METHOD FOR CALIBRATING A GANTRY

FIELD OF THE INVENTION

[0001] The invention pertains to a method for calibrating a controller of a gantry, for example an XY gantry.

BACKGROUND

[0002] Precision stages are components of a motion system that enable highly accurate positioning, crucial in industries like semiconductor manufacturing and metrology. A XY gantry is a specific type of precision stage that allows two-dimensional movement along the X axis direction and the Y axis direction.

[0003] Figure 1 shows a typical XY gantry 100. XY gantry 100 comprises mover 110, a controller and a frame. The frame can be the tracks in which the mover 110 moves along. Responsive to instructions from the controller, mover 110 can be configured to move in a X axis direction and a Y axis direction within the frame, to reach a specific mover position. The mover position has a specific X and Y coordinate. There is normally a discrepancy between the expected mover position and the actual mover position. This is because of inaccuracies with the axis-to-axis perpendicularity of the frame and inaccuracies caused by the thermoelastic deformation of the frame.

[0004] To compensate for this discrepancy, the controller of XY gantry 100 can be calibrated using calibration jig 130. Calibration jig 130 can be aligned with XY gantry 100 as shown in Figure 1. Camera 120 can be centrally located on mover 110. By centrally locating on mover 110, camera 120 can capture images of an area directly below mover 110. This promotes accurate calibration.

[0005] Calibration jig 130 can comprise a plurality of calibration dots 140. The exact gaps between adjacent calibration dots 140 are known and is referred herein as the dot distance. Mover 110 is positioned over calibration dot 140 as shown in Figure 2A. Camera 120 then takes an image of calibration dot 140. Mover 110 is then moved to the right by the dot distance. This will result in mover 110 being positioned above the adjacent calibration dot 140 as shown in Figure 2B. Camera 120 then takes an image of calibration dot 140. Mover 110 is then again moved to the right by the dot distance. This will result in mover 110 being positioned above

the adjacent calibration dot 140 as shown in Figure 2C. Camera 120 then takes an image of calibration dot 140. This process is repeated for every calibration dot.

[0006] The images as captured by the above process are then analyzed. The center of calibration dot 140 in the image is referred herein as the dot center. The dot center is determined based on edge positions of calibration dot 140, where each edge position is determined computationally by a significant change in pixel intensity (from white to black or black to white) in the image. The center of the image is referred herein as the image center. The discrepancy is determined by evaluating the difference between the dot center and the image center.

[0007] The discrepancy for all calibration dots 140 is then used to compute a position error adjustment function. The position error adjustment function is then used to calibrate the controller of XY gantry 100 so that mover 110 can move to the "adjusted" position, which compensates for the aforementioned inaccuracies with the axis-to-axis perpendicularity of the frame and inaccuracies caused by the thermoelastic deformation of the frame.

[0008] However, the above described process does not take into account micro-vibrations in mover 110. The micro-vibrations can cause the images taken by camera 120 to be blurred and to be not sharp. This blurred image will adversely affect the calculation of the dot center of calibration dot 140. When the image is blurred, the change in pixel intensity in the image across an edge of calibration dot 140 will be affected (for example, the transition between the white to black pixels (i.e. "grey" pixels) may be increased). The detected edge position in a blurred image will deviate from the real-world position of the edge. Consequently, the use of the detected edge position will then result in an inaccurate dot center, and thus an inaccurate position error adjustment function.

[0009] There is therefore a need for an improved calibration method that determines a positional adjustment that compensates for the micro-vibrations.

SUMMARY OF INVENTION

[0010] According to an aspect of the invention, a method for calibrating a controller of a XY gantry is disclosed. The XY gantry comprises a frame and a mover configured to move in a X axis direction and a Y axis direction within the frame. The mover comprises a camera centrally located on the mover.

[0011] The method comprises the steps of: aligning a calibration jig with the XY gantry, the calibration jig comprising at least one calibration dot; positioning the mover over the at least one calibration dot; capturing with the camera a plurality of training images of the at least one calibration dot; and selecting a model image from the plurality of training images.

[0012] The method further comprises applying a set of scans to the model image to determine a plurality of model scan portions, the plurality of model scan portions focusing on different edges of the at least one calibration dot, each model scan portion comprising at least one array of pixels; for each model scan portion, determining a model subpixel edge position, the model subpixel edge position representing a subpixel position in the model scan portion where there is a significant change in pixel intensity; and determining a model subpixel edge gradient from the model subpixel edge position, the model subpixel edge gradient representing the change in pixel intensity with respect to subpixel position.

[0013] The method further comprises capturing with the camera a test image of the calibration dot; applying the set of scans to the test image to determine a plurality of test scan portions, the plurality of test scan portions focusing on different edges of the at least one calibration dot, each test scan portion comprising at least one array of pixels; for each test scan portion: determining a test subpixel edge position, the test subpixel edge position representing a subpixel position in the test scan portion where there is a significant change in pixel intensity; and determining a test subpixel edge gradient from the test subpixel edge position, the test subpixel edge gradient representing the change in pixel intensity with respect to subpixel position.

[0014] The method further comprises identifying the model scan portion that focuses on the same edge as the test scan portion; and shifting the model subpixel edge gradient of the identified model scan portion by at least one subpixel towards the test subpixel edge gradient to obtain a corrected model subpixel edge gradient, wherein the corrected model subpixel edge gradient represents the shifted model subpixel edge gradient that exhibits the highest normalized cross-correlation coefficient (NCC) with the test subpixel edge gradient.

[0015] The method further comprises obtaining a denoised subpixel edge position based on the corrected model subpixel edge gradient; determining a denoised dot center of the at least one calibration dot based on the denoised subpixel edge positions of the plurality of test scan portions; calculating a position adjustment function from the denoised dot center of the at least

one calibration dot; and applying the position adjustment function to the controller of the XY gantry.

[0016] Preferably, the step of selecting a model image from the plurality of training images comprises the steps of: for each training image, applying the set of scans to determine a plurality of scan portions, the plurality of scan portions focusing on different edges of the at least one calibration dot, each scan portion comprising at least one array of pixels; for each scan portion, determining a subpixel edge position, the subpixel edge position representing a subpixel position in the scan portion where there is a significant change in pixel intensity; determining a dot center position based on the subpixel edge positions of the plurality of scan portions; and selecting a training image from the plurality of training images as the model image, the selected training image having a dot center position which is median out of the dot center positions of the plurality of training images.

[0017] Preferably, the step of selecting a model image from the plurality of training images further comprises the steps of: for each scan portion in the plurality of scan portions of the selected training image, determining a subpixel edge projection from the subpixel edge position, the subpixel edge projection representing the pixel intensity profile of the scan portion; and determining a NCC of the subpixel edge projection with an ideal edge projection; determining a mean NCC from the NCC of the plurality of scan portions; and if the mean NCC of the selected training image is below a threshold, discarding the selected training image and selecting another training image from the remaining training images as the model image, the another selected training image having a dot center position which is median out of the dot center positions of the remaining training images.

[0018] Preferably, the threshold is at least 0.5.

[0019] Preferably, the mean NCC is determined from the sum of the NCC of the plurality of scan portions divided by the total number of scan portions of the selected training image.

[0020] Preferably, the ideal edge projection is derived from an ideal image of the at least one calibration dot; and wherein the ideal image exhibits no grey pixels at the edges of the at least one calibration dot.

[0021] Preferably, the step of determining a model subpixel edge position comprises the steps of: converting each pixel in the at least one array of pixels to RGB values to obtain at least one

array of RGB values; determining a normalized array of pixel intensity values from the at least one array of RGB values; determining an edge intensity projection from the normalized array of pixel intensity values; applying the first derivative on the edge intensity projection to obtain an edge gradient; and determining the model subpixel edge position from the edge gradient.

[0022] Preferably, the step of determining a model subpixel edge gradient from the model subpixel edge position comprises the steps of: identifying a set of pixel positions proximate to the model subpixel edge position; dividing the set of pixel positions into subpixel positions; plugging the subpixel positions into the edge gradient of the model scan portion to obtain an array of subpixel gradient values of the model scan portion; and determining the model subpixel edge gradient based on the array of subpixel gradient values of the model scan portion.

[0023] Preferably, the step of determining a test subpixel edge position comprises the steps of: converting each pixel in the at least one array of pixels to RGB values to obtain at least one array of RGB values; determining a normalized array of pixel intensity values from the at least one array of RGB values; determining an edge intensity projection from the normalized array of pixel intensity values; applying the first derivative on edge intensity projection to provide an edge gradient; and determining the test subpixel edge position from the edge gradient.

[0024] Preferably, the step of determining a test subpixel edge gradient from the test subpixel edge position comprises the steps of: identifying a set of pixel positions proximate to the test subpixel edge position; dividing the set of pixel positions into subpixel positions; plugging the subpixel positions into the edge gradient of the test scan portion to obtain an array of subpixel gradient values of the test scan portion; and determining the test subpixel edge gradient based on the array of subpixel gradient values of the test scan portion.

[0025] Preferably, the step of determining a subpixel edge position comprises the steps of: converting each pixel in the at least one array of pixels to RGB values to obtain at least one array of RGB values; determining a normalized array of pixel intensity values from the at least one array of RGB values; determining an edge intensity projection from the normalized array of pixel intensity values; applying the first derivative on the edge intensity projection to provide an edge gradient; and determining the subpixel edge position from the edge gradient.

