



Processing of Cryo-EM Movie Data

Z.A. Ripstein^{*,†}, J.L. Rubinstein^{*,†,1}

^{*}Molecular Structure and Function Program, The Hospital for Sick Children, Toronto, ON, Canada

[†]University of Toronto, Toronto, ON, Canada

¹Corresponding author: e-mail address: john.rubinstein@utoronto.ca

Contents

1. Introduction	103
2. Motioncorr	107
3. <i>alignframes_lmbfqs</i> and <i>alignparts_lmbfqs</i>	109
4. Unblur	112
5. Optical Flow	114
6. Particle Polishing in <i>Relion</i>	116
7. Summary	121
8. Future Prospects	122
Acknowledgments	123
References	123

Abstract

Direct detector device (DDD) cameras dramatically enhance the capabilities of electron cryomicroscopy (cryo-EM) due to their improved detective quantum efficiency (DQE) relative to other detectors. DDDs use semiconductor technology that allows micrographs to be recorded as movies rather than integrated individual exposures. Movies from DDDs improve cryo-EM in another, more surprising, way. DDD movies revealed beam-induced specimen movement as a major source of image degradation and provide a way to partially correct the problem by aligning frames or regions of frames to account for this specimen movement. In this chapter, we use a self-consistent mathematical notation to explain, compare, and contrast several of the most popular existing algorithms for computationally correcting specimen movement in DDD movies. We conclude by discussing future developments in algorithms for processing DDD movies that would extend the capabilities of cryo-EM even further.



1. INTRODUCTION

The introduction of direct detector device (DDD) cameras for electron microscopes has led to significant recent advances in the field of single particle electron cryomicroscopy (cryo-EM) (Kühlbrandt, 2014;

Smith & Rubinstein, 2014) and related fields such as electron cryo-tomography. The improved image signal-to-noise ratio (SNR) available from DDDs for ice-embedded protein particles has allowed for atomic models to be calculated directly from cryo-EM maps when protein complexes are sufficiently rigid (Li et al., 2013; Liao, Cao, Julius, & Cheng, 2013) and multiple conformations to be detected in dynamic protein complexes (Zhao, Benlekhir, & Rubinstein, 2015; Zhou et al., 2015). DDDs can improve image SNRs in four distinct ways. First, the inherent detective quantum efficiency (DQE) of the monolithic active pixel sensor is superior to that of photographic film or fiber-optic coupled charged coupled device (CCD) cameras (McMullan et al., 2009). Second, the high frame rate achievable due to the complimentary metal oxide semiconductor (CMOS) technology of the camera allows for individual electrons to be detected and counted in some camera designs, normalizing the contribution from each electron and further boosting the DQE of the camera (Li et al., 2013). Third, the high frame rate of the camera allows beam-induced movement and deformation of the specimen to be, at least partially, corrected by acquiring movies of specimens rather than integrated single exposures (Brilot et al., 2012; Campbell et al., 2012) (Fig. 1). Finally, the ability to collect movies rather than single integrated exposures allows high-resolution features in images to be recorded with low exposure and little radiation damage while low-resolution features, which are expected to be more radiation tolerant, may be imaged with higher total exposure (Baker & Rubinstein, 2010; Baker, Smith, Bueler, & Rubinstein, 2010).

At present, there are three commercial manufacturers of DDD cameras. Direct Electron LP produces the DE series of cameras with physical pixels that are between $6.0 \times 6.0 \mu\text{m}$ and $6.5 \times 6.5 \mu\text{m}$ and with frame rates

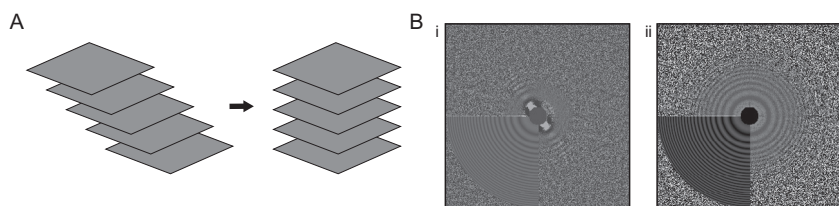


Fig. 1 (A) Cartoon depiction of whole-frame alignment. (B) Thon rings from an averaged DDD movie before (i) and after (ii) motion correction show that high-resolution information is restored by correcting for specimen movement. The bottom left corner of each power spectrum shows a CTF model fit to the Thon rings. CTF parameters were modeled and power spectra generated by CTFFIND4 (Rohou & Grigorieff, 2015).

between 30 and 60 frames per second (fps). FEI Company produces the Falcon cameras, which have $14.0 \times 14.0 \mu\text{m}$ pixels and a thick sensitive layer on the detector, which lead to a high DQE, but currently provide a frame rate of only 18 fps. Electron counting with the current versions of these two cameras is not yet considered practical for single particle cryo-EM but the frame rate is sufficient to allow for correction of beam-induced movement. Gatan Inc. produces the K2 camera, which has $5.0 \times 5.0 \mu\text{m}$ pixels and an internal frame rate of 400 fps, which allows for an approximate electron counting algorithm implemented on a dedicated field-programmable gate array (FPGA) computer. Movie frames that have been processed with the counting algorithm are added together and these sum images can be recorded at up to 40 fps. The different characteristics of the cameras suggest different optimal use strategies (McMullan, Faruqi, Clare, & Henderson, 2014). For integrating cameras such as the Direct Electron DE and FEI Falcon cameras, short movies with high frame rates are optimal. These short movies decrease the overall drift of the specimen holder during movie acquisition. For the Gatan K2 camera, the electron exposure rate must be limited in order to avoid coincidence loss during electron counting, where two or more electrons are counted as a single event due to their overlapping on the sensor during a single frame (Li et al., 2013; Li, Zheng, Egami, Agard, & Cheng, 2013). The slow exposure rate allows significant stage drift during the acquisition of the movie (Fig. 1). This review will describe several algorithms for whole-frame alignment. Cryo-EM of large virus particles with DDDs showed that, in addition to rigid body translation of entire movie frames, the specimen deforms during irradiation (Brilot et al., 2012). For large virus particles there is often sufficient signal in each movie frame to treat the frames of individual particles separately to account for this beam-induced displacement and rotation of particles during acquisition of the DDD movie. However, for smaller particles, algorithms are necessary to track particles during the movie using information available from considering the movie frames together. Therefore, in this review we also describe the current approaches for correcting this anisotropic movement of protein particles in images.

