Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Structural Biology 151 (2005) 196–207
Journal of Structural Biology
www.elsevier.com/locate/yjsbi

Common conventions for interchange and archiving of three-dimensional electron microscopy information in structural biology

J. Bernard Heymann [a,b], Mónica Chagoyen [c], David M. Belnap [a,*]

[a] *Laboratory of Structural Biology Research, National Institute of Arthritis, Musculoskeletal and Skin Diseases, National Institutes of Health, Bethesda, MD 20892, USA*
[b] *Division of Biology, Caltech, 1200 East California Blvd, Pasadena, CA 91125, USA*
[c] *Centro Nacional de Biotecnologia—CSIC, Campus Universidad Autonoma, 28049 Madrid, Spain*

Received 26 April 2005; received in revised form 2 June 2005; accepted 7 June 2005
Available online 5 July 2005

Abstract

Three-dimensional electron microscopy (3DEM) has made significant contributions to structural biology. To accomplish this feat, many image-processing software packages were developed by various laboratories. The independent development of methods naturally implied the adoption of dissimilar conventions—penalizing users who want to take advantage of the wealth of algorithms from different packages. In addition, a public repository of 3DEM research results, the EM Data Bank, is now established. In an era where information exchange is important, standardizing conventions is a necessity. The 3DEM field requires a consistent set of conventions. We propose a set of common conventions named the "3DEM Image Conventions." They are designed as a standardized approach to image interpretation and presentation. In this regard, the conventions serve as a first step on which to build data-exchange solutions among existing software packages and as a vehicle for homogenous data representation in data archives, such as the EM Data Bank.
Published by Elsevier Inc.

Keywords: Cryo-electron microscopy; Electron cryo-microscopy; File formats; Geometrical convention; Geometry; Image processing; Image standards; Imaging; Single-particle analysis; Standard geometry; Standards; Symmetry notation; Tomography; Two-dimensional crystallography

1. Introduction

Biological electron microscopy spans the size range from molecular to cellular structures, providing an important tool to understand the cell and its molecular components. The use of computer image processing techniques greatly enhanced the information gleaned from electron micrographs. Not only were improvements made to the two-dimensional (2D)[1] results obtained from the microscope, image processing made it possible to reconstruct structures in three dimensions.

Three-dimensional electron microscopy (3DEM) has had its greatest usefulness in understanding structures of macromolecular complexes. The structures of several protein complexes were solved to near-atomic resolution by 2D crystallography or helical reconstruction. Images

* Corresponding author. Present address: Department of Chemistry and Biochemistry, Brigham Young University, Provo, Utah 84602, USA. Fax: +1 801 422 0153.
E-mail address: David_Belnap@byu.edu (D.M. Belnap).

[1] *Abbreviations used:* 2D, two-dimensional; 3D, three-dimensional; 3DEM, three-dimensional electron microscopy; CIF, crystallographic information file; EM, electron microscopy; EMDB, EM Data Bank; STAR, Self-Defining Text Archiving and Retrieval; XML, Extensible Markup Language.

1047-8477/$ - see front matter. Published by Elsevier Inc.
doi:10.1016/j.jsb.2005.06.001

of randomly oriented particles ("single-particle" images) have led to many reconstructions at resolutions from ~40 Å to better than 10 Å. Electron tomography has provided detailed 3D insight into viral and cellular structures approaching a resolution of 40-50 Å. In some cases, dynamic processes have been imaged by 3DEM, including a progression of states showing macromolecular metamorphoses.

With such a powerful tool and wealth of information produced, many and diverse techniques arose to process electron micrographs and reconstruct 2D or 3D maps of the specimens. These techniques were implemented in many software packages. The original packages appeared in the 1960s and 1970s, beginning with the work of DeRosier and Klug (1968).[2] The original packages benefited from exchanges of code that still form the backbone of several today. Over the years other packages were developed—often derivatives of the original ones. Many of the conventions adopted in the software packages have similar origins. However, even small differences contributed to confusion about the exact meaning of parameters used in image processing and led to multiple conventions throughout the 3DEM field; and now the interchange of information between packages is a serious productivity restraint. Users are typically limited to one package—because the transfer of parameters between packages is non-trivial and time-consuming. Furthermore, the complexity due to convention differences hampers the automation of large-scale processing.

A significant development in the 3DEM field is the establishment of a public database for electron microscopy data named the Electron Microscopy Data Bank (EMDB) (http://www.ebi.ac.uk/msd; Editorial, 2003; Fuller, 2003; Henrick et al., 2003; Tagari et al., 2002). This effort was initiated in Europe by the BioImage project (Carazo et al., 1999) and completed by the Integrated Information for Macromolecular Structure (IIMS) project. It is currently part of the European Bioinformatics Institute. Such a public forum requires a single set of common conventions to be able to serve the community in a self-consistent manner.

Here we propose common conventions, which we call the "3DEM Image Conventions," as well as a set of recommendations for those involved in the task of encoding information for 3DEM digital images. The 3DEM Image Conventions include basic definitions and a common nomenclature relevant to the representation and interpretation of 3DEM data.

The standard conventions are based on four principles:

- *Generality.* The conventions must be applicable across the field with no exceptions.
- *Commonality.* We selected the most widely used or easiest understandable convention in most cases.
- *Clarity.* The most unambiguous statement of a convention is in mathematical form. Otherwise, we attempt to state the convention as clearly as possible.
- *Fairness.* The conventions may not be based on those of any specific software package, but rather evolve within a public forum, such as the EMDB.

Several of the conventions were proposed previously (Heymann, 2001).

The purposes of this document are threefold:

- To propose an open standard set of conventions for the 3DEM field. Some foreseen uses of this open standard are data exchange between software packages and data archiving in the EMDB. The conventions will be publicly available as a set of formal descriptions and specifications.
- To provide a technical recommendation for use in developing a broader consensus on what is required to ensure 'image data' interoperability among software packages in the 3DEM field and data exchange among the wider structural biology community (encompassing other structural data, such as atomic models).
- To recommend that the 3DEM community adopt a common parameter-file format for data exchange and archiving, and that all software packages read and write this parameter file.