[0026] Preferably, the step of determining a subpixel edge projection from the subpixel edge position comprises the steps of: identifying a set of pixel positions proximate to the subpixel edge position; dividing the set of pixel positions into subpixel positions; plugging the subpixel

positions into the edge intensity projection to obtain an array of subpixel intensity values; and determining the subpixel edge projection based on the array of subpixel intensity values.

[0027] Preferably, the step of identifying a set of pixel positions proximate to the model subpixel edge position, the test subpixel edge position or the subpixel edge position comprises: identifying 4 pixel positions immediately preceding the model subpixel edge position, the test subpixel edge position or the subpixel edge position; and identifying 4 pixel positions immediately following the model subpixel edge position, the test subpixel edge position or the subpixel edge position.

[0028] Preferably, the plurality of training images comprises 32 training images.

[0029] According to another aspect of the invention, a method for using a neural network model to calibrate a controller of a XY gantry is described. The method comprising the steps of aligning a calibration jig with the XY gantry, the calibration jig comprising at least one calibration dot; positioning the mover over the at least one calibration dot; capturing with the camera a set of training images of the at least one calibration dot; and applying a set of scans to each training image in the set of training images to obtain a plurality of scan portions, the plurality of scan portions focusing on different edges of the at least one calibration dot.

[0030] The method further comprises the steps of, for each scan portion in the plurality of scan portions: converting the scan portion into a first interpolated array of pixel intensity values; and feeding as input data, the first interpolated array of pixel intensity values into a neural network model, the neural network model comprising an encoder layer and a decoder layer, the encoder layer configured to create a low-dimension representation of the input data, and the decoder layer configured to recreate the input data.

[0031] The method further comprises the steps of capturing with the camera a set of testing images of the at least one calibration dot; and applying the set of scans to each testing image in the set of testing images to obtain a plurality of testing scan portions, each testing scan portion in the plurality of testing scan portions having a scan position.

[0032] The method further comprises the steps of, for each testing scan portion: converting the testing scan portion into a second interpolated array of pixel intensity values; and feeding as input data, the second interpolated array of pixel intensity values into the neural network model to output a modified interpolated array of pixel intensity values, the modified

interpolated array of pixel intensity values representing a modified edge of the testing scan portion.

[0033] The method further comprises the steps of grouping the testing scan portions that have the same scan position into a group; selecting an ideal modified edge for each group by comparing the modified edges of the testing scan portions within the group; and determining a denoised dot center of the at least one calibration dot from the ideal modified edges.

[0034] The method further comprises the steps of calculating a position adjustment function from the denoised dot center of the at least one calibration dot; and applying the position adjustment function to the controller of the XY gantry.

[0035] Preferably, the neural network model further comprises a loss function; and wherein the loss function implements *MSE Loss* and *Standard deviation Loss Across Images* and is configured to help the decoder layer recreate the input data with as little variability as possible.

[0036] Preferably, the encoder layer comprises interleaving *linear* layers and *ReLU* layers.

[0037] Preferably, the decoder layer comprises interleaving *linear* layers, *ReLU* layers and *sigmoid* layers.

[0038] Preferably, the step of selecting an ideal modified edge for each group by comparing the modified edges of the testing scan portions within the group comprises the steps of: determining a peak first derivative value of each of the modified edges of the testing scan portions within the group; comparing the peak first derivative values of the modified edges; and selecting as the ideal modified edge, the modified edge that exhibits the peak first derivative value which is median out of the peak first derivative values.

[0039] Preferably, each scan portion comprises at least one array of pixels; and the step of converting the training scan portion into a first interpolated array of pixel intensity values comprises the steps of: converting each pixel in the at least one array of pixels to RGB values to obtain at least one array of RGB values; determining a normalized array of pixel intensity values from the at least one array of RGB values; and performing interpolation on the normalized array of pixel intensity values to generate the first interpolated array of pixel intensity values.

[0040] Preferably, each testing scan portion comprises at least one array of pixels; and the step of converting the testing scan portion into a second interpolated array of pixel intensity values comprises the steps of: converting each pixel in the at least one array of pixels to RGB values to obtain at least one array of RGB values; determining a normalized array of pixel intensity values from the at least one array of RGB values; and performing interpolation on the normalized array of pixel intensity values to generate the second interpolated array of pixel intensity values.

[0041] Preferably, the set of training images comprises more images than the set of testing images.

[0042] Preferably, the at least one calibration dot is a plurality of calibration dots.

[0043] Other aspects of the invention will become clearer from the following detailed description of some preferred embodiments thereof.

BRIEF DESCRIPTION OF THE DRAWINGS

[0044] The accompanying figures, where like reference numerals refer to identical or functionally similar elements throughout the separate views and which together with the detailed description below are incorporated in and form part of the specification, serve to illustrate various embodiments, by way of example only, and to explain various principles and advantages in accordance with a present embodiment.

[0045] Figure 1 is an illustration of a calibration jig that is aligned with a XY gantry.

[0046] Figure 2A is an illustration of a mover being positioned over a calibration dot of a calibration jig.

[0047] Figure 2B is an illustration of a mover being positioned over another calibration dot of a calibration jig.

[0048] Figure 2C is an illustration of a mover being positioned over another calibration dot of a calibration jig.

[0049] Figure 3 depicts a method for calibrating a XY gantry, in accordance with certain embodiments.

[0050] Figure 4 depicts a method for determining a denoised dot center for a calibration dot, in accordance with certain embodiments.

[0051] Figure 5 depicts a method for selecting a model image from a plurality of training images, in accordance with certain embodiments.

[0052] Figure 6 is an illustration of a training image of a calibration dot and a scan portion, in accordance with certain embodiments.

[0053] Figure 7 depicts a method for determining a subpixel edge position for each scan portion, in accordance with certain embodiments.

[0054] Figure 8 shows an exemplary transition from a scan portion comprising 3 x 10 array of pixels, to a 3 x 10 array of RGB values, to a 3 x 10 array of pixel intensity values, to a 1 x 10 array of pixel intensity values, to a normalized 1 x 10 array of pixel intensity values, to an edge intensity projection, to an edge gradient, to a subpixel edge position, and to pixel positions proximate to the subpixel edge position, in accordance with certain embodiments.

[0055] Figure 9 depicts a method for determining a subpixel edge intensity projection based on a subpixel edge position, in accordance with certain embodiments.

[0056] Figure 10A is an illustration of a subpixel edge intensity projection and an ideal edge intensity projection, in accordance with certain embodiments.

[0057] Figure 10B is an illustration of a model subpixel edge gradient and a test subpixel edge gradient, in accordance with certain embodiments.

[0058] Figure 10C is an illustration of a shifted model subpixel edge gradient and a test subpixel edge gradient, in accordance with certain embodiments.

[0059] Figure 10D is an illustration of a further shifted model subpixel edge gradient and a test subpixel edge gradient, in accordance with certain embodiments.

[0060] Figure 10E is an illustration of a corrected model subpixel edge gradient and a denoised subpixel edge position, in accordance with certain embodiments.

[0061] Figure 11 depicts a method for determining a model subpixel edge gradient based on a model subpixel edge position, in accordance with certain embodiments.

[0062] Figure 12 depicts a method for obtaining a corrected model subpixel edge gradient, in accordance with certain embodiments.

[0063] Figure 13 shows a flow chart of a method that uses a neural network model to determine the denoised dot center of a calibration dot, in accordance with certain embodiments.

[0064] Figure 14 shows an exemplary transition from a 3 x 10 array of RGB values, to a 3 x 10 array of pixel intensity values, to a 1 x 10 array of pixel intensity values, to a normalized 1 x 10 array of pixel intensity values, and to an interpolated (1 x 10.n) array of pixel intensity values, in accordance with certain embodiments.

[0065] Figure 15 shows an exemplary neural network model, in accordance with certain embodiments.

[0066] Figure 16 shows an exemplary ReLU layer, in accordance with certain embodiments.

[0067] Figure 17 shows an exemplary sigmoid layer, in accordance with certain embodiments.

[0068] Figure 18 shows an exemplary neural network model having a loss function, in accordance with certain embodiments.

[0069] Skilled artisans will appreciate that elements in the figures are illustrated for simplicity and clarity and have not necessarily been depicted to scale. For example, the dimensions of some of the elements in the block diagrams or steps in the flowcharts may be exaggerated relative to other elements to help improve understanding of the present embodiment.

DETAILED DESCRIPTION

[0070] The following detailed description is merely exemplary in nature and is not intended to limit the invention or the application and uses of the invention. Furthermore, there is no intention to be bound by any theory presented in the preceding background of the invention or the following detailed description.

[0071] It is the intent of embodiments of the invention to teach a method for calibrating a controller of a XY gantry. The XY gantry can comprise a frame and a mover. The mover can be configured to move in a X axis direction and a Y axis direction within the frame. The mover can comprise a camera centrally located on the mover.

[0072] The method comprises the steps of aligning a calibration jig with the XY gantry. The calibration jig can comprise calibration dots. The method further comprises the steps of positioning the mover (of the XY gantry) over one calibration dot; and capturing with the camera (located on the mover) a plurality of training images of the calibration dot.

