

Python implementation of distortion correction methods for X-ray micro-tomography

Nghia T. Vo

*Diamond Light Source, Harwell Science and Innovation Campus, Didcot, Oxfordshire,
OX11 0DE, UK
nghia.vo@diamond.ac.uk*

Abstract

The python package (<https://github.com/DiamondLightSource/vounwarp>) is the implementation of the distortion correction methods published in Optics Express [N. T. Vo *et al.*, "Radial lens distortion correction with sub-pixel accuracy for X-ray micro-tomography," 23, 32859-32868 (2015)]. It is useful for calibrating an imaging system which is highly configurable and requires frequent disassembly for maintenance or replacement of parts. The techniques require a single image of a calibration target aligned perpendicular to the optic axis and thus imaged with negligible perspective distortion. The main application of the methods is for synchrotron-based parallel-beam X-ray tomography where specially designed optics is being used for imaging X-rays after conversion to visible light. Here, distortion as small as 1 pixel introduces artifacts in the reconstructed images.

1. Introduction

The methods are used to calculate parameters of a polynomial model of radial lens distortion [1], which are the center of distortion (CoD) and the polynomial coefficients, for correcting the distortion of an imaging system. They are classified as direct techniques because no iterative adjustment of distortion coefficients is necessary. The CoD is calculated independently from the polynomial coefficients. This is useful for bespoke designed systems where the routine mechanical alterations do not alter the lens characteristics but may change the center of the optical axis with respect to the detector. The achievable sub-pixel accuracy relies on the quality of a calibration target which can provide straight and equidistant lines vertically and horizontally. The calibration methods extract these lines, represent them by the coefficients of parabolic fits, and use these coefficients for calculating distortion coefficients. This requires the number of patterns (dots or lines) should be large enough to allow fitting with sub-pixel accuracy. The current package is used for processing a dot pattern target. Pre-processing modules for handling different types of target will be added. The package uses common Python libraries of image processing distributed by Anaconda [2]. In addition to docstrings for every function in the package, this report provides a detailed analysis and explanation of methods which is useful for users or developers.

Figure 0 shows the structure of the python package which has four modules: the LoaderSaver module is used to load an image and save outputs such as an image, a metadata file, or a plot image; the Pre-processing module is used to segment a grid pattern and group points into lines; the

Processing module is for calculating distortion parameters; and the Post-processing module is used to un warp distorted lines, images, or slices and to evaluate the accuracy of the correction results.

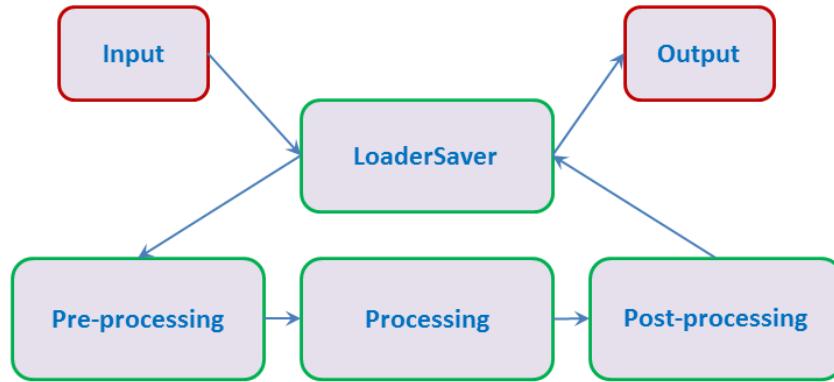


Figure 0. Structure of the package.

2. Analysis procedure

2.1 Basis routine

- Acquire an image of a dot pattern (Fig. 1).

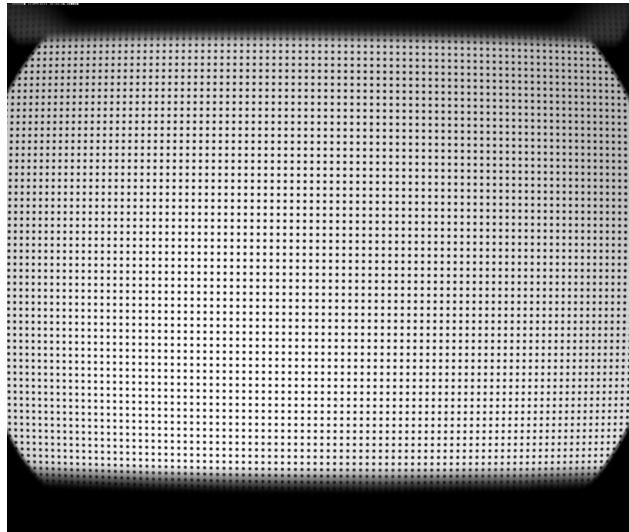


Figure 1. Image of a dot pattern (Visible light target)

- Binarize the image (Fig. 2), calculate the center of mass of each dot, group dots into horizontal lines (Fig. 3) and vertical lines (Fig. 4).

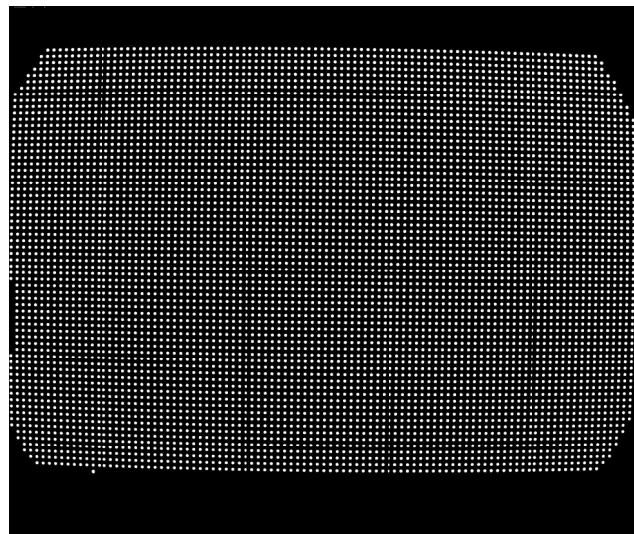


Figure 2. Binary image of the dot pattern.