Our recommendation *does not*:

- Specify a particular design or an implementation of an image-file format or a parameter-file format.
- Compare or list the conventions of current software packages.
- Provide direction on converting parameters from one package to another or on how to extract data from parameter files or data archives. It is our assumption that the responsibility for implementation of the conventions lies with the developers of the various software packages who will write routines to read and write the common parameter file.
- Require software developers to change their internal conventions.
- Provide any tools or data to allow conformance testing.

2. The 3DEM Image Conventions

All of the conventions below relate to digital images and the parameters required to process their content in

[2] For a summary of the history of 3DEM software and a description of 3DEM software packages available in the mid-1990s, please see Carragher and Smith (1996) and other articles in the same issue of J. Struct. Biol. (116 (1), January/February 1996).

a 3DEM experiment. Information associated with images can be characterized as that required to allow a computer program to represent it and that required to interpret its content. Therefore, there is a particular focus on digital 2D projection images and 3D density maps, as they are the primary classes of 2D and 3D images, respectively, that are relevant to a 3DEM experiment.

2.1. Digital image

A digital image is a finite set of *data elements* placed on a *grid*.

2.1.1. Grid

The *grid* (or lattice or matrix) is the set of points in physical space at which a physical quantity is measured or estimated.

2.1.2. Data element (pixel)

Each *data element* in an image is commonly referred to as a *pixel* (in 2D) or a *voxel* (in 3D). Each element is comprised of *coordinates* that define its position on the grid and an *intensity value* that represents a physical measurement (e.g., electron potential in 3D maps).[3]

2.1.3. Types of data-element values

Data element values can be *simple* or *compound*. Simple element types cannot be decomposed into smaller units (e.g., a gray-scale image), while compound types are aggregations of one or more simple types (e.g., three-channel elements encoded as red-green-blue (RGB) values).

2.1.4. Image dimensions

A digital image usually has two or three *dimensions*, though an image can be one-dimensional. Also, a series of 2D or 3D images could be assembled into additional dimensions, for example, as samples in time or other physical or chemical properties.

An image may have any number of elements in each dimension. The number of ordered elements in each dimension is the *size* of that dimension.

2.2. Image content

An image corresponds to a physical entity. Hence, image data elements represent a sampling in space. To correctly interpret an image's contents the following information is necessary to provide unambiguous mapping information between its layout and its real-world context.

2.2.1. Dimensions

2.2.1.1. Dimension type. The *dimension type* is the nature of the physical dimension along each image dimension. In most cases, the first three image dimensions are length. Other measurements can be represented (e.g., time in time-resolved microscopy or frequency in Fourier transforms). Dimension types should be specified. Software packages should explicitly provide for a proper digital image representation (either within the digital data structure or in publicly available documentation).

2.2.1.2. Grid convention. The most widely used grids in image representations, and therefore the ones adopted as the 3DEM convention, are the square (2D) and cubic (3D) grids, where neighboring data elements are orthogonal to each other. Other grids are possible (e.g., face-centered or body-centered cubic lattice), but must be specified as non-orthogonal axes (see Section 2.2.2).

2.2.1.3. Sampling interval, grid step-size. As with separated points on a line, distance exists between data elements (pixels). For regularly sampled dimensions, a *sampling interval* is defined. Adjacent, regularly spaced pixels on the same axis are separated by an integral step of one pixel-unit. This sampling interval also can be expressed as a physical dimension (e.g., the length interval between adjacent elements in a 2D or 3D image).

The *grid step-size* (also called *pixel size* or *voxel size*) is the distance between adjacent data elements along the same axis. Units are given in standard SI units where applicable (see Section 2.2.9). For example, length dimensions are given in meter units (e.g., Ångstroms, nm, or μm). Since each dimension could have a different step-size, a grid step-size for each dimension should be given.

The sampling convention is regularly spaced lengths given as the *grid step-size*. For the case of irregular sampling, a sampling vector is needed (e.g., time-resolved experiments where images are taken at irregular time points). Irregular samplings, if used, should be specified.

2.2.2. Coordinate system

Dimensions in the physical world can be measured via a number of reference coordinate systems. For example, the Cartesian coordinate system has three orthogonal axes with dimensions corresponding to the (x, y, z) coordinates. Spherical and polar coordinate systems have one length and two angular dimensions. Non-orthogonal systems have three non-orthogonal axes (e.g., lattice-based systems in crystallography).

The 3DEM convention is to use *right-handed*, *Cartesian coordinates* with *right-handed rotations* designated

[3] A data element in 3DEM usually originates as the intensity of a 2D sampled area (e.g., from an element of a charge-coupled device (CCD) array or from a scanned area on film). Hence, a data element usually represents the average intensity of the sampled area or volume. A pixel may be considered either as (i) an area or volume centered on the specified coordinate or (ii) a point in space at the specified coordinate.

as *positive*. When an alternative coordinate system is used, it should be specified, and the software package using it must accommodate it in the code and documentation.

2.2.2.1. Orthogonal Cartesian axes. Coordinates are defined by the Cartesian **xyz** axes. Following crystallographic convention, the angles between the axes are defined as α(**y**–**z**), β(**x**–**z**), and γ(**x**–**y**). $\alpha = \beta = \gamma = 90°$ is the standard for image data. (In the context of crystallographic objects, the unit cell parameters define the lattice and reciprocal space representation. One may have non-orthogonal space groups in crystallography, but it is advantageous to use orthogonal coordinates for data exchange and archiving.)

2.2.2.2. Right-handed axis order. The order of the Cartesian axes is right-handed (Fig. 1A). The order is defined by the following cross-product relationships for the orthogonal unit vectors **x**, **y**, and **z**: $\mathbf{z} = \mathbf{x} \times \mathbf{y}$, $\mathbf{y} = \mathbf{z} \times \mathbf{x}$, and $\mathbf{x} = \mathbf{y} \times \mathbf{z}$.

2.2.2.3. Rotations. Positive rotations about any axis or vector are right-handed. For example, if one looks down the positive **z**-axis (i.e., $+z$ pointing at viewer), a positive rotation angle is from the positive **x**-axis to the positive **y**-axis (an anti-clockwise rotation).