[0073] The method further comprises selecting a model image (or the sharpest image least affected by micro-vibrations) from the training images; and applying a set of scans to the model image to determine a plurality of model scan portions. The model scan portions can focus on different edges of the calibration dot. Each model scan portion can comprise at least one array of pixels.

[0074] For each model scan portion (which focus on an edge), the method further comprises determining a model subpixel edge position, the model subpixel edge position representing a subpixel position (or a value that allows identification of a specific subpixel in the model scan portion) where there is a significant change in pixel intensity (indicating the edge); and determining a model subpixel edge gradient from the model subpixel edge position, the model subpixel edge gradient representing the change in pixel intensity with respect to subpixel position.

[0075] The method further comprises capturing with the camera a test image (or raw data) of the calibration dot; and applying the set of scans (as applied to the model image) to the test image to determine a plurality of test scan portions. The test scan portions can focus on different edges of the at least one calibration dot. Each test scan portion can comprise at least one array of pixels.

[0076] For each test scan portion, the method further comprises determining a test subpixel edge position, the test subpixel edge position representing a subpixel position in the test scan portion where there is a significant change in pixel intensity (indicating the edge); determining a test subpixel edge gradient from the test subpixel edge position, the test subpixel edge gradient representing the change in pixel intensity with respect to subpixel position;.

[0077] For each test scan portion, the method further comprises identifying the model scan portion that focuses on the same edge as the test scan portion; and shifting the model subpixel edge gradient of the identified model scan portion by at least one subpixel towards the test subpixel edge gradient to obtain a corrected model subpixel edge gradient. Wherein the corrected model subpixel edge gradient represents the shifted model subpixel edge gradient

that exhibits the highest normalized cross-correlation coefficient (NCC) with the test subpixel edge gradient.

[0078] The method further comprises obtaining a denoised subpixel edge position based on the corrected model subpixel edge gradient; determining a denoised dot center based on the denoised subpixel edge positions of the plurality of test scan portions; calculating a position adjustment function from the denoised dot centers; and applying the position adjustment function to the controller of the XY gantry.

[0079] In essence, by shifting the model subpixel edge gradient to achieve the highest NCC with the test subpixel edge gradient, the disclosed embodiments of the method uses a weighted balance of raw data (from the test image) and exemplary data (from the model image). Advantageously, this results in a positional adjustment that accounts for both the actual operating conditions of the XY gantry and the micro-vibrations.

[0080] Because the model subpixel edge gradient is derived from the model image, the use of the model subpixel edge gradient ensures that the best possible edge data with the least amount of noise is provided for subsequent processing. Yet, the model subpixel edge gradient does not simply replace the test subpixel edge gradient. Instead, the model subpixel edge gradient has been corrected to account for the raw data by shifting it to achieve the highest NCC with the test subpixel edge gradient, and it is the corrected model subpixel edge gradient that is subsequently processed. This ensures that features of the raw data are still retained. In other words, the shifting step therefore retains all the beneficial characteristics of a sharp image without compromising the data representing the actual operating conditions.

[0081] As a result, the denoised subpixel edge position (i.e. the detected edge position) obtained from the corrected model subpixel edge gradient can be closer to the real-world position of the edge, compared to the raw data (i.e. test subpixel edge position).

[0082] In embodiments, the step of selecting a model image from the training images can comprise applying the set of scans to each training image in the training images to determine a plurality of scan portions. The scan portions can focus on different edges of the calibration dot. Each scan portion can comprise at least one array of pixels.

[0083] The step of selecting a model image from the training images can further comprise determining a subpixel edge position for each scan portion, the subpixel edge position

representing a subpixel position in the scan portion where there is a significant change in pixel intensity; determining a dot center position based on the subpixel edge positions of the scan portions; and selecting a training image from the training images as the model image. Wherein the selected training image can have a dot center position which is median out of the dot center positions of the training images.

[0084] Advantageously, by selecting the training image whose dot center is median, one can get the training image that is least affected by the micro-vibrations of the mover out of all training images. In other words, the training image with the least amount of noise. This is because a mover affected by micro-vibrations tend to oscillate or shake around a midpoint, where the midpoint of the oscillation can represent the "true" position of mover.

[0085] In embodiments, the step of selecting a model image from the plurality of training images can further comprise, determining a subpixel edge projection from the subpixel edge position for each scan portion in the scan portions of the selected training image, the subpixel edge projection representing the pixel intensity profile of the scan portion; and determining a NCC of the subpixel edge projection with an ideal edge projection.

[0086] It is also the intent of certain embodiments to use a neural network model to determine the denoised dot center of the calibration dot. The method comprises capturing with the camera training images of the calibration dot and applying a set of scans to each training image to obtain a plurality of scan portions, the plurality of scan portions focusing on different edges of the calibration dot.

[0087] The method further comprises that for each scan portion, converting the scan portion into an interpolated array of pixel intensity values, and then feeding as input data, the interpolated array of pixel intensity values into a neural network model. The neural network model comprising an encoder layer and a decoder layer, the encoder layer configured to create a low-dimension representation of the input data, and the decoder layer configured to recreate the input data.

[0088] The encoder layer can comprise interleaving linear layers and ReLU layers, and the decoder layer can comprise interleaving linear layers, ReLU layers and sigmoid layers. The neural network model can also comprise a loss function which implements *MSE Loss* and *Standard deviation Loss Across Images*.

[0089] The method further comprises capturing with the camera testing images of the calibration dot and applying the set of scans to each testing image to obtain a plurality of testing scan portions, each testing scan portion having a scan position.

[0090] The method further comprises that for each testing scan portion, converting the testing scan portion into an interpolated array of pixel intensity values, and then feeding as input data, the interpolated array of pixel intensity values into the neural network model to output a modified interpolated array of pixel intensity values, the modified interpolated array of pixel intensity values representing a modified edge of the testing scan portion.

[0091] The method further comprises grouping the testing scan portions that have the same scan position into a group; selecting an ideal modified edge for each group by comparing the modified edges of the testing scan portions within the group; and determining a denoised dot center of the at least one calibration dot from the ideal modified edges.

[0092] The neural network model is capable of determining the denoised dot center of the calibration dot as its design is geared towards the input data i.e. the interpolated array of pixel intensity values. As this input data is one-dimensional in nature, the neural network model advantageously only have linear layers, and does not have any convolutional layers. Thus, neural network model is straightforward and has fast processing speed.

[0093] This is in contrast to other models that may require convolutional layers to handle two-dimensional image data, to for example, capture more complex information like shapes and textures. Instead, as the input data is one-dimensional image data, the linear architecture of neural network model is more than capable of optimizing the image data, and determining denoised dot center of the calibration dot.

[0094] Furthermore, the neural network model also prioritizes efficiency. As calibration of the controller of XY gantry usually has to be done on-site, the computation of the denoised dot center of the calibration dot has to be done quickly and efficiently. As the neural network model is based on an autoencoder architecture which invokes a reduction in dimensionality, the linear layers of the neural network model are very efficient, and computation of the denoised dot center of the calibration dot can be performed close to real-time.

[0095] Even further still, autoencoders are unsupervised models. As such, no labelled data is required. This means that the neural network model can learn purely on features it can detect

from the input data which increases its robustness to noise. This is extremely useful in computation of the denoised dot center of the calibration dot as a key objective is to "preserve the edge" of the calibration dot despite the inherent vibration in the XY gantry when the camera is taking the training images and the testing images.

[0096] Autoencoders are inherently supposed to recreate the input data despite the encoding, the decoding, and the inherent vibration present in the training images. Autoencoders are supposed to identify all the key features of the input data despite the noise present which is highly appropriate in this use case.

[0097] Therefore, due to its autoencoder design, the neural network model is ideal in accounting for the micro-vibrations of the XY gantry, and determining the denoised dot center of the calibration dot. The accuracy of the neural network model is further enhanced with the inclusion of the loss function, where the application of both *MSE Loss* and *Standard Deviation Loss Across Images* allows the neural network model to reconstruct the scan portions of the training images, and the testing scan portions of the testing images, with as little variability as possible, despite the inherent noise in the XY gantry.

[0098] Figure 3 shows a flow chart depicting a method for calibrating a controller of XY gantry 100.

[0099] Referring to Figure 3, in step 302, calibration jig 130 can be aligned with XY gantry 100. Calibration jig 130 can comprise at least one calibration dot 140, or a plurality of calibration dots 140. The distance between adjacent calibration dots 140 can be referred to as the dot distance. For example, the dot distance can be 5 mm, 10 mm, 20 mm etc. The dot distance can be consistent across calibration dots 140.

[00100] By aligning calibration jig 130 with XY gantry 100, the edges of calibration jig 130 can be matched with the edges of the frame of XY gantry 100. The edges of calibration jig 130 can be substantially parallel with the edges of the frame.

[00101] In step 304 of Figure 3, mover 110 can be positioned over each calibration dot 140. Mover 110 can be positioned over calibration dot 140 as shown in Figure 2A. Mover 110 can then be moved to the right by the dot distance. This will result in mover 110 being positioned above the adjacent calibration dot 140 as shown in Figure 2B. Mover 110 can then be moved

to the right again by the dot distance. This will result in mover 110 being positioned above the adjacent calibration dot 140 as shown in Figure 2C.

