CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

ClusterAlign: A fiducial tracking and tilt series alignment tool for thick sample tomography

Shahar Seifer* and Michael Elbaum

Chemical and Biological Physics, Weizmann Institute of Science, Rehovot, Israel
*Corresponding author. E-mail: Shahar.seifer@weizmann.ac.il

Abstract

Thick specimens, as encountered in cryo-scanning transmission electron tomography, offer special challenges to conventional reconstruction workflows. The visibility of features, including gold nanoparticles introduced as fiducial markers, varies strongly through the tilt series. As a result, tedious manual refinement may be required in order to produce a successful alignment. Information from highly tilted views must often be excluded to the detriment of axial resolution in the reconstruction. We introduce here an approach to tilt series alignment based on identification of fiducial particle clusters that transform coherently in rotation, essentially those that lie at similar depth. Clusters are identified by comparison of tilted views with a single untilted reference, rather than with adjacent tilts. The software, called ClusterAlign, proves robust to poor signal to noise ratio and varying visibility of the individual fiducials and is successful in carrying the alignment to the ends of the tilt series where other methods tend to fail. ClusterAlign may be used to generate a list of tracked fiducials, to align a tilt series, or to perform a complete 3D reconstruction. Tools to evaluate alignment error by projection matching are included. Execution involves no manual intervention, and adherence to standard file formats facilitates an interface with other software, particularly IMOD/etomo, tomo3d, and tomoalign.

Impact Statement
Fiducial markers such as gold nanoparticles are often added to samples in transmission electron tomography to assist in image alignment prior to 3D reconstruction. Particularly for thick samples, tracking of individual fiducials often fails due to low contrast. ClusterAlign is a software that follows the structure of fiducial clusters, rather than individuals, to serve for alignment of the tilt series and 3D reconstruction.

1. Introduction

Electron tomography offers a powerful set of tools to reveal cellular anatomy in three dimensions. The most popular implementation is that of tilt tomography in the transmission microscope. A set of 2D projection images is recorded, from which a 3D volume representation is reconstructed by one of the available algorithms, which include weighted back projection, direct Fourier inversion, and iterative techniques. An essential intermediate step is that of aligning the 2D images to approximate the ideal rotation of a rigid body around a unique tilt axis. In practice, the mechanical mechanism of rotation always introduces some shifts or drift that must be corrected by translation of the images. Additionally, electron-optical distortions as well as geometric distortions due to beam exposure may need to be corrected for a

successful reconstruction. In order to facilitate the alignment, high contrast nanoparticles such as colloidal gold are often distributed over the specimen in order to assist in correcting or compensating these faults. Tracking the nanoparticles from tilt to tilt across the series serves to establish precisely the tilt axis orientation and to bring the 2D images into alignment around it. As manual tracking is tedious, a number of algorithms have been developed for automatic detection and tracking of the fiducial markers[1–6]. Despite significant advances in software, precise tracking of fiducials can still be difficult to automate, however. Alternatively, a number of fiducial-less alignment algorithms have been developed[7–13]. A recent and successful addition is AreTomo, which is based on successive matching of projections from partial reconstruction to align the tilt series[14]. In such cases, added fiducials may be used as a diagnostic of the alignment quality and indicator of the need for follow-up refinement. The widespread adoption of tomographic acquisition in batch mode accentuates the need for unattended alignment as a first step in data interpretation.

The introduction of scanning transmission electron microscopy[15,16], as well as soft X-ray imaging[17–19] modalities, has expanded the range of specimen thickness accessible to cryo-tomography[20]. The specimen can no longer be considered a thin slab when its thickness may typically reach half the lateral field of view. This introduces specific issues for alignment. The pileup of features in a thick specimen flattens the contrast overall, hindering the effectiveness of fiducial-less alignment, and on the other hand small fiducials may be poorly resolved in transmission through the thick material. This problem often becomes acute at high tilt angles where the projected thickness increases, and moreover, high-contrast features in the sample itself may hide or shadow the fiducials over a limited range of tilt angles. Thus, it becomes difficult even to identify individual particles reliably in every projection and to track the particles across large gaps in tilt views. One popular approach is to build a model of the rotation that can predict the projected position of individual fiducial markers, allowing the user to accept or correct the prediction so as to fill the gaps[5–21]. Other approaches generate overlapping chains of landmarks, each covering a part of the tilt range[8–12], or make use of a local invariant criterion[3]. Further developments are underway, with an emphasis on unattended operation to support the ever-increasing throughput of data acquisition.

Here we offer a complementary strategy to that of following individual fiducial markers. We identify extended clusters that transform geometrically between each of the tilted projections and a single reference view at low tilt angle, where visibility is optimal. Essentially, these are the subsets of fiducials that lie at similar height perpendicular to the view direction, and whose distributions transform with the cosine of the tilt angle. Multiple such clusters may be found and may even overlap, for example, when particles are located on two sides of a slab. The verified cluster vertices are then used as fiducials, rather than the locations of individual particles. Thus, we avoid the need to follow the same particle sequentially from tilt to tilt, and the cluster location is inherently robust to disappearance of individual particles. The relative motions of multiple identified cluster vertices are then used as fiducial markers to constrain the precise position and angle of the tilt axis by least squares fitting. The program can output the list of tracked fiducials in a format compatible with IMOD[5] or Tomoalign[9], a stack containing the aligned tilt series, or a completed 3D reconstruction. In addition, a text file reports the transformations and error estimates generated during the processing.

Alignment error is primarily translational and thus expected to be corrected by solving for $\Delta x, \Delta y$ shifts of the image, but the algorithm is tolerant to moderate global in-plane and out of plane rotations. Note that clusters are defined by a common height with respect to the tilt axis and need not be composed of apparently adjacent fiducials in any given projection. Thus, the aim of the software is to optimize translational alignments under conditions of problematic fiducial visibility without manual intervention.

2. Method

2.1. Software features

A flowchart of the processing appears in Figure 1. The workflow involves three major stages: fiducial particle identification, cluster assignment, and rigid body fitting of the tilt axis. Compatibility is
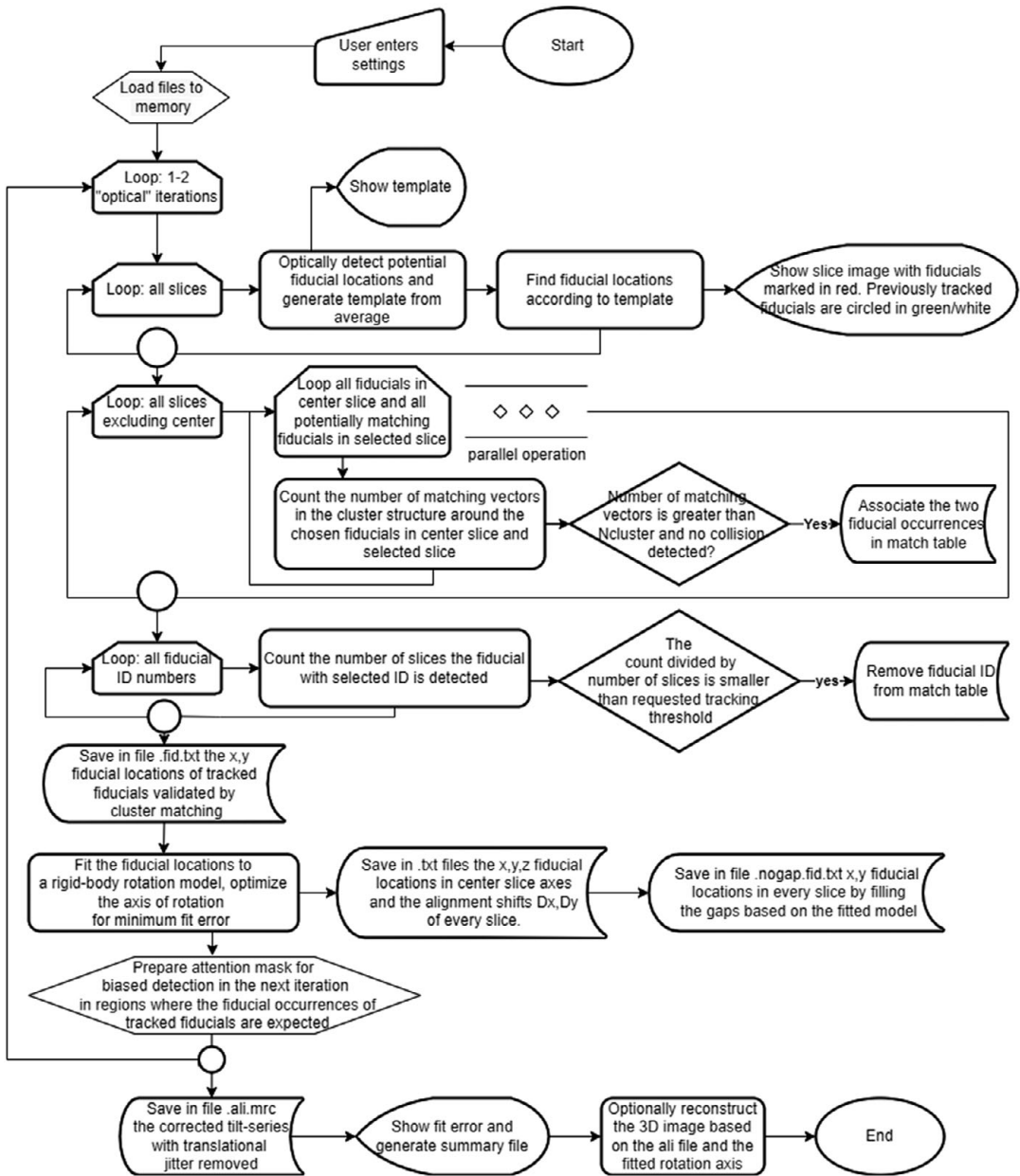
Figure 1. *A flowchart of the ClusterAlign processing workflow.*