2.2.2.4. Standard display orientation. The *standard display orientation* for a Cartesian coordinate system has the x-axis increasing from left to right, the y-axis increasing from bottom to top, and the positive z-axis pointing towards the viewer. If an alternate convention is used, the program should display axes on the screen or describe the alternate convention in the documentation.

2.2.3. Symmetry notation

2.2.3.1. Point-group symmetries. Point-group symmetries are designated with Schoenflies notation (Table 1) (e.g., C3 for cyclic threefold symmetry, D6 for dihedral sixfold symmetry, I for icosahedral symmetry, and so forth).

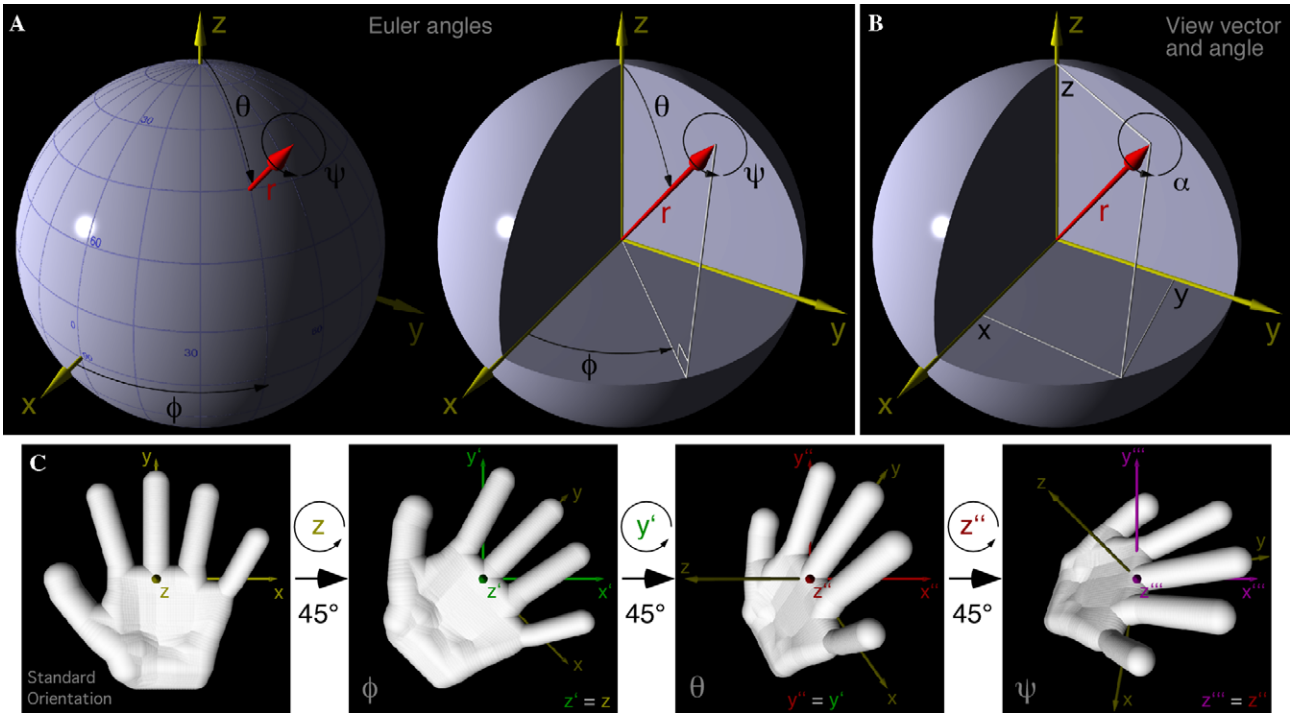

Fig. 1. Definitions of orientation convention about right-handed Cartesian axes **x**, **y**, **z** (yellow) with the origin corresponding to the image origin (origin of the imaged object). (A) *Euler angles*. The position of any vector beginning at the origin can be described by the angles ϕ and θ (in this example, $\phi = \theta = 45°$) The angle ϕ is the angle measured in the anti-clockwise direction between a projection of the vector **r** onto the **x**–**y** plane and the positive **x**-axis. (ϕ is analogous to longitude.) The angle θ is the angle between the vector and the positive **z**-axis. (θ is analogous to latitude.) Rotation about the vector is described by the angle ψ (analogous to compass heading). (B) *View vector and angle*. The vector **r** is the same as the corresponding vector in panel (A) **r** is described by coordinates $\{x, y, z\}$ (labeled in black) in each of the three Cartesian axes **x**, **y**, **z**. The view angle (α) is a rotation about **r**. (C) *Rotation matrices for Euler angles*. A model of a human left hand is used to demonstrate the three angles. As applied in rotation matrices, the convention involves three right-handed rotations about successive orthogonal axes: first, rotation about **z** by ϕ; second, rotation about **y**′ by θ; and third, rotation about **z**″ by ψ. The standard axes {**x**,**y**,**z**} are shown in yellow in each frame. The prime, double-prime, and triple-prime axes are shown in green, red, and magenta, respectively. Note, when the axis of rotation is pointing at the viewer, the coordinate system is rotated anti-clockwise and the object clockwise. The program POV-Ray was used in making this figure (http://www.povray.org).

Table 1
Standard origins and orientations for objects

Symmetry	Notation[a]	Standard origin	Standard orientation
Asymmetric	C1	User-defined	User-defined
Cyclic	C<*n*>	On symmetry axis (*z*-coordinate user-defined)	*z*-axis on symmetry axis
Dihedral	D<*n*>	Intersection of symmetry axes	*z*-axis on principal symmetry axis, 2-fold on *x*-axis
Tetrahedral	T	Intersection of symmetry axes	2-fold axes on *x*, *y*, *z* axes
Octahedral, cubic	O	Intersection of symmetry axes	4-fold axes on *x*, *y*, *z* axes
Icosahedral, dodecahedral	I	Intersection of symmetry axes	2-fold axes on *x*, *y*, *z* axes, front-most 5-fold vertices in *yz* plane[b], front-most 3-fold axes in the *xz* plane[b]
Helical	H<rise>,<angle>, <1/2>, <*n*>	On helical axis, at intersection with dyad axis (if present)	*z*-axis on helical axis, dyad axis on *x* (if present)
Crystal	<sg>	Phase origin of space group	Defined by space group

<sg>, space group notation.
<1/2>, 1 no dyad axis, 2 = dyad axis.
[a] <*n*>: *n*-fold symmetry for cyclic and dihedral point groups, as well as for helices.
[b] With the positive *z*-axis pointing at the viewer and the *x*- and *y*- axes perpendicular to the viewer.