The objective of this chapter is to explain, compare, and contrast the current motion correction algorithms using self-consistent mathematical notation. At present, existing whole-frame and individual-particle motion correction approaches neglect any out-of-plane motion or rotation of the specimen during movie acquisition. Consequently, movement of the specimen or parts of the specimen between movie frames may be treated as a

translation of each frame. Due to differences in the various algorithms described in this chapter, the best way to represent these translations mathematically varies. When referring to the translation of each frame relative to a single reference, for example the first frame or the middle frame of the movie, we write the translation as $-\vec{x}_t = (-\Delta x_t, -\Delta y_t)$ for frame t , where the opposite translation \vec{x}_t is a vector that represents the best shifts, Δx_t and Δy_t , to apply to frame t to bring it into register with the frame that has been defined as stationary. Bringing these frames into register before averaging them optimizes the extraction of high-resolution information. Alternatively, for other algorithms, it is easier to describe the translations as $-\vec{s}_{mn} = (-\Delta x_{mn}, -\Delta y_{mn})$ for the movement from frame m to frame n . The aim of these algorithms is to identify the values of \vec{s}_{mn} to apply to each frame to bring the frames into register with each other. The two different types of unknown are related by $\vec{s}_{mn} = \vec{x}_m - \vec{x}_n$.

Despite the relatively recent wide-spread adoption of DDD cameras in cryo-EM, a variety of different algorithms for aligning movie frames and frame regions have already been described. We begin this review by describing the algorithm in the program *Motioncorr* (Li et al., 2013), a whole-frame alignment approach. *Motioncorr* is intended to be used with the raw movie output of DDD cameras to begin the image analysis procedure. Next, we describe the algorithm used in *alignframes_lmbfgs* and *alignparts_lmbfgs* (Rubinstein & Brubaker, 2015), whole-frame and individual-particle motion correction programs, respectively, that are closely related to each other. *alignframes_lmbfgs* is intended for use with the raw movie output of a DDD camera to allow contrast transfer function (CTF) parameter determination and particle coordinate selection, two of the initial steps in single particle cryo-EM image analysis. *alignparts_lmbfgs* is then used to extract particles from the raw movie output of the camera while simultaneously correcting for whole-frame movement, individual particle movement, and accounting for radiation damage of the particles. Subsequently, we describe two programs intended for whole-frame alignment or alignment of large regions of frames: the Optical Flow method (Abrishami et al., 2015) and *Unblur* (Grant & Grigorieff, 2015), the latter also accounts for the radiation damage that occurs throughout the imaging process. Finally, we describe the particle polishing procedure in *Relion* (Scheres, 2014), an individual particle motion correction algorithm applied to prealigned movie frames after map refinement. We conclude by comparing the attributes of these different algorithms.



2. MOTIONCORR

In the process of implementing methods to use the K2 Summit electron counting DDD, Li and colleagues developed the first popular method for aligning whole movie frames (Li et al., 2013), which has become known as *Motioncorr*. For a movie that consists of T frames, the translations required to bring frames into register to extract high-resolution information are \vec{s}_{mn} with $m \in \{1, T-1\}$ and $n = m+1$. That is, one must know the translation from each frame to the next, $-\vec{s}_{mn}$, in order to apply their inverse, \vec{s}_{mn} . *Motioncorr*, which is currently the most widely used method in the field for whole-frame alignment, determines the translations by calculating the pair-wise cross correlation functions between all of the frames in a movie. To obtain these cross correlation functions, the Fourier transform of each frame is calculated, and in a pixel-wise fashion each Fourier component is multiplied by the complex conjugate of the equivalent Fourier component in the Fourier transform of another frame. The inverse Fourier transform of this product is calculated to give a cross correlation function for each unique pair of movie frames (Bracewell, 1965), yielding $T/2 \times (T-1)$ cross correlation functions for the T frames of the movie. To avoid the effects of fixed pattern noise, images are pretreated with a filter that has the form $\exp\left(\frac{-B}{4d^2}\right)$, where B is a temperature factor and d is resolution. This filter down weights the high spatial frequencies that would cause fixed patterns in the image to dominate the cross correlation function. The maximum in each cross correlation function provides a vector \vec{m}_{mn} that is a measured estimate of how that pair of frames should be aligned. In the absence of noise, the measured displacement vectors from cross correlation (\vec{m} values) are equal to the desired translations (\vec{s} values). For example, in the absence of noise \vec{m}_{34} should equal \vec{s}_{34} while \vec{m}_{14} should equal $\vec{s}_{12} + \vec{s}_{23} + \vec{s}_{34}$. In the presence of noise, these relationships will not always hold true. The large number of \vec{m}_{mn} vectors dramatically overdetermines the $T-1$ needed \vec{s}_{mn} vectors that the method aims to discover. The measured \vec{m}_{mn} values and desired \vec{s}_{mn} values are therefore used to create a system of overdetermined linear equations as shown in Eq. (1), illustrated with an example movie that has four frames. Note that typically many more movie frames would be used, often between 7, the maximum available with a standard Falcon II detector

from FEI, and 100. With 4 frames, 6 independent linear equations can be generated and recorded in the matrices shown:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \vec{s}_{12} \\ \vec{s}_{23} \\ \vec{s}_{34} \end{bmatrix} = \begin{bmatrix} \vec{m}_{12} \\ \vec{m}_{13} \\ \vec{m}_{14} \\ \vec{m}_{23} \\ \vec{m}_{24} \\ \vec{m}_{34} \end{bmatrix} \quad (1)$$

A least squares fit for the values of \vec{s}_{mn} can be obtained from Eq. (1) using established methods from linear algebra (Li et al., 2013). Further improvement of the least squares solutions for the values of \vec{s}_{mn} may be achieved using several approaches implemented in *Motioncorr*. First, cross correlation peaks close to the origin, where their precise location can be masked by a large peak at the origin due to residual fixed-pattern noise, can be avoided by ignoring equations from subsequent or nearly subsequent frames. Second, the maxima in cross correlation functions can be localized to subpixel accuracy using Fourier padding to interpolate the shape of the maximum from the cross correlation function. Third, equations from the matrix can be discarded when the residual of $\vec{m}_{mn} - \sum_{l=m}^{n-1} \vec{s}_{l(l+1)}$ exceeds a predefined amount. This least-squares whole-frame alignment method allowed high-resolution structures to be determined for important biological macromolecules (Cao, Liao, Cheng, & Julius, 2013; Li et al., 2013; Liao et al., 2013). Because of the large number of cross correlation functions that the method is often required to calculate, there is the potential for the method to be quite computationally expensive. Consequently, the original implementation of the program was on a GPU computer. It was pointed out by the authors that the method is not able to align image regions smaller than 2000×2000 pixels for movies acquired using typical conditions, suggesting that the method is not appropriate for aligning individual particle images in order to correct for deformation of the ice layer during imaging. While exploring different frame alignment methods and individual particle alignment methods, one of us (JLR) implemented the algorithm in standard modern Fortran for use with a desktop CPU computer, a program that we now make available online at https://sites.google.com/site/rubinsteingroup/direct-detector-align_lmbfqs. We used the linear algebra library LAPACK to find the least squares fit for values of \vec{s}_{mn} . We found the method to be impressively robust with the 30-frame movies we produced in the laboratory from

a Gatan K2 camera. Because of how overdetermined the system of equations is with 30 frames, little gain was realized from subpixel maximum localization of the cross correlation function or removal of cross correlation functions above a minimum residual.