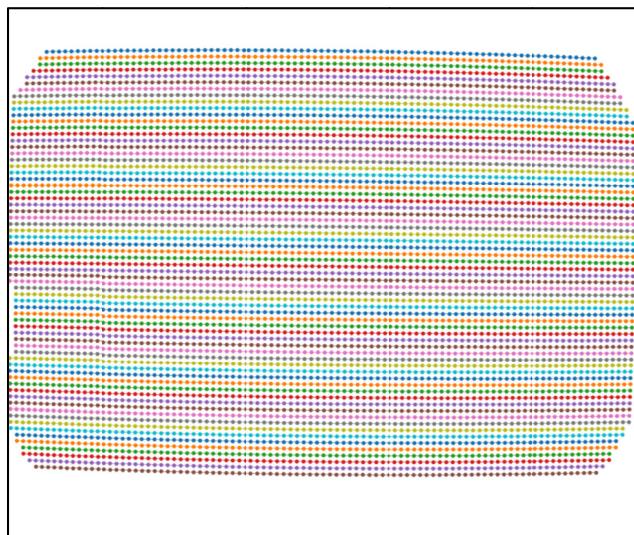


Figure 3. Dots are grouped into horizontal lines (dots on the same line having the same color).

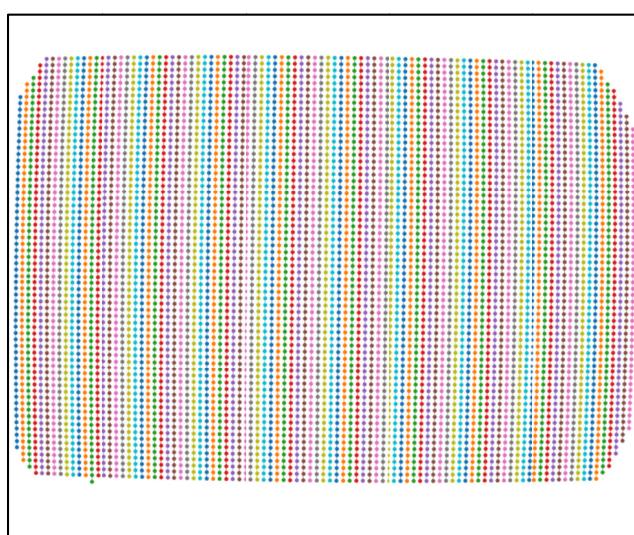


Figure 4. Dots are grouped into vertical lines.

- Calculate the center of distortion, the polynomial coefficients (Tab. 1) of the backward model [1]; apply correction using linear interpolation (Fig. 5); evaluate the residual errors using the distances of corrected dot-centroids to their fitted straight line (Fig. 6).

Table 1: Parameters of the correction model

CoDx (from the left of the image)	1252.18562214
CoDy (from the top of the image)	1008.91664101
A0	1.00035826515
A1	-1.62430961358e-06
A2	-1.35654468765e-08
A3	-2.25582210528e-12
A4	9.58434720477e-16

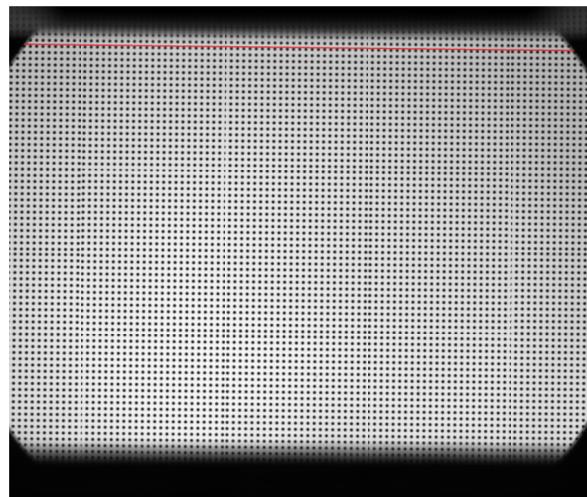


Figure 5. Corrected image.

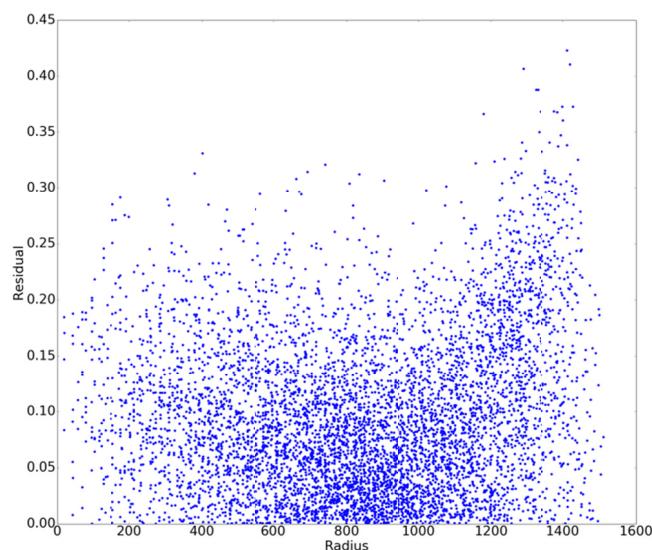


Figure 6. Plot of the distances of the corrected dot-centroids from their fitted straight line against their distances from the CoD.

2.2 Advanced analysis

2.2.1 Pre-processing

2.2.1.1 Binarizing

The basic routine works well if one acquires a nice and clean image of a dot pattern. However, extra pre-processing methods may be needed in the following cases:

- Background is non-uniform (Fig. 7a). To binarize an image using a global thresholding method [3], we need to normalize the image with the background extracted from itself (Fig. 7b). The package provides two ways of generating the background: applying a strong low-pass filter on the image or applying a median filter with a large window size.

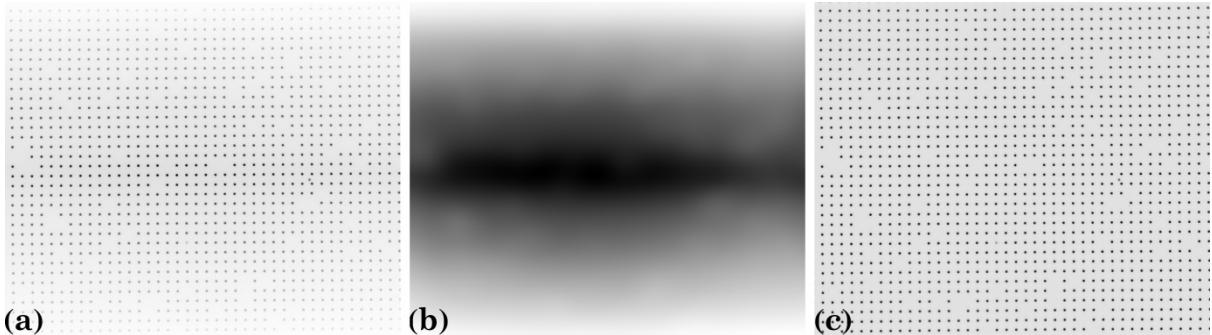


Figure 7. Non-uniform background correction. (a) Image (X-ray target); (b) Extracted background; (c) Corrected image.