[00102] In step 306 of Figure 3, denoised dot centers of calibration dots 140 can be determined. Figure 4 describes in a flow chart a method of determining a denoised dot center of calibration dot 140.

[00103] Referring to Figure 4, in step 402, a plurality of training images 450 can be captured for each calibration dot 140. An example of training image 450 is depicted in Figure 6. Training images 450 can be captured with camera 120 centrally located on mover 110. Any suitable number of training images 450 can be captured. For example, camera 120 can capture 16, 24, 32, 64 training images 450 etc of each calibration dot 140.

[00104] In embodiments, camera 120 can capture one training image 450 of calibration dot 140, and then one training image 450 of the next calibration dot 140 until training images 450 of all calibration dots 140 in calibration jig 130 are captured. This process can then be repeated 32 times.

[00105] Alternatively, camera 120 can capture 32 training images 450 of calibration dot 140, and then 32 training images 450 of the next calibration dot 140 and so on until 32 training images 450 of all calibration dots 140 in calibration jig 130 are captured.

[00106] In step 404 of Figure 4, a model image of calibration dot 140 can be determined from training images 450. Figure 5 describes in a flow chart a method of determining a model image from training images 450.

[00107] Referring to Figure 5, in step 502, a set of scans can be applied to each training image 450. The set of scans are illustrated in Figure 6 and are depicted by arrows. Each scan can be applied across an edge 610 of calibration dot 140. Each scan can have a scan position 640. Each scan can determine one scan portion 650. Each scan portion 650 can have a unique scan position 640. The set of scans can therefore determine a plurality of scan portions 650. The set of scans as shown in Figure 6 comprises of 24 scans (6 on each side), thereby determining 24 scan portions 650. Each of the 24 scan portions 650 can have a unique scan position 640.

[00108] The scans can be in one or more directions. As shown in Figure 6, the scans can comprise left to right scans 602, right to left scans 604, top to bottom scans 606 and bottom to top scans 608.

[00109] Scan portions 650 can focus on edges 610 of calibration dot 140. Each scan portion 650 can focus on one edge 610, therefore scan portions 650 can focus on different edges 610. Each scan portion 650 can comprise at least one array of pixels. Each scan portion 650 can comprise multiple arrays of pixels. Each scan portion 650 can have a user defined size in terms of (H) height x (W) width pixels. For example, each scan portion 650 can be 1 x 10 pixels, 2 x 10 pixels, 3 x 10 pixels, 3 x 20 pixels, 3 x 30 pixels etc. An example of scan portion 650 having a size of 3 x 10 pixels is shown in Figure 6. Since each scan is applied across edge 610, scan portion 650 will contain at least one pixel representing edge 610.

[00110] In step 504 of Figure 5, subpixel edge position 820 can be determined for each scan portion 650. Figure 7 describe in a flowchart a method of determining subpixel edge position 820 for each scan portion 650.

[00111] Referring to Figure 7, in step 702, each pixel of the at least one array of pixels can be converted to RGB values to obtain at least one array of RGB values. A pixel that is completely white has a RGB value of (255, 255, 255). A pixel that is completely black has a RGB value of (0, 0, 0). As a result, scan portion 650 having a size of H x W pixels can turn into a H x W array of RGB values. An example of 3 x 10 array of RGB values 802 is shown in Figure 8.

[00112] In step 704 of Figure 7, an array of pixel intensity values can be determined from the array of RGB values. This determination can be done by averaging the RGB values or greyscaling. An example of 3 x 10 array of pixel intensity values 804 is shown in Figure 8.

[00113] In step 706 of Figure 7, the array of pixel intensity values can be reduced. A H x W array of pixel intensity values can be reduced to a 1 x W array of pixel intensity values by averaging the pixel intensity value of each column. An example of 1 x 10 array of pixel intensity values 806 is shown in Figure 8. By reducing the array of pixel intensity values, random noise present in training image 450 is reduced.

[00114] In step 708 of Figure 7, the array of pixel intensity values can be normalized. The normalization can be done by setting the highest pixel intensity value as 1, and lowest pixel

intensity value as 0. An example of normalized 1 x 10 array of intensity values 807 is shown in Figure 8.

[00115] The skilled person will appreciate that step 702 to step 708 do not need to be carried out according to the described sequence. For example, step 708 can be carried out before step 706.

[00116] In step 710 of Figure 7, edge intensity projection 808 can be determined from the normalized array of pixel intensity values. Edge intensity projection 808 can be a polynomial function ((x)) that best fit all the points derived from plotting the normalized array of pixel intensity values on a graph. An example of edge intensity projection 808 is depicted in Figure 8.

[00117] As shown in Figure 8, edge intensity projection 808 can be graphically represented as a sigmoid curve (or S-shaped curve), where the Y axis represents the normalized intensity and the X axis represents pixel position. The pixel position can be a value that allows identification of a specific pixel within scan portion 650. For example, a pixel position of 0 can represent the leftmost pixel of scan portion 650. It is appreciated that a pixel can be divided into subpixels. Therefore similar to a pixel position, a subpixel position can be a value that allows identification of a specific subpixel within scan portion 650. For example, a subpixel position of 0.00 can represent the leftmost subpixel of scan portion 650.

[00118] Following the curve of edge intensity projection 808 from left to right, normalized intensity starts high (around 1) and slowly decreases. The normalized intensity then drastically decreases at a certain pixel position. The normalized intensity then flattens out. Edge intensity projection 808 can have an inflection point, where the normalized intensity drastically decreases. This inflection point indicates the position of edge 610.

[00119] In step 712 of Figure 7, edge gradient 810 can be obtained by applying the first derivative on edge intensity projection 808 with respect to the pixel position. Edge gradient 810 can be a polynomial function ($f'(x)$). An example of edge gradient 810 is depicted in Figure 8. Edge gradient 810 can represent the extent in which the normalized pixel intensity is changing with respect to the pixel position in edge intensity projection 808. In other words, edge gradient 810 can represent the change in the gradient of edge intensity projection 808.

[00120] As shown in Figure 8, edge gradient 810 can be graphically represented as a parabola with a peak, where the Y axis is the gradient and the X axis is the pixel position.

[00121] In step 714 of Figure 7, subpixel edge position 820 can be determined from edge gradient 810. Subpixel edge position 820 can be a subpixel position in scan portion 450 where there is a significant change in pixel intensity. As shown in Figure 8, subpixel edge position 820 can be the pixel position (or subpixel position) that corresponds to the peak of edge gradient 810. Figure 8 shows that subpixel edge position 820 can be derived from the peak of edge gradient 810 by drawing a vertical line through the peak and finding the pixel position or subpixel position that intersects with the vertical line.

[00122] Referring back to Figure 5, in step 506, dot center 620 of training image 450 can be determined based on subpixel edge positions 820 of scan portions 650 (of that training image 450). Dot center 620 can be determined by any known dot center determination method, such as the least square method. For example, Figure 6 shows that dot center 620 and dot radius 630 can be determined by fitting circle 614 using subpixel edge positions 820 of edges 610.

[00123] In step 508 of Figure 5, select a training image 450 from training images 450 as the model image. The dot center 620 of training images 450 can be compared, and the training image 450 whose dot center 620 is median can be selected as the model image. Dot center 620 can be converted into Euclidian distance to facilitate determination of the median dot center 620.

[00124] It is believed that mover 110 affected by micro-vibrations will oscillate or shake around a midpoint, where the midpoint of the oscillation can represent the “true” position of mover 110. By selecting the training image 450 whose dot center 620 is median, one should get the training image 450 that is captured during the midpoint of the oscillation. That is, the training image 450 that is least affected by the micro-vibrations of mover 110 out of all training images 450.

[00125] In an alternative embodiment, the step of determining the model image of calibration dot 140 can further comprise steps 510 to 520 as shown in Figure 5.

[00126] In step 510 of Figure 5, subpixel edge projection 920 can be determined from subpixel edge position 820 of selected training image 450 (from step 508). Figure 9 describe in a flow chart a method of determining subpixel edge projection 920 from subpixel edge position 820.

[00127] Referring to Figure 9, in step 902, identify a set of pixel positions 830 proximate to subpixel edge position 820. Pixel positions 830 can contain a specific number of pixel positions, such as 2, 3, 4, 5, 6, 7, 8 etc pixel positions proximate to the subpixel edge position 820. For example, Figure 8 shows that when the subpixel edge position 820 is between pixel positions 10 and 11, then pixel positions 830 can be (7, 8, 9, 10, that is, the 4 pixel positions immediately preceding subpixel edge position 820) and (11, 12, 13, 14, that is, the 4 pixel positions immediately following the subpixel edge position 820).

[00128] In step 904 of Figure 9, divide pixel positions 830 into subpixel positions 840. For example, each pixel falling within pixel positions 830 can be divided into 5, 10, 50, or 100 subpixels, to result in a set of subpixel positions 840. Figure 8 illustrates an example where each pixel is divided into 5 subpixels, to result in 40 subpixel positions 840 of (7.0, 7.2, 7.4. 7.6, 7.8, 8.0…..13.0, 13.2, 13.4, 13.6, 13.8, 14.0). In a preferred embodiment, each pixel falling within pixel positions 830 can be divided into 100 subpixels.

