Supplemental Information

1. ***Angular relations necessary to generate point group symmetry related projection directions for orthoaxial projections.***

For C*n* symmetry, in order to generate necessary all symmetry-related projection directions from an orthoaxial projection direction $(\varphi, \theta = 90^\circ, \psi)$ within the unique range, the transformations are (Fig.S2):

1. for *n* even, unique range $\varphi \in [0, 360/n)$: $\varphi_k = k\dfrac{360}{n} + \varphi$, $k = 0, ..., \dfrac{n}{2} - 1$,

2. for *n* odd, unique range $\varphi \in [0, 360/2n)$: $\varphi_k = k\dfrac{360}{n} + \varphi$, $k = 0, ..., n-1$.

For D*n* symmetry, in order to generate necessary all symmetry-related projection directions from an orthoaxial projection direction $(\varphi, \theta = 90, \psi)$ within the unique range, the transformations are (Fig.S3):

1. for *n* even, unique range $\varphi \in [0, 360/2n)$, we have $k = 0, ..., n-1$, and

1.1. for *k* even, $\varphi_k = \dfrac{k}{2}\dfrac{360}{n} + \varphi$

1.2. for *k* odd, additional change of in-plane rotation,

$$\begin{cases} \varphi_k = \left[\dfrac{k}{2}\right]\dfrac{360}{n} - \varphi + 180 \\ \psi_k = \psi + 180 \end{cases}$$

, where [ ] means integer part of the division,

2. for *n* odd, unique range $\varphi \in [0, 360/4n)$, we have $k = 0, ..., 2n-1$, and

2.1. for $k \bmod 4 = 0$, $\varphi_k = \left[\dfrac{k}{4}\right]\dfrac{360}{n} + \varphi,$

2.2. for $k \bmod 4 = 1$, $\begin{cases} \varphi_k = \left[\dfrac{k}{4}\right]\dfrac{360}{n} + \dfrac{360}{2n} + 180 - \varphi \\ \psi_k = \psi + 180 \end{cases}$,

2.3. for $k \bmod 4 = 2$, $\varphi_k = \left[\dfrac{k}{4}\right]\dfrac{360}{n} + \dfrac{360}{2n} + 180 + \varphi$, 2.4. for $k \bmod 4 = 3$, $\begin{cases} \varphi_k = \left[\frac{k}{4}\right]\frac{360}{n} + 2\frac{360}{2n} - \varphi \\ \psi_k = \psi + 180 \end{cases}$.

As can be seen from the above equations, in order to generate all necessary orthoaxial projections given a subset in unique range, we need to consider three operations on projection images: mirror, rotation by 180°, and combined mirror and rotation by 180°. Based on simple representations of four possible orientations (Fig.S1), we can illustrate the equations defining relations between orthoaxial projections using simple schematics (Figs.S2 and S3).

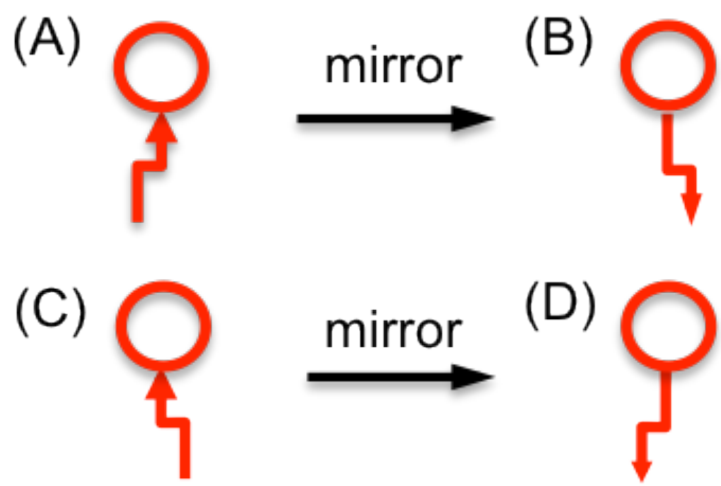

Figure S1: Overview of all four possible orientations of an orthoaxial projection within the unique range: (A) the original projection, (B) mirrored version of the original projection, (C) the 180° in-plane rotated original projection, and (D) mirrored version of the 180° in-plane rotated projection.

