistically results in a score lowering. To this end a conservative threshold of $S^i = 2.5$ was selected which resulted in removal of only 20 bad images corresponding to

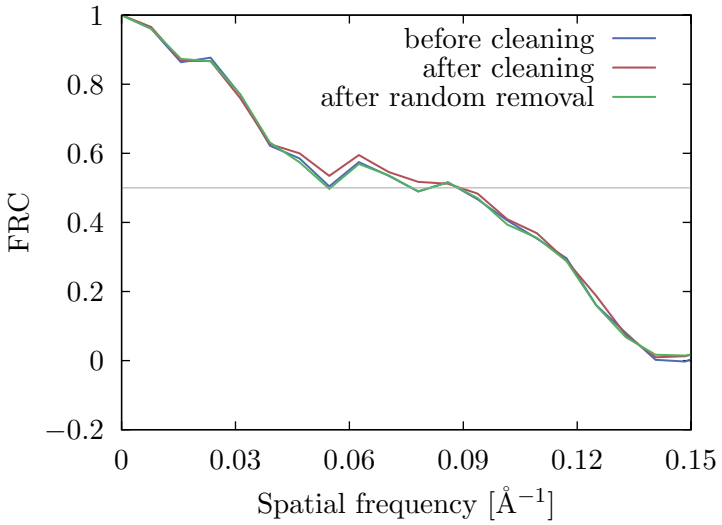

Figure 3.19: Removal of low quality images. Using the Imagic software, 1000 images of a cryo-EM dataset (70S ribosome) were clustered into 50 classes. The blue curve shows the corresponding average FRC. The red curve corresponds to the average FRC after removal of the worst 20 images as identified by Crystalign. In green the same analysis is shown for a random removal of 20 images.

2 % of the total dataset. For comparison 20 images were removed at random and the remaining images scored. Figure 3.19 shows that - also if minor in effect - the total quality was improved after removal of the bad images identified by Crystalign, whereas the random removal had no effect on the overall quality.

3.5 Performance issues in the light of parallel programming

Performance is a big issue for cryo-EM related image processing. The amount of single particle images to be processed is increasing fast. Even worse, with advent of the latest generation of CCD detectors, also the absolute number of pixels per image will increase. To date, images in the range of 64x64 – 128x128 pixels are most commonly processed. However, in near future pixel frames of 256x256 or even 512x512 will be the common format. The quadratic increase of the total pixels by doubling the edge length will quickly swamp algorithms not prepared to handle that. The routines described within this thesis are all developed with this future demands in mind. Whenever possible, parallel code was developed and the most important core functions were implemented twice, once in C++

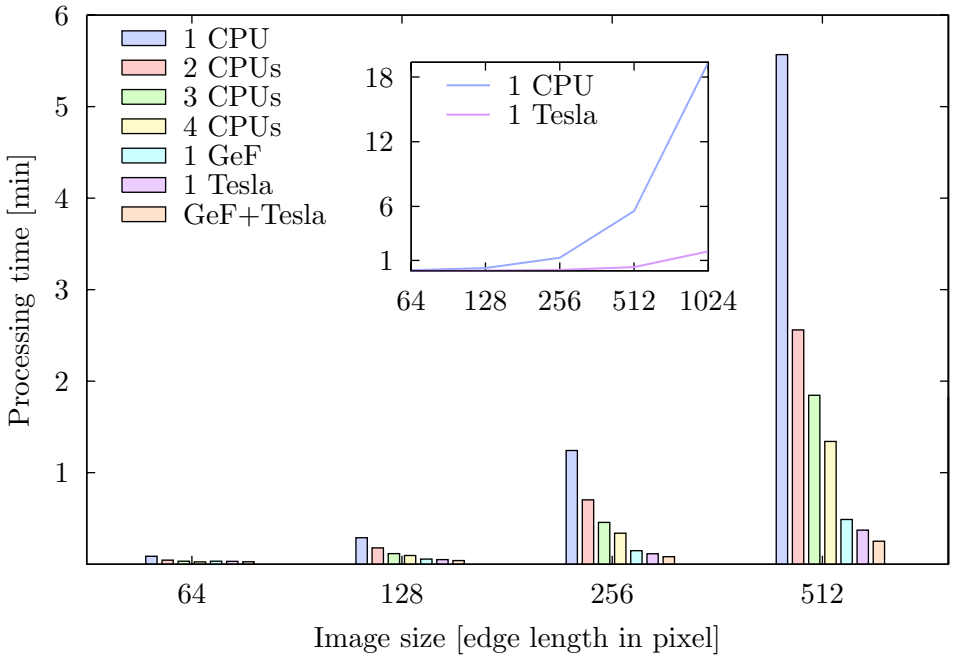

Figure 3.20: Speed measures for a "naive" implementation of the Radon transformation on 24 squared images with different sizes. Images were rotated over a total range of 180 degrees in intervals of 1 degree and subsequently projected down onto one line, resulting in the discrete Radon transform (Sinograms) of the latter. Times are taken for the hardware setup as described in Section 2.1. The inlet plot shows a direct comparison of one CPU *vs.* one NVidia Tesla board.

for standard CPU computation and another time in CUDA for graphics board device computation. A general observation regarding performance related to GPU code, was the strong dependency on the problem size. For small problem sizes the total execution time often was too short in order to weigh out the extra time necessary to set up the parallelization (e.g. host-to-device, device-to-host copies, post-processing of parallel data structures etc.), resulting in only minor overall speed-ups. However, as soon as the problem size increased the power of parallel computation became evident.

A good example of the described problematic is the computation of the discrete Radon transformation (sinogram), as is used for the routines described in this thesis but is also commonly used for image alignment.

Computing a sinogram requires generation of a set of image projections along angular

directions separated by $\Delta\varphi$. Two projections, $\pi/2$ apart, can be obtained by summing the pixels of all images rows and columns, and more are obtained by repeating the sums after stepwise rotation. For a sinogram being sampled in the interval $[0 - \pi]$, n rotations are needed with $n = \pi/(2 \cdot \Delta\varphi)$. As sinograms can be computed for each image individually, the input data set can easily be split and evenly distributed to the number of processors (be it CPU or GPU) available to be processed in parallel. A further, finer grained, level of parallelism can be achieved using the CUDA technology. For that, even the individual pixel operations necessary during rotation and projection, can be processed in parallel on the GPU. Figure 3.20 summarizes the behavior of the sinogram computation under various hardware configurations all available in a single desktop PC. As can be seen in the Figure 3.20 the full power of parallel processing gets unleashed only under increased sample size. Although the problem size gets squared with each data point in Figure 3.20 the processing time for the GPU keeps almost linear. The maximum speed up to be achieved by the GPU computation in comparison to one CPU for this example is by a factor of $\sim$15.