[00129] In step 906 of Figure 9, plug subpixel positions 840 into edge intensity function 808 to obtain an array of subpixel intensity values. Advantageously, this effectively improves resolution by interpolation, i.e. constructing new data based on known data.

[00130] In step 908 of Figure 9, determine subpixel edge intensity projection 920 based on the array of subpixel intensity values. The determination can be based on polynomial curve fitting of any suitable order. The determination can be based on 3rd order polynomial curve fitting, 4th order polynomial curve fitting, 5th order polynomial curve fitting, 6th order polynomial curve fitting, or 7th order polynomial curve fitting etc. Preferably, the determination can be based on 7th order polynomial curve fitting. Higher order polynomial curve fitting generally increases accuracy at the expense of longer computation time.

[00131] Subpixel edge intensity projection 920 can be a polynomial function that best fit all the points derived from plotting the array of subpixel intensity values on a graph. An example of subpixel edge intensity projection 920 is shown in Figure 10A. As shown in Figure 10A, subpixel edge intensity projection 920 can be a sigmoid curve where the Y axis is pixel intensity and the X axis is subpixel position. Subpixel edge intensity projection 920 can therefore represent the pixel intensity profile of scan portion 450.

[00132] Advantageously, by determining subpixel edge intensity projection 920 based on subpixel edge position 820, accuracy and productivity is improved. This is because only a small

portion of scan portion 650 actually captures edge 610. This allows the subsequent processing steps to focus on the pixel positions around subpixel edge position 820 at a higher resolution.

[00133] Referring back to Figure 5, in step 512, a normalized cross-correlation coefficient (hereafter, NCC) can be determined from subpixel edge intensity projection 920. The NCC represents the extent in which subpixel edge intensity projection 920 resembles ideal edge intensity projection 1002. An example of ideal edge intensity projection 1002 is depicted in Figure 10A.

[00134] Ideal edge intensity projection 1002 can be derived from an ideal image of calibration dot 140. The ideal image can exhibit no transition (i.e. no grey pixels) between the black pixels and white pixels at edges 610 of calibration dot 140.

[00135] The NCC can be determined according to the equation below:

[00136]

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Wherein:

x represents normalized intensity from subpixel edge intensity projection 920,

y represents normalized intensity from ideal edge intensity projection 1002,

$\bar{x}$ represents mean intensity from subpixel edge intensity projection 920,

$\bar{y}$ represents mean intensity from ideal edge intensity projection 1002, and

n represents the total number of subpixel positions 840

[00137] The more subpixel edge intensity projection 920 resembles ideal edge intensity projection 1002, the higher the NCC. An NCC of 1 indicates that subpixel edge intensity projection 920 is perfectly correlated with ideal edge intensity projection 1002, while an NCC of 0 indicates no correlation between subpixel edge intensity projection 920 and ideal edge intensity projection 1002.

[00138] In step 514 of Figure 5, a mean NCC can be determined from the NCC of subpixel edge intensity projections 920 of selected training image 450 (from step 508). Since each subpixel edge intensity projection 920 (of scan portion 650) determines a NCC, the mean NCC can be determined based on the sum of all NCC divided by total number of scan portions 650 of selected training image 450.

[00139] In step 516 of Figure 5, the mean NCC of the selected training image 450 can be evaluated for whether it is above a threshold. Examples of threshold are 0.5, 0.6, 0.7 and 0.8, preferably 0.7 and 0.8. The threshold should ensure that if the selected training image 450 is optimal for some reason (such as not being sharp enough, or method performed incorrectly), it is discarded.

[00140] In step 518 of Figure 5, if the mean NCC of the selected training image 450 is above the threshold, then the selected training image 450 can be assigned as the model image.

[00141] In step 520 of Figure 5, if the mean NCC of the selected training image 450 is not above the threshold, then this training image 450 should be discarded, and another training image 450 should be selected as the model image instead. The dot center 620 of training images 450 are once again compared, and the training image 450 whose dot center 620 is median among the remaining training images 450 (less the discarded training image 450) can be selected as the model image.

[00142] Step 516 can then be applied to the newly selected training image 450. If the mean NCC of the newly selected training image 450 is above the threshold, then this newly selected training image 450 can be assigned as the model image. If not, step 520 is repeated.

[00143] Referring back to Figure 4, in step 406, apply the set of scans to the model image to determine model scan portions. The model scan portions can be determined in a similar manner to scan portions 650 as described in step 502 of Figure 5.

[00144] In step 408 of Figure 4, determine a model subpixel edge position and model subpixel edge gradient 1010 for each model scan portion.

[00145] The model subpixel edge position can be determined from the model scan portion. The model subpixel edge position can be determined from the model scan portion in a similar manner as determining subpixel edge position 820 from scan portion 650 as described in steps

702 to 714 in Figure 7. The model subpixel edge position can represent a subpixel position in the model scan portion where there is a significant change in pixel intensity.

[00146] Model subpixel edge gradient 1010 can be determined based on the model subpixel edge position. Figure 11 describe in a flowchart a method of determining model subpixel edge gradient 1010 based on the model subpixel edge position.

[00147] Referring to Figure 11, in step 1102, identify a set of pixel positions proximate to the model subpixel edge position. The set of pixel positions proximate to the model subpixel edge position can be identified in a similar manner as pixel positions 830 in step 902 of Figure 9.

[00148] In step 1104 of Figure 11, divide the pixel positions into subpixel positions. The pixel positions can be divided into subpixel positions in a similar manner as subpixel positions 840 in step 904 of Figure 9.

[00149] In step 1106 of Figure 11, plug the subpixel positions into edge gradient of the model scan portion (derived from step 712) to obtain an array of subpixel gradient values.

[00150] In step 1108 of Figure 11, determine model subpixel edge gradient 1010 based on the array of subpixel gradient values. Model subpixel edge gradient 1010 can represent the change in pixel intensity with respect to subpixel position. An example of model subpixel edge gradient 1010 is depicted in Figure 10B. The determination can be based on polynomial curve fitting of any suitable order as described in the previous embodiments. The determination can be based on 7^{th} order polynomial curve fitting.

[00151] Advantageously, by determining model subpixel edge gradient 1010 based on the model subpixel edge position, accuracy and productivity can be improved. This is because only a small portion of the model scan portion actually captures the edge. This means that the subsequent processing steps can focus on the pixel positions around the model subpixel edge position at a higher resolution, resulting in a more resource-efficient process.

[00152] Referring back to Figure 4, in step 410, a test image of the same calibration dot 140 can be captured with camera 120. The test image can be captured using the same method for capturing training image 450 as described in step 402.

[00153] In step 412 of Figure 4, apply the set of scans to the test image to determine test scan portions. The test scan portions can be determined using the method for determining scan portion 650 as described in the previous embodiments.

[00154] In step 414 of Figure 4, determine a test subpixel edge position and test subpixel edge gradient 1020 for each test scan portion.

[00155] The test subpixel edge position can be determined using the method for determining subpixel edge position 820 from scan portion 450 as described in Figure 7. The test subpixel edge position can represent a subpixel position in the test scan portion where there is a significant change in pixel intensity.

[00156] Test subpixel edge gradient 1020 can be determined from the test subpixel edge position in a similar manner as determining model subpixel edge gradient 1010 as described in steps 1102 to 1108 in Figure 11. Test subpixel edge gradient 1020 can represent the change in pixel intensity with respect to subpixel position. An example of test subpixel edge gradient 1020 is depicted in Figure 10B. Test subpixel edge gradient 1020 can be less symmetrical compared to model subpixel edge gradient 1010. This asymmetry can be due to effects of micro-vibrations and/or noise in the raw data.

[00157] In step 416 of Figure 4, for each test scan portion, shift model subpixel edge gradient 1010 of a corresponding model scan portion by at least one subpixel towards test subpixel edge gradient 1020 to obtain corrected model subpixel edge gradient 1030. Corrected model subpixel edge gradient 1030 can represent the model subpixel edge gradient 1010 that has been shifted which exhibits the highest NCC with test subpixel edge gradient 1020.

[00158] A model scan portion can be corresponding to a test scan portion if they are focused on the same edge 610 of calibration dot 140. That is, the model scan portion and the test scan portion are obtained from the scan at the same scan position 640. Prior to step 416, the model scan portion that is focused on the same edge 610 as the test scan portion can be identified.

[00159] The highest NCC can be determined by a recursive process involving shifting model subpixel edge gradient 1010 one subpixel at a time and then calculating the NCC after each shift. As an example, Figure 10B depicts model subpixel edge gradient 1010 before any shifting is done. Model subpixel edge gradient 1010 can then be shifted by one subpixel to the left, to result in the profile as shown in Figure 10C. The NCC of the shifted model subpixel edge

gradient 1010 with the test subpixel edge gradient 1020 is then calculated. Model subpixel edge gradient 1010 can be further shifted by one subpixel to the left (that is two subpixels to the left from the original position in Figure 10B), to result in the profile as shown in Figure 10D. The NCC of the newly shifted model subpixel edge gradient 1010 with the test subpixel edge gradient 1020 is then calculated. This process can be repeated for a predetermined number of subpixels, for example, 30, 50, 100 subpixels etc. This process can be repeated for the all the subpixels within one pixel, that is, if each pixel has been divided into 100 subpixels, this process is repeated for 100 subpixels.