- Image is contaminated (Fig. 8). The package provides two methods of removing non-dot objects after the binarization step. In the first approach, the median size of the dots (MS) is determined, then only objects with sizes in the range of $(MS-R*MS; MS+R*MS)$ are kept where R (ratio) is a parameter. In the second approach, the ratio between the largest axis and the smallest axis of the best-fit ellipse is used [4]. Objects with ratios out of the range $(1.0; 1.0+R)$ are removed.

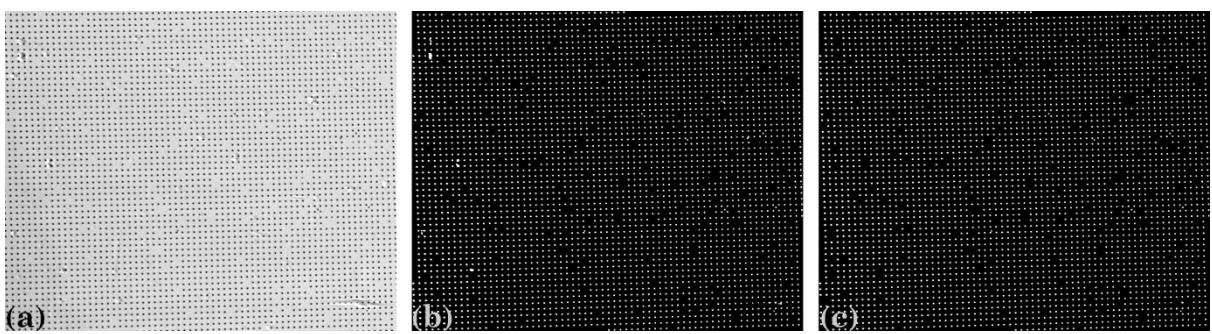


Figure 8. Processing a contaminated image.

(a) Image (X-ray target); (b) Binary image; (c) Image with non-dot objects removed.

- There are misplaced dots in the image. X-ray dot patterns made in-house may have dots placed in wrong positions as shown in Fig. 9. The misplaced dot is identified by using its distances to four nearest dots. If none of the distances is in the range of $(MD-R*MD; MD+R*MD)$, where MD is the median distance of two nearest dots, the dot is removed. This method, however, should not be used for image with strong distortion where the distance of two nearest dots changes significantly against

their distances from the CoD (Fig. 1). A more generic approach to tackle the problem is presented in the next section.

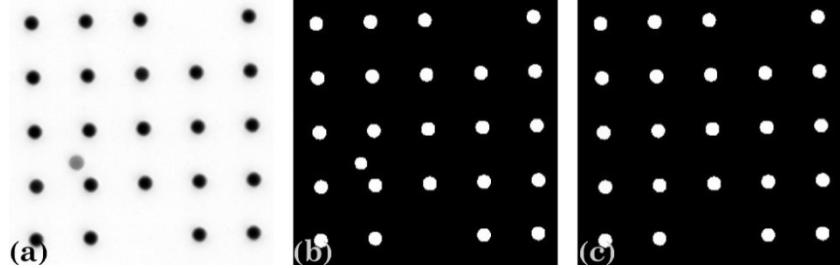


Figure 9. (a) Image with a misplaced dot; (b) Binary image; (c) Misplaced dot removed.

In the previous methods, the median size of the dot and the median distance of two nearest dots are used to select dots. These values are determined by: selecting the middle part of the image where there is least distortion; binarizing this ROI; calculating the size of each dot; calculating the distance between a dot and its nearest dot for all dots; using the median value of the sizes and the distances. The median value is used instead of the mean value to avoid the influence of the non-dot objects, misplaced dots, or missing dots.

2.2.1.2 Grouping

In this step, the center of mass (CoM) of each dot is calculated and used to group dots into horizontal lines and vertical lines. Grouping methods search the neighbours of a dot to decide that they belong to the same group or not. The search window is defined by the distance of two nearest dots, the slope of the grid, the parameter R , and the acceptable number of missing dots. The slope of the grid is coarsely estimated using the Radon transform [5]. Next, dot-centroids around a line which goes through the centroid of the middle dot of the ROI and has the same slope as the previous result are used for the linear fit. The slope of the fitted line is the accurate estimation of the grid slope. Note that the R value ($0.0 \rightarrow 1.0$) or the acceptable number of missing dots may need to be adjusted to avoid the problem of lines broken as shown in Fig. 10.

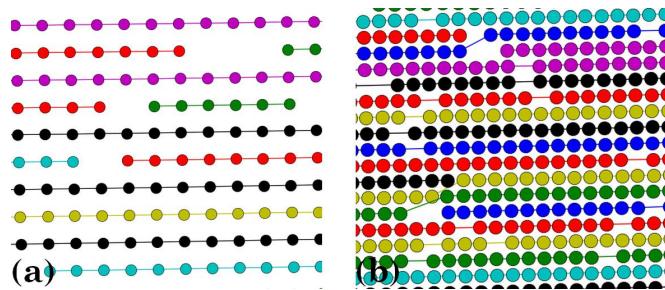


Figure 10. Incorrect grouping demonstration: (a) If the acceptable number of missing dots is small; (b) If R is large.

As mentioned in the previous section, there is a different way of removing misplaced dots after they are grouped. Dot-centroids in the same group are fitted with a parabolic curve. Any dot-centroid with the distance larger than a certain value (in pixel unit) from its fitted position is removed (Fig. 11).

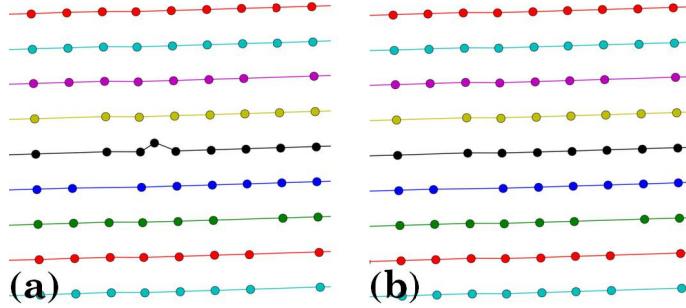


Figure 11. Misplaced dot removal: (a) Before; (b) After.