Chapter 4

Discussion

4.1 Classification - Iterative *vs.* multivariate data analysis

In order to judge the potential of the new classification strategy it is of importance to understand the fundamental differences of this method compared to the existing multivariate data analysis (MDA) based approaches. The most striking difference lies in the context upon which the decisions of class membership are taken. In the case of MDA, the context is set up *once* in the beginning by defining a new reduced coordinate system with basis-vectors corresponding to features of highest variance relative to the *whole* dataset[1]. Classification finally is done on the images expressed in this reduced coordinate system. This strategy has some striking advantages, as it prevents severe misclassifications in separating images according to in significance decreasing order of features, which are found relative to all features present in the whole data set. Furthermore, by taking only few significant features a drastic data reduction is achieved (typically more than 70-fold) which reduces computation time correspondingly. However, a disadvantage of this approach is that during feature extraction every pixel of each image comes into the evaluation with the same weight. In other words, images of bad intrinsic quality will equivalently take part in the feature extraction process and thus will add noise which may hide fine differences (i.e. features with less total variance in respect of all data) possibly existing between images of very good intrinsic quality. Strategies to overcome this problem by applying multiple MDA analyses on ever similar image subsets are under development (e.g. “Cluster Tracking” (Fu et al., 2007)).

[1]Technically this most commonly is achieved by means of Principle Component Analysis, Correspondence Analysis, Self Organizing Maps (Pascual-Montano et al., 2001), etc.

In contrast, for the classification strategy described here no fixed context is set up. In fact, several contexts are generated (as many as there are classes) which evolve and adapt during optimization. As described in Section 2.3.1.1 each image is evaluated with respect to the class it is currently in. The biggest advantage of this strategy over the MDA approach, is that the classification decision is taken according to a local context, thus allowing to decide on finer or rougher criteria depending on the intrinsic quality of the context. Hence, with increasingly improving context the classification criteria also adaptively get increasingly sharper. An obvious disadvantage of this strategy is the start-up phase, in the beginning the contexts (classes) are of small size and hence the classification criteria are fuzzy. If for this very reason *not* well fitting images are added in the beginning the context will not sharpen but rather blur. The situation thus is in danger of self-enhancing its unwanted behavior and may lead to severe misclassifications[2].

Another fundamental difference is based on the decision of how many classes with which amount of members should be generated. For the MDA approach this decision can not easily be done in an automatic fashion. As the feature extraction is a data relative process, no generic threshold is easily found that would define how many of the features (Eigenimages) are still describing the signal component or already analyze noise statistics. This problem is well known and methods to circumvent this problem have been designed. Frank et al. (1993) for example, describe a method that finds a significance threshold by producing an artificially generated noise-only control out of the same data. In essence all approaches will depend on a relative identification of noise and signal given the input data only. Thus, the problem simply remains that at no time an absolute quality measure contributes to the decision, of which images are going to be classified together. As an exaggerated theoretical situation of a misleading signal/noise decomposition one could imagine an inproperly working CCD. This chip may have a region of so called "hot-pixels" on the CCD camera which would introduce a systematic intensity increase for all images collected at this very region. In terms of MDA the absence and presence of this intensity increase on different images would be identified as a strong signal feature relative to all images available. An FRC analysis for images clustered in this way would however show weak results as the few hot pixels would represent only a minor, local correlation.

For the iterative process described here a mixture of relative signal/noise decomposition

[2]Crystallographers will call this "precipitation"

and absolute scoring is used. This duality is expressed in the two scoring functions S and S^i (see Section 2.3.1.1). The absolute score S being a variant of the Q-Factor is Fourier-based and thus data independent, whereas the S^i score is a statistical analysis relative to the current class context with a dynamic detection and down-weighting of noise components in Fourier space (which implies that individual image scores S^i can not be compared amongst different classes). The advantage of this duality is that an absolute threshold for class-quality may be defined (compare (2.4)) which allows for unsupervised evolution of number and sizes of classes and additionally allows inter-dataset quality comparisons. The challenge is to find an appropriate combination of these two measures under varying class sizes. Throughout this thesis several ideas have been implemented with respect to this issue. None of them were completely satisfactory nor is the currently demonstrated one, which tends to produce too small classes. This issue is regarded to be one of the key problems of the new approach and is still under active investigation.

Concerning the fundamental differences in the principle design of the new classification strategy, surprisingly good results are obtained in comparison to the established procedures. This result encourages further investigation and improvement of this new method. Some ideas that will be implemented and tested in near future are outlined in the following sections.

4.1.1 Modified scoring function and unit cell setup

It was mentioned above that a key issue for classification quality is believed to be the correct combination of a relative (statistical) and an absolute scoring. To this end another, possibly superior, scoring function utilizing a variant of the SSNR (see Equation (1.23)) will be practically investigated in future and is theoretically introduced in the following outline. Starting from the raw input images the preparation process for each image will be the following:

1) Image masking (circular mask) and normalization

2a) Optionally: Auto/Self-correlation to achieve translational invariance

3) Computation of the discrete Radon transform (sinogram)

3a) Optionally: Auto/Self-correlation to achieve rotational invariance

4) 1D-FFT on all rows (i.e. for constant polar angle φ)

5) Multiplication of each row with a response function to balance the uneven sampling in Fourier space

6) Transposition of the complex image (i.e. exchange of rows and columns)

The discrete Radon transformation in combination with the subsequent row-wise 1D-Fourier transformation can be understood as the polar sampling of the 2D Fourier transform of the image. Polar samplings of evenly spaced grids always lead to uneven, radius dependent sampling, with a relative oversampling for small radii (low spatial frequencies) and a relative undersampling at larger radii (high spatial frequencies). The sampling difference has a linear relationship and is thus easily corrected for by multiplication with a ramping function in Fourier space[3] (or equivalently by convolution of the ramping function in real space). After final transposition, the image rows represent evenly sampled Fourier coefficients of same spatial frequency, well suited for efficient scoring.

The total score S could be computed using the SSNR (1.23):

$$\mathrm{S} = max(\boldsymbol{k}) \quad \text{for} \quad \mathrm{SSNR}(\boldsymbol{k}) > t \tag{4.1}$$

by finding the maximum resolution $|\boldsymbol{k}| = k$ in terms of the highest spatial frequency available upon an appropriate threshold t (commonly a value of $t = 4$ is reported to be appropriate (Unser et al., 1987)).

The single image scores could be found similar to Equations (2.5)-(2.8) but weighted by the SSNR for the corresponding frequency instead by the Q-Factor:

$$\mathrm{S}^i = \frac{d_i - \bar{d}}{\sigma_d} \tag{4.2}$$

with

$$d_i = \sum_{k=1}^{K} \mathrm{SSNR}(\boldsymbol{k}) \cdot (F_i(\boldsymbol{k}) - \bar{F}(\boldsymbol{k}))^2 \tag{4.3}$$

where

$$\bar{d} = \frac{1}{N} \sum_{i}^{N} d_i \,, \qquad \sigma_{d_i} = \sqrt{\frac{(d_i - \bar{d})^2}{N-1}} \tag{4.4}$$

and

$$\bar{F}(\boldsymbol{k}) = \frac{1}{N} \sum_{i=1}^{N} F_i(\boldsymbol{k}) \tag{4.5}$$

[3] Electron microscopists will know this correction from the filtered back-projection reconstruction

with $F_i(\boldsymbol{k})$ being the Fourier transform of the i'th image, K the maximum frequency (Nyquist), and N the total number of images per class.