3. *alignframes_lmbfgs* and *alignparts_lmbfgs*

The programs *alignframes_lmbfgs* and *alignparts_lmbfgs* (Rubinstein & Brubaker, 2015) align whole movie frames and individual particles, respectively. In order to produce a robust and computationally efficient method, Rubinstein and Brubaker pose the problem of finding \vec{x}_t for the T frames as an optimization problem, either for whole movie frames or parts of frames corresponding to individual particle images. Optimization of functions, particularly where the partial derivatives of the objective function with respect to each of the parameters can be calculated analytically, is a well developed area in computer science. In this optimization approach, a single objective function that depends on all of the \vec{x}_t values is minimized by adjusting the values of \vec{x}_t . Shifting of frames in Fourier space does not require interpolation and consequently the approach is performed with Fourier transforms of images rather than the images themselves. In Fourier space, image translation is equivalent to a phase change by ϕ_{jt} , and each Fourier component of the shifted frame is given by $F_{jt}(\cos\phi_{jt} + i\sin\phi_{jt})$ where F_{jt} is the j^{th} Fourier component of the t^{th} unshifted frame. Alternatively, this shifted Fourier component can be written $F_{jt}S_{jt}$ where $S_{jt} = (\cos\phi_{jt} + i\sin\phi_{jt})$. The amount of phase change is given by

$$\phi_{jt} = k_x(j) \cdot x_t \frac{2\pi}{N} + k_y(j) \cdot y_t \frac{2\pi}{N} \quad (2)$$

where N is the extent in pixels in both the x and y direction of the $N \times N$ image, and $k_x(j)$ and $k_y(j)$ are the distance of the j^{th} Fourier component from the origin in the k_x and k_y directions, respectively. The objective function used in the *alignframes_lmbfgs* and *alignparts_lmbfgs* programs is the negative of the sum of the unnormalized correlations of each shifted frame with the sum of the shifted frames:

$$O(\Theta) = -Re \sum_{t=1}^T \sum_{j=1}^J \left[F_{jt}^* S_{jt}^* \sum_{t'=1}^T F_{jt'} S_{jt'} \right] \quad (3)$$

where O is a function of Θ , the translations of all frames, Re indicates the real part of the expression and $*$ indicates the complex conjugate of a complex number. This objective function is useful because its partial derivatives with respect to x_t and y_t , the x - and y -components of the desired \vec{x}_t vectors, can be calculated easily, giving

$$\frac{\partial O(\Theta)}{\partial x_a} = -Re \sum_{j=1}^J \frac{2\pi i k_x(j)}{N} \left[F_{ja} S_{ja} \sum_{t=1}^T F_{jt}^* S_{jt}^* - F_{ja}^* S_{ja}^* \sum_{t=1}^T F_{jt} S_{jt} \right] \quad (4)$$

and

$$\frac{\partial O(\Theta)}{\partial y_a} = -Re \sum_{j=1}^J \frac{2\pi i k_y(j)}{N} \left[F_{ja} S_{ja} \sum_{t=1}^T F_{jt}^* S_{jt}^* - F_{ja}^* S_{ja}^* \sum_{t=1}^T F_{jt} S_{jt} \right]. \quad (5)$$

Due to the availability of these derivatives, often called gradients in the field of optimization, the optimum in this objective function can be found in a computationally efficient manner with any of a variety of gradient-based optimization algorithms. The *alignframes_lmbfgs* and *alignparts_lmbfgs* programs make use of the limited memory Broyden–Fletcher–Goldfarb–Shanno (lm-bfgs) algorithm (Byrd, Lu, Nocedal, & Zhu, 1995) to optimize Eq. (3), giving the programs the “lmbfgs” part of their names.

The assumption that true frame or particle trajectories are unlikely to undergo sudden and dramatic changes in direction can be enforced by penalizing changes in $\partial x_t / \partial t$ and $\partial y_t / \partial t$. If $\partial x_t / \partial t$ and $\partial y_t / \partial t$ are constant ($\partial^2 x_t / \partial t^2$ and $\partial^2 y_t / \partial t^2$ are 0), the expected value for $(\vec{x}_t - \vec{x}_{t-1})$ is $(\vec{x}_{t-1} - \vec{x}_{t-2})$. Deviation from this expected linear trajectory can be penalized by an amount $\lambda ([\vec{x}_t - \vec{x}_{t-1}] - [\vec{x}_{t-1} - \vec{x}_{t-2}])^2$. The overall penalty imposed on the objective function to encourage smoothness is then given by

$$P(\Theta) = \sum_{t=3}^T \lambda [(x_t - 2x_{t-1} + x_{t-2})^2 + (y_t - 2y_{t-1} + y_{t-2})^2] \quad (6)$$

where λ is a user selected weighting parameter. This penalty is known as second order smoothing because it penalizes finite difference approximations of the second derivatives of x_t and y_t with respect to t , $\partial^2 x_t / \partial t^2$ and $\partial^2 y_t / \partial t^2$. The penalty function described in Eq. (6) is added to the objective function in Eq. (3) to obtain the overall objective function that is minimized. The contribution to the penalty function in Eq. (6) from shifting of the a^{th} frame

when $a \in [3, T-2]$ is $\lambda[(\vec{x}_a - 2\vec{x}_{a-1} + \vec{x}_{a-2})^2 + (\vec{x}_{a+1} - 2\vec{x}_a + \vec{x}_{a-1})^2 + (\vec{x}_{a+2} - 2\vec{x}_{a+1} + \vec{x}_a)^2]$ and consequently the first derivative of Eq. (6) with respect to x_a is given by

$$\frac{\partial P(\Theta)}{\partial x_a} = \begin{cases} 2\lambda(x_a - 2x_{a+1} + x_{a+2}), & a = 1, \\ 2\lambda(-2x_{a-1} + 5x_a - 4x_{a+1} + x_{a+2}), & a = 2, \\ 2\lambda(x_{a-2} - 4x_{a-1} + 6x_a - 4x_{a+1} + x_{a+2}), & a \in [3, T-2], \\ 2\lambda(x_{a-2} - 4x_{a-1} + 5x_a - 2x_{a+1}), & a = T-1, \\ 2\lambda(x_{a-2} - 2x_{a-1} + x_a), & a = T. \end{cases} \quad (7)$$

And

$$\frac{\partial P(\Theta)}{\partial y_a} = \begin{cases} 2\lambda(y_a - 2y_{a+1} + y_{a+2}), & a = 1, \\ 2\lambda(-2y_{a-1} + 5y_a - 4y_{a+1} + y_{a+2}), & a = 2, \\ 2\lambda(y_{a-2} - 4y_{a-1} + 6y_a - 4y_{a+1} + y_{a+2}), & a \in [3, T-2], \\ 2\lambda(y_{a-2} - 4y_{a-1} + 5y_a - 2y_{a+1}), & a = T-1, \\ 2\lambda(y_{a-2} - 2y_{a-1} + y_a), & a = T. \end{cases} \quad (8)$$

The derivative of the smoothed objective function is therefore the sum of the values from Eqs. (4) and (7) for the derivative with respect to x_a , and the sum of the values from Eqs. (5) and (8) for the derivative with respect to y_a . Because of this second order smoothness constraint, *alignparts_lmbfsgs* is best used to extract particle images from unaligned movies frames. Otherwise, changes in trajectory introduced into the frame-to-frame trajectories of individual particles by whole-frame alignment could be penalized by individual particle motion correction, turning a straight particle trajectory into one that has changes in direction or speed or otherwise altering the true trajectory.