2.2.3.2. Helical symmetries. Helical symmetry is designated by four symmetry operators:

- A translation (rise per subunit).
- A rotation (rotation angle per subunit).
- Presence or absence of a dyad axis perpendicular to the helix axis.
- *n*-fold symmetry about the helix axis.

The following notation is suggested: H<rise>, <angle>, <1/2>, <*n*> (e.g., H24.8, 67.1, 2, 1 indicates a helix with a rise of 24.8 Å and a rotation of 67.1° per unit, a dyad axis, and no rotational symmetry about the helix axis).

2.2.3.3. Crystallographic (space-group) symmetries. Space-group symmetries (for 2D or 3D crystals) are designated by Hermann–Mauguin or Schoenflies notation or space-group number (Hahn, 2002; Rossmann and Arnold, 2001).

2.2.4. Image origin

The first data element (pixel) in a 2D or 3D image has the coordinates {0,0} or {0,0,0}, respectively, in storage or representation space. However, the object within an image has inherent properties defining a reference point for its position in real space, such as a center-of-mass or a symmetry origin. This reference point for the imaged object is defined as the *image origin*. It will be the reference point ({0,0} or {0,0,0} for 2D and 3D data, respectively) of physical data-space—for the imaged object. It may or may not be located at the first data element. The image origin needs to be encoded to, for example, allow the correct translational placement of a 2D image in a 3D reconstruction calculation and allow correct placement of atomic coordinates of macromolecules in a 3D image.

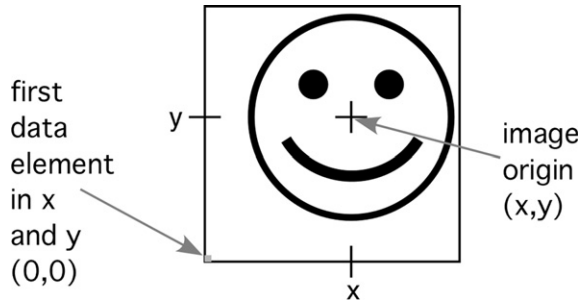

Fig. 2. Example of the standard origin definition in a two-dimensional image. The zeroth (first) data element for *x* and *y* is the first stored pixel in the image {0,0}, shown here in the bottom left corner. The image origin is labeled. It is given in pixel coordinates {*x*, *y*}. The origin for this asymmetric object is arbitrarily defined here as the center of the area outlined by the circle.

The *image origin* is specified as the offset relative to the first element (“0” in each dimension) measured as data-element units (pixels) along each dimension (Fig. 2). The offset can be a non-integer value.

The *standard image origin* is defined by the symmetry of the object (Table 1). In the case of symmetric objects, it is the intersection of symmetry axes or the phase origin for a crystallographic object. There is no common convention for origin position in the case of asymmetric objects. The origin may be, for example, the center-of-mass or the center of a specific volume.

2.2.5. Image orientation

Image orientation is a 3D geometrical parameter required for describing the projection direction for micrographs, the projection direction for particle images in micrographs, and the orientation of 3D maps. The multiple ways of defining orientations in the 3DEM field significantly hampers data interchange.

Despite their common use, Euler angles do not give an unambiguous standard because of multiple definitions. To give an unambiguous standard, *Euler angles* and *image orientation* are defined in terms of "*view*," composed of a *view vector* and a *rotation angle* around the view vector. The view vector is a vector defined within the Cartesian coordinate system. Euler angles—consistent with the most common usage in the 3DEM field—are defined relative to this vector and angle.

The *image orientation* is the view of the imaged object with respect to a standard orientation and origin. Orientation values are given in one of two inter-convertible notations (Fig. 1, Appendix A):

(i) Three *Euler angles*, phi (ϕ), theta (θ), psi (ψ)
Any rotation in 3D Euclidean space may be described as a product of three successive axial rotations. The three angles describing the three successive rotations are called *Euler angles*. Different conventions use different axes for the three rotations. In the 3DEM conventions, any *image orientation* may be given by (ϕ, θ, ψ) where
- the first rotation is an angle ϕ about the z-axis, i.e., in the xy-plane,
- the second rotation is an angle θ about the y'-axis, i.e., the new y-axis and in the $x'z'$-plane,
- the third rotation is an angle ψ about the z''-axis, i.e., the new z-axis and in the $x''y''$-plane.

(ii) *View vector and angle*, a vector **r** and a rotation about that vector (α)
The vector **r** is a unit vector, length $(r) = 1$, described by coordinates $\{x, y, z\}$ in each of the three Cartesian axes **x**, **y**, **z**. It is a directional vector only.[4] The *view angle* (α) is a rotation about **r** (Appendix B).

Further considerations for the definition of *image orientation*:

- All angles are given in degrees.
- The relationship between the *Euler angles* and the *view vector and angle* is

$$x = \cos\phi\ \sin\theta, \quad \phi = \arctan\left(\tfrac{y}{x}\right),$$
$$y = \sin\phi\ \sin\theta, \quad \theta = \arccos(z),$$
$$z = \cos\theta, \quad \psi = \alpha - \phi,$$
$$\alpha = \phi + \psi \quad \text{if } x = y = 0, \text{ then } \phi = \theta = 0^\circ.$$

This definition is consistent with the most common definition for the transformation from a Cartesian to spherical coordinate system.[5] Note that ϕ and ψ cover a full cycle (i.e., -180° to 180° or 0° to 360°), θ ranges from 0° ($+z$-axis) to 180° ($-z$-axis), and ϕ is undefined if $\theta = 0^\circ$ or 180°, but it is set to 0°.