The described modified scoring may be superior to the presented one as the SSNR is believed to have a better statistical performance than Q-Factor, DPR or FRC especially for smaller class sizes. Moreover, as Unser et al. (1987) pointed out, the SSNR directly relates to the Fourier based resolution criteria commonly used in crystallography, hence may fit the optimization approach of growing crystals *in-silico* best.

Another improvement - not in scoring - but in computational performance will certainly be achieved by implementing another, faster algorithm for the Radon-transform as described in Lanzavecchia et al. (1996). This algorithms makes uses of the relationship between Radon- and Fourier transform, i.e. uses a 2D FFT approach with a polar sampling in Fourier space to compute the sinograms.

4.1.2 Taking the best from two worlds - Ideas for a hybrid approach

As outlined above the biggest problem with the current optimization strategy is in the very beginning. Crystals are still small and thus weak in their ability to decide for well fitting unit cells. This problem could be overcome by a seeding approach, i.e. the optimization process is started on already existing small crystals of sufficient quality. The preparation of those seeding crystals may be done by a MDA approach like PCA directly on the complex unit cells or on the real input images. An advantage of using the complex unit cells for PCA would be an already in dimensionality reduced input dataset as high spatial frequencies could be omitted in the first place. Using a sufficiently high enough and fixed number of Eigenimages an initial classification resulting in crystals of ~ 10 members in size could be performed. Subsequently, crystals could be sorted by quality (making use of the absolute class score S) and the worst fraction of crystals could be solvated, such that the optimization process as described in Section 2.3.2.2 is initialized with partly filled containers for both sides, the best seeding crystals in the crystal map and the worst, into unit cells solvated ones as part of the linear queue of unit cells. Of course, any already existing MDA approach with appropriately adjusted parameters could be initially used for testing, and the Crystalign part could be implemented similarly to the clean module, as a “refine” module.

4.2 The *align* in Crystalign

In Section 1.4 it was formulated that the separation of image classification and alignment is believed to be in danger of introducing model-bias. However, any approach that would exhaustively align and classify all combinations of images including any combination of translation and rotation per image is deemed to fail (even on the latest available hardware) due to the combinatorial explosion of available possibilities. Therefore, the only practical feasible strategy is separation. It is the uncoupling of this two processes which is left as a solution to the problem, and a way to do so is to classify using only features which are not used for alignment. Section 3.2.1.1 shows that utilizing the new scoring functionality image discrimination (the basis for classification) even under presence of relatively high amounts of noise is theoretically possible. Although finer details may not be separated, this does not harm the overall strategy. Alignment of images that are different in details only will still be meaningful and sufficiently bias-free. Resolution lost during alignment and averaging of slightly different images can easily be reconstituted by another cycle of classification, yet on already aligned images.

As for all described scoring procedures, sinograms of each image are computed and a sinogram-based alignment like the one described by Lanzavecchia et al. (1996) will be implemented and tested for performance in future.

4.3 Streamlining the process from raw data to 3D structure

Irrespective the experimental method used, the elucidation of a 3D structure is a complex and complicated process. Only concerning the computational part of the process, the amount of available expert routine's (available for all disciplines, i.e. NMR, Crystallography, and cryo-EM) is overwhelming. The correct combination of the routines and the adjustment of each routines' individual parameters for maximum performance require in-depth knowledge of the underlying theory and an up-to-date overview about all software available. Furthermore, to connect individual routines into a global work flow requires programming skills at least on a scripting level in order to let the routines communicate with each other (i.e. fit input and output formats etc.). In context of the advanced experimental methods, which too require a serious training, it seems impossible to equip oneself with all skills necessary to solve a 3D structure of an biological macromolecule independently and in a reasonable amount of time.

For this reason the crystallographic community started to think of the structure elucidation process as one global process, hence time was invested to minimize all user-interactions needed and replace them with intelligent automatic decision making and parameter adjustment. Those ideas peaked in the so-called *Collaborative Computational Project* (Collaborative Computational Project, 1994) which not only enabled beginners in the field to be able to use existing software and solve structures but also scientifically added more consistency, reliability and reproducibility to the whole process. Besides other specialized software pipelines (e.g. Auto-Rickshaw (Panjikar et al., 2005) for improved synchrotron data collection at EMBL Hamburg) at least one other package is becoming more and more powerful and more widely used and is called PHENIX (Adams et al., 2002). To the author's opinion the success of PHENIX is a result of essential design goals met in this software:

i) Leading scientists in the field have decided to work with each other and cooperate on a joint software, rather than everyone crafting an individual solution.

ii) The software is rather complete (thanks to i) in also small details that are still necessary for structure solution. This completeness avoids the need for using any other software in between which would break the project flow and would add a huge amount of user-interaction to the process.

iii) The software can be used by beginners, however leaves experts the possibility to adjust the default parameters.

iv) Internally the software is designed in a well structured object-oriented pattern, which allows for easy extension by external software developers. Again the design is flexible, such that beginners in programming may use a high level scripting interface (here: Python) and experts may add fast performing modules in C++.

A similar product for cryo-EM, based on the PHENIX environment is currently under active development. The so-called SPARX (Hohn et al., 2007) suite aims to combine already existing excellent software from cryo-EM (e.g. EMAN (Ludtke et al., 1999), SPIDER (Frank et al., 1996)) and from X-ray crystallography.

However, in comparison to crystallography, a big obstacle for all pipelines in cryo-EM is the lack of several modules needed for automation, the most important being:

i) Automatic solution of the orientation problem, which needs the collection of tilted datasets and - more challenging - routines for robust and automated picking of tilt-pairs (i.e. projections of the same particle on raw images differing by the tilt angle only).

ii) Automatic detection and removal of images with bad quality throughout the whole refinement process.

iii) Bias-free ensemble refinement of several 3D structures representing the inherent heterogeneity of the dataset.

iv) Objective validation of the quality and correctness of the final 3D structures.

Only those pipelines which will feature all the described modules will be able to seriously set a milestone in automatic processing. It is thus the highest prioritized objective to close those gaps. With the efforts underlying this thesis at least the second point can be canceled from the list already.

Appendix A

Mathematical Fundamentals

A.1 Fourier theory

The Fourier transformation is a linear mapping of a function into a different representation. This representation can be understood as a linear combination of sine waves with different frequencies, amplitudes and phases. Physically, if the original function describes a process in a specific domain, the Fourier transform will describe the same information in the reciprocal domain (commonly referred to as *time domain* and *frequency domain*, respectively). Being a linear transformation one can go back and forth (indicated by the "$\Longleftrightarrow$" symbol) between the two representations (here f and F, respectively) by means of the Fourier transform equations:

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{2\pi i k x} dx \quad \Longleftrightarrow \quad f(x) = \int_{-\infty}^{\infty} F(k) e^{-2\pi i k x} dk \tag{A.1}$$

If x for example is measured in seconds, then k in equation (A.1) will be in cycles per seconds (i.e. in Hertz).