For individual particle alignment, *alignparts_lmbfsgs* makes use of the additional constraint that nearby particle images are unlikely to have significantly different trajectories, which is achieved by local averaging of trajectories. Once raw trajectories are determined, locally averaged trajectories are calculated according to

$$\vec{x}_{nt}' = \frac{\sum_{m=1}^M w_{nm} \vec{x}_{mt}}{\sum_{m=1}^M w_{nm}} \quad (9)$$

where \vec{x}_{nt}' is the smoothed displacement vector for the n^{th} particle in the t^{th} frame and \vec{x}_{mt} is the original displacement vector for the m^{th} particle in the t^{th} frame. The weight w_{nm} is given by

$$w_{mn} = \exp\left(\frac{-d_{mn}^2}{2\sigma^2}\right) \quad (10)$$

where d_{mn} is the distance between the m^{th} and n^{th} particles and σ is a user set parameter that determines the extent to which the smoothing is applied. This Gaussian weighting is equivalent to the local averaging used for fitting linear trajectories in *Relion* (Scheres, 2014). Because of the Gaussian form of Eq. (10), 95 % of the weight for a particle trajectory will come from the trajectories within 2σ pixels of that particle. With $\sigma=0$ there is no local averaging. As the value for σ increases the method forces all particle trajectories in an image to be the same, as they would be if only whole-frame alignment were performed. This method of individual particle alignment by gradient-based optimization with enforced smoothness and correlation in the trajectories of nearby particles allows for the movement of individual particles in movies to be determined quickly and accurately with minimal computational cost (Fig. 2A).

As proposed even before the widespread availability of DDDs (Baker et al., 2010), the *alignparts_lmbfgs* algorithm adjusts the contribution from each frame for the exposure and resolution dependent fading of information due to radiation damage. However, rather than the critical exposure information obtained from the study of 2D crystals (Baker et al., 2010), *alignparts_lmbfgs* uses the critical exposure curves determined from the study of icosahedral virus particles (Grant & Grigorieff, 2015).



4. UNBLUR

The program *Unblur* (Grant & Grigorieff, 2015) also uses a cross correlation approach for frame alignment. Similar to *alignframes_lmbfgs*, *Unblur* aligns noisy individual frames from a movie to the high SNR average of frames. Unlike *alignframes_lmbfgs*, *Unblur* does not calculate a combined objective function or the gradients of the objective function. Instead, it iteratively aligns each individual frame to an average of frames that does not include the frame being aligned. Shifts for the individual frames are found by identifying the peak in a normalized cross correlation function, which for the a^{th} frame is given by:

$$CC_a = \frac{\text{Re} \sum_{j=1}^J \left(F_{ja}^* \left(\sum_{t=1, \neq a}^T F_{jt} \right) \right)}{\sqrt{\sum_{j=1}^J (F_{ja})^2 \sum_{j=1}^J \left(\sum_{t=1, \neq a}^T F_{jt} \right)^2}}. \quad (11)$$

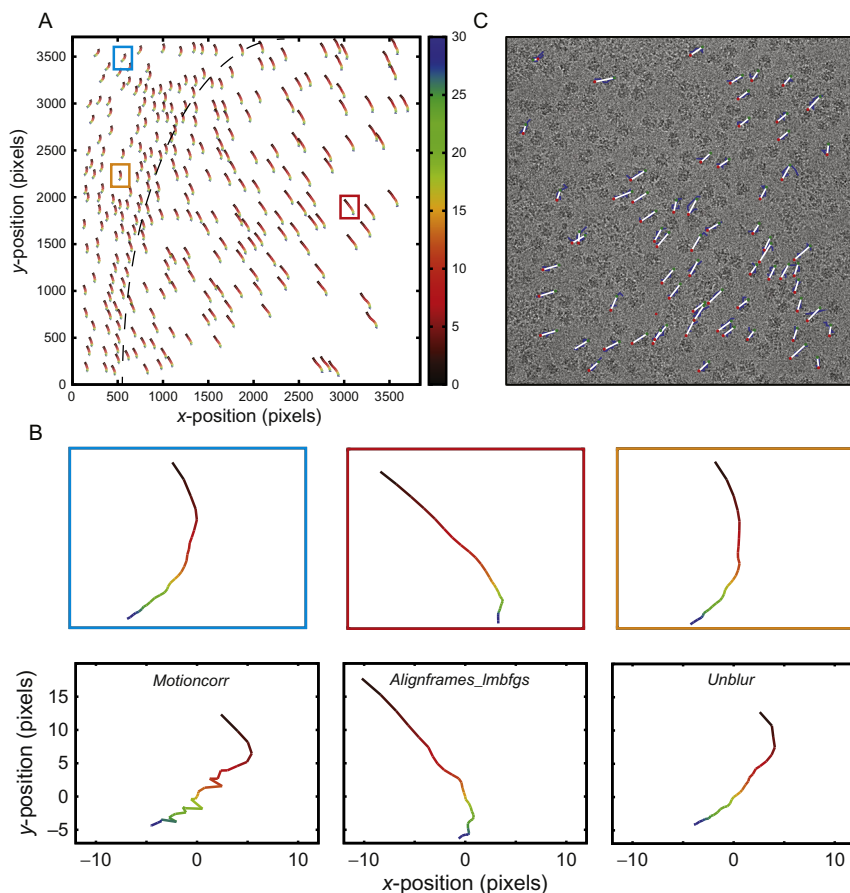


Fig. 2 (A) A plot of individual particle trajectories from *alignparts_lmbfbs* applied to unaligned movie frames. Each line in the plot indicates the trajectory of a single particle from frame 1 (black) to frame 30 (blue), exaggerated by a factor of 5. The broken line indicates the approximate edge of the carbon support film, with particles to the left of the line lying on the carbon support and particles to the right being in unsupported ice. (B) Inspection of individual particle trajectories from three regions of the micrograph shows smoothed particle trajectories that differ across the micrograph (three upper panels, color coded such that they correspond to the three small boxes in panel A). The whole-frame alignment programs *Motioncorr*, *alignframes_lmbfbs*, and *Unblur* all find somewhat different solutions to the alignment problem that produce similar power spectra from the average of aligned frames, each of which approximately matches the individual particle trajectories in different regions of the micrograph (lower panels). (C) Linear particle trajectories calculated using *Relion* particle polishing applied to whole-frame aligned movies. Particle trajectories start at the green dot and move to the red dot. Trajectories are from a different movie than used for parts A and B and are exaggerated by a factor of 50. Source: Reproduced with permission from reference Scheres, S. H. (2014). Beam-induced motion correction for sub-megadalton cryo-EM particles. *Elife*, 3, e03665.