- The *standard image orientation* is defined as the view $= \{x, y, z, \alpha\} = \{0, 0, 1, 0^\circ\}$ and the Euler angles $\phi = \theta = \psi = 0^\circ$. In the case of symmetric objects or enforced symmetry, the *standard image orientation* is also defined by the symmetry of the object (Table 1, Fig. 3).

2.2.6. Positive density for objects

Imaging in the electron microscope gives both positive (negative stain) and negative (frozen-hydrated) density values for objects, justifying the use of positive or negative contrast, respectively, in reconstructions. However, it is more useful to have a consistent sign of contrast. The convention in X-ray crystallography is that objects have positive density, with many programs requiring positive contrast. The 3DEM convention is the same. Imaged objects should be presented with positive densities (contrast).

2.2.7. Definition of defocus

Defocus is the focal setting of the recorded electron image given in length units, and it is used with a positive sign for underfocused images and a negative sign for overfocused images. Defocus is specified as an average value followed by a deviation from the average and an angle measured anticlockwise from the x-axis to specify astigmatism. An astigmatic image would be given as, for example, average = 2.01 μm, deviation = 0.05 μm, angle = 49°. An image with no astigmatism would be, for example, average = 2.01 μm, deviation = 0.0 μm, angle = 0°.

2.2.8. Electron dose

Electron dose is the number of electrons per unit area. It refers to the number of electrons, in the transmission electron microscope, that impinge upon an illuminated area of the specimen.

2.2.9. Units

Each parameter that is quantifiable in terms of SI units must, in addition to its value, contain a factor that designates the value in terms of the standard SI unit. For example, units of length must be accompanied by a term that states the meter units (i.e., μm = 10^{-6} m, nm = 10^{-9} m, Å = 10^{-10} m, and so forth). Several parameters must have the units defined in this way (Table 2).

[4] Although **r** is defined as a unit vector, for user convenience **r** can be given as arbitrary numbers $\{x_0, y_0, z_0\}$ and then normalized: $r_0^2 = x_0^2 + y_0^2 + z_0^2$; $x = x_0/r_0$; $y = y_0/r_0$; $z = z_0/r_0$; $r = 1$. For example, {1,1,1} could be given instead of {0.57735, 0.57735, 0.57735}.

[5] ϕ = *azimuth angle* or *longitude* in spherical coordinates; θ = *zenith angle* or *co-latitude* in spherical coordinates.

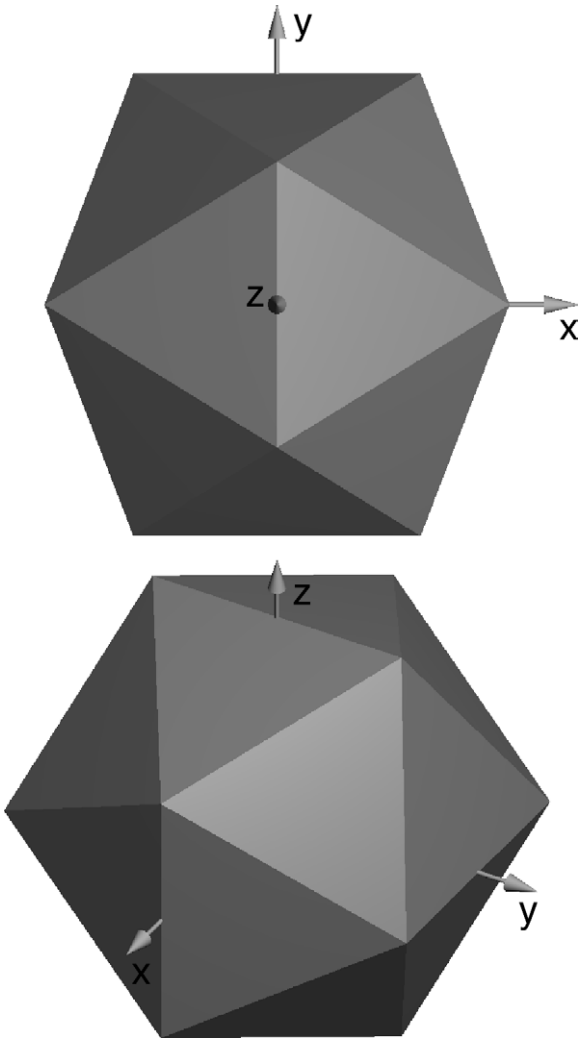

Fig. 3. Standard orientation for icosahedral and dodecahedral particles. Both objects have 60-fold symmetry with 15 twofold axes, 6 fivefold axes, and 10 threefold axes. The standard orientation has three twofold axes coincident with the **x**, **y**, and **z** axes. Once oriented in this manner, these objects can be oriented in one of two ways, 90° different about any Cartesian axis. With the positive **z**-axis pointing at the viewer, the standard convention is for the line between the two closest vertices (fivefold) to be oriented parallel to the **y**-axis, as shown. The program POV-Ray was used in making this figure (http://www.povray.org).

Table 2
Units to be used for various parameters

Parameter	SI unit	Suggested usage unit
Aberration coefficients (spherical, C_s; chromatic, C_c)	Meter	mm
Defocus	Meter	μm
Electron dose (electrons/area)	Meter^{-2}	Å^{-2}
Pixel size/sampling	Meter	Å or nm
Time	Seconds	Seconds
Voltage	Volt	kV

2.2.10. Relative-magnification factor

The *relative-magnification factor* refers to the size of an imaged particle or reconstruction compared to a reference projection or map. For example, if a particle image is 95% the size of a standard particle image (e.g., the average particle size), its relative-magnification factor is 0.95. A particle 1% larger than the average or reference would have a relative-magnification factor of 1.01. To correct for relative magnification, an image must be resized by 1/(relative-magnification factor).

2.2.11. Sign of Fourier transforms

Forward transforms (real-space to reciprocal space) have a negative exponent. Backward transforms have a positive exponent. This convention is necessary to preserve the correct sign of phases and the handedness in the back-transformed images.

