Software Standards in Electron Tomography

May 2024

1 Introduction

Electron tomography is a powerful imaging technique to obtain high-resolution, three-dimensional structures of biological samples. This method involves collecting a series of two-dimensional images of the sample at different tilt angles, typically in a transmission electron microscope (TEM). A comprehensive set of images is acquired by tilting the sample incrementally, which are then computationally reconstructed into a 3D volume. This reconstruction provides detailed insights into the ultrastructure of cellular components, such as organelles, macromolecular complexes, and even individual proteins, allowing researchers to visualize and analyze their spatial organization and interactions within the context of the cellular environment. Electron tomography is particularly valuable for studying the architecture of cells and tissues, bridging the gap between traditional electron microscopy and other imaging modalities like X-ray crystallography or cryo-electron microscopy.

Public databases with electron tomography data are crucial for advancing scientific research by providing open access to high-quality, standardized datasets that facilitate reproducibility and collaborative studies. These databases, such as the CryoET data portal of the Chan Zuckerberg Initiative (CZI, https://cryoetdataportal.czscience.com/), the Electron Microscopy Data Bank (EMDB, https://www.ebi.ac.uk/emdb/), and the Electron Microscopy Public Image Archive (EMPIAR, https://www.ebi.ac.uk/empiar/), serve as essential repositories for researchers to deposit and retrieve tomographic data. Comprehensive datasets are essential for validating research findings and training new computational methods. The availability of many datasets allows for extensive testing and refinement of novel algorithms and structures, accelerating scientific progress and fostering a collaborative research environment. Additionally, as software and analytical tools improve, these archived datasets offer opportunities for reanalysis, potentially leading to discoveries and insights that were missed initially.

Including comprehensive metadata with the electron tomography datasets is vital, as it provides detailed information about the experimental conditions, sample preparation, imaging parameters, and data processing protocols. This contextual information ensures that the data can be accurately interpreted, compared, and reused by other researchers, enhancing the overall utility and impact of the shared data. Essential metadata elements include acquisition order, exposure settings, imaging geometry, and defocus values. These pieces of metadata are critical for ensuring that subsequent analyses are reliable and reproducible. However, having standardized descriptions of these parameters is crucial to maximize their utility across different research groups and software platforms. Without such standardization, metadata reported by one software might be misinterpreted or mishandled by another, leading to errors and inconsistencies in data analysis. Establishing and adhering to standardized metadata formats ensures that datasets are universally understandable and usable, fostering more effective data sharing and collaborative research.

There are several metadata reporting and standardization levels, each crucial for different data acquisition and analysis stages. At the most basic level, metadata standards are needed for individual acquisition frames, capturing parameters such as exposure, defocus, and geometry for each frame. The next level involves metadata for the tilt series, which includes information on the acquisition order and relative orientation. This level ensures that the series of images can be accurately reconstructed into a 3D tomogram. Finally, annotations of tomograms, which involve identifying and labeling specific structures within the 3D volume, require standardized descriptions to facilitate consistent interpretation and comparison across studies.

It is paramount to provide each metadata level with a precise geometrical meaning that can be universally understood. Accurate geometrical meaning ensures that each data acquisition and reconstruction step is clearly defined and interpretable across different software platforms and research studies.

This document lays the groundwork for a standardized geometrical interpretation of metadata associated with cryo-electron tomography. A comparable initiative was previously undertaken for single-particle analysis, as detailed in (Sorzano et al., 2014). The successful implementation of such standards necessitates engagement from a broad spectrum of stakeholders, extending beyond the research community to include equipment manufacturers and data acquisition and analysis software developers. This endeavor is crucial for enhancing the precision and applicability of electron tomography data across diverse scientific disciplines and research groups.

2 Standard definitions

2.1 Acquisition information

Acquisition details describing the microscope, the camera, and the acquisition session are crucial to fully exploit the exchange of tilt series movies or images.

STANDARD: When tilt series movies or images are exchanged, the following parameters must be specified: voltage (kV), spherical aberration (mm), amplitude contrast (fraction normalized to 1), nominal defocus (Å), and nominal phase shift (degrees; if not given assumed to be 0).
OPTIONAL: Other session parameters, such as the energy filter or stage position parameters, may be given to help understand the acquisition setup. However, the description of this second set of parameters is much more variable, and they do not participate in the standard.

Dark and gain images, as well as a defect file, may be provided when available. The defect file should specify the (x, y) indices of pixels to be avoided during image processing (refer to Sec. 2.3 for the standard definition of X and Y).

The dark and gain images must be such that the corrected image is computed as

$$I_{corrected}(x, y) = \frac{I_{acquired}(x, y) - I_{dark}(x, y)}{I_{gain}(x, y)} \tag{1}$$

If a dark image is not provided, it will be presumed to have a value of 0 throughout. Similarly, if a gain image is not provided, it will be assumed to have a value of 1 everywhere.

Although this document does not go in specific file formats, `mdoc` files could be the basis for the interchange of information.