In the discrete case (A.1) can be written as:

$$F(k) = \sum_{k=0}^{N-1} f(x) e^{\frac{2\pi i k x}{N}} \quad \Longleftrightarrow \quad f(x) = \sum_{k=0}^{N-1} F(k) e^{\frac{-2\pi i k x}{N}} \tag{A.2}$$

Symmetries present in one domain will lead to special relationships in the other domain. Table A.1 summarizes those correspondences between the two domains. Other important correspondences between Fourier pairs are:

$$f(ax) \Longleftrightarrow \frac{1}{|a|} F\left(\frac{k}{a}\right) \qquad \text{time scaling} \tag{A.3}$$

$$\frac{1}{|b|} f\left(\frac{x}{b}\right) \Longleftrightarrow F(bk) \qquad \text{frequency scaling} \tag{A.4}$$

$$f(x - x_0) \Longleftrightarrow F(k) e^{2\pi i k x_0} \qquad \text{time shifting} \tag{A.5}$$

$$f(x) e^{-2\pi i k_0 x} \Longleftrightarrow F(k - k_0) \qquad \text{frequency shifting} \tag{A.6}$$

With two functions $f(x)$ and $g(x)$, and their corresponding Fourier transforms $F(k)$ and $G(k)$ two combinations of special interest can be formed.

Table A.1: Correspondence between symmetries in the two Fourier related domains

If...	**then...**
$f(x)$ is real	$F(-k) = [F(k)]^*$
$f(x)$ is imaginary	$F(-k) = -[F(k)]^*$
$f(x)$ is even	$F(-k) = F(k)$
$f(x)$ is odd	$F(-k) = -F(k)$
$f(x)$ is real and even	$F(k)$ is real and even
$f(x)$ is real and odd	$F(k)$ is imaginary and odd
$f(x)$ is imaginary and even	$F(k)$ is imaginary and even
$f(x)$ is imaginary and odd	$F(k)$ is real and odd

A.1.1 Convolution

The convolution of the two functions, denoted $(f * g)(x)$, is defined by:

$$(f * g)(x) \equiv \int_{-\infty}^{\infty} f(\xi)g(x - \xi)\, d\xi \tag{A.7}$$

$(f * g)(x)$ is a function in the time domain and $(f * g)(x) = (g * f)(x)$. It turns out that the convolution can be written as a simple transform pair,

$$(f * g)(x) \Longleftrightarrow F(k)G(k) \quad \text{convolution theorem} \tag{A.8}$$

In words, the Fourier transform of the convolution is the product of the individual Fourier transforms.

A.1.2 Correlation

The correlation of two functions, denoted $(f \star g)(x)$, is defined by:

$$(f \star g)(x) \equiv \int_{-\infty}^{\infty} f(\xi + x)g(\xi)\, d\xi \tag{A.9}$$

The correlation is a function of x, which is called the *lag*. It therefore lies in the time domain, and turns out to be a member of the Fourier transform pair:

$$(f \star g)(x) \Longleftrightarrow F(k)G^*(k) \quad \text{correlation theorem} \tag{A.10}$$

if f and g are real functions. This shows that multiplying the Fourier transform of one function by the complex conjugate of the Fourier transform of the other gives the Fourier transform of their correlation. The correlation of a function with itself is called its *autocorrelation*. In this case (A.9) becomes the transform pair

$$(f \star f)(x) \equiv \int_{-\infty}^{\infty} |F(x)|^2 \quad \text{Wiener-Khinchin theorem} \tag{A.11}$$

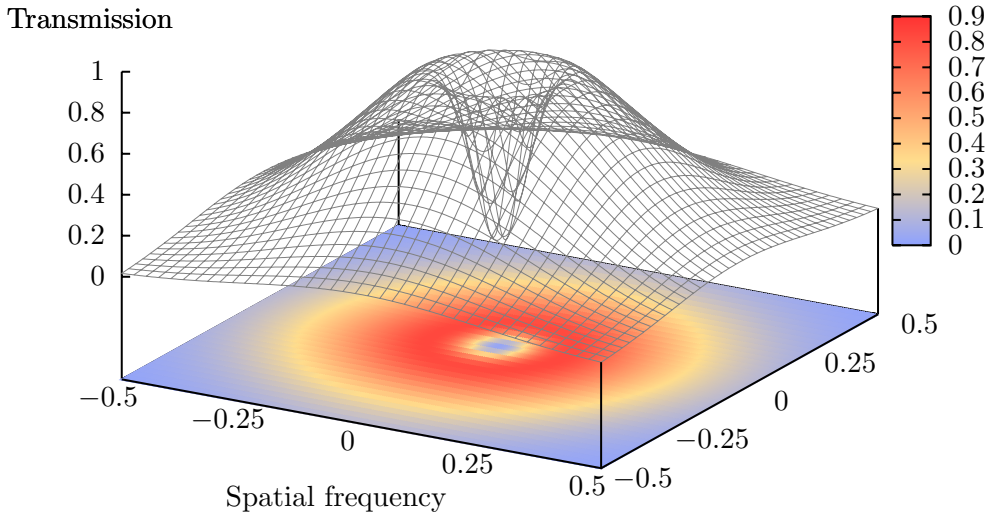

Figure A.1: Typical 2D-Gaussian band-pass filter profile. Plotted is the filter profile with parameters $trans = 0.0$, $freqLow = 0.15$, and $freqHigh = 0.70$ (see text for more details). The frequency cut-offs define spatial frequencies for which the transmission is reduced to e^{-1} and are indicated in yellow.

A.1.3 Power spectrum

The *total power* in a signal is of course the same whether it is computed in the time or frequency domain. It is known as the *Parseval's theorem*:

$$\text{total power} \equiv \int_{-\infty}^{\infty} |f(x)|^2 \, dt = \int_{-\infty}^{\infty} |F(k)|^2 \, dk \tag{A.12}$$

What is used here as the power spectrum is the power contained in the frequency interval between 0 ("zero frequency" of D.C.) to $+\infty$ which commonly is called the *one-sided power spectral density (PSD)* of the function f:

$$P_f(k) \equiv |F(k)|^2 + |F(-k)|^2 \qquad 0 \leq k < \infty \tag{A.13}$$

When f(x) is real, the two terms in (A.13) are equal such that $P_f(k) = 2|F(k)|^2$.

