CBCT Geometric Calibration

By - Shreya Pande
Paul Scherrer Institute

Aim/Object:

To measure the geometry parameters from a sparse projection data set of a stack of beads.

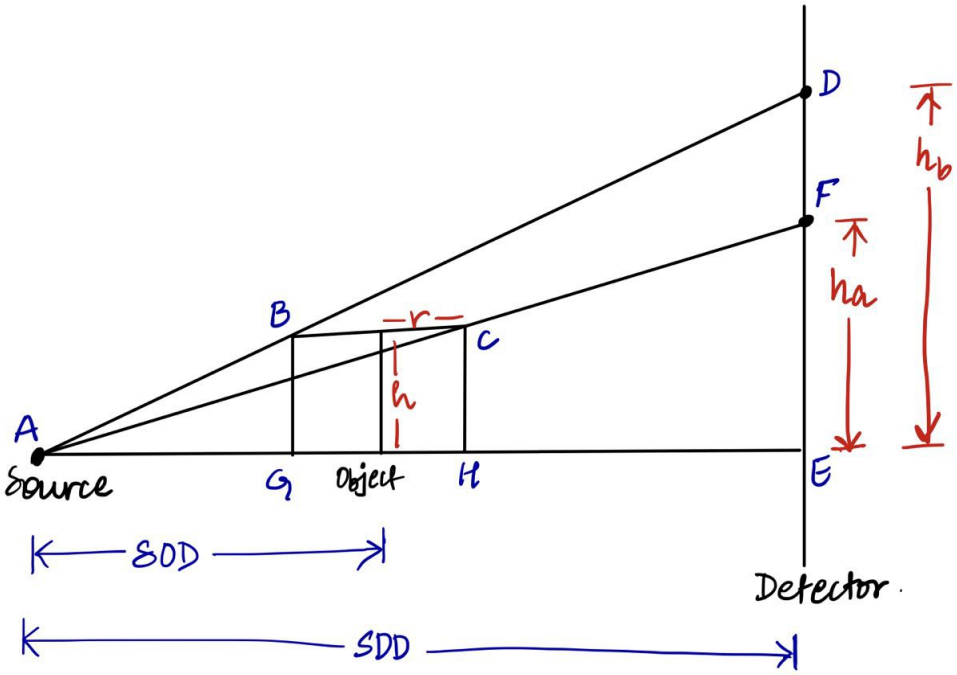
In cone-beam tomography, the projections are distorted by the perspective the divergent beam is producing. This means that geometry is essential to the successful reconstruction of tomography. The center of rotation and the pixel size (this is only needed to scale the attenuation coefficients) which is the only information required for parallel beam reconstruction. For cone-beam reconstruction, you further need to know the distances from the source to the sample (object), the distance from the source to the detector (SDD). These parameters define the magnification of the configuration and would in principle be sufficient for the reconstruction. The beam usually doesn't hit the detector perpendicularly, therefore we also need to know the position of the piercing point on the detector px and py.

Steps Deployed in Geometric Calibration:

1. Import the necessary libraries and the images captured by the X-ray tomography. This includes the dark current images, open beam images and the main projection images.
2. The first step is to normalize the projections of the images using the dark current and the open beam images. Image texture is one of the very important factors for a good reconstruction. This is often distorted by the noise during scanning and other artifacts that may be present. Thus, we need to normalize the images. This is followed by flipping the image for an upright orientation.
3. The next step is to crop the image by converting the pixels of all unwanted elements into 0 such that they are not taken into consideration into the steps ahead for reconstruction. (This step is optional i.e depending on the magnification used, the extent of artifacts in the projection images varies depending on magnification used.)
4. This is followed by baseline removal such that only part left are the beads from the projection. This is done by setting an appropriate threshold to separate the rest of the image components from the beads and hence removing them by filtering them out.
5. This is followed by erosion of the leftover beads to remove any excess noise close to the beads and then performing watershed fragmentation on them.
6. The next step is to identify the beads from the image of the segmented beads in order to track their trajectories across the many images taken.
   Method for bead detection deployed:
   - The centers of the watershed segmented beads are identified using skimage regionprops.

    - The centres obtained in the regionprops table are then converted to a 2D coordinate system form for easier understanding.
7. This is followed by writing a function buildbeads that tracks the path of a specific bead across all the projects and stores the centres in an array for further processing.
8. Once the trajectory of the beads has been identified, the aim is to find ellipses that best fit the paths. In some cases, the bead trajectories obtained are not optimum and hence, need to be dropped from this process since the ellipse identified corresponding to them would not be an accurate representation of the bead patten and hence may create errors in the calculation of tomography parameters ahead. The reason for these distorted bead trajectory can be attributed to the interference from other beads or the noise in the area of the beads.
9. The ellipses' parameters i.e center coordinate, the lengths of the major and minor axis, and the tilt in the ellipse orientation are calculated by using the EllipseModel() package by python to which the input parameter is the array containing the centers of specific beads across all the projects. Then, a check on the center points is performed wherein points at a distance of 10 units more than the larger of the two axes are identified as outliers and hence removed from the beads array and hence, the ellipse parameters are calculated again.
10. The center of rotation and tilt is calculated by using the best fit line for the center of all the trajectories of the beads through multiple projections. A polynomial of degree 1 is approximated to fit the trend of the centers. The 2 main parameters obtained from the calculation give the tilt and the center of rotation axis coordinate.
11. The magnification is calculated by using one of the projection images. The mean dsitance between the centres of the beads is calculated and then multiplied by the pixel pitch i.e. 0.139mm to obtain the magnification.
12. Piercing Point is the minima of all the elliptical trajectories. The minor axis for the ellipse at the piercing point is 0. Thus, the strategy applied is to plot a graph of minor axis length versus the height of the ellipse i.e the y-coordinate of the centre of the ellipse. The best linear fit line on being extrapolated gives the y-coordinate of the piercing point. Further using the best linear fit line for the centres of the ellipses, we can obtain the x-coordinate of the piercing point.

13. This is done by using two similar triangle geometries to calculate the various heights of the beads and hence the SOD and SDD.



4 triangles that can be identified from the above are: ΔABG, ΔADE, ΔACH, ΔAFE. The similar triangles' pairs are ΔABG and ΔADE, ΔACH and ΔAFE. By using similarity, we have the following equations:

By using similarity, we have the following equations:

$$\frac{AG}{AE} = \frac{BG}{DE}$$

and

$$\frac{AH}{AE} = \frac{CH}{FE}$$

$$\frac{SOD - r}{SDD} = \frac{h}{h_b}$$

and

$$\frac{SOD + r}{SDD} = \frac{h}{h_a}$$

After extracting value of h, we equate the two equations as follows:

$$h_b(SOD - r) = h_a(SOD + r)$$

$$\frac{(SOD - r)}{(SOD + r)} = \frac{h_a}{h_b}$$

$$SOD \cdot h_b - r \cdot h_b = SOD \cdot h_a + r \cdot h_a$$

$$SOD \cdot (h_b - h_a) = r \cdot (h_b + h_a)$$

$$SOD = \frac{(h_b + h_a) \cdot r}{(h_b - h_a)}$$

Using magnification we find earlier we calculate the SDD using the following relation between SDD, SOD and $magnification(mag)$:

$$mag = \frac{SDD}{SOD}$$

$$SDD = mag \cdot SOD$$