2.2 Points and vectors

Before proceeding, it is essential to establish a common understanding of several fundamental geometrical concepts. The transformations outlined in this section are applied to 3D points, typically expressed in Cartesian coordinates as $(x, y, z) \in \mathbb{R}^3$. It is crucial to differentiate between a 3D point and a 3D vector. While both can be represented by three numerical values (x, y, z), they belong to different mathematical spaces. A 3D point is an element of an affine space, whereas a 3D vector resides in a vector space. An affine space comprises a set A (in our context, $A = \mathbb{R}^3$), a vector space V (here, $V = \mathbb{R}^3$), and a faithful and transitive group action of V on A. This structure allows us to manipulate points (in the affine space) and vectors (in the vector space), enabling operations such as adding a vector to a point to yield a new point or subtracting two points to produce a vector. Conceptually, a point represents a position in a space, while a vector represents a direction.

The geometrical transformations we define are applied to points representing Coulomb potentials' positions within macromolecular structures. We denote the Coulomb potential, which influences the electrons forming the image in an electron microscope, as a function $V(\mathbf{r}) : \mathbb{R}^3 \to \mathbb{R}$, where $\mathbf{r}$ is a point in space.

The approximation of this function is commonly referred to as a tomogram, a volume, or a map. Analogously, we define an image as a function $I(\mathbf{s}) : \mathbb{R}^2 \to \mathbb{R}$.

STANDARD: All coordinates, $\mathbf{r}$ and $\mathbf{s}$, are expressed in Å.

When transforming points, using homogeneous coordinates is conventional. These are obtained by appending a 1 to the standard coordinate values. We denote a point in homogeneous coordinates as $\tilde{\mathbf{r}} = (x, y, z, 1)$, distinguishing it from its non-homogeneous form $\mathbf{r} = (x, y, z)$. Note that $\tilde{\mathbf{r}} \in \mathbb{R}^3 \times \{1\}$. Depending on the context, we may use either $\mathbf{r}$ or $\tilde{\mathbf{r}}$ as the argument of a volume function, with both yielding the same intensity value for the corresponding point.

2.3 2D Coordinate system

When dealing with an image of dimensions $N_x \times N_y$ pixels, it is essential to consider the indexing conventions of different programming languages. Some languages (like C, C++, Python, Java) use zero-based indexing, ranging from 0 to $N_i - 1$ ($i = x, y$; see Fig. 1), while others (such as Fortran, MATLAB) use one-based indexing, ranging from 1 to N_i. We will refer to these pairs as "physical indexes".

However, when defining geometrical transformations, it is often preferable not to be restricted by this physical indexing. Instead, we can arbitrarily set the origin in any location. A typical choice is to place the origin of the coordinate system at the volume's center rather than at a corner. The origin is such that the top-left corner has an index $(-\lfloor \frac{N_x}{2} \rfloor, -\lfloor \frac{N_y}{2} \rfloor)$, where $\lfloor x \rfloor$ denotes the floor function (see Fig. 1). Its coordinates are $(-\lfloor \frac{N_x}{2} T_x \rfloor, -\lfloor \frac{N_y}{2} T_y \rfloor)$, where T_x and T_y are the pixel size in Å along the X and Y axes. We will refer to these coordinates with arbitrary origin as "logical coordinates", and the rest of this article works with them rather than with physical indexes.

STANDARD: Each image should report its top-left pixel's (x, y) logical coordinate and the pixel size in both directions. The top-left pixel is defined as the one with minimum X (left) and minimum Y (top) coordinates. Each image, either a frame or an image of a tilt series, should be accompanied by its stage angle, nominal defocus (Å), acquisition order (1, 2, ...), time stamp, and the accumulated dose before its acquisition in e$^-$/Å^2.

Note that the standard does not specify how the image should be stored in memory or disk. A few well-established standards exist: MRC (Short et al., 2023), TIFF, HDF5, EER (Guo et al., 2020). However, whenever a coordinate is reported, it must be consistent with this standard.

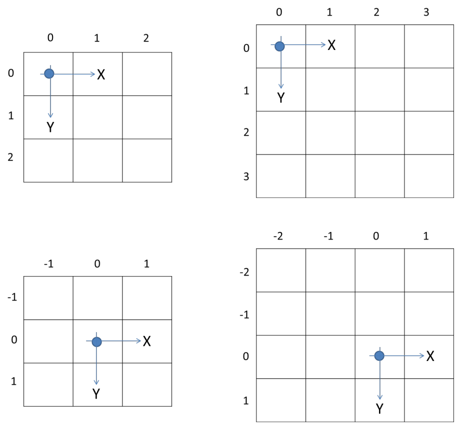

Figure 1: Top row: index of the pixel center for 3 × 3 and 4 × 4 images when the image origin is in one corner of the image (physical indexes). Bottom row: index of the pixel center of the same images when the image origin is in the middle of the image (logical coordinates).