3. Recommendations for encoding information associated with images

This section addresses some practical issues involved with programming of an image-processing package. Some of these points could be handled differently in different software packages and could even be invisible to the user. Nevertheless, these points represent important considerations for programmers.

3.1. Statement of conventions

To enable data exchange, interoperability, and user understanding, each software package and database should have its conventions clearly defined in the documentation. These definitions should be publicly and easily accessible.

3.2. Image physical structure (storage layout)

Any image format should unambiguously and explicitly identify the correspondence of data elements as stored in the file structure (e.g., 1D array of integers) to the corresponding data elements in the image (e.g., a 3D image), by providing the following information:

- Number of dimensions.
- Data element type; compound data types should be described in terms of underlying simple types.
- Dimensional mapping (e.g., sequential order of dimensions).

Additionally, to correctly interpret image content an image format should provide:

- Underlying coordinate system.
- Labeling of dimension type and axis order.

- Grid information, including grid type and parameters (e.g., sampling or pixel size in each dimension in the case of regularly sampled grids).
- Image origin position.

The following storage order is common and is encouraged to be adopted (it is consistent with the FORTRAN convention for storing multi-dimensional arrays). Nevertheless, it is recognized that file formats may differ considerably from the outline described below. Such formats could easily be included in a package because the input/output routines could be made to handle such differences.

3.2.1. Data-element type

The fastest varying component is the data-element type (i.e., compound data types, such as RGB images, which store simple data types in adjacent locations).

3.2.2. Dimension 1

The second fastest varying components lie along the first dimension, which is usually the *x*-axis.

3.2.3. Dimension 2

The third fastest varying components lie along the second dimension, which is usually the *y*-axis.

3.2.4. Dimension 3

The fourth fastest varying components lie along the third dimension, which is usually the *z*-axis.

3.2.5. Additional dimensions

If present, additional dimensions with varying purposes (e.g., temporal dimension) would be the fifth, sixth, etc., fastest varying.

3.3. Binary images

File formats that store data elements as binary data must also consider the following issues:

3.3.1. Endianness

An endianness[6] test should be encoded into the software reading an image file. Endianness should not be an issue that users need to deal with. Therefore, an image written in either endianness should be read properly and converted to the native endianness on a computer. Writing files with a specific endianness is discouraged. Every file should be written in the native endianness of the processor used.

[6] Endianness defines the byte-order of multi-byte data types (such as short, int, or float in C) and is determined by the hardware (e.g., PC and Alpha processors are little-endian while SGI and Macintosh processors are big-endian).

Table 3
Data types for images

C type	FORTRAN type	Bits
unsigned char	INTEGER*1 or BYTE	8
short	INTEGER*2	16
int	INTEGER*4	32
long long	INTEGER*8	64
float	REAL*4	32
double	REAL*8 or DOUBLE PRECISION	64
struct{float,float}	COMPLEX*8	64
struct{double,double}	COMPLEX*16	128

Note. the C type "long" and the FORTRAN types "INTEGER" and "REAL" (without size specifiers) are 32- or 64-bit depending on whether the code was compiled as 32- or 64-bit. These types are intended for indices and pointers in programs and should be avoided in specifications of image elements written into image files (to avoid incompatibilities between 32- and 64-bit programs).

3.3.2. IEEE floating-point numbers

The use of commonly adopted standards in the computer world is encouraged. In this line, floating-point numbers should be in IEEE format (IEEE, 1985).

3.3.3. Standard bit sizes

For consistency in stored data, where the programmer may set the size of numbers, the size of real and integer numbers should be consistent (Table 3). Complex numbers are formed from pairs of one of the simpler types (e.g., two int or two REAL*4).

4. Discussion

We propose a set of common conventions for use in the field of 3DEM. These conventions are intended to facilitate data exchange between different software packages in the 3DEM community, as well as providing a homogenous framework for image data archival that will ease data sharing and comparison. Our motivations are: (1) to promote 3DEM data sharing and exchange within the 3DEM community and in the broader structural and molecular biology domains, and (2) to improve software interoperability.

To achieve a wide solution for image data exchange, every package should be able to read and write at least one image format that conforms to the common set of conventions, providing the corresponding conversions to internal conventions and data structures. In addition, the specification of origins and orientations must be consistent to allow for unambiguous interpretation of the image content. This can only be achieved if the conventions are adopted.

We do not recommend that any software package change its internal conventions. Indeed, to do so is completely unnecessary. All we recommend is that each package read and write the standard parameter file format. During reading or writing each package can convert to or from its own internal format.

This set of conventions and recommendations will likely be transparent to most investigators in the 3DEM field, as they experience the interpretation of digital image data through the use of pertinent programs. Software hides the complexity of digital data representation and interpretation (through input/output or read/write routines, as well as internal data structures) and provides means to process and transform that information in a given way to produce the result desired by the user. As long as data are not communicated between different software packages, conventions are usually not an issue.

The standard conventions proposed and the recommendations for encoding image information are therefore most relevant for those developing new methods, maintaining or creating software packages, building integrated frameworks for 3DEM processing, or those involved in the management or archiving of 3DEM data. They may not, therefore, have a direct impact on average users, but provide a basis on which to build ongoing and future practical solutions to promote software interoperability.

Enforcement of the standard conventions in the EM Data Bank, as well as its adoption in new archives and databases, will improve the usefulness of these resources. Unnecessary delays and confusion—particularly for novice users—will result if conventions are not enforced. Submitted data should conform to the 3DEM Image Conventions, and database policies should enforce the standards at the time of submission.

4.1. Management of the 3DEM image conventions

The EM Data Bank seems to us to be the obvious "keeper" of the conventions. As such, the EMDB would post the conventions on their website.

Undoubtedly there will be times when the conventions will need to be amended. We propose that there be a "3DEM Standards Committee" established. This committee would be comprised of representatives from various laboratories in the field and a chair nominated and elected by the group. Proposals to amend the conventions would be submitted to the committee for approval or rejection.