2.2.2 Processing

It is important to be aware that different methods use different coordinate systems. In the image coordinate system, the origin is at the top-left corner of the image. For example, the CoM of a dot refers to this origin [6]. In the plotting coordinate system, the origin is at the bottom-left. The parabolic fits of horizontal lines refers to the bottom-left origin, however the one of vertical lines refers to the top-left origin with coordinates swapped. This is necessary to avoid the numerical problem of fitting lines nearly perpendicular to the axis.

2.2.2.1 Calculating the center of distortion

Points on each group are fitted to parabolas in which the horizontal lines are represented by

$$y = a_i x^2 + b_i x + c_i, \quad (1)$$

and vertical lines by

$$x = a_j y^2 + b_j y + c_j \quad (2)$$

where i, j are the index of the horizontal lines and vertical lines respectively. The rough estimate of the CoD is explained in the Fig. 12 where (x_0, y_0) is the average of the axis intercepts c of two parabolas between which the coefficient a changes sign. The slopes of the red and green line are the average of the b coefficients of these parabolas.

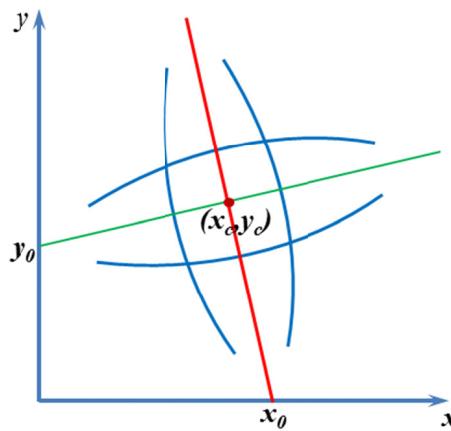


Figure 12. Coarse CoD is the intersection of the red and the green line.

For accurately determining the CoD coordinates, they are varied inside the bounds of the above parabolas and metrics are calculated as the following steps: The point having minimum distance to the current CoD is located for each parabola. The horizontal and vertical parabolas yield two sets of points. Each set of points is fitted to a straight line. The sum of the intercepts of two fitted straight lines (Fig. 13) is the metric of the estimated CoD. The best CoD is the one having the minimum metric.

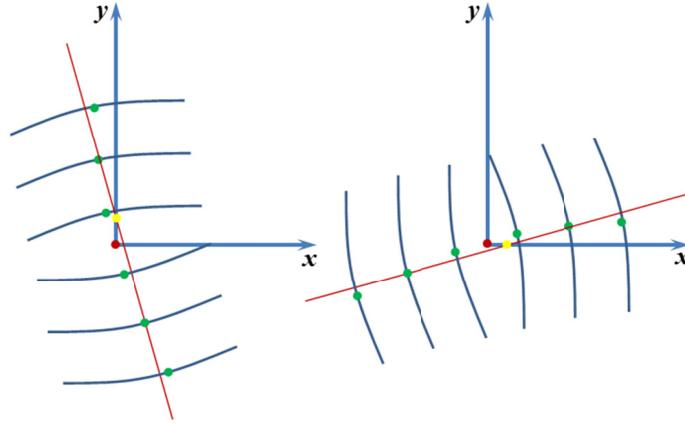


Figure 13. Accurate determination of the CoD.

However, this routine of finding accurate CoD is very sensitive to the perspective distortion (which may be useful for applications needing to detect this type of distortion). Figure 14 shows the metric maps of two cases: one without perspective distortion and one with small perspective distortion which show the global minimum inside the box and outside the box, respectively. In practice, it is found that the coarse CoD is accurate enough for both cases.

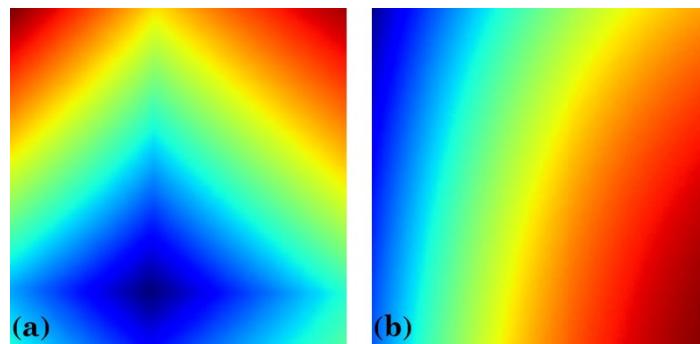


Figure 14. Metric map of the CoD search: (a) without perspective distortion; (b) with perspective distortion.

2.2.2.2 Calculating undistorted intercepts

Undistorted intercepts, c_i^u and c_j^u , of undistorted lines used for calculating distortion coefficients can be determined without distortion correction. Using the assumption that the undistorted, uniform line spacing can be extrapolated from the area near the CoD having negligible distortion. The undistorted intercepts are constructed by extrapolating from a few lines around the CoD as

$$c_i^u = \text{sgn}(c_i) \times |(i - i_0) \Delta c| + c_{i_0} \quad (3)$$

where the $\text{sgn}()$ function returns the value of -1, 0, or 1 corresponding to its input of negative, zero, or positive value. $i0$ is the index of the line closest to the CoD. $\overline{\Delta c}$ is the average of the difference of c_i near the CoD. $\overline{\Delta c}$ can be refined further by varying it around an initial guess and find the minimum of $\sum_i (c_i - c_i^u)^2$ which also is provided in the package. The same routine is used for the vertical lines as

$$c_j^u = \text{sgn}(c_j) \times |(j - j_0)\overline{\Delta c}| + c_{j_0} \quad (4)$$

2.2.2.3 Calculating distortion coefficients of the forward model

In the forward (FW) model, the relationship between an undistorted dot-centroid (x_u, y_u) and a (distorted) dot-centroid (x, y) is described as

$$\frac{r_u}{r} = \frac{x_u}{x} = \frac{y_u}{y} = k_0 + k_1 r + k_2 r^2 + k_3 r^3 + \dots + k_n r^n \equiv F(r) \quad (5)$$

where r_u and r are their distances from the CoD in the undistorted image and in the distorted image, respectively; k_n are the coefficients. Knowing a_i , c_i , c_i^u , x , y , and r , the distortion coefficients are calculated by solving a system of linear equations as the following; one for each dot on both horizontal and vertical lines

$$\begin{bmatrix} \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & r & r^2 & \dots & r^n \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & r & r^2 & \dots & r^n \\ \vdots & \vdots & \vdots & \dots & \vdots \end{bmatrix} \begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ \vdots \\ k_n \end{bmatrix} = \begin{bmatrix} \vdots \\ c_i^u / (a_i x^2 + c_i) \\ \vdots \\ c_j^u / (a_j y^2 + c_j) \\ \vdots \end{bmatrix} \quad (6)$$

The FW model is useful to evaluate the accuracy of the polynomial model of the radial lens distortion, or to evaluate if the distortion of an imaging system is significant (*i.e* residual larger than 1 pixel) for the need of correction.