2.4 3D Coordinate system

Establishing a consensus on geometrical matters must begin with a mutual understanding of the coordinate system that provides the framework for expressing various geometrical elements. Heymann et al. (2005) proposed a right-handed coordinate system. The defining characteristic of a right-handed coordinate system is the following set of relationships between the vectors that constitute the coordinate system:

$$\begin{aligned} \mathbf{X} \times \mathbf{Y} &= \mathbf{Z} \\ \mathbf{Y} \times \mathbf{Z} &= \mathbf{X} \\ \mathbf{Z} \times \mathbf{X} &= \mathbf{Y}. \end{aligned} \tag{2}$$

The term “right-handed” derives from the fact that for any cross-product $\mathbf{A} \times \mathbf{B} = \mathbf{C}$, if one curls the fingers of their right hand from **A** to **B** (with fingertips pointing towards **B**), the right thumb will align with the direction of **C**.

In 3DEM, it is often beneficial to visualize the coordinate system related to the electron microscope. The coordinate system’s origin is positioned to coincide with the center of the macromolecule under reconstruction, presumed to be at the level of the sample holder. Electrons travel from negative **Z** to positive **Z** (with **Z** aligned with the microscope column, see Fig. 2). **Y** points towards the microscopist, while **X** points to the microscopist’s right. It is worth noting that an observer within this coordinate system would perceive no difference between the two representations in Fig. 2. The choice between these depictions may depend on which is more intuitive for the operation.

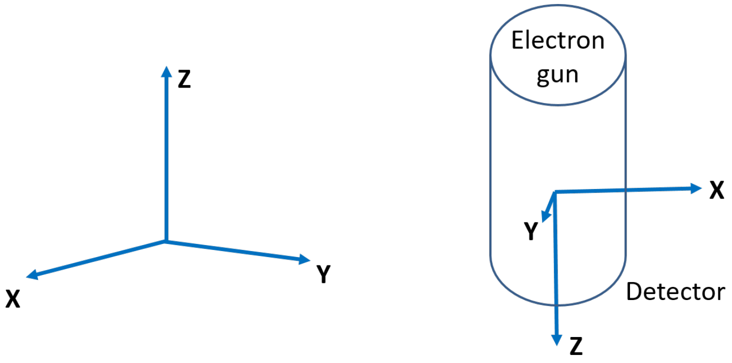

Figure 2: Left: Right-handed coordinate system established as a convention in Heymann et al. (2005). **Z** is vertical, in the paper plane, and **X** and **Y** stand out of the paper, toward the reader. Right: The same coordinate system is used in a different orientation and put in context to the electron microscope.

2.5 Geometrical transformations

Matrix operations on homogeneous coordinates can be employed to represent geometrical transformations:

$$\tilde{\mathbf{r}}_{\tilde{A}} = \tilde{A}\tilde{\mathbf{r}}, \tag{3}$$

In this equation, $\tilde{\mathbf{r}} \in \mathbb{R}^3 \times \{1\}$ denotes the homogeneous coordinate of the point undergoing transformation, $\tilde{\mathbf{r}}_{\tilde{A}} = (x, y, z, 1) \in \mathbb{R}^3 \times \{1\}$ represents its transformed counterpart in homogeneous coordinates, and $\tilde{A}$ is a 4×4 invertible matrix of real numbers, structured as follows:

$$\tilde{A} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}. \tag{4}$$

The tilde notation on matrix A serves as a reminder that it operates on homogeneous coordinates. Matrix $\tilde{A}$ incorporates both translations (represented by vector $\mathbf{t}$) and other transformations such as rotations, reflections, shearing, and scaling (encoded in matrix R). The application of Eq. (4) to any point results in:

$$\mathbf{r}_{\tilde{A}} = R\mathbf{r} + \mathbf{t}, \tag{5}$$

which combines a linear transformation of the point's coordinates with a translation. The matrix $\tilde{A}$ is termed an affine transformation . It possesses several fundamental properties: it maintains the linearity of straight lines (transformed points from a straight line remain on a straight line), preserves distance ratios along a straight line (e.g., the transformed midpoint between two points remains the midpoint of the transformed points), and conserves the parallelism of lines. However, it does not preserve the length of line segments or the angles between adjacent segments.

This definition in 3D can be easily adapted to 2D by simply making the R and $\mathbf{t}$ a 2D matrix and vector.

Geometrical transformations align different types of data in electron microscopy: images with other images, images with volumes, or volumes with other volumes. While we will delve into the specifics of each application later, it is beneficial to introduce their general usage here. Consider a 2D transformation represented by $\tilde{A}$. We define the transformed image $I_{\tilde{A}}(\tilde{\mathbf{s}})$ as follows:

$$I_{\tilde{A}}(\tilde{\mathbf{s}}) = I(\tilde{A}^{-1}\tilde{\mathbf{s}}) \tag{6}$$

We use $\mathbf{r}$ for 3D points and $\mathbf{s}$ for 2D points to clarify the notation. As images are defined in 2D, we use $\mathbf{s}$ for them. Using the inverse of $\tilde{A}$ is for mathematical convenience.