4.2. A roadmap for data exchange

The 3DEM Image Conventions can be formalized in a number of image formats. The ultimate goal in data interchange would be to have a single, comprehensive image format with all of the conventions embedded. The creation and adoption of a standard file format for data exchange and archiving is outside the scope of the present document.

Nevertheless, there is a need to adopt a practical solution. For example, as a matter of pragmatism, the CCP4 format is currently being used in the emerging EM Data Bank. However, as with other image formats currently used in the field, it is limited in scope and unsuitable for many required applications. For example, three limitations of the CCP4 format are: (1) a limited header size with fixed fields, (2) the origin is based on the crystallographic unit cell and requires whole numbers, which is unsuitable for 3DEM, and (3) the official specification is for a single-image 3D format, making it unsuitable for storing multiple images (the common use of packing 2D images into slices of a 3D volume is not stated in the specification, and generates ambiguity in interpreting the data). We envision two alternatives.

The first alternative is to use available image formats to store binary data, while providing interpretive data in a separate text file. It is clear that the image layout specification is absolutely required to make the image legible to programs and must therefore be encoded with the binary data. However, if interpretive information is encoded in a separate parameter file, a user can use the majority of image formats as is (including the popular TIFF format). In several packages in the 3DEM field, text parameter files are used routinely. The interpretive information in these parameter files includes (but is not limited to) the following:

- Image type: micrograph, individual particle, reconstruction, Fourier transform, power spectrum, synthetic image, and so forth.
- Color model.
- Origin.
- Orientation.
- Magnification, sampling, and pixel size.
- Symmetry.
- Unit cell parameters.
- Microscope parameters: microscope identification, voltage, defocus, aberration coefficients, beam divergence angle, amplitude contrast, and so forth.

The two text formats we have found most useful are the Self-Defining Text Archiving and Retrieval (STAR) format (used in crystallography as CIF and in macromolecular biology as mmCIF) and Extensible Markup Language (XML) format (which is becoming the basis for information exchange on the World Wide Web and is used in the EM Data Bank). Both of these formats are tag-based, which allows them to be extensible and backwards compatible (i.e., additions of new tags do not make old parameter files incompatible, although the change of existing tags may). In this way, we propose that the community develop a consistent text-based file format, based either on the STAR or XML formats, or both, incorporating the conventions stated here to establish a useful exchange mechanism for images. All 3DEM software packages and data archives should be able to import and export parameter data via the standard parameter file.

As part of the development of the EM Data Bank, an effort is underway to create a common parameter file for 3DEM. The file developed for the EMDB could also be the medium of exchange between computer software packages.

The second alternative is to adopt a neutral format or produce a new format acceptable to most. This is a unique opportunity to deal with the limitations of the existing formats and come up with a set of requirements for such an image format. The majority of the existing formats are fixed, i.e., not extensible. This means that changing a single field in terms of location, content, or meaning in the file constitutes a new format. However, resistance in the field towards the use of new image formats severely inhibits this approach.

5. Conclusions

The diversity of software in the electron microscopy community is highly desirable to allow the development of new algorithms and ways to process electron micrographs and reconstructions.

However, the lack of common standards makes interchange of information between different packages highly problematic. Here we propose a set of conventions that will impose consistency of data presentation, thus promoting data interchange and automation of complex protocols for micrograph processing.

Believing that the main obstacle to acceptance of standards is political and not technical, we have sought to avoid simply selecting the convention of one prominent 3DEM package. The fact that a package must only read and write the standard parameter file is a change for every package and means that the standard parameters can be changed within each package to match the package's internal convention. This is a change all should be able to accept.

5.1. Adherence to the standard

The majority opinion is that standards are in principle desirable, and there is a common consensus on many of the details. However, to show adherence to the standard, each software package or program should provide the following:

- Conventions internal to the package stated clearly in documentation.
- A mechanism for importing and exporting parameters conforming to the standard conventions (such as database XML or STAR files).

It is our sincere hope that the 3DEM community will adopt the 3DEM Image Conventions and a common parameter file for use in data exchange and EM Data Bank archiving. Of course with any change there are difficulties. However, in the long run, acceptance of common conventions will enhance productivity in the 3DEM laboratories and the usefulness of the EM Data Bank.

Acknowledgments

We thank Alasdair Steven and Jose-Maria Carazo for support and encouragement. We thank the following individuals who have given input and critique: Matthew Asplund, Felicia Betancourt, Bridget Carragher, David DeRosier, Herbert Edelsbrunner, Joachim Frank, Wen Jiang, Salvatore Lanzavecchia, Cathy Lawson, Steven Ludtke, Roberto Marabini, Carlos Pérez-Roca, Ansgar Phillipsen, Carlos Oscar Sánchez-Sorzano, Michael Schatz, Alex Stewart, Benes Trus, Willy Wriggers, and Takuo Yasunaga. Thanks to the participants of the IIMS Workshop (European Bioinformatics Institute, November 2002), 3DEM developers workshop (European Bioinformatics Institute, May 2004), and the EMDB team at the EBI (particularly Kim Henrick, Richard Newman, and Mohamed Tagari). In addition to these people we thank many others who have given encouragement and support to this effort. The EMDB development is funded by the European Commission as the IIMS project (QLRI-CT-2000-3137) under the RTD program "Quality of Life and Management of Living Resources."