[00160] In the example above, because the NCC derived from shifting the model subpixel edge gradient 1010 by two subpixels to the left is the highest (out of all shifts that have been done), the model subpixel edge gradient 1010 that has been shifted by two subpixels to the left is assigned as the corrected model subpixel edge gradient 1030.

[00161] Figure 12 depicts how the corrected model subpixel edge gradient 1030 can be obtained. As shown, test subpixel edge gradient 1020 can have gradients (a, b, c … h) at a set of subpixel positions. While shifted model subpixel edge gradient 1010 (the model subpixel edge gradient 1010 which exhibits the highest NCC as determined in the previous embodiments) can have gradients (ab', bc', cd'… hi') at the same set of subpixel positions. As shown in Figure 12, the gradients of test subpixel edge gradient 1020 can be replaced by the corresponding gradients of shifted model subpixel edge gradient 1010. That is, "a" replaced by "ab'", "b" replaced by "bc'" etc. As a result, corrected model subpixel edge gradient 1030 is obtained.

[00162] In an embodiment, the process of shifting model subpixel edge gradient 1010 one subpixel at a time and then calculating the NCC after each shift can end when the NCC starts to decrease. This is because this indicates that the highest NCC has been found.

[00163] The skilled person would appreciate that the shift can be done in the left direction and/or right direction, depending on the relative position of the test subpixel edge gradient 1020.

[00164] If model subpixel edge gradient 1010 is represented by a 7^{th} order polynomial function a + bx + cx^2 … hx^7, then model subpixel edge gradient 1010 that is shifted by 1 subpixel in the left direction can be represented by the function a + b(x+1) + c(x+1)^2 … h(x+1)^7.

[00165] The NCC can be calculated according to the equation as described in previous embodiments, but replacing the substituents. For example:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Wherein:

x represents gradient from model subpixel edge gradient 1010 that is shifted,

y represents gradient from test subpixel edge gradient 1020,

$\bar{x}$ represents mean gradient from model subpixel edge gradient 1010 that is shifted,

$\bar{y}$ represents mean gradient from test subpixel edge gradient 1020, and

n represents the total number of subpixel positions

[00166] It is appreciated that the NCC can represent the extent of overlay between the shifted model subpixel edge gradient 1010 and the test subpixel edge gradient 1020.

[00167] Advantageously, by shifting the model subpixel edge gradient 1010 to achieve the highest NCC with the test subpixel edge gradient 1020, the method uses a weighted balance of the raw data and the exemplary data (provided by the model image) to arrive at a positional adjustment that best accounts for both the actual operating conditions of XY gantry 100 and the micro-vibrations.

[00168] As the model subpixel edge gradient 1010 is derived from the model image (which is the sharpest training image 450 that is least affected by the micro-vibrations), the use of the model subpixel edge gradient 1010 ensures that the best possible edge data with the least amount of noise is provided for subsequent processing. Yet, features of the raw data are not ignored, i.e. the model subpixel edge gradient 1010 does not simply replace the test subpixel edge gradient 1020. Instead, the model subpixel edge gradient 1010 has been corrected to account for the raw data (by shifting it to achieve the highest NCC with test subpixel edge gradient 1020), and it is the corrected model subpixel edge gradient 1030 that is subsequently processed. This ensures that features of the raw data are still retained. The shifting step therefore retains all the beneficial characteristics of a sharp image without compromising the data representing the actual operating conditions.

[00169] In step 418 of Figure 4, denoised subpixel edge position 1050 can be obtained based on the corrected model subpixel edge gradient 1030. Denoised subpixel edge position 1050 can

be obtained using the method for determining subpixel edge position 820 from edge gradient 810. As shown in Figure 10E, denoised subpixel edge position 1050 can be a subpixel position that correspond to the peak of corrected model subpixel edge gradient 1030. The denoised subpixel edge position 1050 can be closer to the real-world position of edge 610, compared to the test subpixel edge position.

[00170] In step 420 of Figure 4, for each calibration dot 140, a denoised dot center can be determined based on denoised subpixel edge positions 1050. The denoised dot center can be determined using the method for determining dot center 620 based on subpixel edge positions 820 as described in previous embodiments. The denoised dot center can be performed by known dot center determination methods. In an embodiment, to facilitate determination of the denoised dot center by known dot center determination methods, denoised subpixel edge position 1050 can be converted from a subpixel position to a pixel position. For example, if denoised subpixel edge position 1050 has a subpixel position of 4.12, then it can be converted to a pixel position of 4.

[00171] Advantageously, by denoising dot center 620, the method improves the capability at which camera 120 can measure dot center 620 repeatably. The capability can be measured by the following steps:

Taking 32 images of a calibration dot 140,

Measuring dot center 620 and dot radius 630, and

Computing the standard deviation of the 32 dot centers 620 and dot radii 630.

It was observed that prior to denoising, the standard deviation of dot center 620 is many times worse than dot radii 630. After denoising, the standard deviation of the denoised dot center is around the same as the dot radii 630. This means that the denoising results in repeatable measurements of the dot center 620 position.

[00172] Referring back to Figure 3, in step 308, a position adjustment function can be calculated from the denoised dot center of calibration dot 140. The position adjustment function can be calculated based on the discrepancy between the image center (i.e. the center of the image) and the denoised dot center. The method can then be repeated for other calibration dots 140 in calibration jig 130 to determine the denoised dot centers of the other calibration dots

140 in calibration jig 130. The denoised dot centers of the other calibration dots 140 can then also be used to calculate the position adjustment function.

[00173] In step 310 of Figure 3, the position adjustment function can then be applied to the controller of XY gantry 100. The position adjustment function can be used to move mover 110 to the "adjusted" position.

[00174] In embodiments, the denoised dot centers of calibration dots 140 can be determined via machine learning. In particular, the denoised dot centers of calibration dots 140 can be determined via a neural network model.

[00175] Figure 13 shows a flow chart of a method that uses a neural network model to determine the denoised dot center of calibration dot 140.

[00176] In step 1302 of Figure 13, prepare a set of training images 450 for calibration dot 140. Similar to the previously described embodiments, training images 450 can be the images taken by camera 120 of calibration dot 140. The set of training images 450 can comprise 8, 16, 24, 32, 40 or 48 training images 450, or any reasonable number of training images 450, so as to train the neural network model.

[00177] In step 1304 of Figure 13, for each training image 450, prepare scan portions 650. Similar to the previously described embodiments, scan portions 650 can be the portions of training images 450 which focuses on an edge 610 of calibration dot 140. Each scan portion 650 can comprise multiple arrays or at least one array of pixels. Each scan portion 650 can comprise scan position 640 (as shown in Figure 6).

[00178] In step 1306 of Figure 13, for each scan portion 650, convert each pixel of scan portion 650 into multiple arrays of RGB values. For example, as a result of this step, scan portion 650 having a size of H x W pixels can produce an H x W array of RGB values. As an example, a 3 x 10 array of RGB values 1402 is illustrated in Figure 14.

[00179] In step 1308 of Figure 13, determine multiple arrays of pixel intensity values from the multiple arrays of RGB values. This determination can be done by averaging the RGB values or greyscaling. For example, a RGB value (230, 220, 210) can be averaged and result in a pixel intensity value of (220). As an example, a 3 x 10 array of pixel intensity values 1404 is illustrated in Figure 14.

[00180] In step 1310 of Figure 13, reduce the multiple arrays of pixel intensity values to a single array of pixel intensity values. For example, a H x W array of pixel intensity values can be reduced to a 1 x W array of pixel intensity values by averaging the pixel intensity value of each column. As an example, a 3 x 10 array of pixel intensity values 1404 is reduced to a 1 x 10 array of pixel intensity values 1406, as illustrated in Figure 14. Reducing the multiple arrays of pixel intensity values to a single array of pixel intensity values is beneficial as it can help reduce the random noise in scan portion 650.

[00181] In step 1312 of Figure 13, normalize the single array of pixel intensity values. For example, the normalization can be done by setting the highest pixel intensity value as 1, and the lowest pixel intensity value as 0. As an example, the normalized single array of pixel intensity values 1408 is illustrated in Figure 14.

[00182] In step 1314 of Figure 13, perform interpolation on the normalized single array of pixel intensity values to generate an interpolated array of pixel intensity values. Interpolation involves estimating the pixel intensity values of *n* subpixels from each pixel. As an example, the interpolated array of pixel intensity values 1410 is illustrated in Figure 14.

[00183] As a result of interpolation, the pixel intensity values of *n* subpixels can be derived from each pixel. For example, if *n* is 100, the (1x10) array will become an interpolated (1x1000) array (i.e. each of the 10 pixels will result in 100 subpixels). Any suitable interpolation technique known in the art can be used. Performing interpolation on the array of pixel intensity values is beneficial as this can increase the resolution and precision of the input data.

[00184] Though steps 1302 to 1314 have been described in numbered sequence, it would be apparent to the skilled person that variations to the numbered sequence is possible. For example, step 1312 (normalization) can be performed before step 1310 (reduction of array).

[00185] In step 1316 of Figure 13, feed the interpolated array of pixel intensity values 1410 into neural network model 1500.