A.2 Fourier based Gaussian image filtering

Gaussian image filters are 2D Gaussian functions in the frequency domain. One commonly differentiates three types of filtering, *low-pass* (higher frequencies are downweighted), *high-pass* (lower

frequencies are down weighted), and *band-pass* (lower and higher frequencies are down weighted). Profiles as used within this thesis are generated by (only shown for square images for clarity):

$$G(k_x, k_y) = \left[1 - p \cdot e^{-\left(\frac{k_x^2}{l_x} + \frac{k_y^2}{l_y}\right)}\right] e^{-\left(\frac{k_x^2}{h_x} + \frac{k_y^2}{h_y}\right)} \tag{A.14}$$

with

$$p = 1 - trans \tag{A.15}$$

$$l_{x,y} = (freqLow \cdot r_{x,y})^2 \tag{A.16}$$

$$h_{x,y} = (freqHigh \cdot r_{x,y})^2 \tag{A.17}$$

where $r_{x,y}$ describes the x and y radius of the original image, respectively. Three parameters are used to define the profile, *trans* (residual transmission of low frequencies), *freqLow*, and *freqHigh* (low- and high frequency cutoff, respectively). A profile for a typical band-pass filter is illustrated in Figure A.1. The filtered image $f_{fil}(x, y)$ is computing from the original 2D image f(x,y) by the convolution with the filter profile:

$$f_{fil}(x, y) = \mathfrak{F}^{-1}[F(x, y)G(x, y)] \tag{A.18}$$

where $F(x, y)$ denotes the 2D Fourier transform of $f(x, y)$ and $\mathfrak{F}^{-1}$ indicates a reverse 2D Fourier transformation.

A.3 Fourier-slice theorem

The Fourier-slice theorem (also known as *projection theorem*) states that the Fourier transform of the projection of a 2D function $f(x, y)$ onto a single line is equal to a slice through the origin of the 2D Fourier transform of that function which is parallel to the projection line. The theorem can be extended to N dimensions and is of special use for cryo-EM, as the 2D projection images can be reconstructed to its 3D object by assembling their corresponding 2D slices in a 3D Fourier space and subsequently computing the inverse 3D Fourier transform to yield a real representation of the 3D object.

For clarity the proof of the theorem will be shown for the 2D case and with the projection line taken as the x-axis of the 2D function $f(x, y)$. The proof can easily be extended for higher dimensions and other projection lines. Let $f(x, y)$ denote a 2D function then the projection of onto the x axis is $p(x)$ where

$$p(x) = \int_{-\infty}^{\infty} f(x, y)\, dy.$$

The Fourier transform of $f(x, y)$ is

$$F(k_x, k_y) = \iint_{-\infty}^{\infty} f(x, y) e^{-2\pi i(xk_x + yk_y)}\, dx\, dy.$$

The slice $s(k_x)$ is then

$$\begin{aligned} s(k_x) = F(k_x, 0) &= \int_{-\infty}^{\infty}\int f(x,y)e^{-2\pi x k_x}\,dx\,dy \\ &= \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f(x,y)\,dy\right] e^{-2\pi x k_x}\,dx \\ &= \int_{-\infty}^{\infty} p(x)e^{-2\pi x k_x}\,dx \end{aligned}$$

which is just the Fourier transform of $p(x)$.

A.4 Radon transform

The 2D Radon transform (in its discrete form also known as *sinogram*) of a function $f(\boldsymbol{r})$ can be defined as:

$$f_{radon}(p,\xi) = \int f(\boldsymbol{r})\delta(p - \xi^T\boldsymbol{r})\,d\boldsymbol{r} \tag{A.19}$$

where $\boldsymbol{r} = (x,y)^T$ and $\delta(p - \xi^T\boldsymbol{r})$ represents a line defined by the direction of the (normal) unit vector ξ. One can think of the 2D Radon transform as a systematic stack of 1D projections of the original image under different rotation angles (defined by the direction of the vector ξ). The Radon transform is, like the Fourier transform, an integral transformation, and thus both transforms are related. A more detailed mathematical treatment is out of the scope of this thesis but is referred to Helgason (1999).

List of Symbols and Abbreviations

1D	One-dimensional
2D	Two-dimensional
3D	Three-dimensional
Å	Ångstrom (1 Å = 1^{-10} m)
AC	Auto-Correlation
APC	Anaphase Promoting Complex
ART	Algebraic Reconstruction Technique
CA	Correspondence Analysis
CCC	Cross Correlation Coefficient
CCD	Charge Coupled Device
CPU	Central Processing Unit
cryo-EM	Cryogenic Electron Microscopy
CUDA	Compute Unified Device Architecture
DFT	Discrete Fourier Transformation
DPR	Differential Phase Residual
FFT	Discrete Fast Fourier Transformation
FRC	Fourier Ring Correlation
FSC	Fourier Shell Correlation
GPU	Graphical Processing Unit
HAC	Hierarchical Ascending Classification
HDF	Hierarchical Data Format
HPC	High Performance Computing
MDA	Multivariate Data Analysis
MPI	Message Passing Interface
NVCC	NVidia C Compiler
PCA	Principal Component Analysis
PhCTF	Phase-Contrast Transfer Function
RCT	Random Conical Tilt
SC	Self-Correlation
SIRT	Simultaneous Iterative Reconstruction Technique
SNR	Signal-To-Noise Ratio
SSNR	Spectral Signal-To-Noise Ratio
STL	Standard Template Library
TEM	Transmission Electron Microscopy

Bibliography

P. D. Adams, R. W. Grosse-Kunstleve, L. W. Hung, T. R. Ioerger, A. J. McCoy, N. W. Moriarty, R. J. Read, J. C. Sacchettini, N. K. Sauter, and T. C. Terwilliger. Phenix: building new software for automated crystallographic structure determination. *Acta Crystallogr D Biol Crystallogr*, 58 (Pt 11):1948–1954, Nov 2002.

S. S. Alhir. *UML in a Nutshell.* O'Reilly, 1998.

D. P. Anderson. Boinc: A system for public-resource computing and storage. In *GRID '04: Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing*, pages 4–10, 2004.

J. P. Benoit and J. Doucet. Diffuse scattering in protein crystallography. *Q Rev Biophys*, 28(2): 131–169, May 1995.

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Res*, 28(1):235–242, Jan 2000.

M. Biadene, I. Hazemann, A. Cousido, S. Ginell, A. Joachimiak, G. M. Sheldrick, A. Podjarny, and T. R. Schneider. The atomic resolution structure of human aldose reductase reveals that rearrangement of a bound ligand allows the opening of the safety-belt loop. *Acta Crystallogr D Biol Crystallogr*, 63(Pt 6):665–672, Jun 2007. DOI 10.1107/S0907444907011997. URL `http://dx.doi.org/10.1107/S0907444907011997`.

B. Böttcher, S. A. Wynne, and R. A. Crowther. Determination of the fold of the core protein of hepatitis b virus by electron cryomicroscopy. *Nature*, 386(6620):88–91, Mar 1997. DOI 10.1038/386088a0. URL `http://dx.doi.org/10.1038/386088a0`.

A. T. Brünger. Free r value: a novel statistical quantity for assessing the accuracy of crystal structures. *Nature*, 355(6359):472–475, Jan 1992.