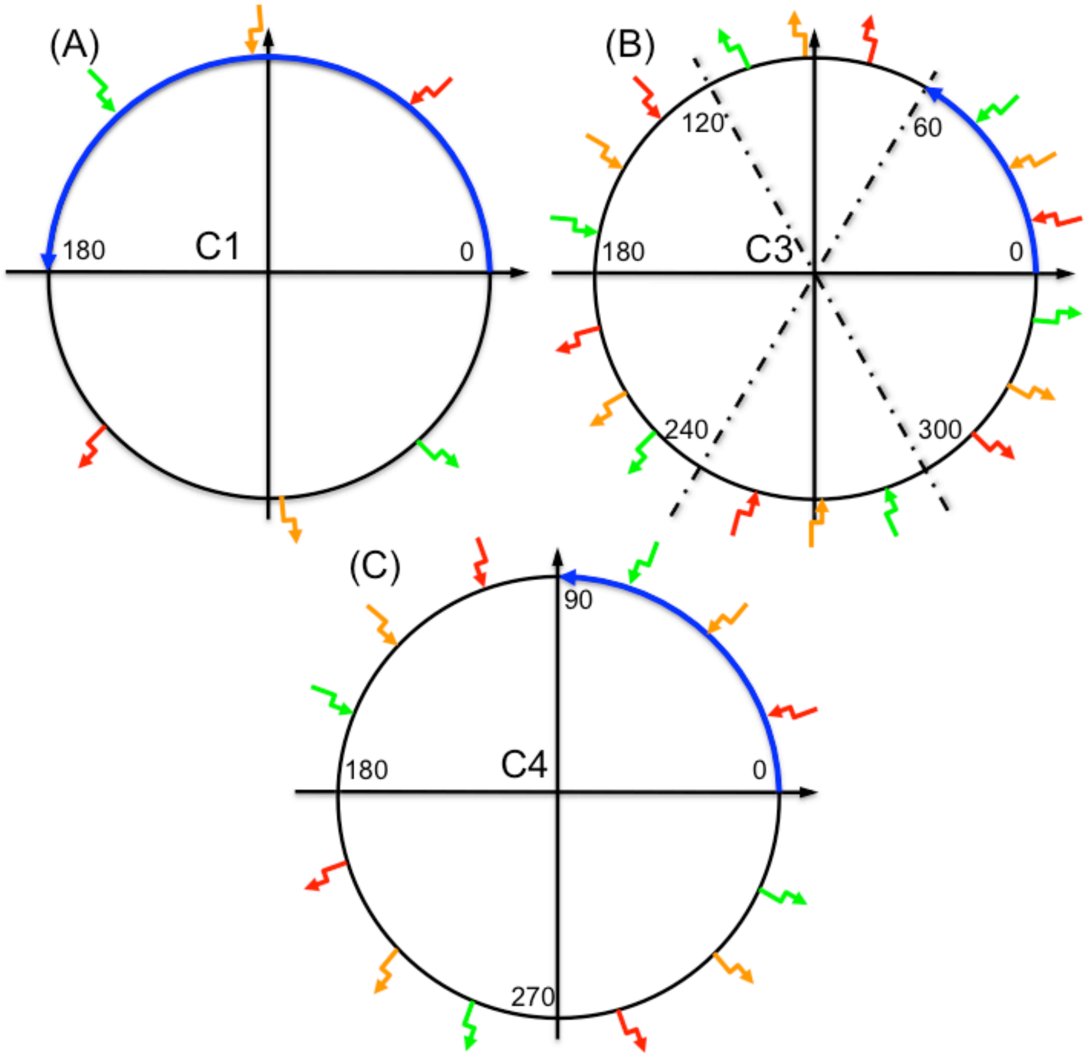

Figure S2: Schematic diagram of relations between orthoaxial projections within the full angular range ($\varphi \in [0,360^\circ)$) for (A) C1 symmetry, (B) C3 symmetry, and (C) C4 symmetry. The blue arc represents the unique angular range. The progression of φ angle within the unique range is illustrated by three different-color arrows. Arrows outside the unique range with same color denote projections related to the same projection of the unique range.