Let us consider the simple example of image translation. In this case,

$$\tilde{A} = \begin{pmatrix} I & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}.$$

where I is the 2×2 identity matrix. Then, this transformation shifts the input image as seen by the following equation and in Fig. 3:

$$I_{\tilde{A}}(\tilde{\mathbf{s}}) = I(\tilde{A}^{-1}\tilde{\mathbf{s}}) = I(\mathbf{s} - \mathbf{t}) \tag{7}$$

We see that $\tilde{A}$ takes points from the I image and reveals their location in $I_{\tilde{A}}$. While $\tilde{A}^{-1}$ takes points from $I_{\tilde{A}}$ and reveal their location in I.

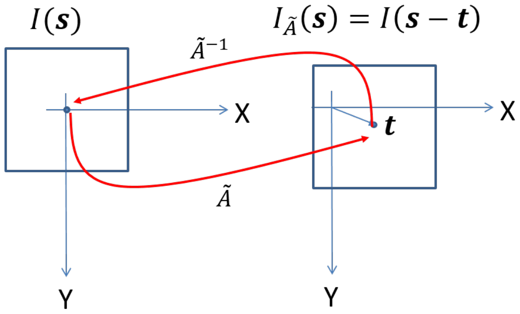

Figure 3: An image is shifted by a the vector $\mathbf{t} = (2, 1)^T$.

Transformation composition

Consider a scenario where we apply a transformation matrix $\tilde{A}_1$ to an image $I(\tilde{\mathbf{s}})$, resulting in a transformed image $I_{\tilde{A}_1}(\tilde{\mathbf{s}})$, defined as $I_{\tilde{A}_1}(\tilde{\mathbf{s}}) = I(\tilde{A}_1^{-1}\tilde{\mathbf{s}})$.

Subsequently, we apply a second transformation matrix $\tilde{A}_2$ to this new image, yielding $I_{\tilde{A}_2}(\tilde{\mathbf{s}}) = I_{\tilde{A}_1}(\tilde{A}_2^{-1}\tilde{\mathbf{s}})$. These two transformations can be consolidated into a single matrix operation by observing that:

$$I_{\tilde{A}_2}(\tilde{\mathbf{s}}) = I_{\tilde{A}_1}(\tilde{A}_2^{-1}\tilde{\mathbf{s}}) = I(\tilde{A}_1^{-1}\tilde{A}_2^{-1}\tilde{\mathbf{s}}) = I((\tilde{A}_2\tilde{A}_1)^{-1}\tilde{\mathbf{s}}). \tag{8}$$

Consequently, the composite transformation can be expressed as $\tilde{A} = \tilde{A}_2\tilde{A}_1$.

It must be noted that we can decompose the matrix $\tilde{A}$ as the multiplication of two other matrices:

$$\tilde{A} = \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} I & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} R & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} = \tilde{T}\tilde{R}, \tag{9}$$

where $\tilde{T}$ represents a pure translation and $\tilde{R}$ is a combination of rotations, scalings, mirrors, and shearings. Then, $\tilde{A}$ first implies applying $\tilde{R}$ and then a translation. Note that matrix multiplication is not commutative and, therefore, $\tilde{T}\tilde{R} \neq \tilde{R}\tilde{T}$.

Transformation properties

If R is a rotation matrix, then it is orthonormal, that is, $R^TR = RR^T = I$, where I is the identity matrix. Or, in other words, its transpose is its inverse.

The modules of the eigenvalues of R reveal the scaling change along some axes. Suppose one of the modules is smaller or larger than 1. In that case, the image is enlarged or compressed along some direction (the direction it is enlarged or compressed depends on the corresponding eigenvector).

If one of the eigenvalues is real-valued and has a negative sign, then there is a mirroring along the direction associated with its eigenvector.

2.6 Frame alignment

Currently, most frame alignment algorithms used to produce images incorporate local deformations. However, the methods for describing these local deformations vary among software packages. Some utilize polynomials in space and time Zheng et al. (2017), while others employ splines Tegunov and Cramer (2019); Strelak et al. (2020). This diversity in approaches makes direct comparisons challenging. Nevertheless, all these algorithms compute a global shift between frames, providing a common ground for comparison.

To facilitate this comparison, let us consider that each frame has its global transformation, denoted by $\tilde{A}_i$. Typically, these transformations account only for shifts, excluding rotations. Using this framework, we can express the generation of an image, only compensating for the global shift, as follows:

$$I_{avg}(\tilde{\mathbf{s}}) = \frac{1}{N}\sum_{i=1}^{N} I_{\tilde{A}_i}(\tilde{\mathbf{s}}). \tag{10}$$

where N is the number of frames, and $I_{\tilde{A}_i}$ are the aligned frames under the transformation $\tilde{A}_i$. The reference frame is also different for each program. However, we can quickly identify it because its transformation is the identity matrix.