2.2.2.4 Calculating distortion coefficients of the backward-from-forward model

In the backward (BW) model, the relationship between an undistorted point (x_u, y_u) and a distorted point (x, y) is inverted as

$$\frac{r}{r_u} = \frac{x}{x_u} = \frac{y}{y_u} = k_0^b + k_1^b r_u + k_2^b r_u^2 + k_3^b r_u^3 + \dots + k_n^b r_u^n. \quad (7)$$

To calculate k_n^b one needs to know r_u which can be determined from r using the FW model (Eq. (5)).

The distortion coefficients are given by solving

$$\begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ 1 & rF(r) & \dots & r^n F(r)^n \\ \vdots & \vdots & \dots & \vdots \end{bmatrix} \begin{bmatrix} k_0^b \\ k_1^b \\ \vdots \\ k_n^b \end{bmatrix} = \begin{bmatrix} \vdots \\ 1/F(r) \\ \vdots \end{bmatrix} \quad (8)$$

2.2.2.5 Calculating distortion coefficients of the backward model

The coefficients of the BW model can be determined directly without using the results of the FW by solving the following equation

$$\begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ 1 & r/F_i & \dots & r^n/F_i^n \\ \vdots & \vdots & \dots & \vdots \\ 1 & r/F_j & \dots & r^n/F_j^n \\ \vdots & \vdots & \dots & \vdots \end{bmatrix} \begin{bmatrix} k_0^b \\ k_1^b \\ \vdots \\ k_n^b \end{bmatrix} = \begin{bmatrix} \vdots \\ F_i \\ \vdots \\ F_j \\ \vdots \end{bmatrix} \quad (9)$$

where $F_i = \frac{a_i x^2 + c_i}{c_i^u}$ and $F_j = \frac{a_j y^2 + c_j}{c_j^u}$. In practice, using distortion coefficients up to the fifth order is accurate enough, as there is no significant gain in accuracy with higher order.

2.2.3 Post-processing

In this step, distortion correction is applied and evaluated using the following methods

2.2.3.1 Unwarping a line of points using the FW model.

The coordinates of undistorted points of a line is determined straightforward using the FW model (Eq. (5)). The accuracy of the correction model can then be evaluated by checking the straightness of the corrected lines. The results of unwarping lines in Fig. 3 and Fig. 4 are shown in Fig 15.

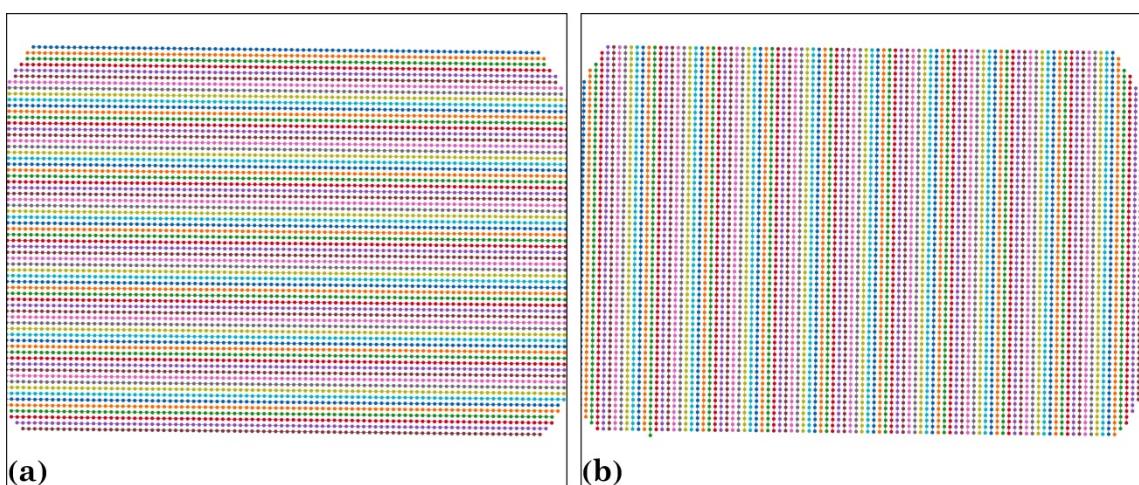


Figure 15. Unwarped lines of dots corresponding to: (a) Fig. 3; (b) Fig. 4.

2.2.3.2 Unwarping a line of points using the BW model.

It is impossible to calculate x_u, y_u directly from Eq. (7) with the polynomial order higher than 2. Here, a workaround using numerical optimization techniques is proposed. Given r and distortion coefficients, r_u is determined by minimizing the following function

$$\left(r - r_u \left(k_0^b + k_1^b r_u + k_2^b r_u^2 + k_3^b r_u^3 + \dots k_n^b r_u^n \right) \right)^2 \quad (10)$$

then x_u and y_u are given by

$$x_u = \frac{r_u}{r} x ; \quad y_u = \frac{r_u}{r} y \quad (11)$$

2.2.3.3 Evaluating the straightness of a line of points

The straightness of a line of points is evaluated by using the distances of the points from their fitted straight line. This method is important to evaluate the accuracy of the correction model, or to check that the distortion is significant for the need of correction. Figure 16 shows the plot of the distances of dot-centroids from their fitted straight lines against their distances from the origin before and after correction (for all vertical lines in Fig. 4 and Fig. 15b, respectively). As can be seen, the residual up to 9-pixel is reduced to sub-pixel (0.35) after the lines are unwarped. Note that the origin of the coordinate system before the correction is at the top-left corner of the image.