The coordinates of the peak in this cross correlation function, $(\Delta x_a, \Delta y_a)$, are the components of the desired measured shift, \vec{x}_a , that would bring the frames into register with each other. Frames are assigned shifts sequentially and then the shifts are fit to either a spline or polynomial. This process of frame alignment by cross correlation, fitting of a smooth trajectory to the measured shifts, and application of smoothed shifts to each frame in order to calculate an improved average of frames is repeated until the estimates of shifts fall below a user-specified threshold, such as 0.5 pixels.

The fitting of the data to a spline or polynomial in *Unblur* smooths the trajectories, thereby removing sudden changes in trajectory that are due to noise in the images. Fitting of cubic splines is equivalent to minimizing the expression (Craven & Wahba, 1979):

$$\sum_{t=1}^T (\vec{x}_t - f(t))^2 + \lambda \int_{t=1}^T f''(t) dt \quad (12)$$

where the first term $\sum_{t=1}^T (\vec{x}_t - f(t))^2$ represents a least squares optimization of the fitted function $f(t)$ to the data \vec{x}_t . In this case \vec{x}_t are the raw measured shifts, and $f(t)$ is the desired smoothed trajectory. In this model the measured values are assumed to come from the desired function plus noise, $\vec{x}_t = f(t) + \epsilon_t$, where ϵ_t represents the noise. The second half of the equation accomplishes the same goal as the second order smoothing in *alignparts_lmbfgs* and *alignframes_lmbfgs*, which penalizes unlikely sudden changes in frame or particle trajectories, respectively. Here λ is a weighting factor for smoothing akin to that used in *alignparts_lmbfgs* and *alignframes_lmbfgs*. However, in *Unblur* the spline fitting algorithm implicitly calculates values for λ using a cross validation approach (Craven & Wahba, 1979), which takes into account the inherent noisiness of the signal. Finally, while averaging frames, *Unblur* performs exposure weighting to compensate for radiation damage (Baker et al., 2010) using values of critical exposures that were measured directly from single particle cryo-EM data, and a novel weighting scheme (Grant & Grigorieff, 2015).



5. OPTICAL FLOW

The Optical Flow approach (Abrishami et al., 2015) uses changes in pixel intensity to track the movement of frames relative to each other. Much like the least-squares approach implemented in the program *Motioncorr* (Li et al., 2013) and described in Section 3, the aim of the Optical Flow

method is to find \vec{s}_{mn} values, the frame-to-frame displacements in specimen that occur during movie acquisition. However, Optical Flow has been used to calculate \vec{s}_{mn} values not only for whole frames but also for subregions of frames. Unlike the methods described above that have used cross correlation to measure the shift between frames, the Optical Flow implementation described by Abrishami and colleagues uses the difference in pixel intensities between subsequent frames to estimate translations. Specifically, the problem is phrased as a first order Taylor expansion of the change in pixel intensities I as a function of image shift. This formulation of the problem allows for computation of derivatives of the intensity of image pixels with respect to image translation:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) \cong I(x, y, t) + \frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t \quad (13)$$

where $I(x, y, t)$ is the intensity of the pixel at coordinates (x, y) in the t^{th} frame and Δx and Δy are the calculated shifts for this pixel. The assumption is made that the total intensity of frames does not change, such that $I(x + \Delta x, y + \Delta y, t + \Delta t) \cong I(x, y, t)$. Consequently, Eq. (13) becomes the set of linear equations:

$$\frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t = 0 \quad (14a)$$

$$\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v = -\frac{\partial I}{\partial t} \quad (14b)$$

where $u = \frac{\Delta x}{\Delta t}$ and $v = \frac{\Delta y}{\Delta t}$. In the case where two subsequent frames, m and n , are compared, $\Delta t = 1$ and the translation vector $\vec{s} = (\Delta x, \Delta y)$, which is equivalent to the $\vec{s}_{mn} = (x_{mn}, y_{mn})$ notation used elsewhere in this chapter. To calculate a translation vector for each pixel (x_0, y_0) one assumes the shifts to be locally correlated at that pixel, and Eq. (14b) is solved for a window of coordinates centered on the pixel. The problem of calculating a translation vector for each pixel in the window is similar to the problem solved by *Motioncorr*, as the large number of pixels in the window provides a large set of overdetermined linear equations of the form $A\vec{s}_{mn} = B$. This process is performed for each pixel (x, y) in the image, with the result being a vector field the size of the image, containing a vector \vec{s}_{mn} for each pixel in the input

image. For clarity we denote this optical flow shift measurement process here as \circledast and the resulting vector field as OF_z .

Again, because of the noisy nature of the images, this alignment alone is insufficient to robustly calculate the shifts in a single step. Thus, as in the case of *Unblur*, the alignment is iterated to produce more reliable measured translation vectors. In the implementation of the Optical Flow approach, Abrishami et al. proposed iterative alignment in a pyramidal scheme of averaging frames. For example, with a 16 frame movie the alignment would proceed as follows:

1. Average all 16 unaligned frames: $I(x, y) = \sum_{t=1}^{16} I_t(x, y)$
2. Split the movie into 2^k groups of frames, where k is the current iteration, and average. That is: in the first iteration the frames are grouped and averaged as $I(x, y) = \sum_{t=1}^8 I_t(x, y)$ and $I(x, y) = \sum_{t=9}^{16} I_t(x, y)$
3. Align these group averages with the global average by calculating optical flows, ie,

$$OF_{1-8} = \sum_{t=1}^8 I_t \circledast \sum_{t=1}^{16} I_t \quad (15)$$

4. Use the measured OF to align each group of frames, then calculate the new aligned average of frames using the shifted frames
5. Repeat steps 2–4 until each group of frames has only 1 frame in it.