STANDARD: Each program should report the $\tilde{A}_i$ matrix for each frame for the interchange of frame alignment information.

2.7 Tilt series alignment and tomogram reconstruction

Let us define the projection operator P that projects any tomogram, $V(\mathbf{r})$, tilted or not, along the Z direction:

$$I(\tilde{\mathbf{s}}) = P\{V(\mathbf{r})\} = \int\limits_{-\infty}^{\infty} V(\tilde{H}_z^T \mathbf{s}) dz \tag{11}$$

where

$$\tilde{H}_z^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & z \\ 0 & 0 & 1 \end{pmatrix} \tag{12}$$

This way, H_z is a matrix that transforms 2D coordinates $\mathbf{s}$ into 3D coordinates $\mathbf{r}$.

At this point, let us extend Eq. 6 to 3D by simply allowing $\tilde{A}$ to be a 3D transformation matrix applied to a 3D map, V (the tomogram). Then, the j-th image within a tilt series is coming from a tilted version of the tomogram that can be calculated as

$$V_{\tilde{A}_j}(\tilde{\mathbf{r}}) = V(\tilde{A}_j^{-1}\tilde{\mathbf{r}}) \tag{13}$$

and the j-th image is its projection along Z

$$I_{\tilde{A}_j}(\tilde{\mathbf{s}}) = P\{V_{\tilde{A}_j}(\mathbf{r})\} = \int\limits_{-\infty}^{\infty} V(\tilde{A}_j^{-1}\tilde{H}_z^T \mathbf{s}) dz \tag{14}$$

STANDARD: Each program should report the $\tilde{A}_j$ matrix for each tilted image to interchange the information of how to align it to a given tomogram.

The matrix $\tilde{A}_j$ provides a change of basis matrix that transforms 3D points from the tomogram, $\tilde{\mathbf{r}}$, into 3D points in a coordinate system within the projection image that we will refer to as $\tilde{\mathbf{r}}_j$, such that

$$\tilde{\mathbf{r}}_j = \tilde{A}_j \tilde{\mathbf{r}} \tag{15}$$

The projection of this point onto the image of the tilt series is simply

$$\tilde{\mathbf{s}} = \tilde{H}_0 \tilde{\mathbf{r}}_j = \tilde{H}_0 \tilde{A}_j \tilde{\mathbf{r}} \tag{16}$$

where $\tilde{H}_0$ is defined as in Eq. 12 with $z = 0$.

One of the notable features of the $\tilde{A}_j$ matrix is its third column because it gives the projection direction of the image in the coordinate system of the tomogram.

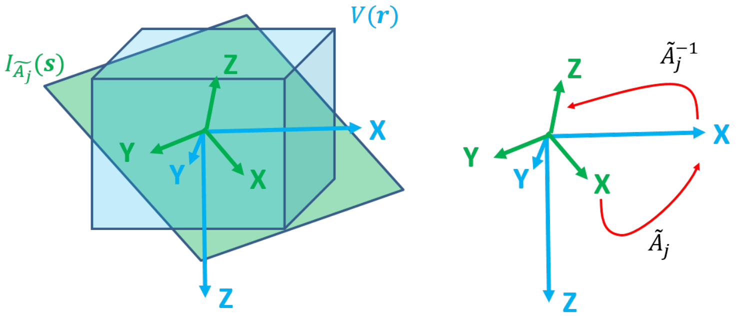

Figure 4: Left: The coordinate origin in the tomogram, $V(\mathbf{r})$, and the j-th image of the tilt series, $I_{\tilde{A}_j}(\tilde{\mathbf{s}})$, are the same. Right: The matrix $\tilde{A}_j$ provides the matrix needed to go from the tilt series coordinates (green) to the tomogram coordinates (blue), and viceversa.

In the simplest case in which the transformation only implies rotations around the Y axis, we would have

$$\tilde{A}_j = \begin{pmatrix} \cos(\theta_j) & 0 & \sin(\theta_j) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_j) & 0 & \cos(\theta_j) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{17}$$

If we represent the sample tilting within the electron microscope as shown in Fig. 2, then we get the result shown in Fig. 5. This should also be the effect of the tilting once the tilt axis is aligned with Y. However, the $\tilde{A}_j$ is a wealthy description of the projection geometry. It can accommodate the arbitrary location of the tilt axis, even if this is not perpendicular to the electron beam.

STANDARD: When the tilt axis is aligned with Y (normally the standard for aligned images), a positive tilt angle moves the positive X axis against the electron beam direction.