maintained with IMOD file formats and conventions. The software is written in C#, makes use of the EMGU/OpenCV library,[22,23] and is available as open source under the GNU-GPLv3 license (see Data Availability Statement). Processing 800 fiducials in a tilt series of 57 frames of $2k \times 2k$ frame size requires 4 min on a 24-core Intel Xeon-based computer, and the reconstruction using Nvidia RTX 3060 GPU with binning of 2 takes an additional 4 min.

The key input parameters appear in Table 1 and the user interface in Figure 2. Most parameters are self-explanatory. Significance of those related to cluster assignment is explained below. Practical details appear in the user manual (see Data Availability Statement).

Table 1. *Settings.*

Parameter	Purpose
File path	Path for input/output files
Tilt series name	Name of the tilt series stack file of the images (*.mrc)
Tilt angle file	Name of a text file that contains tilt angles in degrees, one value per line according to the order of images in the tilt series file (e.g., *.rawtlt file)
Rotation axis	X or Y, according to approximate rotation axis (X is horizontal), or Angle $= 0$–$179°$ to request pre-rotation of the images that aligns the rotation axis with Y
Cluster size	Maximum radius of a cluster in pixels
Ncluster	Minimum count of particle members to comprise a cluster. Enter "-1" for automatic adjustment
Fiducial size	Average size expected of fiducials in pixels
Fiducial location tolerance	The tolerance allowed for matching relative locations of cluster members, measured in % relative to fiducial size (typically between 100–300%)
Max. number of fiducials	Maximum number of particles to be detected in a frame, often set to twice the actual number of particles to achieve acceptable sensitivity
Center slice	The number of the slice to use as the reference or center tilt (if -1 then automatically the slice of smallest tilt angle is chosen)
Tracking threshold (%)	Minimum threshold in percentage of occurrences from available slices required to accept validity of a fiducial "trajectory"
Optical iteration	No for a single pass, Yes to repeat the entire image analysis in attempt to increase the tracking threshold, while increasing detection sensitivity at predicted locations of tracked fiducials
Cosine sampling?	True when the aspect ratio of pixels is adjusted according to $\cos(\theta)$
Alignment tolerance	Maximum alignment shift allowed
Fiducials are bright?	False for bright field images and True for dark field

The software supports optional loading of external fiducial files and pre-aligned stacks for refinement, optional reconstruction in single or dual axis, and loading of an attention mask for special uses explained in the user manual. There is also support for cosine sampling tomograms, which can be acquired by varying the aspect ratio of the scan pattern[24].

The user may choose to run the analysis in two iterations to enhance precisions. The information that is carried between iterations is global and does not include particular trajectories. The knowledge gained in the first iteration is used to update the tracking threshold, to generate attention maps, and to determine fiducial size. The predicted locations of fiducial according to a rigid body model are displayed during the second iteration.

The output of the program is a text-based fiducial list of the form used by Tomoalign (.fid.txt), which can be converted to the binary IMOD format (.fid) using the command point2model. An aligned tilt series is generated either by imposing only translation corrections without adjustment of the tilt axis (_jali.mrc), or in standard form (_ali.mrc) globally aligned with tilt axis vertical and centered. Optionally, the package can output a Simultaneous Iterative Reconstruction Technique (SIRT) reconstruction based on the Astra toolbox[25,26], which corresponds to the exact fitted rotation axis. (This option is available only on computers equipped with GPU and Matlab.) The default generated tilt series (_jali.mrc) best matches our Matlab tomogram reconstruction using rotation axis properties found in file (.output.txt), while the standard form is intended for compatibility with other reconstruction software such as IMOD or tomo3d[27]. No attempt is made to optimize view angles that deviate from the mechanical tilts, nor local alignments to compensate for specimen distortion. Optimizations such as focused local alignment or
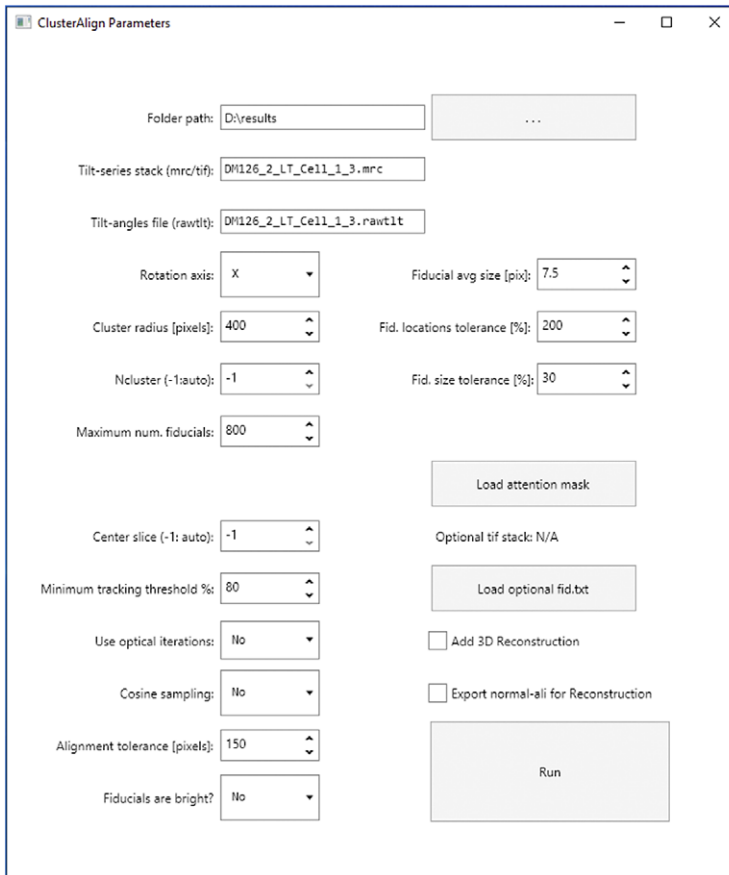
Figure 2. *The ClusterAlign user interface. See the user manual for more information.*

global distortion corrections may be performed in other dedicated software such as Tomoalign[9], EMClarity[10], or constrained reconstruction model[28]. Furthermore, the Astra toolbox can reconstruct from several rotation axes at once. We provide a Matlab module for dual axis reconstruction that can be used after separate alignment of each tilt series in ClusterAlign. The method involves rotating the second series by 90° and passing to Astra the 3D center coordinates of each tilt series (example is shown in the Supplementary Materials).

2.2. Particle localization

Detection and localization of particles is an initial step before the actual cluster analysis. The locations of particles are found based on template matching using an average generated from the measured data, an approach that has been used in RAPTOR[1]. The main difference from previous works is in the method of threshold calculation and in the way particle candidates are selected to generate a template.