Appendix A. Rotation matrices for the defined Euler angles

The three *Euler angles* are defined as three right-handed rotations in the following order (Fig. 1C; Radermacher, 1992; van Heel, 1987):

(i) The first rotation is a rotation about the standard **z**-axis by the angle ϕ. This rotation results in new axes, $\mathbf{x}'$, $\mathbf{y}'$, and $\mathbf{z}'$ ($\mathbf{z}' = \mathbf{z}$). The rotation matrix $R(\phi)$ is given by

$$R(\phi) = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

(ii) The second rotation is a rotation about the $\mathbf{y}'$-axis by the angle θ. This rotation results in new axes, $\mathbf{x}''$, $\mathbf{y}''$, and $\mathbf{z}''$ ($\mathbf{y}'' = \mathbf{y}'$). The rotation matrix $R(\theta)$ is given by

$$R(\theta) = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}.$$

(iii) The third rotation is a rotation about the $\mathbf{z}''$-axis by the angle ψ. This rotation results in new axes, $\mathbf{x}'''$, $\mathbf{y}'''$, and $\mathbf{z}'''$ ($\mathbf{z}''' = \mathbf{z}''$). The rotation matrix $R(\psi)$ is given by

$$R(\psi) = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The three rotations can be combined into one rotation matrix, which is given by

$$\begin{aligned} R(\psi,\theta,\phi) &= R(\psi)R(\theta)R(\phi) \\ &= \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \cos\psi\cos\theta\cos\phi - \sin\psi\sin\phi & \cos\psi\cos\theta\sin\phi + \sin\psi\cos\phi & -\cos\psi\sin\theta \\ -\sin\psi\cos\theta\cos\phi - \cos\psi\sin\phi & -\sin\psi\cos\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta \\ \sin\theta\cos\phi & \sin\theta\sin\phi & \cos\theta \end{pmatrix}. \end{aligned}$$

Note that ϕ, θ, and ψ can be computed directly from the rotation matrix (with matrix elements designated from left to right: r_{11}, r_{12}, r_{13} in the top row; r_{21}, r_{22}, r_{23} in the middle row; and r_{31}, r_{32}, r_{33} in the bottom row), see also Penczek et al. (1994):

$\theta = \arccos(r_{33})$.

For $\theta \neq 0°$, 180°:

$\phi = \arctan 2(r_{32}, r_{31})$,

$\psi = \arctan 2(r_{23}, -r_{13})$.

For $\theta = 0°$, 180°:

$\phi = 0°$,

$\psi = \arctan 2(-r_{21}, r_{11})$.

Here arctan 2 is the inverse tangent function that is based on two variables (i.e., y, x, respectively, in two dimensions) and returns values in a full 360° range.

Appendix B. Relationship of Euler angle ψ and view angle α

The *view angle* (α) is *not* equivalent to the Euler angle ψ (except when $\phi = 0°$), though it may appear so because both α and ψ are rotations about the same orientation vector ($\{x, y, z\}$ or $\{\phi \neq 0°, \theta \neq 0°\}$, see Figs. 1A and B).

The difference comes from the path taken to get to that vector from the standard vector (the positive z-axis or $\{\phi = 0°, \theta = 0°\}$). The *Euler angle* convention takes two rotations (ϕ and θ) about orthogonal axes (z and y', respectively). The *view vector* convention takes only one rotation (θ) about an axis in the x–y plane that has an angle $\beta = \phi + 90°$. After these actions, the angles ψ and α, respectively, are defined as 0°.

However, unless $\phi = 0°$, the two views are not equivalent. They are rotated in-plane with respect to each other. To relate the *view vector* convention to the *Euler angle* convention, the β angle must be taken into account: $\psi = -(\beta - 90°) = -\phi$ and, therefore, $\alpha = \phi + \psi = 0°$. Rotations after this point change the value of α or ψ.

To illustrate the relationship of α and ψ, place your right hand in front of you palm up with your fingers pointing straight ahead. Rotate the hand 90° ($\phi = -90°$) so your fingers point to your left side, then rotate the hand 90° ($\theta = 90°$) so your thumb is vertical and your palm is facing your abdomen. This is the *Euler angle* convention ($\psi = 0°$). Now, put your hand in the original position and rotate the hand 90° ($\theta = 90°$) so your thumb is vertical and your palm is facing your left side. This is the *view vector* convention ($\alpha = 0°$). Note the difference in the in-plane rotation of the hand. The ψ angle for this view is 90°.

Note that orientation vectors along the positive or negative z-axis ($\{\phi = 0°, \theta = 0°\} = \{x = 0, y = 0, z = 1\}$ or $\{\phi = 0°, \theta = 180°\} = \{x = 0, y = 0, z = -1\}$, respectively), are included in the special case where ψ and α are equivalent angles.

References

Carazo, J.M., Stelzer, E.H.K., Engel, A., Fita, I., Henn, C., Machtynger, J., McNeil, P., Shotton, D.M., Chagoyen, M., de Alarcón, P.A., Fritsch, R., Heymann, J.B., Kalko, S., Pittet, J.J., Rodriguez-Tomé, P., Boudier, T., 1999. Organising multi-dimensional biological image information: The BioImage database. Nucleic Acids Res. 27, 280–283.

Carragher, B., Smith, P.R., 1996. Advances in computational image processing for microscopy. J. Struct. Biol. 116, 2–8.

DeRosier, D.J., Klug, A., 1968. Reconstruction of three dimensional structures from electron micrographs. Nature 217, 130–134.

Editorial., 2003. A database for 'em. Nat. Struct. Biol. 10, 313.

Fuller, S.D., 2003. Depositing electron microscopy maps. Structure 11, 11–12.

Hahn, T. (Ed.), 2002. International Tables for Crystallography, Vol. A. Space-Group Symmetry, fifth ed. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Henrick, K., Newman, R., Tagari, M., Chagoyen, M., 2003. EMDep: a web-based system for the deposition and validation of high-resolution electron microscopy macromolecular structural information. J. Struct. Biol. 144, 228–237.

Heymann, J.B., 2001. Bsoft: image and molecular processing in electron microscopy. J. Struct. Biol. 133, 156–169.

IEEE Standard for Binary Floating-Point Arithmetic, 1985. in ANSI/IEEE Std 754-1985 (IEEE, New York).

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

Radermacher, M., 1992. Three-dimensional reconstruction of single particles in electron microscopy. In: Häder, D. (Ed.), Image Analysis in Biology. CRC Press, Boca Raton, pp. 219–249.

Rossmann, M., Arnold, E. (Eds.), 2001. International Tables for Crystallography, Vol. F. Crystallography of Biological Macromolecules. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Tagari, M., Newman, R., Chagoyen, M., Carazo, J.-M., Hendrick, K., 2002. New electron microscopy database and deposition system. Trends Biochem. Sci. 27, 589.

van Heel, M., 1987. Angular reconstitution: a posteriori assignment of projection directions for 3D reconstruction. Ultramicroscopy 21, 111–123.