[00186] As shown in Figure 15, neural network model 1500 can comprise input layer 1502, encoder layer 1504, decoder layer 1510 and output layer 1518. Neural network model 1500 can be a backpropagation based neural network.

[00187] Input layer 1502 can supply input data to encoder layer 1504. The input data can be the interpolated array of pixel intensity values 1410.

[00188] Encoder layer 1504 can comprise linear layer 1506 and rectifier activation function (henceforth referred to as ReLU layer) 1508. Linear layer 1506 and ReLU layer 1508 can be interleaved.

[00189] Linear layer 1506 can be configured to receive the input data from input layer 1502. The function of linear layer 1506 is to perform dimensionality reduction of the input data and reduce the variables in the input data. Linear layer 1506 can perform the dimensionality reduction of the input data by generating weights *W* and multiplying weights *W* to the input data to get an output.

[00190] Weights *W* can represent how important each subpixel in the interpolated array of pixel intensity values 1410 is in recreating the output. The larger the weight *W*, the more important that subpixel is to the training of neural network model 1500.

[00191] Weights *W* can be updated via backpropagation and can implement the function :

$$w_{new} = w_{old} - \eta . \nabla L$$

Wherein :
η = learning rate
∇L = loss gradient

[00192] The function of ReLU layer 1508 is to add non-linearity to the input data by adding sparsity to the input data which further encourages neural network model 1500 to identify only the most useful/key features (i.e. most important subpixels in the interpolated array of pixel intensity values 1410) without potentially relying on noise/false patterns in the input data. ReLU layer 1508 can implement the function *f(x) = max(0.x)*. An example of ReLU layer 1508 is shown in Figure 16.

[00193] Linear layer 1506 and ReLU layer 1508 of encoder layer 1504 thus work in concert to create a low-dimension representation of the input data and capture the key features of the input data. The output of encoder layer 1504 is referred to as encoder output. Encoder layer 1504 feeds the encoder output to decoder layer 1510.

[00194] Decoder layer 1510 comprises linear layer 1512, rectifier activation function (henceforth referred to as ReLU layer) 1514, and sigmoid layer 1516. Linear layer 1512, ReLU layer 1514 and sigmoid layer 1516 can be interleaved.

[00195] The output of decoder layer 1510 is referred to as decoder output. The function of linear layer 1512 is to receive the encoder output, and increase its data dimensionality so that the decoder output is the same size as the input data.

[00196] ReLU layer 1514 works in a similar way as ReLU layer 1508, and its function is to add non-linearity to the decoder output, which further encourages neural network model 1500 to recreate the input data.

[00197] Linear layer 1512 and ReLU layer 1514 of decoder layer 1510 thus work in concert to recreate the input data (i.e. make the decoder output resemble the input data). In other words, the objective of decoder layer 1510 is to recreate the interpolated array of pixel intensity values 1410 (i.e. the scan portions 650 of the training images 450) which had been fed into neural network model 1500.

[00198] The function of sigmoid layer 1516 is to normalize the decoder output. Sigmoid layer 1516 can implement the function :

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

An example of sigmoid layer 1516 is shown in Figure 17.

[00199] In embodiments, neural network model 1500 can comprise loss function 1802 as shown in Figure 18. Loss function 1802 can be applied just after decoder layer 1510, and can be configured to receive the decoder output. The purpose of loss function 1802 is to ascertain how well neural network model 1500 is performing in reconstructing the input data.

[00200] Loss function 1802 can apply both *MSE Loss* (Mean Squared Loss) and *Standard Deviation Loss Across Images*.

[00201] *MSE Loss* refers to the mean squared error of training images 450 and the reconstructed scan portions 650 taken across all training images 450.

[00202] *MSE Loss* can implement the function :

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Wherein :

n = number of interpolated array of pixel intensity values 1410

y_i = original values of interpolated array of pixel intensity values 1410

$\hat{y}_i$ = predicated values of interpolated array of pixel intensity values 1410

[00203] The purpose of *MSE Loss* is to evaluate how well neural network model 1500 learned to reconstruct scan portions 650 of training image 450 after passing the interpolated array of pixel intensity values 1410 through neural network model 1500.

[00204] The use of *MSE Loss* can help to ensure the decoder output is as close in similarity to the input data as possible and thus help to increase the accuracy of neural network model 1500.

[00205] The purpose of *Standard deviation loss across images* is to determine the standard deviation between all the reconstructed scan portions 650 of training images 450 after passing the interpolated array of pixel intensity values 1410 through neural network model 1500. In other words, the uncertainty of neural network model 1500 is evaluated. *Standard deviation loss across images* acts as an additional penalty (beyond that of *MSE Loss*) to account for variability in the decoder output.

[00206] *Standard deviation loss across images* can implement the function :

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{N}}$$

Wherein :

σ = standard deviation across interpolated array of pixel intensity values 1410

N = total number of interpolated array of pixel intensity values 1410

x_i = original value of interpolated array of pixel intensity values 1410

μ = mean of values of interpolated array of pixel intensity values 1410

[00207] The purpose of determining the standard deviation loss across the reconstructed scan portions 650 of training images 450 is to teach neural network model 1500 to prioritise important features in the input data, so that neural network model 1500 can learn how to

reconstruct scan portions 650 of training images 450 with as little variability as possible. In other words, to find the denoised and "true" edge 610, in light of the inherent noise in XY gantry 100. The benefit of determining the standard deviation loss across the reconstructed scan portions 650 of training images 450 is increased sharpness/contrast of the reconstructed scan portions 650 of training images 450.

[00208] Neural network model 1500 has been trained after steps 1304 to 1316 have been repeated for all training images 450.

[00209] In step 1318 of Figure 13, prepare a set of testing images for calibration dot 140. Similar to training images 450 described in step 1302, testing images can be the images taken by camera 120 of calibration dot 140. The set of testing images can comprise 4, 8, 16, or 32 testing images, or any reasonable number of testing images, so as to test neural network model 1500. The set of testing images can comprise less images than the set of training images 450.

[00210] In step 1320 of Figure 13, apply steps 1304 to 1316 to each testing image. For the sake of brevity, these steps would not be repeated in detail here. A skilled person would understand that the application of these steps to the set of testing images can result in testing scan portions for each testing image being prepared.

[00211] Testing scan portions can be similar to scan portions 650 and can be the portions of testing images which focuses on an edge 610 of calibration dot 140. Each testing scan portion can comprise multiple arrays or at least one array of pixels. Each testing scan portion can have scan position 640 (as shown in Figure 6). Scan position 640 can indicate the unique pixel position of testing scan portion. For example, if eight testing scan portions are prepared from one testing image, each of the eight testing scan portions will have a unique scan position 640.

[00212] Each of the testing scan portions can then go through the RGB conversion (step 1306), pixel intensity value conversion (step 1308), reduction of array (step 1310), normalization (step 1312) and interpolation (step 1314) steps to result in an interpolated array of pixel intensity values 1410. Each of the interpolated array of pixel intensity values 1410 can then be fed into neural network model 1500.

[00213] In step 1322 of Figure 13, after the interpolated array of pixel intensity values 1410 are fed into neural network model 1500, neural network model 1500 can output a modified interpolated array of pixel intensity values for each testing scan portion. Each modified

interpolated array of pixel intensity values can represent a modified edge of a testing scan portion.

[00214] In step 1324 of Figure 13, the testing scan portions that have the same scan position 640 can be grouped into a group.

[00215] In step 1326 of Figure 13, an ideal modified edge can be selected for each group by comparing the modified edges of the testing scan portions within the group. The purpose of this step is find the "best" modified edge for each group (and thereby each scan position 640) out of the set of testing images. With the completion of this step, an ideal modified edge would have been selected for each scan position 640.

[00216] The ideal modified edge for each group can be selected by determining the peak first derivative value of each of the modified edges of the testing scan portions within the group, comparing the peak first derivative values of the modified edges, and selecting as the ideal modified edge, the modified edge that exhibits the peak first derivative value which is median out of the peak first derivative values of the modified edges.

[00217] The modified edge that exhibits the median value can be chosen as the "best" modified edge because it is believed that mover 110 affected by micro-vibrations will oscillate or shake around a midpoint, where the midpoint of the oscillation can represent the "true" position of mover 110. By selecting the testing scan portion whose modified edge is median, one should get the testing image that is captured during the midpoint of the oscillation. That is, the testing image that is least affected by the micro-vibrations of mover 110 out of all the testing images.

[00218] In step 1328, the denoised dot center of calibration dot 140 can be determined from the ideal modified edges. The denoised dot center of calibration dot 140 can be determined from the ideal modified edges via any dot center determination algorithm known in the art e.g. the least-squares method.

[00219] As described in previous embodiments, a position adjustment function can then be calculated from the denoised dot center of calibration dot 140. The method can then be repeated for other calibration dots 140 in calibration jig 130 to determine the denoised dot centers of the other calibration dots 140 in calibration jig 130. The denoised dot centers of the other calibration dots 140 can then also be used to calculate the position adjustment function.

[00220] The position adjustment function can then be applied to the controller of XY gantry 100 to calibrate the controller of XY gantry 100.