The main distinctive feature that allows initial marking the center of particles is the "convoluted divergence," which is calculated as follows:

$$G(M) = \sqrt{(\ker_x \circledast \nabla_x M)^2 + (\ker_y \circledast \nabla_y M)^2},$$

where $\nabla_x M$ is the gradient in the x direction of the raw gray image. Similarly $\nabla_y M$ is the gradient for the y direction. $\ker_x$ denotes a small matrix with nonzero elements over positions of a ring that envelopes the

range of possible fiducial particle sizes as entered in the user settings. The values of the nonzero elements are $(\ker_x)_{i,j} = j/\sqrt{i^2+j^2}$, $(\ker_y)_{i,j} = i/\sqrt{i^2+j^2}$, where i and j are in reference to the center element of the matrix. $(A \circledast B)$ denotes convolution of A with B.

The next step is to dilate (i.e., replace pixel values by maxima values of) the image of convoluted divergence, $G(M)$, using a mask with elements 1 over a filled circle the size of the largest particle expected, and elements 0 outside the circle. The locations where $G(M) = \text{Dilated}\{G(M)\}$ are the peaks of such spots and are considered as candidate fiducial locations. Although the implementation is different, the detection algorithm bears a resemblance to the radiality transform used in the SRRF fluorescence localization microscopy method[29].

Images that have been pre-aligned by cross-correlation or other means contain data-free margins with a sharp gradient at the edges. The program identifies the margins based on regions with zero values of $G(M)$ and removes these margins from further processing.

For each projection in the tilt series, the raw image *M[n]* is band pass filtered between spatial periods of 1/3 and 20 of the fiducial size, as defined in the user settings panel. The result after further manipulations is a regularized gray image $M^\dagger$*[n]*. Regions at the sides of $M^\dagger$ that are outside the expected projection of a thin layer in the center tilt are masked since any fiducial found there will not be traceable to one that appears in the center slice.

A threshold of the peaks in $M^\dagger$ should determine which of the candidate locations of fiducial particles are to be accepted for the next screening process. The maximum allowed number of peaks is determined by the user-defined "maximum number of fiducials" (NfidMax). Initial threshold is calculated by the triangle type threshold algorithm found in EMGU/OpenCV. Then, $M^\dagger$ is temporarily multiplied by a mask of spots around the locations of candidate fiducials and the number of contours is counted using the FindContours function. If the count is larger than the maximum allowed number of fiducials, then the threshold is increased incrementally and the process is repeated until the number of fiducials is reduced to fall within the expected range. If such iterations fail, the threshold is determined based on the histogram of gray levels and the total expected area of fiducials.

In the next step, the average image of potential fiducials is generated based on their selected locations. Since we wish to reduce artifacts at sharp edges, the fiducials are filtered according to isotropy. An identified fiducial is considered sufficiently isotropic for inclusion in the average if the normalized correlation between its image and its transposed image is greater than 0.4. This criterion also filters small clusters of particles from consideration. The average fiducial image serves as a template, which is correlated with $M^\dagger$ to generate a map H (using the MatchTemplate function in EMGU/OpenCV of type "correlation-normalized").

The peaks in the correlation map H are considered the candidates for precise locations of the fiducial particles. In order to select the best candidates counted up to NfidMax, the process of finding the suitable threshold is repeated as described above, this time based on the gray levels of $M^{\dagger 2} \times H$ and a mask of the candidate locations convolved with maximum expected fiducial size. The final locations are selected from those candidate locations according to the threshold.

This rather elaborate protocol achieves a high sensitivity (up to 99%) of nanoparticle detection in the sample, excluding most of the artifacts (false negative expected between 1 and 6%) while optimizing the precision of locations reported for the selected particles.

2.3. Fiducial registration by clusters

With a list of detected particle locations in each projection image, the next step is to track them throughout the tilt series. The conventional approach is to track fiducials that are displaced in a direction perpendicular to the tilt axis in adjacent tilt images, working incrementally through the stack. For thick specimens, this approach is not robust because of the very large displacements observed, and because of the possibility for a tracked particle to disappear behind a dense object in the specimen. We apply here an alternative strategy based on recognition of particle clusters.
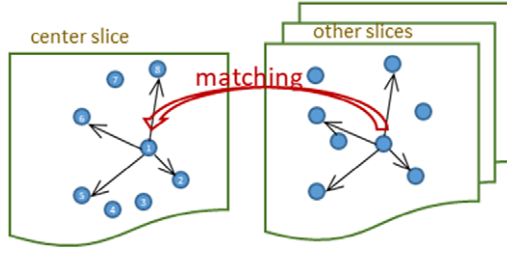
Figure 3. *Tracking a fiducial across different projections is based on comparing the normalized vectors pointing from each of the detected particle to the surrounding particles in all possible clusters, and approving the cluster association if a sufficient number of vectors match the corresponding patten in the zero-tilt reference image.*

Each particle location in every slice is selected as a vertex, from which vectors are drawn to all neighboring particles within the user-defined maximum cluster radius. Vector lengths are normalized in the component perpendicular to the tilt axis by the cosine of the tilt angle. The set of vectors emerging from each vertex is then compared between the tilted images and the common center reference image (see Figure 3). Matches are accepted within a tolerance defined in the user settings, typically to 200% of the average fiducial size. (The tolerance allows for some displacement in the z plane.) When the number of matching vectors is greater than a threshold number NCluster (which may be defined by the user or automatically) the vertex is considered a candidate for tracking of a given fiducial at a given slice. If these candidates are indeed unique and are found in a sufficient number of slices (in terms of a user-defined "tracking threshold"), the vertex fiducials are registered as locations of a bona fide fiducial for tracking. Note that recognition of the vector pattern is robust to translations, so the strategy of comparison between tilted projections and an untilted reference does not require a pre-alignment of the tilt series.

Tracking fiducials is the most demanding task in terms of computation time, so the calculations are handled by multi-threading of several CPU cores. Since the success of pairing depends sometimes on the order of the search, the process is not completely deterministic, and thus we may observe a small variability of results for certain ranges of parameters. In particular, collisions may occur where a given particle may be assigned at different tilts to different clusters. Such collisions are detected and removed in advance of the registration of the matches.

2.4. Rigid body analysis

The goal is to align the images by removing the translational shifts $\left(\Delta x_{(n)}, \Delta y_{(n)}\right)$ in every slice n, which originate in mechanical jitter or optical distortions. Thus, to find the translational shifts it is necessary to fit the available information on the set of fiducial particles p at slice n, $(x^p_{(n)}, y^p_{(n)})$, with a model. The model considered for rigid body rotation with shifts is

$$\begin{pmatrix} x^p_{(n)} \\ y^p_{(n)} \\ z^p_{(n)} \end{pmatrix} = A^{\vec{\omega}}_{(\theta_n)} \begin{pmatrix} x^p_c \\ y^p_c \\ z^p_c \end{pmatrix} - \begin{pmatrix} \Delta x_{(n)} \\ \Delta y_{(n)} \\ 0 \end{pmatrix}. \tag{1}$$

We denote the center index c as the one parallel to the tilt angle $\theta = 0$. For each vertex, the values x^p_c and y^p_c are known from the projected image at $\theta = 0$. Similarly, at every projection angle θ_n the locations $x^p_{(n)}, y^p_{(n)}$ are known with respect to the common lab frame. The rotation axis is defined as

$$\vec{\omega} = \begin{pmatrix} \cos\psi \sin\varphi \\ \cos\psi \cos\varphi \\ \sin\psi \end{pmatrix},$$

where φ denotes the azimuth of the rotation axis projected on the x-y plane in the center framework, and ψ is the out of plane angle. Assuming that the sample (and stage) rotates by a tilt angle θ_n around a general fixed axis $\vec{\omega}$, the transformation matrix based on[30] is found

$$
A_{(\theta)} = \cos\theta\, \mathbb{I} + (1-\cos\theta)\begin{pmatrix} \cos^2\psi\sin^2\varphi & \cos^2\psi\cos\varphi\sin\varphi & \cos\psi\sin\psi\sin\varphi \\ \cos^2\psi\cos\varphi\sin\varphi & \cos^2\psi\cos^2\varphi & \cos\psi\sin\psi\cos\varphi \\ \cos\psi\sin\psi\sin\varphi & \cos\psi\sin\psi\cos\varphi & \sin^2\psi \end{pmatrix}
$$
$$
+ \sin\theta\begin{pmatrix} 0 & -\sin\psi & \cos\psi\cos\varphi \\ \sin\psi & 0 & -\cos\psi\sin\varphi \\ -\cos\psi\cos\varphi & \cos\psi\sin\varphi & 0 \end{pmatrix}.
$$

(2)

We solve iteratively for the height of fiducial p in the sample, z_c^p, and for the projected stage translations $\Delta x[n]$, $\Delta y[n]$ based on the inverse of equation (1).