B. Busche. New algorithms for automated processing of electron-microscopic data. Master's thesis, Applied Computer Science, Max-Planck-Institute for Biophysical Chemistry, 2009.

R. Cambie, K. H. Downing, D. Typke, R. M. Glaeser, and J. Jin. Design of a microfabricated, two-electrode phase-contrast element suitable for electron microscopy. *Ultramicroscopy*, 107(4-5):329–339, 2007. DOI 10.1016/j.ultramic.2006.09.001. URL `http://dx.doi.org/10.1016/j.ultramic.2006.09.001`.

J. Z. Chen and N. Grigorieff. Signature: a single-particle selection system for molecular electron microscopy. *J Struct Biol*, 157(1):168–173, Jan 2007. DOI 10.1016/j.jsb.2006.06.001. URL `http://dx.doi.org/10.1016/j.jsb.2006.06.001`.

N. . Collaborative Computational Project. The ccp4 suite: programs for protein crystallography. *Acta Crystallogr D Biol Crystallogr*, 50(Pt 5):760–763, Sep 1994.

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

D. de Rosier and A. Klug. Reconstruction of three dimensional structures from electron micrographs. *Nature*, 217:130–134, 1968.

M. Diaconu, U. Kothe, F. Schlünzen, N. Fischer, J. M. Harms, A. G. Tonevitsky, H. Stark, M. V. Rodnina, and M. C. Wahl. Structural basis for the function of the ribosomal l7/12 stalk in factor binding and gtpase activation. *Cell*, 121(7):991–1004, Jul 2005. DOI 10.1016/j.cell.2005.04.015. URL `http://dx.doi.org/10.1016/j.cell.2005.04.015`.

J. Frank. Single-particle imaging of macromolecules by cryo-electron microscopy. *Annu Rev Biophys Biomol Struct*, 31:303–319, 2002. DOI 10.1146/annurev.biophys.31.082901.134202. URL `http://dx.doi.org/10.1146/annurev.biophys.31.082901.134202`.

J. Frank. *Three-Dimensional Electron Microscopy*. Oxford University Press, Inc., 2006.

J. Frank, A. Verschoor, and M. Boublik. Computer averaging of electron micrographs of 40s ribosomal subunits. *Science*, 214(4527):1353–1355, Dec 1981.

J. Frank, P. Penczek, and W. Liu. Alignment, classification, and three-dimensional reconstruction of single particles embedded in ice. *Scanning Microsc Suppl*, 6:11–20; discussion 20–2, 1992.

J. Frank, W. Chiu, and R. Henderson. Flopping polypeptide chains and suleika's subtle imperfections: analysis of variations in the electron micrograph of a purple membrane crystal. *Ultramicroscopy*, 49(1-4):387–396, Feb 1993.

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(1):190–199, 1996. DOI 10.1006/jsbi.1996.0030. URL `http://dx.doi.org/10.1006/jsbi.1996.0030`.

J. Fu, H. Gao, and J. Frank. Unsupervised classification of single particles by cluster tracking in multi-dimensional space. *J Struct Biol*, 157(1):226–239, Jan 2007. DOI 10.1016/j.jsb.2006.06.012. URL `http://dx.doi.org/10.1016/j.jsb.2006.06.012`.

E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns - Elements of Reusable Objet-Oriented Software*. Addison-Wesley, 2005.

P. Gilbert. Iterative methods for the three-dimensional reconstruction of an object from projections. *J Theor Biol*, 36(1):105–117, Jul 1972.

M. M. Golas, C. Böhm, B. Sander, K. Effenberger, M. Brecht, H. Stark, and H. U. Göringer. Snapshots of the rna editing machine in trypanosomes captured at different assembly stages in vivo. *EMBO J*, 28(6):766–778, Mar 2009. DOI 10.1038/emboj.2009.19. URL `http://dx.doi.org/10.1038/emboj.2009.19`.

N. Grigorieff. Frealign: high-resolution refinement of single particle structures. *J Struct Biol*, 157 (1):117–125, Jan 2007. DOI 10.1016/j.jsb.2006.05.004. URL `http://dx.doi.org/10.1016/j.jsb.2006.05.004`.

G. Harauz and M. van Heel. Direct 3d reconstruction from projections with initially unknown angles. *Pattern Recognition and Image Processing*, 2:279–288, 1986.

S. Helgason. *The Radon Transform Second Edition*. Birkhäuser, 1999.

F. Herzog, I. Primorac, P. Dube, P. Lenart, B. Sander, K. Mechtler, H. Stark, and J.-M. Peters. Structure of the anaphase-promoting complex/cyclosome interacting with a mitotic checkpoint complex. *Science*, 323(5920):1477–1481, Mar 2009. DOI 10.1126/science.1163300. URL `http://dx.doi.org/10.1126/science.1163300`.

M. Hohn, G. Tang, G. Goodyear, P. R. Baldwin, Z. Huang, P. A. Penczek, C. Yang, R. M. Glaeser, P. D. Adams, and S. J. Ludtke. Sparx, a new environment for cryo-em image processing. *J Struct Biol*, 157(1):47–55, Jan 2007. DOI 10.1016/j.jsb.2006.07.003. URL `http://dx.doi.org/10.1016/j.jsb.2006.07.003`.

Z. Huang, P. R. Baldwin, S. Mullapudi, and P. A. Penczek. Automated determination of parameters describing power spectra of micrograph images in electron microscopy. *J Struct Biol*, 144(1-2): 79–94, 2003.

L. Joyeux and P. A. Penczek. Efficiency of 2d alignment methods. *Ultramicroscopy*, 92(2):33–46, Jul 2002.

W. Kauzmann. *Quantum Chemistry*. New York: Academic Press, 1957.

J. C. Kendrew, G. Bodo, H. M. Dintzis, R. G. Parrish, H. Wyckoff, and D. C. Philipps. A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. *Nature*, 181(4610): 662–666, Mar 1958.

M. Kessel, M. Radermacher, and J. Frank. The structure of the stalk surface layer of a brine pond microorganism: correlation averaging applied to a double layered lattice structure. *J Microsc*, 139(Pt 1):63–74, Jul 1985.

Khronos-Group. Opencl - the open standard for parallel programming of heterogeneous systems. 2008. URL `http://www.khronos.org/news/press/releases/khronos_launches_heterogeneous_computing_initiative`.

A. Kidera, K. Inaka, M. Matsushima, and N. Go. Normal mode refinement: crystallographic refinement of protein dynamic structure. ii. application to human lysozyme. *J Mol Biol*, 225(2): 477–486, May 1992.

S. Lanzavecchia, L. Tosoni, and P. L. Bellon. Fast sinogram computation and the sinogram-based alignment of images. *Comput Appl Biosci*, 12(6):531–537, Dec 1996.

J. Lepault and J. Dubochet. Electron microscopy of frozen hydrated specimens: preparation and characteristics. *Methods Enzymol*, 127:719–730, 1986.