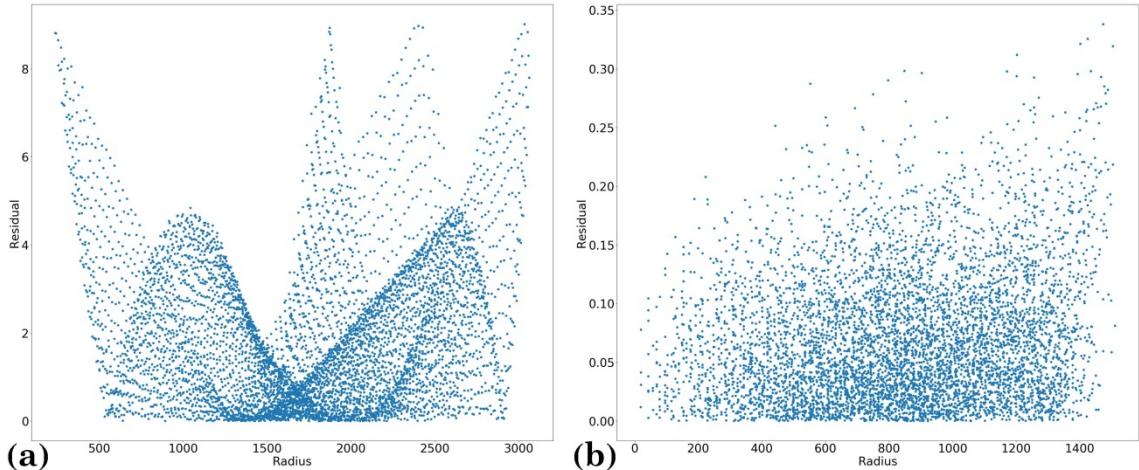


Figure 16. Residual plot: (a) Before correction; (b) After correction.

2.2.3.4 Unwarping an image using the FW model

Image is treated as a grid of points. Forward mapping of a point from a distorted image may not fall on a grid point of an undistorted image as shown in Fig. 17. To find the values on the grid points of the undistorted image, we need to interpolate from the values at the closest points. Find the closest points to a given grid point is computationally costly. A simple correction using nearest interpolation results the problem of vacant pixels (Fig. 18a) or overlapping pixels (Fig. 18b) depending on the distortion of the barrel type or the pincushion type.

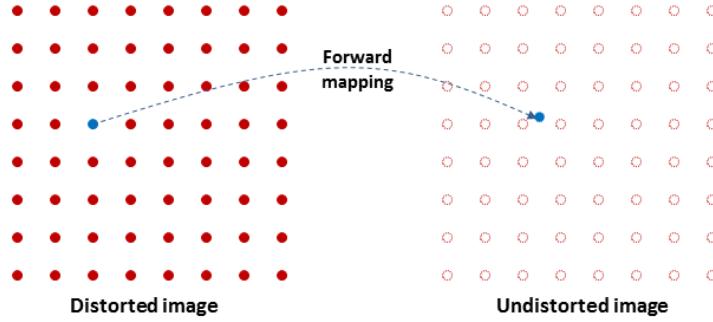


Figure 17. Demonstration of the forward mapping.

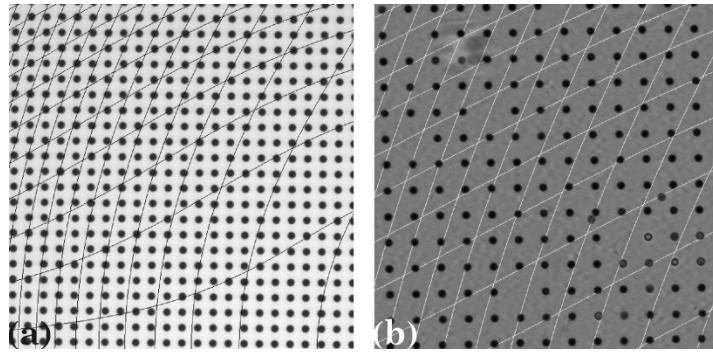


Figure 18. (a) Vacant pixels problem; (b) Overlapping pixels problem.

2.2.3.5 Unwarping an image using the BW model

Backwards mapping a grid point of the undistorted image into the coordinate system of the distorted image maintains the known values on the grid (Fig. 19). This makes it simple to find the closest points for interpolation (Fig. 5).

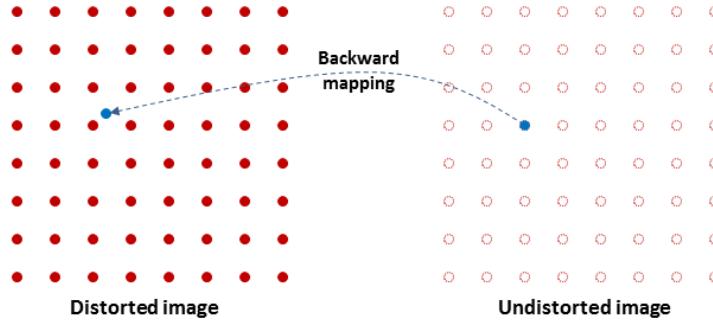


Figure 19: Demonstration of the backward mapping.

Given (x_u, y_u) of a grid point in the undistorted image, the CoD (x_{CoD}, y_{CoD}) and the distortion coefficients, k_n^b , of the BW model, the calculation routine is as follows:

- Translate the coordinates $x_u = x_u - x_{CoD}$; $y_u = y_u - y_{CoD}$.
- Calculate $r_u = \sqrt{x_u^2 + y_u^2}$.
- Calculate $r = r_u (k_0^b + k_1^b r_u + k_2^b r_u^2 + k_3^b r_u^3 + \dots k_n^b r_u^n)$.

- Calculate $x = x_u \frac{r}{r_u}$; $y = y_u \frac{r}{r_u}$
- Translate the coordinates $x = x + x_{CoD}$; $y = y + y_{CoD}$.
- Find 4 nearest grid points of the distorted image by combining two sets of $\lfloor \text{floor}(x); \text{ceil}(x) \rfloor$ and $\lfloor \text{floor}(y); \text{ceil}(y) \rfloor$. Clip values out of the range of the grid.
- Interpolate the image value at (x, y) using values of 4 nearest points. Assign the result to the point (x_u, y_u) in the undistorted image.

2.2.3.6 Unwarping slices of a 3D dataset

In parallel-beam tomography, a set of 2D distorted projection (image) need to be corrected and resliced to a set of 2D sinograms [7] before they can be reconstructed independently. In practice, we may want to extract just few corrected sinograms to evaluate the quality of the correction using reconstructed images. The package provides a method of extracting undistorted slices directly as demonstrated in Fig. 20.