Obtaining x_c^p and y_c^p is trivial if a projection at tilt angle $\theta = 0$ is available. Otherwise, we calculate the locations of each particle in the center frame from a tilt angle θ_0 nearest to zero as follows:

$$
\begin{pmatrix} x_c^p \\ y_c^p \\ \blacksquare \end{pmatrix} = \left(A_{(\theta_0)}^{\vec{\omega}}\right)^{-1}\begin{pmatrix} x^p[\theta_0] \\ y^p[\theta_0] \\ \left(A_{\theta_0}^{\vec{\omega}}\right)_{3,3} z_c^p[p.i] \end{pmatrix}.
$$

(3)

The lateral shifts are defined zero at tilt angle θ_0 and the value of the fiducial height z_c^p in equation (3) is initialized to zero in the first iteration. Subsequently, it is later propagated from previous iterations (denoted $p.i$).

For convenience, we search for the axis of rotation $\vec{\omega}$ with deviation of up to $15°$ from either the x or y axes as defined in the user settings (larger in plane rotations should be compensated in advance by specifying approximate angle φ in the settings). In the following, we denote angle brackets with subscript n as the average of different slices n (either mean or median depending on the number of samples). The elements of the rotation matrix A are indexed with respect to vector (x,y,z). Accordingly, the height of the fiducial z_c^p should be extracted from the differences in y or x positions of the fiducial p between the center slice and other slices

$$
z_c^p\left(\vec{\omega} = \hat{x}\right) = \left\langle \left(y_{(n)}^p + \Delta y[n] - \left(A_{\theta_n}^{\vec{\omega}}\right)_{2,1} x_c^p - \left(A_{\theta_n}^{\vec{\omega}}\right)_{2,2} y_c^p\right)/\left(A_{\theta_n}^{\vec{\omega}}\right)_{2,3}\right\rangle_n
$$

$$
z_c^p\left(\vec{\omega} = \hat{y}\right) = \left\langle \left(x_{(n)}^p + \Delta x[n] - \left(A_{\theta_n}^{\vec{\omega}}\right)_{1,1} x_c^p - \left(A_{\theta_n}^{\vec{\omega}}\right)_{1,2} y_c^p\right)/\left(A_{\theta_n}^{\vec{\omega}}\right)_{1,3}\right\rangle_n.
$$

Thus, we generalize the calculation using

$$
z_c^p = \frac{\left(A_{\theta_n}^{\vec{\omega}}\right)_{1,3} z_c^p\left(\vec{\omega} = \hat{y}\right) + \left(A_{\theta_n}^{\vec{\omega}}\right)_{2,3} z_c^p\left(\vec{\omega} = \hat{x}\right)}{\left(A_{\theta_n}^{\vec{\omega}}\right)_{1,3}^2 + \left(A_{\theta_n}^{\vec{\omega}}\right)_{2,3}^2}.
$$

(4)

The translational shifts (jitter) are calculated by iterations based on equation (1) as follows:

$$
\Delta x_{(n)} = \left\langle \left(A_{\theta_n}^{\vec{\omega}}\right)_{1,1} x_c^p + \left(A_{\theta_n}^{\vec{\omega}}\right)_{1,2} y_c^p + \left(A_{\theta_n}^{\vec{\omega}}\right)_{1,3} z_c^p - x_{(n)}^p\right\rangle_p
$$

$$
\Delta y_{(n)} = \left\langle \left(A_{\theta_n}^{\vec{\omega}}\right)_{2,1} x_c^p + \left(A_{\theta_n}^{\vec{\omega}}\right)_{2,2} y_c^p + \left(A_{\theta_n}^{\vec{\omega}}\right)_{3,3} z_c^p - y_{(n)}^p\right\rangle_p.
$$

(5)

In equation (5), we take a "med-average" of different fiducials p. Namely, we sort the values by order and calculate the average of 20% of the values located in the center of the ordered list, thus rejecting false fiducial points while at the same time reducing the standard deviation by arithmetic average.

The fitting error is calculated as

$$\text{error} = \langle (x^p[n] + \Delta x[n] - x_{\text{sim}}[p,\, n])^2 + (y^p[n] + \Delta y[n] - y_{\text{sim}}[p,\, n])^2 \rangle_{n,p}^{1/2} \tag{6}$$

where "sim" stands for simulated values, and the vector is calculated as

$$\begin{pmatrix} x_{\text{sim}}[p,\, n] \\ y_{\text{sim}}[p,\, n] \\ z_{\text{sim}}[p,\, n] \end{pmatrix} = A_{(\theta[n])}^{\vec{\omega}} \begin{pmatrix} x_c^p \\ y_c^p \\ z_c^p \end{pmatrix}.$$

The process is repeated for different parameters of φ and ψ in a range of $\pm 15°$ with respect to the assumed orientation of the rotation axis. By minimizing the fitting error, those parameters and the final solution of $\Delta x[n]$, $\Delta y[n]$ are determined.
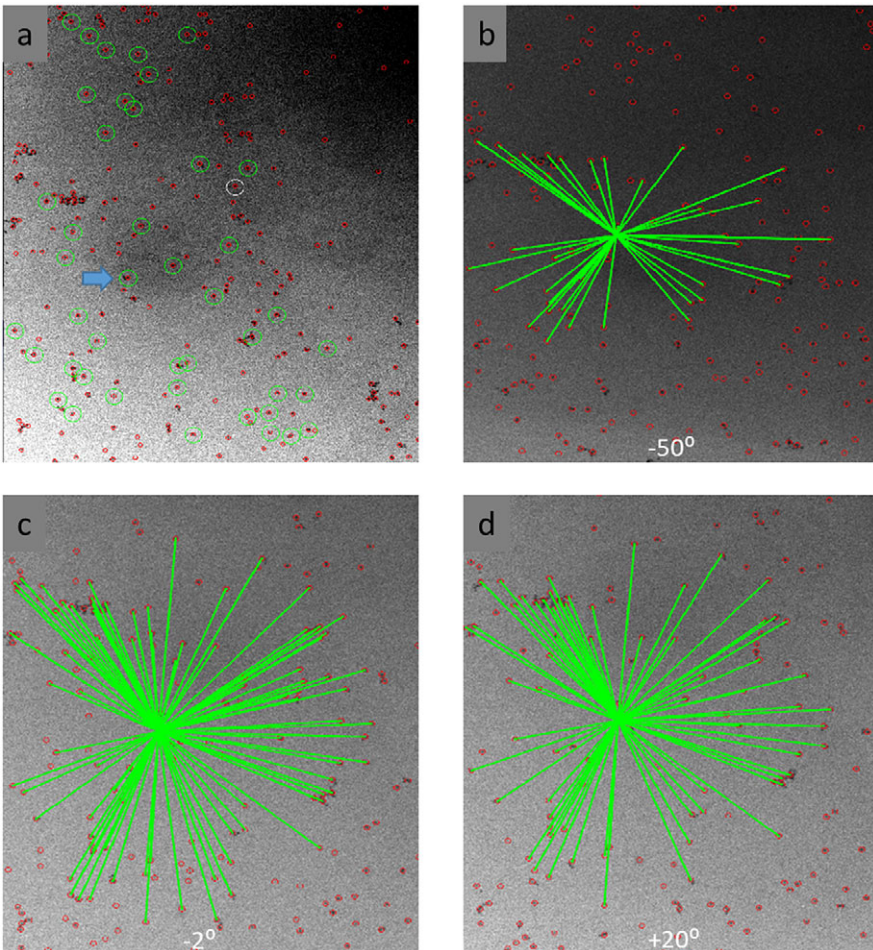

Figure 4. *Fiducial detection and cluster assignment. (a) Detected fiducials are marked in red, while the particles outlined in green have been identified as belonging to a common cluster in the untilted view. Panels (b–d) show views at −50°, −2°, and +20°, respectively.*

3. Results and Demonstration

We have confirmed that the image analysis is able to detect most of the fiducials that could be recognized by eye. Figure 4a shows an example where the detected particles are circled in red, and highlights in green about one fifth of them. Specifically, these tracked fiducials are validated as cluster vertices and will be used to optimize the series alignment. Figure 4b–d shows the assignment of fiducial particles to a single cluster viewed from three angles: $-50°$, $-2°$, and $+20°$. The specimen is a vitrified cell of Plasmodium falciparum imaged by cryo-scanning transmission electron tomography (CSTET) in bright field mode with sampling 2 nm/pixel in a Titan Krios microscope (Thermo Fisher Scientific) at 300 kV acceleration[31]. The single-tilt series covered a range from $-56°$ and $+56°$ in steps of $2°$. Gold nanoparticles are scattered over the sample to serve as fiducials, but due to the specimen thickness of 1 μm they are just marginally visible by eye at the higher tilts.