Acquisition geometry, missing wedge, and tilt gaps

The missing wedge in cryo-electron tomography comes from the fact that the sample cannot be tilted within the microscope more than a given amount. Typ-

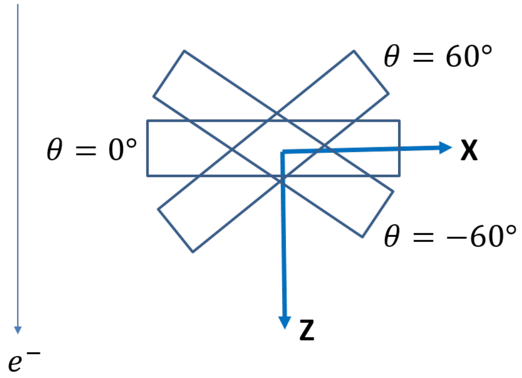

Figure 5: A sample is tilted by different angles θ. The Y coordinate points towards the reader (see Fig. 2).

ical tilt series go from -60° to +60° in tilt. But other configurations can be found. For instance, when FIB-SEM lamellas are visualized by cryoET, the sample is already tilted to the electron beam even if the sample holder is at 0$\circ$ tilt.

Additionally, the quality of the reconstructed tomogram depends on the angular distance between consecutive tilts, which may not be uniformly distributed throughout the tilt range.

Finally, the tilt axis may not be perfectly perpendicular to the electron beam; there might be slight misadjustments in the tilt axis among the different images, etc. All these effects result in a complicated tilting geometry (Turonova et al., 2016).

The standard successfully addresses all these subtleties because an independent alignment matrix, $\tilde{A}_j$, is provided for each image in the tilt series.

Still, there might be even more sophisticated effects that are not contemplated. For instance, the grid might be locally bent or have independent local deformations. In all these situations, the projection geometry of each region is different depending on its location within the grid, and the single alignment matrix, $\tilde{A}_j$, is a first-order approximation to the projection geometry. More sophisticated approaches are, for instance, provided by Fernandez and Li (2021), but they are out of the definition of this standard.

2.8 Contrast Transfer Function (CTF)

This tilting convention has an impact on the defocus. CryoET typically works in underfocus. The standard way to report the average defocus is with a negative sign and the units in Å (Sorzano et al., 2007; Rohou and Grigorieff, 2015). When the sample is tilted, the defocus closer to the image formation plane has larger (less negative) values than those further away (see Fig. 6).

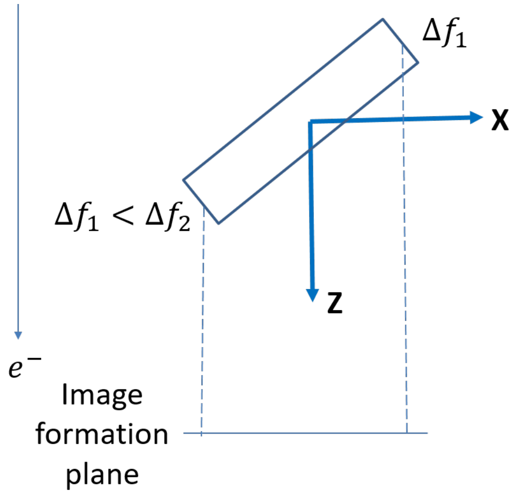

Figure 6: Consequences of the tilting convention on the defocus values.

STANDARD: The defocus of a tilt series is reported for the 0-tilt image without any alignment to the tomogram. The defocus orientation can be corrected for 3D reconstruction by applying $\tilde{A}_0$. Points farther from the image formation plane have a more negative defocus than points closer to it.

We must also specify how to report the CTF for the 0°-tilt image.

STANDARD: For the 0°-tilt image, the CTF should be determined by two defoci in Å, called Δf_U and Δf_V, and the angle that Δf_U makes with the X frequency axis (see Fig. 7). If not given, it will be assumed a phase shift of 0°.

Programs may compute more parameters to describe the CTF, but they do not participate in the standard.

2.9 Point annotations

Given a tomogram, $V(\mathbf{r})$, we may specify a 3D point annotation by the logical coordinates of the point of interest. Let us refer to it as $\mathbf{r}_n$. To do so, each tomogram must be accompanied by the logical coordinate of its top-left-back voxel.

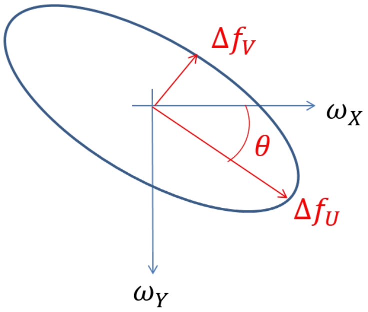

Figure 7: Standard definition of the defoci for the 0°-tilt image.

STANDARD: Each tomogram should report the (x, y, z) logical coordinate of its top-left-back voxel and the voxel size in each direction. The top-left-back voxel is defined as the one with minimum X (left), minimum Y (top), and minimum Z (back) coordinates.