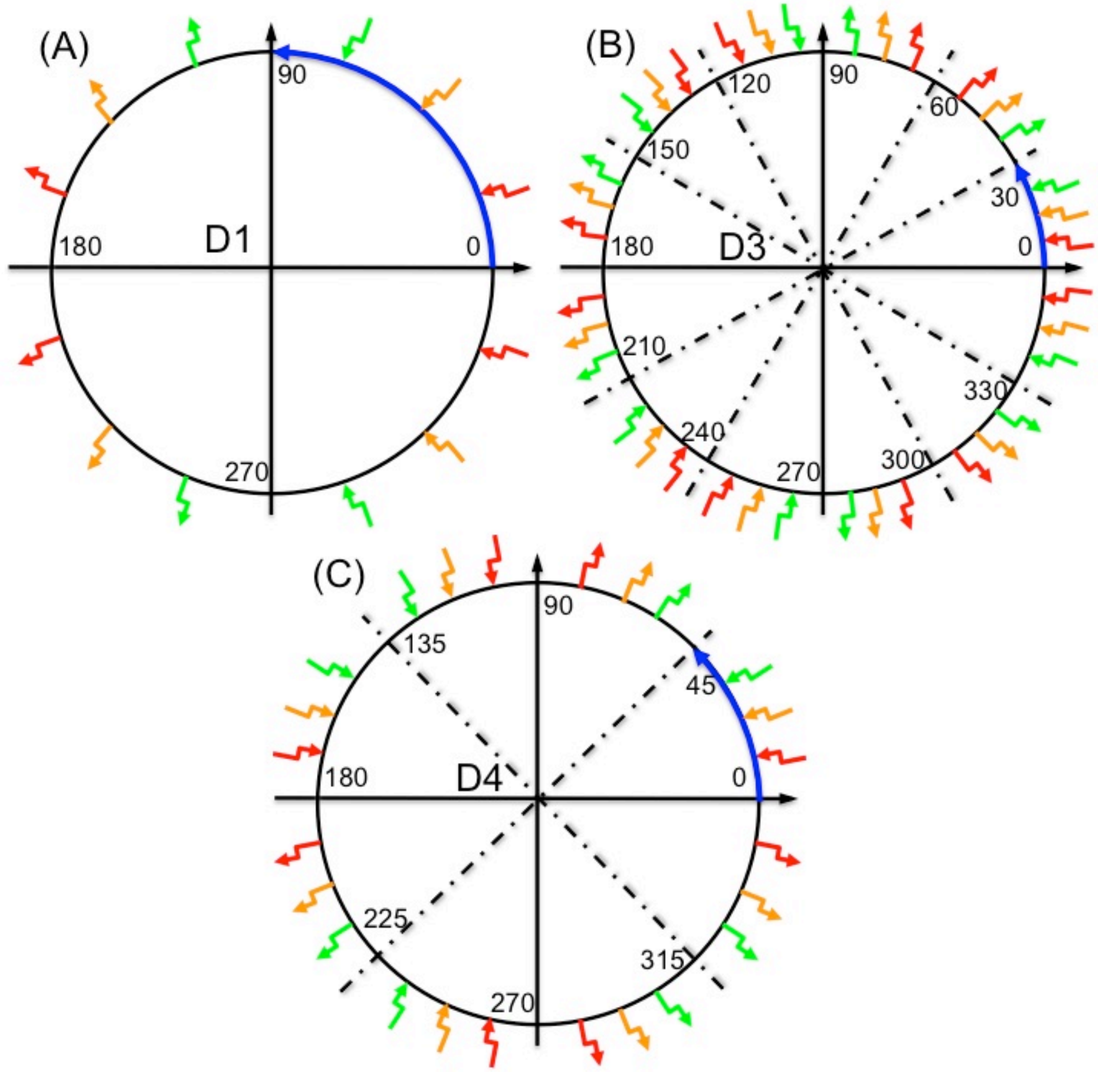

Figure S3: Schematic diagram of relations between orthoaxial projections within the full angular range ($\varphi \in [0,360^{\circ})$) for (A) D1 symmetry, (B) D3 symmetry, and (C) D4 symmetry. The blue arc represents the unique angular range. The progression of φ angle within the unique range is illustrated by three different-color arrows. Arrows outside the unique range with same color denote projections related to the same projection of the unique range.

2. *A Python program that computes pixel adjustment factor given original pixel size, rise, and segment length*:

```
#!/usr/bin/env python

def match_pixel_rise(dz, px, nz, rele = 0.1):
        #  find pixel size closest to the given one (px)
        #    such that rise (dz) is approximately equal to integer number of pixels
        #  Input:
        #       dz - axial rise [Angstrom]
        #       px - pixel size [Angstrom]
        #       nz - z length of the segment [pixel]
        #       rele - relative error of the approximation
        #  Output:
        #       q  - pixel size adjustment factor
        #       error - relative error of resulting approximation

        dnz = nz*px
        # odd number of full disks in the segment
        ndisk = 2*((int(dnz/dz)-1)//2) + 1
        # However, I believe we should only use half of them,
        #    because error increase counting from the center of the volume
        ndisk = (int(dnz/dz)-1)//2

        q = 1.0
        for i in xrange(900000):
                q = 1.0 - 0.000001*i
                error = ((int(ndisk*dz/q/px) - ndisk*dz/q/px)/px)**2
                if( error < rele ):  return q,error
        return -1.0,-1.0

#  execute program
dz = 27.6
px = 1.31
nz = 220

o =  match_pixel_rise(dz, px, nz, 0.001)
print o

#  The resampling is done using function resample of SPARX system:

#vn = resample( vo ,  1.0/o[0] )
```