The user settings (Table 1, Figure 2) are as follows: cluster radius 400 pixels, fiducial size 7.5 pixels $\pm 30\%$, fiducial location tolerance 200%, maximal number of fiducials 800, tracking threshold of 80%, without second optical iteration (no repeat of the analysis), and with a generous alignment tolerance of 150 pixels. The minimum count of fiducials in a cluster, Ncluster, was varied with each run. Association between a fiducial appearance in the center slice and its appearance in another slice is approved if the number of relevant matching vectors is equal or greater than Ncluster. The tracking of a given fiducial is recorded only if its location as a vertex center is found in a number of tilt slices that exceeds the tracking threshold times the total number of slices (here, e.g., we ignore fiducials that could be found in less than 46 slices out of 57). In addition, we neither count nor report fiducial locations that can be associated with multiple fiducials in the center slice, the process which is called collision blockage. For the purpose of demonstration, a flag called "ignore_collisions" in the code was altered between true and false to test the influence of rejecting fiducials that match with more than one fiducial in the center slice. Moreover, when the user sets Ncluster = Auto, the threshold value Ncluster is determined independently for each fiducial and is determined as the maximum that yields only one association in fiducial location in a number of slices according to the tracking threshold.

Following the description above, the tilt axis position and orientation was solved iteratively from the locations of tracked fiducials across the tilt series. The aligned series was then reconstructed by our Matlab program using SIRT3D in ASTRA toolbox, activated in the GUI window of ClusterAlign. The 3D slices shown in Figure 5 demonstrate details of the parasite as well as two separate layers of fiducials.

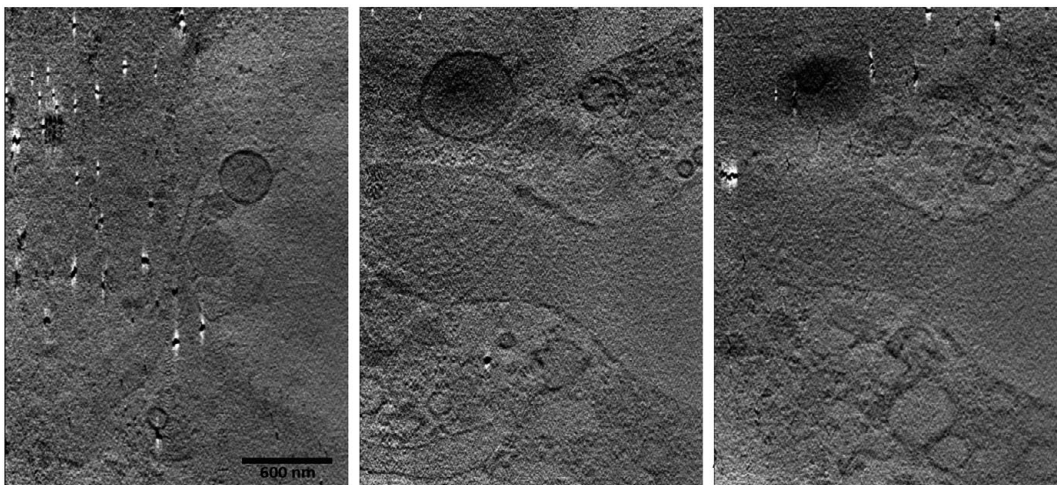**Figure 5.** *SIRT3D reconstruction after automated alignment with ClusterAlign, showing 3 layers of the sample at different heights. Fiducials appear in planes both below (left) and above (right) the cellular content.*
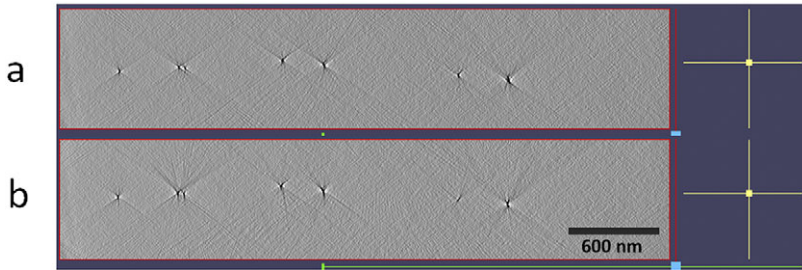
Figure 6. *YZ sections through reconstructions (weighted back projection in IMOD) after automated alignment using ClusterAlign (a) or IMOD (b).*

The symmetry of intensity projections (spines) in the YZ plane radiating from each fiducial is a qualitative mark of a successful alignment. Figure 6 compares reconstruction using automated alignment in IMOD versus ClusterAlign. For a fair comparison, both reconstructions were performed using IMOD. The ClusterAlign result is visually superior. Manual refinement may be applied to either workflow, but of course that contradicts the goal of hands-off processing.

Fiducials located significantly apart in z locations cannot form clusters that sustain their structure during rotation, as viewed in the image plane. Even if we observe several sets of fiducial points "traveling" to different directions during rotation due to differences in z coordinate, the algorithm will associate cluster members for each vertex of similar height while ignoring the nonmembers. Collisions are not likely to occur unless the uniqueness in structure of each cluster is compromised. However, anticipating possible warping and distortions we limit the radius of the clusters when possible. This also serves to accelerate the processing. The rationale is that nearby fiducials (e.g., those on the same side of the sample) are likely to spread at similar height in z. We test this assumption for the dataset of Figure 5 using files generated by ClusterAlign that report on the accepted fiducial locations in the central frame of axes. The plot in Figure 8 was produced using the dataset of Figure 7 and Ncluster = auto; it shows that indeed the average and minimum difference in z location is smaller for nearby fiducials. Additional examples are shown in the Supplementary Materials.
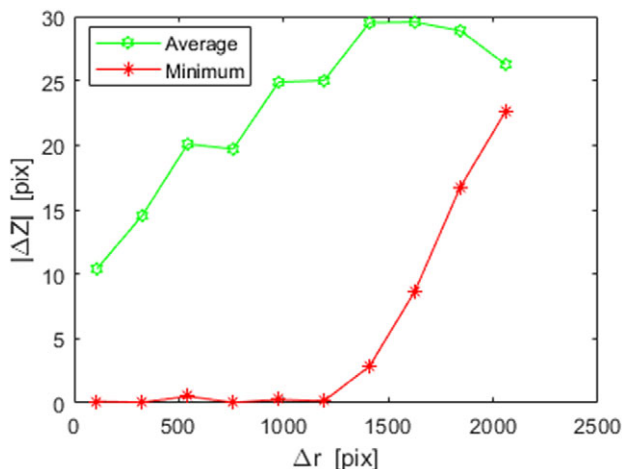