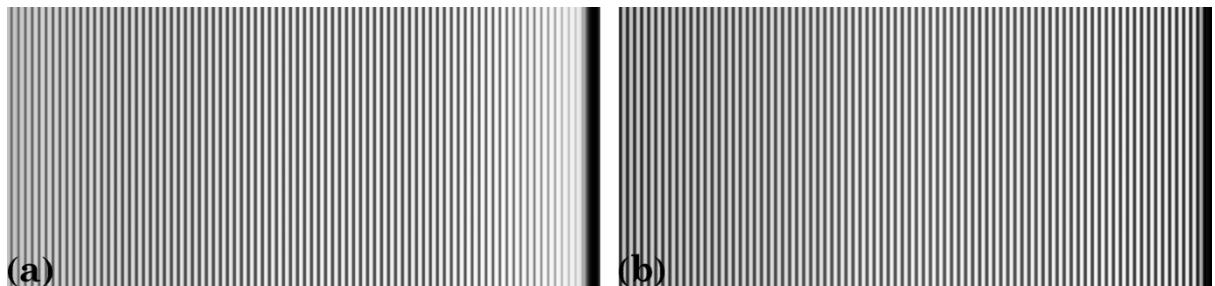


Figure 20. Slice of a 3D data made of identical dot pattern images: (a) Before correction; (b) After correction.

3. Application and further development

Demonstration of using the distortion correction methods for a tomographic data collected at the beamline I12-JEEP [8], Diamond Light Source (DLS), UK are as follows;

- A dot pattern image (visible light target) was acquired (Fig. 21). Distortion coefficients are calculated using the BW model with sub-pixel accuracy (Fig. 22).

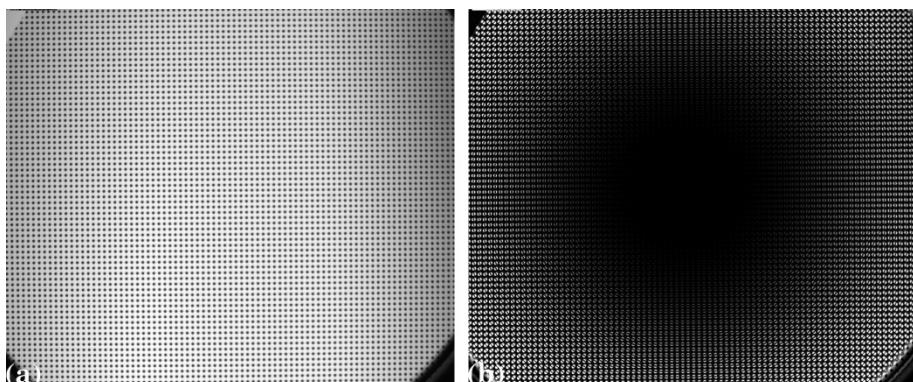


Figure 21. (a) Dot pattern image; (b) Difference between the distorted image and the undistorted image.

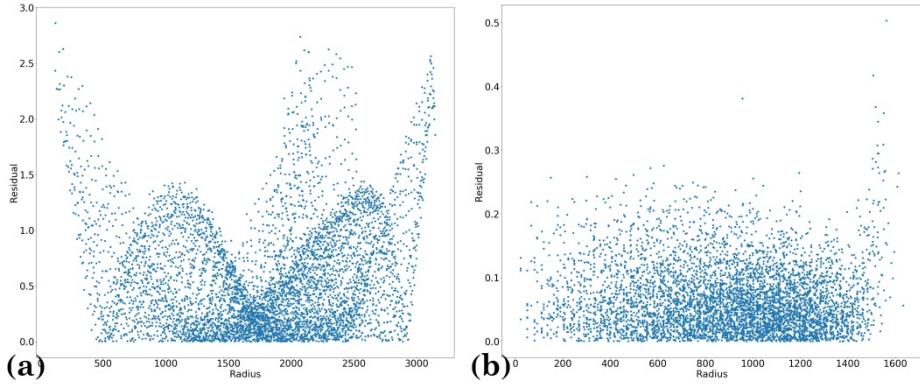


Figure 22. Residual plot: (a) Before correction; (b) After correction.

- Fig. 23 shows an X-ray projection of a sample (acquired by replacing the dot target with a scintillator) and its difference from the corrected image.

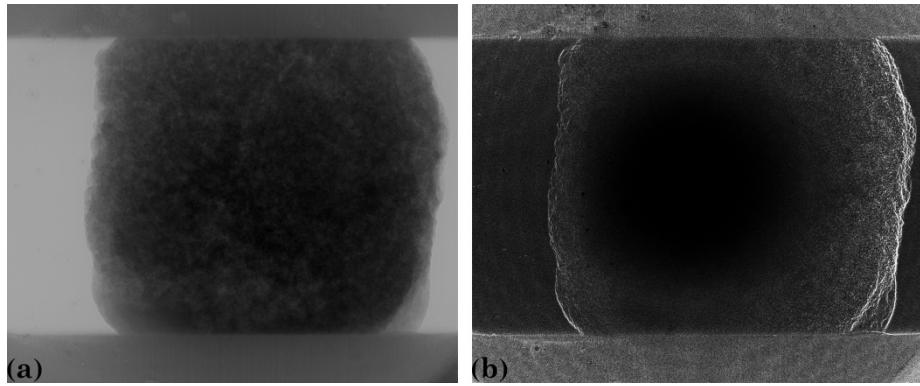


Figure 23. (a) Flat-field correction of an X-ray projection of a sample; (b) Difference between the original image and the corrected image.

- Although the residual distortion is just a few pixels, it has a clear impact on reconstructed images as shown in Fig. 24. A simple indication of the artifacts introduced by the distortion is that the reconstructed area close to the CoD (around the middle of an image) is artifact-free (Fig. 24b) but the area further away from the CoD shows streak artifacts (Fig. 24c). All these artifacts are removed after correction as can be seen in Fig. 25, where the shape of small pores is reconstructed properly (Fig. 25c).