6. PARTICLE POLISHING IN *RELION*

“Particle polishing” is the individual particle motion correction procedure implemented in the program *Relion*. The name is derived from a catch phrase that became popular at the 2007 Gordon Research Conference on Three Dimensional Electron Microscopy. When Dr. Xuekui Yu of UCLA was asked what he meant when he said he used the “good” particles to calculate his 3.9 Å map of cytoplasmic polyhedrosis virus from a film dataset (Yu, Jin, & Zhou, 2008), he replied that he used particles that looked “shiny.” After the identification of particle movement during imaging as a cause of information loss in single particle cryo-EM, particles that were not “shiny” could be made “shiny” by motion correction, which is analogous to “polishing” them. The method developed by Scheres (2014) differs from the other methods described in this review in that it aims to correct for particle movement using 2D projections of a 3D reference map to align individual

particles in images, rather than performing this task using information from just the images. Consequently, unlike the methods described above, polishing is performed *after* a 3D map is calculated, rather than before, and is tightly integrated into the workflow of the *Relion* software package. During 3D map refinement, the Euler angles that relate the orientation of each particle image to the 3D map are estimated using particle images that are the averages of movie frames after whole-frame alignment. Once estimates of all of these orientations are obtained, particle polishing relaxes the assumption that the whole-frame alignment adequately describes the optimal alignment for individual particles in movies. Instead, it attempts to improve the translational alignment of particles in individual frames to projections of the 3D map. These translations are estimated by calculating cross correlation functions between map projections and particle images in movie frames to provide the appropriate values $-\vec{x}_t$. The low SNR in individual movie frames is accounted for in a few different ways. First, polishing is typically used with running averages of frames rather than individual frames, thus increasing the SNR over individual frames. In most cases, the trajectories for individual particle images estimated in this way are still noisy. Consequently, polishing makes the assumption that particle trajectories are linear and that particles move at a constant velocity. The initial estimate of each linear constant-velocity trajectory is obtained by performing regression of the estimates of $-\vec{x}_t$ to a straight line for the each particle. Finally, neighboring particles are assumed to have correlated trajectories. This correlation is enforced by replacing each particle trajectory with a weighted average of particle trajectories. The weighting function, w_p , falls off as a Gaussian with the distance between particles, much as in *alignparts_lmbfgs*. This fit to a linear trajectory with a Gaussian weighting over neighboring particles is achieved by minimizing the function:

$$O(\Theta)_n = \sum_m^M w_m \sum_t^T \left(\vec{x}_{mt} - \left(\vec{\alpha}_n + \vec{\beta}_n t \right) \right)^2 \quad (16)$$

where $O(\Theta)_n$ is the overall objective function that describes the linear trajectory of particle n , M is the number of particles in the field of view, and $m \in \{1, M\}$ with $m \neq n$ denotes all neighboring particles. The Gaussian weights used for the neighboring particles, w_m , are set by having the user define a value for σ and are the same as the Gaussian weighting for local correlation in *alignparts_lmbfgs* shown in Eq. (10). The values of \vec{x}_{mt} are the measured shifts for the neighboring particle m and $\vec{\alpha}_n$ and $\vec{\beta}_n$ are the intercept and

slope of the straight line trajectory for particle n , where $\vec{\alpha}_n = (x_0, y_0)$, and $\vec{\beta}_n = \left(\frac{dx}{dt}, \frac{dy}{dt}\right)$. It is important to note that in this version of particle polishing only x and y translations are considered, as opposed to earlier versions that accounted for particle rotation. In this way, each particle ends up with its own unique linear trajectory, with all of the trajectories tending to be the same when σ is large and $w_m \approx 1$. An example of these linear trajectories superimposed on a micrograph is shown in Fig. 2C.

In addition to particle motion correction, one of the key features of particle polishing is weighting each movie frame (or running averages of frames) in a frequency dependent manner to best reflect the information degradation caused by rapid movements and radiation damage. Rather than using estimates of information loss as a function of electron exposure (Baker et al., 2010), Polishing uses a data-driven approach with a filter estimated for each frame that has the form $\exp\left(\frac{B_t}{4d^2} + C_t\right)$, where B_t is a relative temperature factor for frame t , d is the resolution, and C_t is the resolution independent component of the signal in the frame. Values of B_t and C_t are estimated for each frame by calculating 3D maps from individual frames. The advantage of this approach vs experimental measurement of radiation damage is that it is insensitive to inaccurate measurement of electron exposure in each frame and removes the assumption that all macromolecules suffer from radiation damage in the same way. A relative weighting for each frame is used according to:

$$\frac{W_t\left(\frac{1}{d}\right)}{W_\Sigma\left(\frac{1}{d}\right)} = \sqrt{\frac{FSC_t\left(\frac{1}{d}\right) - FSC_t\left(\frac{1}{d}\right)FSC_\Sigma\left(\frac{1}{d}\right)}{FSC_\Sigma\left(\frac{1}{d}\right) - FSC_t\left(\frac{1}{d}\right)FSC_\Sigma\left(\frac{1}{d}\right)}} \quad (17)$$

where $FSC_t\left(\frac{1}{d}\right)$ and $FSC_\Sigma\left(\frac{1}{d}\right)$ are the Fourier shell correlations between the two independently refined maps from halves of the individual frame datasets and the averaged frames dataset, respectively. A type of Guinier plot

is produced by plotting $\frac{W_t\left(\frac{1}{d}\right)}{W_\Sigma\left(\frac{1}{d}\right)}$ against $\frac{1}{d^2}$, which reveals the decrease in

signal of the individual frame maps relative to the map from the average of frames. From this relative Guinier plot, the slope B_i and intercept C_i are used to calculate relative B-factors for individual frames, and the following weighting scheme as a function of spatial frequency $\frac{1}{d}$ is applied:

$$w_i\left(\frac{1}{d}\right) = \frac{\exp\left(\frac{B_i}{4d^2} + C_i\right)}{\sum_{i'} w_{i'}\left(\frac{1}{d}\right)}. \quad (18)$$

Consequently, if the signal from an individual frame map decreases faster than that of the map from the average of frames then the relative B-factor will be negative and that frame will be down weighted in a resolution dependent manner (note the $\frac{1}{d^2}$ term in Eq. (18)). In comparison, if the signal in an individual frame map is greater than the signal in the map from the average of frames then the relative B-factor will be positive and the frame will be up weighted relative to other frames. Note that in contrast to the form of most temperature factors, $\exp\left(\frac{-B}{4d^2}\right)$, the relative B-factors used here have the opposite sign, and positive and negative relative B-factors up weight and down weight high spatial frequencies, respectively, while positive and negative B-factors in general do the opposite. This weighting scheme is normalized and consequently the sum of the weights for all spatial frequencies is unity, so that the weighted average has the same power as the unweighted average and the weighting does not sharpen or dampen the map (Fig. 3A). After these weights have been estimated for the frames, the interpolated linear trajectory shifts are applied to the frames, and the weighted average of frames is calculated. Note that the first few frames are often down weighted, which is most likely due to large movements of the sample upon initial irradiation that causes the loss of high-resolution information. After the first few frames the effect of radiation damage becomes dominant and the curve looks similar to the weighting scheme based on measurements of radiation damage used by *Unblur* and *alignparts_lmbfqs* (Fig. 3B). The utility of these weighting approaches and individual particle motion correction is shown in Fig. 3C and D, where the improvement in map quality is evident.