Figure 7. *Fiducials that lie nearby in the plane are likely to lie at similar height (more examples are shown in the Supplementary Materials).*
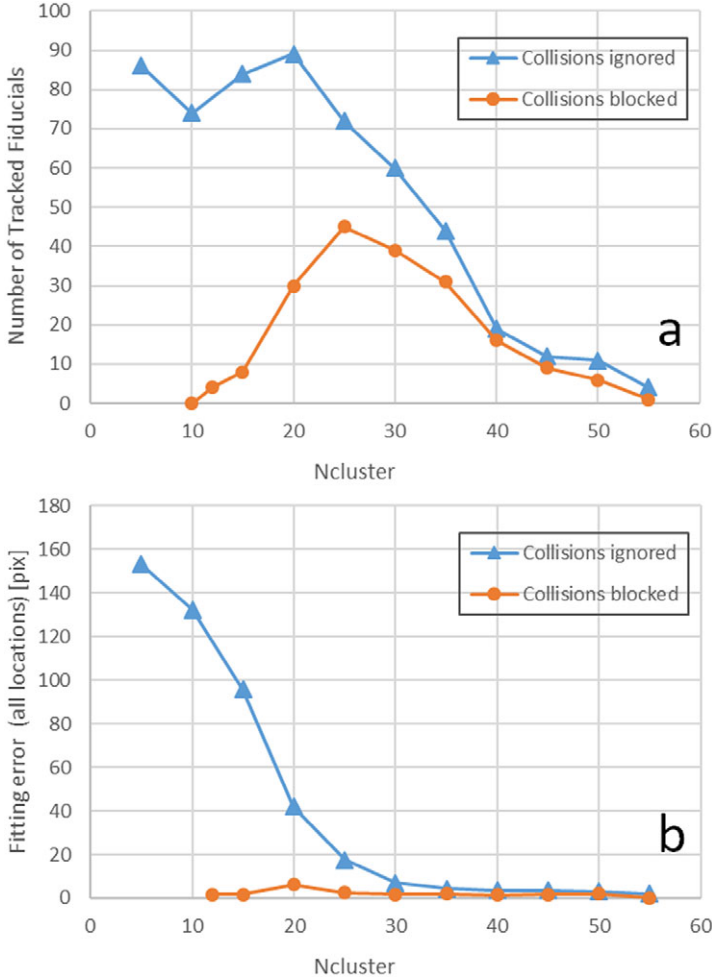
Figure 8. *Effects of cluster size. (a) The number of tracked fiducials reaches a maximum when ambiguous assignments (collisions) are ignored. This feature can be used to assign the cluster size automatically. (b) Blocking of collisions has a strong effect on fitting error for smaller cluster sizes.*

In the following, we present quantitative measures to evaluate the software operation. First, we need a numerical measure of the fiducial visibility. A convenient estimate of the signal to noise ratio related to fiducial detection is found by averaging 2D images of many detected fiducials and their surrounding background. Adapting from[32], we define the contrast to noise ratio (CNR) based on $\langle I_f^2 \rangle$, the average template of all potential fiducials within $\pm 30°$ tilt angles (applicable to the center tilt within 5% accuracy), and the standard deviation (STD) of the raw image as

$$\text{CNR} = \frac{\langle I_f^2 \rangle_{\max}^{1/2} - \langle I_f^2 \rangle_{\min}^{1/2}}{\text{STD}(I)}$$

The difference between the maximum and minimum of the average fiducial image represents the difference between object and background. The square over the image intensity is justified by the observed improvement in fiducial detection compared to linear averaging, consistent with their intensities standing out locally over the specimen. For the example at hand we found CNR = 1.0. This is a rather low value, indicating a difficult dataset for conventional fiducial-based tracking methods. A case of strong

contrast in a plastic-embedded section (training data for IMOD reconstruction, https://bio3d.colorado.edu/imod/doc/etomoTutorial.html) obtained a CNR = 3.5.

Alignment precision depends on the number of tracked fiducials (validated by cluster centers) as well as the fidelity of their localization. As seen in Figure 8a, their number depends inversely on the number of cluster members (NCluster) when collisions, for example, ambiguous assignments in the zero tilt image, are ignored. When collision suppression is enforced, the number of tracked fiducials finds a maximum, as small clusters fare poorly in the collision test. These tendencies repeat in all datasets that we have studied, and imply quite reasonably that the confidence in the discrimination of tracked fiducials is higher when they are supported on a larger cluster count, and that the collision suppression is an effective filter to retain the reliable subset of tracked fiducials. In the example shown the number of tracked fiducial is maximum (45) for NCluster = 25. The maximum in cluster count provides a means to optimize the cluster size automatically in the face of an unknown total number of fiducial particles. We can evaluate the fitting error as the root mean square (RMS) distance in (x,y) between a reported fiducial location and its predicted location according to a fitted rigid body rotation model, for all accepted (tracked) fiducial points. The RMS of the fitting error per slice is stored in a file during execution. Figure 8b shows that the fitting error depends sensitively on the collision blocking but not on the cluster size when colliding assignments are suppressed.

As an independent analysis tool, we evaluate alignment errors based on projection matching, namely, the evaluation of projection shifts, calculated based on peak locations in cross-correlation between the corrected (aligned) tilt images and the reprojected tilt images from a 3D reconstruction. We calculate reprojections from a general reconstruction, leaving out only slices with fitting error exceeding 15 pixels. In comparison, the calculation of resolution based on the value of cross-correlation between image and reprojection is more delicate and requires leaving out the one tilt angle from each dedicated reconstruction[33]. The Matlab code for alignment error assessment is available under the name clusteralign_test.m and makes use of the Astra toolbox[25,26] for weighted back projection reconstruction and projection. (Iterative reconstruction such as SIRT should not be used in this evaluation[11]). Sub-pixel resolution is achieved using Fast Fourier Transform interpolation code[34]. Figure 9a shows that for raw tilt series data, which includes rough misalignments, the quantification of alignment error using projection matching (circles) is similar to the errors based on fitting fiducial locations to a rigid body model (triangles). Since the reconstruction of the unaligned stack is blurry, the cross-correlation at some points is too low to provide numerical values for comparison. However, considering some inaccuracy in both the fiducial locations and correlation, the evaluation is consistent. Figure 9b presents RMS of error in fit of reported fiducial locations to rigid body model after alignment by various methods. The fiducial locations are reported in fid.txt file, per tilt slice, both by ClusterAlign and IMOD. In the case of IMOD, we observe that the tracking is successful, but some of the reported locations are fictitious, that is, they are based on a model and not actually detected. In contrast, the fiducial locations reported by ClusterAlign are found independently of the rigid body model and appear to have lower residues. In the first iteration of ClusterAlign, which obtained 65 tracked fiducials, except for one error point there is improvement over the IMOD report. In the second optical iteration of ClusterAlign, which obtained six tracked fiducials in this case, all large errors are removed and the RMS fit errors of the fiducial locations are mostly below 1 pixel. While the number 6 may appear alarmingly small, it represents six vertices of clusters containing tens of identified nanoparticles, all of which contribute to determination of the tilt axis. It should be noted that IMOD provides further tools to correct manually the locations errors, but our analysis refers to automated batch mode of a difficult tilt series. The batch mode process in IMOD reports on 2.8 nm (1.4 pixels) weighted mean residual error. In Figure 9c, we show the actual alignment errors evaluated based on projection matching evaluation performed on the aligned files. In ClusterAlign, most of the errors are below 0.5 pixels and on average about half the errors from IMOD. In IMOD, three slices at high tilt angles could not be aligned; where the coarse cross-correlation based on image contrast fails, the fiducial tracking cannot recover. These tilts would normally be excluded from the reconstruction, but precisely these high tilt images contribute to resolution in the axial (z) direction.
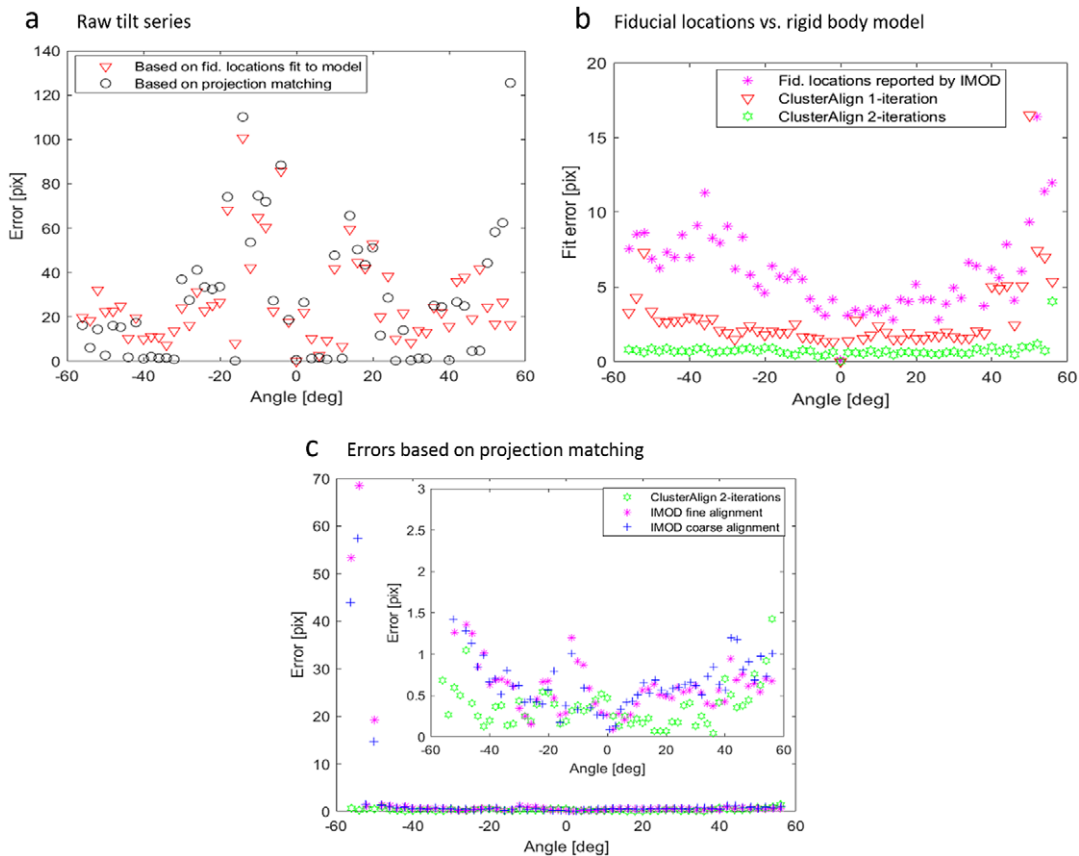
Figure 9. *Alignment errors. (a) The quantification of alignment errors using projection matching (circles) is in correlation with errors based on fitting fiducial locations to a rigid body model (triangles) in the case of a raw tilt series having large misalignments. (b) After alignment, the RMS of errors in fitting the fiducial locations reported in fid.txt file to a rigid body model, per tilt slice. (c) Alignment errors based on projection matching evaluation. The inset shows a zoom-in on the points in the outer chart.*