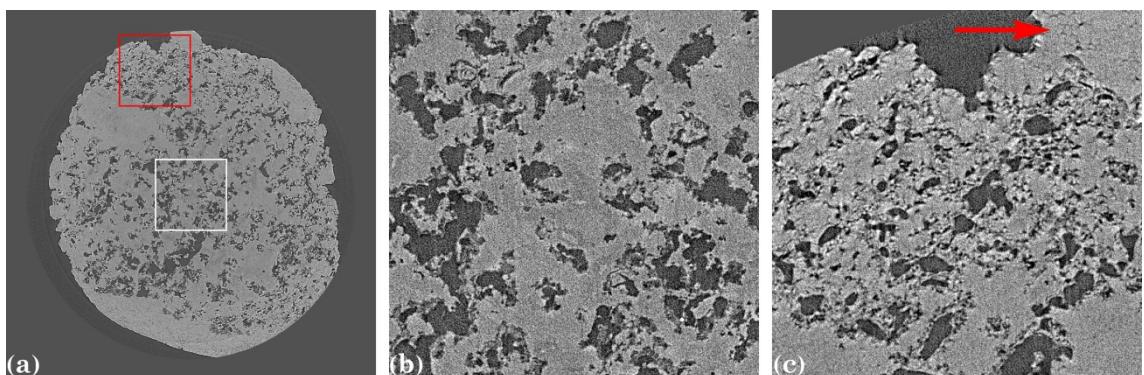


Figure 24. (a) Reconstructed slice at the middle row of Fig. 23a without distortion correction; (b) Magnified view from the white frame in (a) shows no artifacts; (c) Magnified view from the red frame in (a) shows streak artifacts.

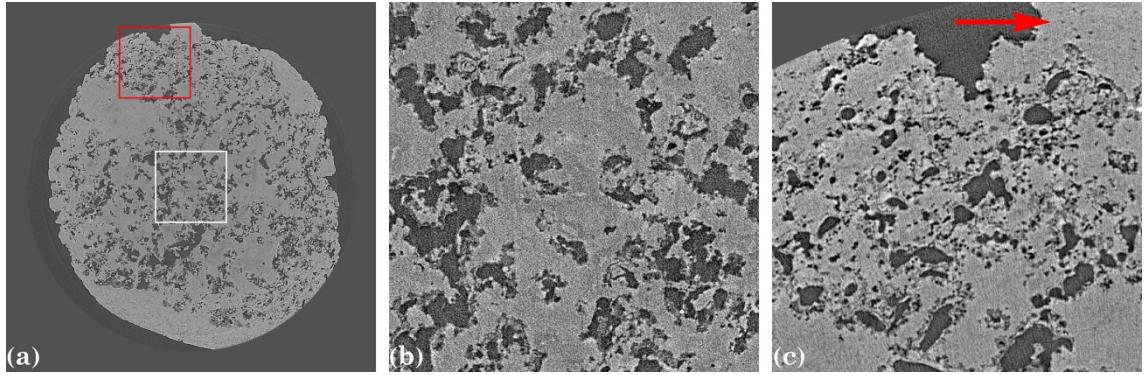


Figure 25. (a) Reconstructed slice at the middle row of Fig. 23a after distortion correction; (b) Magnified view from the white frame in (a) shows no artifacts; (c) Magnified view from the red frame in (a) shows no artifacts.

Exchanging a dot pattern with a scintillator is not always trivial. Depending on the optics design of an X-ray imaging system, it may be necessary to use an X-ray target (Fig. 7 and Fig. 8) as used at the beamline I13, DLS [9]. Some pre-processing methods provided in the package are to handle the issues caused by these bespoke targets. Further development of pre-processing methods to handle different types of patterns will be added. Figure 26 shows an example of processing a line pattern image which the methods of detecting lines was used.

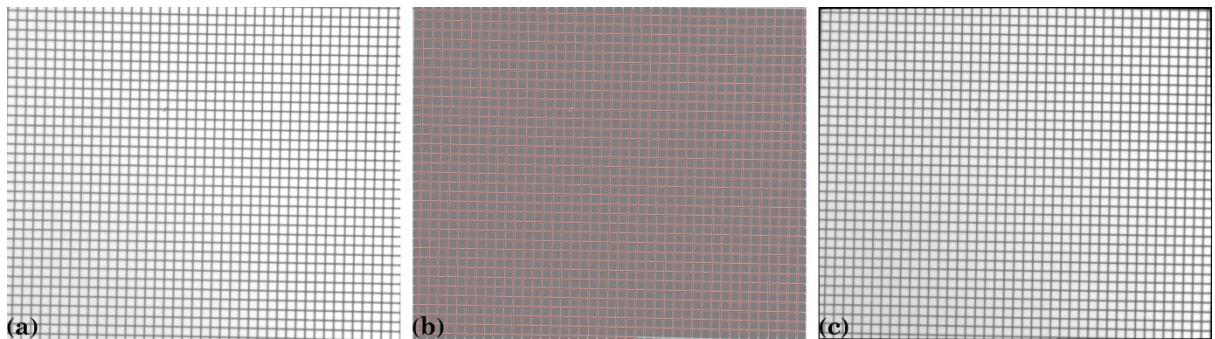


Figure 26. (a) Line pattern image; (b) Detected line image; (c) Corrected image.

Acknowledgments

This work was carried out with the support of the Diamond Light Source. The author thanks Robert Atwood, Michael Drakopoulos, Andrew Bodey, Malte Storm, and Shashidhara Marathe for providing pattern images used in this report.

References and links

- [1] Vo N. T., Atwood R., and Drakopolous M., "Radial lens distortion correction with sub-pixel accuracy for X-ray micro-tomography," Opt. Express 23, 32859-32868 (2015).
- [2] <https://conda.io/docs/index.html>
- [3] http://scikit-image.org/docs/dev/api/skimage.filters.html#skimage.filters.threshold_otsu
- [4] <http://scikit-image.org/docs/dev/api/skimage.measure.html#skimage.measure.regionprops>
- [5] <http://scikit-image.org/docs/dev/api/skimage.transform.html#skimage.transform.radon>
- [6] https://docs.scipy.org/doc/scipy-0.16.1/reference/generated/scipy.ndimage.measurements.center_of_mass.html
- [7] Vo N. T., Drakopoulos M., Atwood R. C., and Reinhard C., "Reliable method for calculating the center of rotation in parallel-beam tomography," Opt. Express 22, 19078-19086 (2014).
- [8] Drakopoulos, M., Connolley, T., Reinhard, C., Atwood, R., Magdysyuk, O., Vo, N., Hart, M., Connor, L., Humphreys, B., Howell, G., Davies, S., Hill, T., Wilkin, G., Pedersen, U., Foster, A., De Maio, N., Basham, M., Yuan, F. & Wanekl, K., "I12: the Joint Engineering, Environment and Processing (JEEP) beamline at Diamond Light Source", J. Synchrotron Rad. 22, 828-838 (2015).
- [9] Rau C., Wagner U., Pešić Z., and De Fanis A., "Coherent imaging at the Diamond beamline I13," Phys. Status Solidi A, 208, 2522-2525 (2011).