A. E. Leschziner and E. Nogales. Visualizing flexibility at molecular resolution: analysis of heterogeneity in single-particle electron microscopy reconstructions. *Annu Rev Biophys Biomol Struct*, 36:43–62, 2007. DOI 10.1146/annurev.biophys.36.040306.132742. URL `http://dx.doi.org/10.1146/annurev.biophys.36.040306.132742`.

Y. Liang, E. Y. Ke, and Z. H. Zhou. Imirs: a high-resolution 3d reconstruction package integrated with a relational image database. *J Struct Biol*, 137(3):292–304, Mar 2002.

S. J. Ludtke, P. R. Baldwin, and W. Chiu. Eman: semiautomated software for high-resolution single-particle reconstructions. *J Struct Biol*, 128(1):82–97, Dec 1999. DOI 10.1006/jsbi.1999.4174. URL `http://dx.doi.org/10.1006/jsbi.1999.4174`.

M. Lüttich. *Analytische Methoden zur hochauflösenden Strukturbestimmung in der Kryo-Elektronen-Mikroskopie*. PhD thesis, Georg-August University Göttingen, 2007.

E. Majorovits, B. Barton, K. Schultheiss, F. Pérez-Willard, D. Gerthsen, and R. R. Schröder. Optimizing phase contrast in transmission electron microscopy with an electrostatic (boersch) phase plate. *Ultramicroscopy*, 107(2-3):213–226, 2007. DOI 10.1016/j.ultramic.2006.07.006. URL `http://dx.doi.org/10.1016/j.ultramic.2006.07.006`.

L. Mandelkern. *Crystallization of Polymers, 2nd ed., Volume 2: Kinetics and Mechanisms*. Cambridge University Press: Cambridge, 2001a.

L. Mandelkern. *Crystallization of Polymers, 2nd ed., Volume 1: Equilibrium Concepts*. Cambridge University Press: Cambridge, 2001b.

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(1-2):53–65, Apr 1998.

M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, 1998. ISSN 1049-3301. DOI http://doi.acm.org/10.1145/272991.272995.

J. A. Mindell and N. Grigorieff. Accurate determination of local defocus and specimen tilt in electron microscopy. *J Struct Biol*, 142(3):334–347, Jun 2003.

W. V. Nicholson and R. M. Glaeser. Review: automatic particle detection in electron microscopy. *J Struct Biol*, 133(2-3):90–101, 2001. DOI 10.1006/jsbi.2001.4348. URL `http://dx.doi.org/10.1006/jsbi.2001.4348`.

NVidia. *NVIDIA CUDA Compute Unified Device Architecture - Programming Guide*. NVidia Cooperation, 2009. URL `http://www.nvidia.com/object/cuda_develop.html`.

E. V. Orlova, P. Dube, J. R. Harris, E. Beckman, F. Zemlin, J. Markl, and M. van Heel. Structure of keyhole limpet hemocyanin type 1 (klh1) at 15 a resolution by electron cryomicroscopy and angular reconstitution. *J Mol Biol*, 271(3):417–437, Aug 1997. DOI 10.1006/jmbi.1997.1182. URL `http://dx.doi.org/10.1006/jmbi.1997.1182`.

F. P. Ottensmeyer, J. W. Andrew, D. P. Bazett-Jones, A. S. Chan, and J. Hewitt. Signal to noise enhancement in dark field electron micrographs of vasopressin: filtering of arrays of images in reciprocal space. *J Microsc*, 109(3):259–268, Apr 1977.

S. Panjikar, V. Parthasarathy, V. S. Lamzin, M. S. Weiss, and P. A. Tucker. Auto-rickshaw: an automated crystal structure determination platform as an efficient tool for the validation of an x-ray diffraction experiment. *Acta Crystallogr D Biol Crystallogr*, 61(Pt 4):449–457, Apr 2005. DOI 10.1107/S0907444905001307. URL `http://dx.doi.org/10.1107/S0907444905001307`.

S.-Y. Park and S. Hariri. A high performance message-passing system for network of workstations. *J. Supercomput.*, 11(2):159–180, 1997.

A. Pascual-Montano, L. E. Donate, M. Valle, M. Bárcena, R. D. Pascual-Marqui, and J. M. Carazo. A novel neural network technique for analysis and classification of em single-particle images. *J Struct Biol*, 133(2-3):233–245, 2001. DOI 10.1006/jsbi.2001.4369. URL `http://dx.doi.org/10.1006/jsbi.2001.4369`.

P. Penczek, M. Radermacher, and J. Frank. Three-dimensional reconstruction of single particles embedded in ice. *Ultramicroscopy*, 40(1):33–53, Jan 1992.

P. A. Penczek, R. A. Grassucci, and J. Frank. The ribosome at improved resolution: new techniques for merging and orientation refinement in 3d cryo-electron microscopy of biological particles. *Ultramicroscopy*, 53(3):251–270, Mar 1994.

P. A. Penczek, R. Renka, and H. Schomberg. Gridding-based direct fourier inversion of the three-dimensional ray transform. *J Opt Soc Am A Opt Image Sci Vis*, 21(4):499–509, Apr 2004.

M. Radermacher. Three-dimensional reconstruction of single particles from random and nonrandom tilt series. *J Electron Microsc Tech*, 9(4):359–394, Aug 1988. DOI 10.1002/jemt.1060090405. URL `http://dx.doi.org/10.1002/jemt.1060090405`.

M. Radermacher. *Electron Tomography*, chapter Weighted back-preojection methods, pages 91–115. Plenum, New York, 1992.

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(Pt 2):113–136, May 1987.

M. Radermacher, T. Ruiz, H. Wieczorek, and G. Grüber. The structure of the v(1)-atpase determined by three-dimensional electron microscopy of single particles. *J Struct Biol*, 135(1):26–37, Jul 2001. DOI 10.1006/jsbi.2001.4395. URL `http://dx.doi.org/10.1006/jsbi.2001.4395`.

L. Reimer and H. Kohl. *Transmission Electron Microscopy Physics of Image Formation*. Springer, 2007.

M. G. Rossmann. *International Tables for Crystallography Volume F*, chapter Chapter 1.2. Historical background, pages 4–9. Kluwer Academic Publishers, 2006.

E. Ruska. Die fruehe entwicklung der elektronenlinsen und der elektronenmikroskopie. *Acta Historica Leopoldina*, 12, 1979.

P. Salamon, P. Silbani, and R. Frost. *Facts, Conjectures, and Improvements for Simulated Annealing*. New York: SIAM Press, 2002.

L. Salmon, G. Bouvignies, P. Markwick, N. Lakomek, S. Showalter, D.-W. Li, K. Walter, C. Griesinger, R. Brüschweiler, and M. Blackledge. Protein conformational flexibility from structure-free analysis of nmr dipolar couplings: Quantitative and absolute determination of backbone motion in ubiquitin13. *Angewandte Chemie International Edition*, 48(23):4154–4157, 2009.