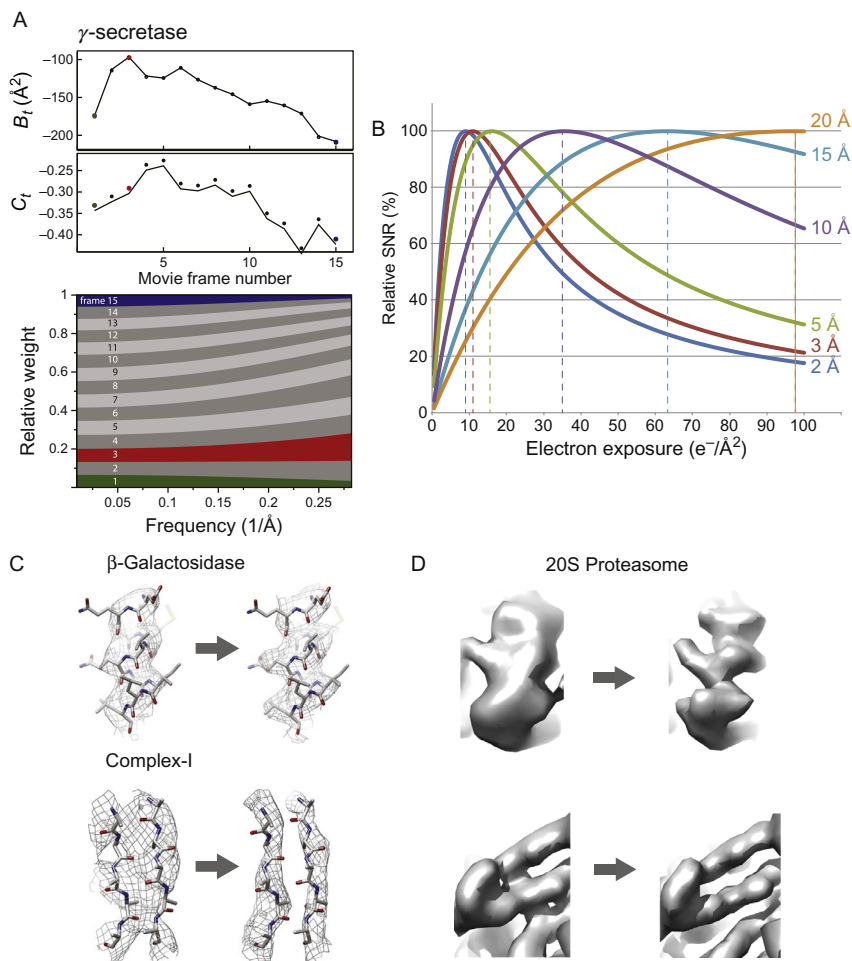


Fig. 3 (A) An example of measurements of B_t and C_t from *Relion* particle polishing. Lower panel shows the resulting weights of frames used. (B) Optimal exposure weighting curve used in *alignparts_lmbfgs* and *Unblur*. (C) Examples of map improvement upon motion correction and frame weighting with *Relion* particle polishing (Scheres, 2014). (D) Examples of map improvement upon motion correction and exposure weighting in *alignparts_lmbfgs*. Source: (A) and (C) Reproduced with permission from Scheres, S. H. (2014). Beam-induced motion correction for sub-megadalton cryo-EM particles. *Elife*, 3, e03665. (B) Figure adapted from Baker, L. A., Smith, E. A., Bueler, S. A., & Rubinstein, J. L. (2010). The resolution dependence of optimal exposures in liquid nitrogen temperature electron cryomicroscopy of catalase crystals. *Journal of Structural Biology*, 169 (3), 431437. doi: 10.1016/j.jsb.2009.11.014 using the critical exposures measured by Grant and Grigorieff (2015) for virus particles at 300 kV.



7. SUMMARY

Each of the different techniques described above has a number of advantages. *Motioncorr*, *alignframes_lmbfgs*, *Unblur*, and Optical Flow are all whole-frame alignment methods. *Motioncorr* benefits from highly over-determined information from computing the pairwise cross correlation function between each pair of images in a movie. The disadvantage of the method is that noisy frames are compared to other noisy frames, which can make detection of signal difficult for low contrast images. The large number of calculations required makes CPU implementations of the least-squares algorithm somewhat slow, but this problem has been overcome by implementing the algorithm for GPU processing in *Motioncorr*. *alignframes_lmbfgs*, *Unblur*, and Optical Flow compare frames to the average of frames, the latter having a higher SNR than an individual frame and consequently these methods may perform somewhat better for low contrast images than *Motioncorr*. Alignment of a low SNR frame to an average of frames is expected to be more accurate than alignment of a low SNR frame to another low SNR frame, but the latter provides over-determination of parameters while the former does not. *Unblur* is arguably the simplest method of the three, repeatedly aligning frames to an average, which produces better alignments and a better average, while simultaneously forcing the trajectories of frames to be smooth. The Optical Flow method tackles this problem using a pyramidal scheme for aligning frames to the average of frames and the optical flow approach rather than a cross correlation approach. *alignframes_lmbfgs* makes use of the well-established field of derivative-based optimization, which allows it to quickly and robustly find the optimal alignment for all of the frames. *alignframes_lmbfgs* enforces a smooth trajectory for frames by including a penalty for nonsmooth trajectories in the objective function that is optimized. In comparing the output of these whole-frame alignment algorithms, it is evident that the calculated shifts measured by the different algorithms do not always agree for a given micrograph (Fig. 2B, lower panels). However, each approach produces a whole frame trajectory that is similar to a region of the individual particle trajectories (Fig. 2B, upper panel) showing that the algorithms pick up on different parts of the micrograph, thereby correcting some but not all of the particle motion. Consequently, it is clear that there is no correct answer for a whole frame trajectory when there is distortion of the specimen during imaging and the discrepancies between the different methods justifies the

need for individual particle motion correction when attempting to obtain high-resolution structures from cryo-EM.

Particle polishing in *Relion* and *alignparts_lmbfgs* are the two methods that attempt to determine the trajectories for individual particles in the dataset. Their fundamental difference is that *alignparts_lmbfgs* translationally aligns the frames for particles only making reference to the average of frames for the particle (ie, at the beginning of the image analysis process) while polishing uses a map projection after estimating the pose or Euler angles for the particle (ie, at the end of the image analysis process). The map projection used by *Relion* will have a better SNR than the average of frames, which would provide polishing with an advantage over *alignparts_lmbfgs*. On the other hand, the average of frames used by *alignparts_lmbfgs* will always match exactly the appearance of the particle in the frames, while for polishing this agreement depends on accurate estimation of Euler angles, CTF parameters, particle conformation, and the absence of any contamination or signal from water in the raw image, which should provide an advantage to *alignparts_lmbfgs* over polishing. Both methods use a similar constraint that nearby particles must have similar trajectories. An additional difference between the methods is that polishing applies a linear and constant velocity trajectory for each particle while *alignparts_lmbfgs* allows variable direction and velocity, which if estimated accurately should be an advantage for *alignparts_lmbfgs*. However, polishing uses a powerful data-driven approach to determining the relative importance of each frame while *alignparts_lmbfgs* relies on physically measured fading of information for its weighting scheme. A final difference is in computation cost, with *alignparts_lmbfgs* being computationally inexpensive and polishing usually requiring significant computer resources.