Another CSTET sample tested (HFF-1 cell) shows a good fiducial visibility (CNR = 2.8) and a copious amount of fiducials. The alignment errors based on projection matching evaluation are shown in Figure 10. The alignment settings used maximum fiducial number 2,000 and average particle size 6.5 pixel. After the first optical iteration ClusterAlign found 389 tracked fiducials, and thus the same number of points was requested in etomo using the batch alignment. After a second optical iteration in ClusterAlign, 16 tracked fiducials remained. The median error increased from about 0.2 pixels to 0.5 pixels, but all the significant errors at high tilts were reduced below 1.5 pixels. For comparison, IMOD performance was either comparable or slightly better in most points, but failed significantly at 5 points of high tilt angles. The correlation between coarse and fine alignment is again dominant. We expect that the performance of ClusterAlign at sub-pixel errors should ideally depend on the fitting error divided by the square root of the number of analyzed points per slice, which is in principle the number of tracked fiducials. Thus, the increase in the median error after the second iteration is a consequence of decrease in the number of tracked fiducials. The latter is a consequence of managed cluster rejections aiming to increase the tracking threshold and achieve higher slice coverage.
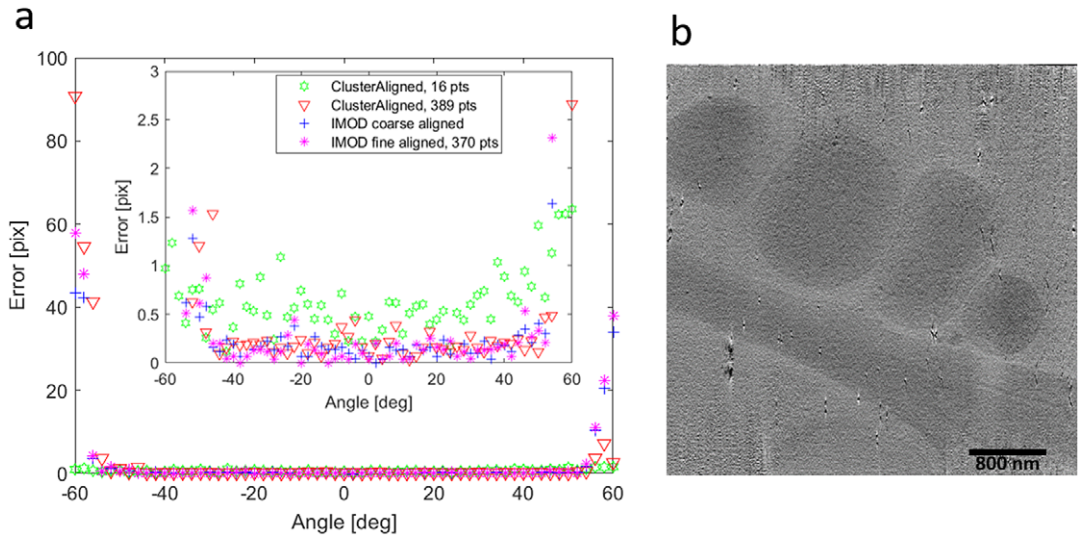
Figure 10. *(a) Alignment errors based on projection matching method for a second dataset. The alignment in ClusterAlign, based on 389 tracked fiducial points after first iteration, is comparable to IMOD alignment. After second iteration in ClusterAlign, 16 tracked fiducials are left, and while the sub-pixel performance is reduced the significant errors at high tilts are eliminated. The inset shows a zoom-in on the points in the outer chart. (b) A section through the reconstruction generated automatically in ClusterAlign.*

4. Discussion and Conclusion

ClusterAlign is a tool for fiducial-based tomographic reconstruction. It offers automated cluster tracking, alignment, and reconstruction that is tolerant to gaps in tilt views and to skew in the rotation axis. Hence, it is especially suited for analysis of thick samples. The software is freely available and should add to the growing toolchest for tomographic reconstruction. Ongoing improvement in tilt series alignment methods carries several positive implications. Foremost, proper alignment at high tilt angles is crucial for depth resolution. Excluding high tilt contributions from reconstruction is acceptable for workflows involving sub-tomogram averaging, where the missing information is obtained from multiple particles in complementary orientations, but the exclusion will be detrimental in reconstruction of unique structures. Second, avoiding the need for manual refinements is essential in order to keep up with ever more rapid data collection. Reconstruction quality depends on sub-pixel precision, as misalignments are projected to the entire thickness of reconstruction. ClusterAlign is especially useful for tomograms that are difficult to align and reconstruct by other means. In addition, we offer several tools to compare performance with other alignment algorithms. These include import and export of fiducial location files in a format compatible with tomoalign and convertible to IMOD .fid, evaluation of the aligned stack on the basis of projection matching, and reporting locations of fiducials without alignment corrections (by setting alignment tolerance to zero). The software also extracts the 3D position of the fiducials and thus provides a tool for characterization of fiducial spread in the sample.

The idea of cluster analysis for fiducial tracking was introduced in TxBr[6,35] and Raptor[1], where clusters are compared between neighboring tilt view for improving the cross-correlation approach. In TxBr, the 3D locations of the fiducials are estimated based on triangulation of two tilt views, but the accuracy depends on successful alignment by cross-correlation. Later, affine-invariant vector products based on four fiducial points were used to provide a unique fingerprint of any such cluster under rotation.[3] The advantage in this elegant approach is that any four points could join a cluster, irrespective of the location in *z*. ClusterAlign is using solely fiducial particle locations for alignment. It searches for unique patterns in many

points at once and is robust for tracking in the case of thick samples, with large lateral movements and poor visibility of fiducials at high tilt angles. A possible extension is to take into account in the second iteration the z values that are known for tracked fiducials in the first. Based on the differences in z, it would then be possible to predict accurately the dependence of the vector length on tilt angle and thus include clusters members irrespective of their height in z; however, this approach has not been implemented so far due to cost in calculation time. Finally, we can comment that the ClusterAlign approach seems to imitate the human perception of tracking fiducials based on their relations to nearby points.

Data Availability Statement. Source code and user manual are available at https://github.com/Pr4Et/ClusterAlign and a ClickOnce Windows application installer is available at https://sourceforge.net/projects/clusteralign/files/.

Supplementary Materials. To view supplementary material for this article, please visit http://doi.org/10.1017/S2633903X22000071.