B. Sander, M. M. Golas, and H. Stark. Automatic ctf correction for single particles based upon multivariate statistical analysis of individual power spectra. *J Struct Biol*, 142(3):392–401, Jun 2003.

B. Sander, M. M. Golas, E. M. Makarov, H. Brahms, B. Kastner, R. Lührmann, and H. Stark. Organization of core spliceosomal components u5 snrna loop i and u4/u6 di-snrnp within u4/u6.u5 tri-snrnp as revealed by electron cryomicroscopy. *Mol Cell*, 24(2):267–278, Oct 2006. DOI 10.1016/j.molcel.2006.08.021. URL `http://dx.doi.org/10.1016/j.molcel.2006.08.021`.

W. O. Saxton and W. Baumeister. The correlation averaging of a regularly arranged bacterial cell envelope protein. *J Microsc*, 127(Pt 2):127–138, Aug 1982.

M. Schatz. *Invariante Klassifizierung elektronenmikroskopischer Aufnahmen von eiseingebetteten biologischen Makromolekülen*. PhD thesis, Fachbereich Physik der Freien Universität Berlin, 1992.

S. H. W. Scheres, R. Nunez-Ramirez, Y. Gomez-Llorente, C. S. Martín, P. P. B. Eggermont, and J. M. Carazo. Modeling experimental image formation for likelihood-based classification of electron microscopy data. *Structure*, 15(10):1167–1177, Oct 2007. DOI 10.1016/j.str.2007.09.003. URL `http://dx.doi.org/10.1016/j.str.2007.09.003`.

O. Scherzer. The theoretical resolution limit of the electron microscope. *J. Appl. Phys.*, 20:20, 1949.

M. Schmeisser. *New computational methods for 3D structure determination of macromolecular complexes by single particle cryo-electron microscopy*. PhD thesis, Georg-August University Göttingen, 2009.

M. Schmeisser, B. C. Heisen, M. Luettich, B. Busche, F. Hauer, T. Koske, K.-H. Knauber, and H. Stark. Parallel, distributed and gpu computing technologies in single-particle electron microscopy. *Acta Crystallogr D Biol Crystallogr*, 65(Pt 7):659–671, Jul 2009. DOI 10.1107/S0907444909011433. URL `http://dx.doi.org/10.1107/S0907444909011433`.

T. R. Schneider. What can we learn from anisotropic temperature factors ? *Proceedings of the CCP4 Study Weekend (E. Dodson, M. Moore, A. Ralph, and S. Bailey, eds.)*, pages 133–144, 1996.

R. R. Schroder, B. Barton, H. Rose, and G. Brenner. Constrast enhancement by anamorphotic phase plates in an aberration corrected tem. *Microscopy and Microanalysis*, 13:136–137, 2007.

H. Stark and R. Lührmann. Cryo-electron microscopy of spliceosomal components. *Annu Rev Biophys Biomol Struct*, 35:435–457, 2006. DOI 10.1146/annurev.biophys.35.040405.101953. URL `http://dx.doi.org/10.1146/annurev.biophys.35.040405.101953`.

G. Tang, L. Peng, P. R. Baldwin, D. S. Mann, W. Jiang, I. Rees, and S. J. Ludtke. Eman2: an extensible image processing suite for electron microscopy. *J Struct Biol*, 157(1):38–46, Jan 2007. DOI 10.1016/j.jsb.2006.05.009. URL `http://dx.doi.org/10.1016/j.jsb.2006.05.009`.

T. Thüne and J. Badger. Thermal diffuse x-ray scattering and its contribution to understanding protein dynamics. *Prog Biophys Mol Biol*, 63(3):251–276, 1995.

M. Unser, B. L. Trus, and A. C. Steven. A new resolution criterion based on spectral signal-to-noise ratios. *Ultramicroscopy*, 23(1):39–51, 1987.

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(13):3557–3567, Jul 2002. DOI 10.1093/emboj/cdf326. URL `http://dx.doi.org/10.1093/emboj/cdf326`.

M. van Heel and J. Hollenberg. *Electron Microscopy at Molecular Dimensions*, chapter The stretching of distorted images of two-dimensional crystals., pages 256–260. Springer Verlag, Berlin/NewYork, 1980.

M. van Heel and M. Stöffler-Meilicke. Characteristic views of e. coli and b. stearothermophilus 30s ribosomal subunits in the electron microscope. *EMBO J*, 4(9):2389–2395, Sep 1985.

M. van Heel, G. Harauz, E. V. Orlova, R. Schmidt, and M. Schatz. A new generation of the imagic image processing system. *J Struct Biol*, 116(1):17–24, 1996. DOI 10.1006/jsbi.1996.0004. URL `http://dx.doi.org/10.1006/jsbi.1996.0004`.

P. van Laarhoven and E. Aarts. *Simmulated Annealing: Theory and Applications*. Berlin: Springer Verlag, 1987.

R. Wade and J. Frank. Electron microscopic transfer functions for partially coherent axial illumination and chromatic defocus spread. *Optik*, 49:81–92, 1977.

F. Zernike. Phase contrast, a new method for the microscopic observation of transparent objects. *Physica*, 9:686, 1942.

Y. Zhu, B. Carragher, R. M. Glaeser, D. Fellmann, C. Bajaj, M. Bern, F. Mouche, F. de Haas, R. J. Hall, D. J. Kriegman, S. J. Ludtke, S. P. Mallick, P. A. Penczek, A. M. Roseman, F. J. Sigworth, N. Volkmann, and C. S. Potter. Automatic particle selection: results of a comparative study. *J Struct Biol*, 145(1-2):3–14, 2004.

Curriculum Vitae

Burkhard C. Heisen

born 30^{th} July 1980 in Wolfhagen, Germany

Im Winkel 9, 37077 Göttingen
Phone +49 (0) 551 8208048
E-Mail bheisen@gwdg.de

Education

2006 - present	Max Planck Institute for Biophysical Chemistry, Göttingen, Germany: Three-dimensional Electron Cryomicroscopy. PhD thesis: *New Algorithms for Macromolecular Structure Determination*
2005 - 2006	Georg-August-University, Göttingen, Germany: Structural Chemistry. Master Thesis: *New algorithms for crystal structure investigation*
2004 - 2005	Georg-August-University, Göttingen, Germany: International MSc/PhD program
2004	Stockholm University/KTH, Sweden: Studies in Molecular Biology, Scientific research project: *Transmembrane helix prediction using a genetic programming approach*
2001 - 2004	Universtity Lübeck, Germany: Molecular Biotechnology (BSc.).
1987 - 2000	Wilhelm-Filchner School, Wolfhagen, Germany (Abitur).

Scholarships

2005 - 2008	Stipend Georg-Christoph-Lichtenberg Stipend
2004 - 2005	Stipend International Max-Planck Research School

Entrepreneurship

2007 - present	Co Founder and Executive Director of *BitConf - Conference Software Solutions*