8. FUTURE PROSPECTS

Since the advent of DDDs, the benefit of motion correction by alignment of movie frames has been apparent (Brilot et al., 2012). At present, the large transverse translations in the x - y plane are corrected by the movie processing algorithms described in this review, but challenges still remain. Chief among these would be the ability to correct for sample rotations and movement in the z -direction, which are known to occur during the complex flexing of samples in the electron beam. Additionally, one can conceive of different image acquisition schemes that enable motion correction during the initial period of rapid specimen movement, while maintaining the ability to align subsequent frames with less severe motion.

ACKNOWLEDGMENTS

We thank Sjors Scheres for providing the images used in parts of Figs. 2 and 3. We are grateful to Alexis Rohou, Niko Grigorieff, Timothy Grant, Xueming Li, Sjors Scheres, and Tony Crowther for a critical reading of this manuscript. Z.A.R. was supported by a scholarship from the Hospital for Sick Children and J.L.R. was supported by a Canada Research Chair. This work was supported by operating grant MOP 81294 from the Canadian Institutes of Health Research and the Natural Sciences and Engineering Research Council Discovery Grant 401724-12.

REFERENCES

- Abrihami, V., Vargas, J., Li, X., Cheng, Y., Marabini, R., Sorzano, C. O. S., & Carazo, J. M. (2015). Alignment of direct detection device micrographs using a robust optical flow approach. *Journal of Structural Biology*, 189(3), 163–176. <http://dx.doi.org/10.1016/j.jsb.2015.02.001>. <http://www.sciencedirect.com/science/article/pii/S1047847715000313>.
- Baker, L. A., & Rubinstein, J. L. (2010). Chapter fifteen—radiation damage in electron cryomicroscopy. *Methods in Enzymology*, 481, 371–388.
- Baker, L. A., Smith, E. A., Bueler, S. A., & Rubinstein, J. L. (2010). The resolution dependence of optimal exposures in liquid nitrogen temperature electron cryomicroscopy of catalase crystals. *Journal of Structural Biology*, 169(3), 431–437. <http://dx.doi.org/10.1016/j.jsb.2009.11.014>.
- Bracewell, R. (1965). *The Fourier transform and its applications*. New York: McGraw-Hill.
- Brilot, A. F., Chen, J. Z., Cheng, A., Pan, J., Harrison, S. C., Potter, C. S., ... Grigorieff, N. (2012). Beam-induced motion of vitrified specimen on holey carbon film. *Journal of Structural Biology*, 177(3), 630–637. <http://dx.doi.org/10.1016/j.jsb.2012.02.003>.
- Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5), 1190–1208.
- Campbell, M. G., Cheng, A., Brilot, A. F., Moeller, A., Lyumkis, D., Veesler, D., ... Grigorieff, N. (2012). Movies of ice-embedded particles enhance resolution in electron cryo-microscopy. *Structure*, 20(11), 1823–1828. <http://dx.doi.org/10.1016/j.str.2012.08.026>.
- Cao, E., Liao, M., Cheng, Y., & Julius, D. (2013). TRPV1 structures in distinct conformations reveal activation mechanisms. *Nature*, 504(7478), 113–118. <http://dx.doi.org/10.1038/nature12823>.
- Craven, P., & Wahba, G. (1979). Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik*, 31, 377–403.
- Grant, T., & Grigorieff, N. (2015). Measuring the optimal exposure for single particle cryo-EM using a 2.6 Å reconstruction of rotavirus VP6. *ELife*, 4, 1–19. <http://dx.doi.org/10.7554/eLife.06980>. <http://elifesciences.org/lookup/doi/10.7554/eLife.06980>.
- Kühlbrandt, W. (2014). The resolution revolution. *Science*, 343(6178), 1443–1444.
- Li, X., Mooney, P., Zheng, S., Booth, C. R., Braunfeld, M. B., Gubbens, S., ... Cheng, Y. (2013). Electron counting and beam-induced motion correction enable near-atomic-resolution single-particle cryo-EM. *Nature Methods*, 10(6), 584–590. <http://dx.doi.org/10.1038/nmeth.2472>.
- Li, X., Zheng, S. Q., Egami, K., Agard, D. A., & Cheng, Y. (2013). Influence of electron dose rate on electron counting images recorded with the K2 camera. *Journal of Structural Biology*, 184(2), 251–260.
- Liao, M., Cao, E., Julius, D., & Cheng, Y. (2013). Structure of the TRPV1 ion channel determined by electron cryo-microscopy. *Nature*, 504(7478), 107–112. <http://dx.doi.org/10.1038/nature12822>.

- McMullan, G., Faruqi, A., Clare, D., & Henderson, R. (2014). Comparison of optimal performance at 300 keV of three direct electron detectors for use in low dose electron microscopy. *Ultramicroscopy*, 147, 156–163.
- McMullan, G., Faruqi, A., Henderson, R., Guerrini, N., Turchetta, R., Jacobs, A., & Van Hoften, G. (2009). Experimental observation of the improvement in MTF from back-thinning a CMOS direct electron detector. *Ultramicroscopy*, 109(9), 1144–1147.
- Rohou, A., & Grigorieff, N. (2015). CTFFIND4: Fast and accurate defocus estimation from electron micrographs. *Journal of Structural Biology*, 192(2), 216–221.
- Rubinstein, J. L., & Brubaker, M. A. (2015). Alignment of cryo-EM movies of individual particles by optimization of image translations. *Journal of Structural Biology*, 192(2), 188–195.
- Scheres, S. H. (2014). Beam-induced motion correction for sub-megadalton cryo-EM particles. *Elife*, 3, e03665.
- Smith, M. T., & Rubinstein, J. L. (2014). Beyond blob-ology. *Science*, 345(6197), 617–619.
- Yu, X., Jin, L., & Zhou, Z. H. (2008). 3.88 Å structure of cytoplasmic polyhedrosis virus by cryo-electron microscopy. *Nature*, 453(7193), 415–419.
- Zhao, J., Benlekhir, S., & Rubinstein, J. L. (2015). Electron cryomicroscopy observation of rotational states in a eukaryotic V-ATPase. *Nature*, 521(7551), 241–245.
- Zhou, A., Rohou, A., Schep, D. G., Bason, J. V., Montgomery, M. G., Walker, J. E., ... Rubinstein, J. L. (2015). Structure and conformational states of the bovine mitochondrial ATP synthase by cryo-EM. *Elife*, 4, e10180.