References

1. Amat F, Moussavi F, Comolli LR, Elidan G, Downing KH & Horowitz M (2008) Markov random field based automatic image alignment for electron tomography. *J Struct Biol* **161**, 260–275. https://doi.org/10.1016/j.jsb.2007.07.007.
2. Han R, Zhang F & Gao X (2018) A fast fiducial marker tracking model for fully automatic alignment in electron tomography. *Bioinformatics* **34**, 853–863. https://doi.org/10.1093/bioinformatics/btx653.
3. Han R, Wang L, Liu Z, Sun F & Zhang F (2015) A novel fully automatic scheme for fiducial marker-based alignment in electron tomography. *J Struct Biol* **192**, 403–417. https://doi.org/10.1016/j.jsb.2015.09.022.
4. Heymann JB (2018) Guidelines for using Bsoft for high resolution reconstruction and validation of biomolecular structures from electron micrographs. *Protein Sci* **27**, 159–171. https://doi.org/10.1002/pro.3293.
5. Mastronarde DN & Held SR (2017) Automated tilt series alignment and tomographic reconstruction in IMOD. *J Struct Biol* **197**, 102–113. https://doi.org/10.1016/j.jsb.2016.07.011.
6. Lawrence A, Bouwer JC, Perkins G & Ellisman MH (2006) Transform-based backprojection for volume reconstruction of large format electron microscope tilt series. *J Struct Biol* **154**, 144–167. https://doi.org/10.1016/j.jsb.2005.12.012.
7. Castaño-Díez D, Scheffer M, Al-Amoudi A & Frangakis AS (2010) Alignator: a GPU powered software package for robust fiducial-less alignment of cryo tilt-series. *J Struct Biol* **170**, 117–126. https://doi.org/10.1016/j.jsb.2010.01.014.
8. Chen M, Bell JM, Shi X, Sun SY, Wang Z & Ludtke SJ (2019) A complete data processing workflow for cryo-ET and subtomogram averaging. *Nat Methods* **16**, 1161–1168. https://doi.org/10.1038/s41592-019-0591-8.
9. Fernandez J-J, Li S, Bharat TAM & Agard DA (2018) Cryo-tomography tilt-series alignment with consideration of the beam-induced sample motion. *J Struct Biol* **202**, 200–209. https://doi.org/10.1016/j.jsb.2018.02.001.
10. Himes BA & Zhang P (2018) emClarity: software for high-resolution cryo-electron tomography and subtomogram averaging. *Nat Methods* **15**, 955–961. https://doi.org/10.1038/s41592-018-0167-z.
11. Houben L & Bar Sadan M (2011) Refinement procedure for the image alignment in high-resolution electron tomography. *Ultramicroscopy* **111**, 1512–1520. https://doi.org/10.1016/j.ultramic.2011.06.001.
12. Messaoudi C, Boudier T, Sorzano COS & Marco S (2007) TomoJ: tomography software for three-dimensional reconstruction in transmission electron microscopy. *BMC Bioinform* **8**, 288. https://doi.org/10.1186/1471-2105-8-288.
13. Noble AJ & Stagg SM (2015) Automated batch fiducial-less tilt-series alignment in Appion using Protomo. *J Struct Biol* **192**, 270. https://doi.org/10.1016/j.jsb.2015.10.003.
14. Zheng S, Wolff G, Greenan G, *et al.* (2022) AreTomo: an integrated software package for automated marker-free, motion-corrected cryo-electron tomographic alignment and reconstruction. *J Struct Biol* **6**, 100068. https://doi.org/10.1101/2022.02.15.480593.
15. Hohmann-Marriott MF, Sousa AA, Azari AA, *et al.* (2009) Nanoscale 3D cellular imaging by axial scanning transmission electron tomography. *Nat Methods* **6**, 729–731. https://doi.org/10.1038/nmeth.1367.

16. Wolf SG, Houben L & Elbaum M (2014) Cryo-scanning transmission electron tomography of vitrified cells. *Nat Methods* **11**, 423–428. https://doi.org/10.1038/nmeth.2842.

17. Do M, Isaacson SA, McDermott G, Le Gros MA & Larabell CA (2015) Imaging and characterizing cells using tomography. *Arch Biochem Biophys* **581**, 111–121. https://doi.org/10.1016/j.abb.2015.01.011.

18. Groen J, Conesa JJ, Valcárcel R & Pereiro E (2019) The cellular landscape by cryo soft X-ray tomography. *Biophys Rev* **11**, 611–619. https://doi.org/10.1007/s12551-019-00567-6.

19. Kapishnikov S, Berthing T, Hviid L, *et al.* (2012) Aligned hemozoin crystals in curved clusters in malarial red blood cells revealed by nanoprobe X-ray Fe fluorescence and diffraction. *Proc Natl Acad Sci* **109**, 11184–11187. https://doi.org/10.1073/pnas.1118134109.

20. Elbaum M (2018) Expanding horizons of cryo-tomography to larger volumes. *Curr Opin Microbiol* **43**, 155–161. https://doi.org/10.1016/j.mib.2018.01.001.

21. Kremer JR, Mastronarde DN & McIntosh JR (1996) Computer visualization of three-dimensional image data using IMOD. *J Struct Biol* **116**, 71–76. https://doi.org/10.1006/jsbi.1996.0013.

22. Bradski G (2000) The openCV library. *Dr. Dobb's J Softw Tools* **120**, 122–125.

23. Emgu CV (2022) http://www.emgu.com/wiki/index.php/Main_Page, accessed 14 August 2022.

24. Seifer S, Houben L & Elbaum M (2021) Flexible STEM with simultaneous phase and depth contrast. *Microsc Microanal* **27**, 1476–1487. https://doi.org/10.1017/S1431927621012861.

25. Palenstijn WJ, Batenburg KJ & Sijbers J (2011) Performance improvements for iterative electron tomography reconstruction using graphics processing units (GPUs). *J Struct Biol* **176**, 250–253. https://doi.org/10.1016/j.jsb.2011.07.017.

26. van Aarle W, Palenstijn WJ, De Beenhouwer J, *et al.* (2015) The ASTRA toolbox: a platform for advanced algorithm development in electron tomography. *Ultramicrosc* **157**, 35–47. https://doi.org/10.1016/j.ultramic.2015.05.002.

27. Agulleiro J-I & Fernandez J-J (2015) Tomo3D 2.0 – exploitation of advanced vector eXtensions (AVX) for 3D reconstruction. *J Struct Biol* **189**, 147–152. https://doi.org/10.1016/j.jsb.2014.11.009.

28. Han R, Li L, Yang P, Zhang F & Gao X (2021) A novel constrained reconstruction model towards high-resolution subtomogram averaging. *Bioinformatics* **37**, 1616–1626. https://doi.org/10.1093/bioinformatics/btz787.

29. Gustafsson N, Culley S, Ashdown G, Owen DM, Pereira PM & Henriques R (2016) Fast live-cell conventional fluorophore nanoscopy with ImageJ through super-resolution radial fluctuations. *Nat Commun* **7**, 12471. https://doi.org/10.1038/ncomms12471.

30. Korn GA & Korn TM (2000) Mathematical handbook for scientists and engineers. In *Definitions, Theorems, and Formulas for Reference and Review.* New York, NY: Dover Publications.

31. Mullick D, Rechav K, Leiserowitz L, Dzikowski R, Regev-Rudzki N & Elbaum M (2022) Diffraction contrast in cryo-scanning transmission electron tomography reveals the boundary of hemozoin crystals in situ. *Faraday Discuss*. https://doi.org/10.1039/D2FD00088A.

32. Luo W, Yoo S, Wu QJ, Wang Z & Yin F-F (2008) Analysis of image quality for real-time target tracking using simultaneous kV-MV imaging. *Med Phys* **35**, 5501–5509. https://doi.org/10.1118/1.3002313.

33. Cardone G, Grünewald K & Steven AC (2005) A resolution criterion for electron tomography based on cross-validation. *J Struct Biol* **151**, 117–129. https://doi.org/10.1016/j.jsb.2005.04.006.

34. Schabel M (2021) *N-dimensional Fourier interpolation, MATLAB Central File Exchange*. https://www.mathworks.com/matlabcentral/fileexchange/22665-n-dimensional-fourier-interpolation, accessed 14 August 2022.

35. Phan S, Boassa D, Nguyen P, *et al* (2016) 3D reconstruction of biological structures: automated procedures for alignment and reconstruction of multiple tilt series in electron tomography. *Adv Struct Chem Imag* **2**, 8. https://doi.org/10.1186/s40679-016-0021-2.