A 3D point annotation may be qualified by any number of suitable features, and in general, we will refer to the set of 3D point annotations as

$$\mathcal{A}_V = \{(\tilde{\mathbf{r}}_n, f_n)\} \tag{18}$$

where the nature of the feature f_n can be varied:

- Discrete label: We may label these points as belonging to the "cellular membrane", "nuclear membrane", ... "ribosome in state 1", "ribosome in state 2", ... "microtubule no. 1", "microtubule no. 2", ... A 3D point annotation may have multiple labels of this kind, depending on the different dictionaries that we define for a given tomogram.

- Scalar value: We may label these points depending on any scalar feature that we may find sensible, e.g., the size of a box surrounding the point that defines a subtomogram, the local average defocus, resolution, intensity, or the probability of belonging to a membrane.

- Vector values: The labels may be vectors. For instance, we may associate each point with two defoci and the angle they make with the Y axis, the direction of the normal to a given structure (e.g., the cellular membrane at that location) in the coordinate system of the tomogram, the probability of belonging to a structure of type A, B, or C, or the eigenvalues of the inertia moment tensor at that location.

- Matrix: The labels may also be matrices. One of the most interesting examples is when the 3D annotation represents the center of a subtomogram. Then, we may align these subtomograms to a reference structure using an alignment matrix $\tilde{A}_n$ as we did in Eq. 13. This time, the reference volume plays the role of V, and the role of $V_{\tilde{A}_n}$ is played by the subtomogram as extracted from the tomogram.

- Tensor: Finally, the labels can be represented as a tensor containing relevant content specific to the application.

There are several important consequences of this definition:

1. The set of annotations, $\mathcal{A}_V$, makes full sense in the context of its tomogram, $V(\mathbf{r})$.
2. A set of subtomograms within a tomogram is nothing more than a set of 3D annotations with features such as the box size, the subtomogram class as a discrete label or as a vector of probabilities to belong to specific classes, and, optionally, one or several alignment matrices.
3. A set of subtomograms missing their cellular context, that is, with no reference to V or $\mathbf{r}_n$ is suboptimal as we miss their relative position within the cell, to the water-air interface, their height (and local defocus) within the tomogram, etc.

The same concept of 3D point annotations could be extended to 2D point annotations. The 2D points can be genuinely defined in the tilt series images, such as the center of gold beads or any other fiducial marker throughout the tilt series. However, 2D points can also result from projecting 3D points onto the tilt series, as in the per-particle, per-tilt approach (Chen et al., 2019). In this case, rather than defining a subtomogram by a single 3D point $\mathbf{r}_n$, it would be characterized by a series of 2D points $\mathbf{s}_{j,n}$ corresponding to its projections across the tilt series (Eq. 16)

$$\tilde{\mathbf{s}}_{j,n} = \tilde{H}_0 \tilde{A}_j \tilde{\mathbf{r}}_n \tag{19}$$

2D point annotations would be referred to a tilt series T and, specifically, to its j-image:

$$\mathcal{A}_T = \{(\{\tilde{\mathbf{s}}_{j,n}, f_{j,n}\}, f_n)\} \tag{20}$$

The set now has features at the 2D point level in the tilt image (j, n) and the object level (n).

2.10 Point structure annotations

Creating a more complex hierarchical structure of annotations is possible when working with a collection of N point annotations. These higher-level annotations could represent various biological structures, such as filaments (curves), membranes (surfaces), or cellular organelles (volumetric structures). To geometrically define these structures, we need to establish their spatial relationships. In this standard, we accomplish this by using an adjacency matrix to represent a graph. In this graph, each point in the set serves as a node, while the edges describe a single characteristic of the entire point structure. The graph may be directional or non-directional; this property must be explicitly stated. Additionally, the graph can incorporate weighted edges or even labeled edges. However, it is essential to note that all edge features must be of the same type within a single graph. If multiple features need to be associated with each edge, this should be implemented by creating various adjacency matrices, each defining a separate graph for each feature type.

2.11 Volumetric or image annotations

Tomograms and micrographs can be enhanced with volumetric or image annotations. These annotations might include local resolution estimates for each voxel in a tomogram, probabilities of belonging to specific structures, or estimated absorption at particular energy levels. Tomogram segmentations also fall into this category of annotations. Segmentations can be classified into two types: 1) semantic segmentations, where each voxel is categorized into a class (e.g., identifying a voxel as part of a microtubule), and 2) instance segmentations, where each voxel is assigned to a specific object (e.g., identifying a voxel as part of Microtubule 1).

While we will focus on explaining these concepts about maps, it is essential to note that they are equally applicable to images. The label file associated with these annotations does not necessarily need to match the shape and voxel size of the referenced tomogram, although this is often the case for simplicity. However, as stipulated in the standard for map interchange, the label file must specify the coordinates of its top-left-back voxel and the voxel size in all dimensions.

Labels can be represented as integers that serve as indices to an accompanying dictionary. This approach allows a single annotation file to contain multiple annotations (such as instance segmentations of all microtubules in a sample) without spatial overlap. Alternatively, labels can be real-valued. The labels' meaning must be clearly defined in the accompanying documentation, defining a dictionary for the integer annotations and a text for the real-valued ones.

2.12 Hierarchy of annotations

Based on the annotations already described, we propose a hierarchy of them with a higher biological meaning. The sets of point annotations, point structures, and volumetric annotations are called primitive annotations. Each primitive annotation would have a name indicating its meaning (e.g., gold beads, nuclear membrane, microtubules, microtubule 1, ...)

Then, the hierarchy of annotations can be constructed using the following construction rules:

$$\begin{array}{rcl} \text{Annotation} & \leftarrow & \text{Primitive Annotation}|\text{Composite Annotation} \\ \text{Composite Annotation} & \leftarrow & \{\text{Annotation}+\} \end{array} \tag{21}$$

where | means "or", $\{\cdot\}$ represents a set, and + means 1 or more.

This way, we can define annotations made of Primitive Annotations or any nested structure of sets of Primitive and Composite Annotations. The nested nature must be understood as the relationship “is part of”.

Finally, Composite Annotations can relate to each other in a directed graph with relationships that must be taken from an ontology and that describe their relations, for example, “is adjacent to”, “is connected to”, “interacts with”, “transports to”, “transports from”, “anchors to”, ...

3 Conclusions

This article does not develop a specific syntax for specifying all these elements. Instead, we provide a clear and precise geometrical interpretation of the parameters involved. By establishing a standardized framework for understanding the geometric aspects of metadata associated with electron tomography, we aim to facilitate consistent and accurate data sharing and analysis across different research groups and software platforms. This approach ensures that while the syntax may vary, the underlying geometrical principles remain uniform, enhancing the reliability and reproducibility of scientific findings.

Developing tools to establish compliance with the proposed standard would be immensely beneficial. Such tools would enable researchers and developers to verify that their software adheres to this document’s standardized geometrical and metadata interpretations. These tools would facilitate seamless data exchange and integration across different platforms and research groups by ensuring compatibility. Additionally, compliance verification tools would help identify discrepancies and areas for improvement, fostering the development of robust and interoperable software solutions.

References

Chen, M., Bell, J. M., Shi, X., Sun, S. Y., Wang, Z., and Ludtke, S. J. (2019). A complete data processing workflow for cryo-et and subtomogram averaging. *Nature methods*, 16(11):1161–1168.

Fernandez, J.-J. and Li, S. (2021). Tomoalign: A novel approach to correcting sample motion and 3d ctf in cryoet. *J. Structural Biology*, 213(4):107778.

Guo, H., Franken, E., Deng, Y., Benlekbir, S., Singla Lezcano, G., Janssen, B., Yu, L., Ripstein, Z., Tan, Y., and Rubinstein, J. (2020). Electron-event representation data enable efficient cryoEM file storage with full preservation of spatial and temporal resolution. *IUCrJ*, 7(5).

Heymann, J. B., Chagoyen, M., and Belnap, D. M. (2005). Common conventions for interchange and archiving of three-dimensional electron microscopy information in structural biology. *J. Structural Biology*, 151:196–207.

Rohou, A. and Grigorieff, N. (2015). Ctffind4: Fast and accurate defocus estimation from electron micrographs. *J. Structural Biology*, 192(2):216–221.

Short, J., Palmer, C., Burnley, T., Winn, M., Zhang, Q., Venkataram Prasad, B., Chen, S., Crowther, R., Unwin, P., and Henderson, R. (2023). MRC2020: improvements to ximdisp and the mrc image-processing programs. *IUCrJ*, 10:579–583.

Sorzano, C. O. S., Jonic, S., Núñez Ramírez, R., Boisset, N., and Carazo, J. M. (2007). Fast, robust and accurate determination of transmission electron microscopy contrast transfer function. *J. Structural Biology*, 160:249–262.

Sorzano, C. O. S., Marabini, R., Vargas, J., Otón, J., Cuenca-Alba, J., Quintana, A., de la Rosa-Trevín, J. M., and Carazo, J. M. (2014). *Computational Methods for Three-Dimensional Microscopy Reconstruction*, chapter Interchanging geometry information in electron microscopy single particle analysis: mathematical context for the development of a standard, pages 7–42. Springer.

Strelak, D., Filipovic, J., Jiménez-Moreno, A., Carazo, J. M., and Sorzano, C. O. S. (2020). Flexalign: An accurate and fast algorithm for movie alignment in cryo-electron microscopy. *Electronics*, (9):1040.

Tegunov, D. and Cramer, P. (2019). Real-time cryo-electron microscopy data preprocessing with Warp. *Nature methods*, 16:1146–1152.

Turonova, B., Marsalek, L., and Slusallek, P. (2016). On geometric artifacts in cryo electron tomography. *Ultramicroscopy*, 163:48–61.

Zheng, S. Q., Palovcak, E., Armache, J.-P., Verba, K. A., Cheng, Y., and Agard, D. A. (2017). Motioncor2: anisotropic correction of beam-induced motion for improved cryo-electron microscopy. *Nature methods*, 14:331–332.