[00221] Thus, neural network model 1500 is capable of determining the denoised dot center of calibration dot 140. The design of neural network model 1500 is geared towards the input data i.e. the interpolated array of pixel intensity values 1410. As this input data is one-dimensional in nature, neural network model 1500 advantageously only have linear layers, and does not have any convolutional layers. Thus, neural network model 1500 is straightforward and has fast processing speed.

[00222] This is in contrast to other models that may require convolutional layers to handle two-dimensional image data, to for example, capture more complex information like shapes and textures. Instead, as the input data is one-dimensional image data, the linear architecture of neural network model 1500 is more than capable of optimizing the image data, and determining denoised dot centers of the calibration dots 140.

[00223] Furthermore, neural network model 1500 also prioritizes efficiency. As calibration of the controller of XY gantry 100 usually has to be done on-site, the computation of the denoised dot centers of the calibration dots 140 have to be done quickly and efficiently. As neural network model 1500 is based on an autoencoder architecture which invokes a reduction in dimensionality, the linear layers of neural network model 1500 are very efficient, and computation of the denoised dot centers of calibration dots 140 can be performed close to real-time.

[00224] Even further still, autoencoders are unsupervised models. As such, no labelled data is required. This means that neural network model 1500 can learn purely on features it can detect from the input data which increases its robustness to noise. This is extremely useful in computation of the denoised dot centers of the calibration dots 140 as a key objective is to "preserve the edge" of the calibration dots 140 despite the inherent vibration in the XY gantry 100 when camera 120 is taking training images 450 and the testing images.

[00225] Autoencoders are inherently supposed to recreate the input data despite the encoding, the decoding, and the inherent vibration present in training images 450. Autoencoders are supposed to identify all the key features of the input data despite the noise present which is highly appropriate in this use case.

[00226] Therefore, due to its autoencoder design, neural network model 1500 is ideal in accounting for the micro-vibrations of XY gantry 100, and determining the denoised dot centers of calibration dots 140. The accuracy of neural network model 1500 is further enhanced with the inclusion of loss function 1802, where the application of both *MSE Loss* and *Standard Deviation Loss Across Images* allows neural network model 1500 to reconstruct scan portions 650 of training images 450, and the testing scan portions of the testing images, with as little variability as possible, despite the inherent noise in XY gantry 100.

[00227] The present methodologies described herein are fully intended to be operable on a variety of computer systems. Any hardware platform suitable for performing the methodologies described here is suitable for use with the invention. The program code required for a computer processor to execute instructions to perform the present methodologies described herein can be stored in any suitable computer-readable storage media.

[00228] Computer-readable storage media refer to any medium or media that participate in providing instructions to a CPU for execution. Such media can take many forms, such as but not limited to, non-volatile and volatile media including optical or magnetic disks and dynamic memory, respectively. Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, a hard disk, magnetic tape, any other magnetic medium, a CD-ROM disk, digital video disk (DVD), any other optical medium, RAM, PROM, EPROM, a FLASHEPROM, any other memory chip or cartridge.

[00229] Various forms of transmission media may be involved in carrying one or more sequences of one or more instructions to a CPU for execution. A bus carries the data to system RAM, from which a CPU retrieves and executes the instructions. The instructions received by system RAM can optionally be stored on a fixed disk either before or after execution by a CPU.

[00230] Unless specifically stated otherwise, and as apparent from the following, it will be appreciated that throughout the present specification, discussions utilizing terms such as “applying”, “identifying”, ‘converting”, “tagging”, “feeding”, “grouping”, “incrementing”, “determining”, “assigning”, “approving”, “selecting”, “sending”, “calculating”, “determining”, “replacing”, “generating”, “initializing”, “outputting”, “comparing”, “performing”, “shifting”, “obtaining”, “dividing”, “plugging”, or the like, refer to the action and processes of a computer system, or similar electronic device, that manipulates and transforms data represented as physical quantities within the computer system into other data similarly represented as physical

quantities within the computer system or other information storage, transmission or display devices.

[00231] In the application, unless specified otherwise, the terms "comprising", "comprise", and grammatical variants thereof, intended to represent "open" or "inclusive" language such that they include recited elements but also permit inclusion of additional, non-explicitly recited elements.

[00232] It will be apparent that various other modifications and adaptations of the application will be apparent to the person skilled in the art after reading the foregoing disclosure without departing from the spirit and scope of the application and it is intended that all such modifications and adaptations come within the scope of the appended claims.

CLAIMS

1. A method for calibrating a controller of a XY gantry, the XY gantry comprising a frame and a mover configured to move in a X axis direction and a Y axis direction within the frame, the mover comprising a camera centrally located on the mover, the method comprising the steps of:

aligning a calibration jig with the XY gantry, the calibration jig comprising at least one calibration dot;

positioning the mover over the at least one calibration dot;

capturing with the camera a plurality of training images of the at least one calibration dot;

selecting a model image from the plurality of training images;

applying a set of scans to the model image to determine a plurality of model scan portions, the plurality of model scan portions focusing on different edges of the at least one calibration dot, each model scan portion comprising at least one array of pixels;

for each model scan portion :

determining a model subpixel edge position, the model subpixel edge position representing a subpixel position in the model scan portion where there is a significant change in pixel intensity; and

determining a model subpixel edge gradient from the model subpixel edge position, the model subpixel edge gradient representing the change in pixel intensity with respect to subpixel position;

capturing with the camera a test image of the calibration dot;

applying the set of scans to the test image to determine a plurality of test scan portions, the plurality of test scan portions focusing on different edges of the at least one calibration dot, each test scan portion comprising at least one array of pixels;

for each test scan portion :

determining a test subpixel edge position, the test subpixel edge position representing a subpixel position in the test scan portion where there is a significant change in pixel intensity;

determining a test subpixel edge gradient from the test subpixel edge position, the test subpixel edge gradient representing the change in pixel intensity with respect to subpixel position;

identifying the model scan portion that focuses on the same edge as the test scan portion;

shifting the model subpixel edge gradient of the identified model scan portion by at least one subpixel towards the test subpixel edge gradient to obtain a corrected model subpixel edge gradient, wherein the corrected model subpixel edge gradient represents the shifted model subpixel edge gradient that exhibits the highest normalized cross-correlation coefficient (NCC) with the test subpixel edge gradient; and

obtaining a denoised subpixel edge position based on the corrected model subpixel edge gradient;

determining a denoised dot center of the at least one calibration dot based on the denoised subpixel edge positions of the plurality of test scan portions;

calculating a position adjustment function from the denoised dot center of the at least one calibration dot; and

applying the position adjustment function to the controller of the XY gantry.

2. The method of claim 1, wherein the step of selecting a model image from the plurality of training images comprises the steps of :

for each training image,

applying the set of scans to determine a plurality of scan portions, the plurality of scan portions focusing on different edges of the at least one calibration dot, each scan portion comprising at least one array of pixels;

for each scan portion, determining a subpixel edge position, the subpixel edge position representing a subpixel position in the scan portion where there is a significant change in pixel intensity;

determining a dot center position based on the subpixel edge positions of the plurality of scan portions; and

selecting a training image from the plurality of training images as the model image, the selected training image having a dot center position which is median out of the dot center positions of the plurality of training images.

3. The method of claim 2, wherein the step of selecting a model image from the plurality of training images further comprises the steps of :

for each scan portion in the plurality of scan portions of the selected training image,

determining a subpixel edge projection from the subpixel edge position, the subpixel edge projection representing the pixel intensity profile of the scan portion; and

determining a NCC of the subpixel edge projection with an ideal edge projection;

determining a mean NCC from the NCC of the plurality of scan portions; and

if the mean NCC of the selected training image is below a threshold, discarding the selected training image and selecting another training image from the remaining training images as the model image, the another selected training image having a dot center position which is median out of the dot center positions of the remaining training images.

4. The method of claim 3, wherein the threshold is at least 0.5.

5. The method of claim 3 or claim 4, wherein the mean NCC is determined from the sum of the NCC of the plurality of scan portions divided by the total number of scan portions of the selected training image.

6. The method of any one of claims 2 to 5, wherein the ideal edge projection is derived from an ideal image of the at least one calibration dot; and wherein the ideal image exhibits no grey pixels at the edges of the at least one calibration dot.

7. The method of any one of claims 1 to 6, wherein the step of determining a model subpixel edge position comprises the steps of:

converting each pixel in the at least one array of pixels to RGB values to obtain at least one array of RGB values;

determining a normalized array of pixel intensity values from the at least one array of RGB values;

determining an edge intensity projection from the normalized array of pixel intensity values;

applying the first derivative on the edge intensity projection to obtain an edge gradient; and

determining the model subpixel edge position from the edge gradient.

8. The method of claim 7, wherein the step of determining a model subpixel edge gradient from the model subpixel edge position comprises the steps of:

identifying a set of pixel positions proximate to the model subpixel edge position;

dividing the set of pixel positions into subpixel positions;

plugging the subpixel positions into the edge gradient to obtain an array of subpixel gradient values of the model scan portion; and

determining the model subpixel edge gradient based on the array of subpixel gradient values of the model scan portion.

9. The method of any one of claims 1 to 6, wherein the step of determining a test subpixel edge position comprises the steps of:

converting each pixel in the at least one array of pixels to RGB values to obtain at least one array of RGB values;

determining a normalized array of pixel intensity values from the at least one array of RGB values;

determining an edge intensity projection from the normalized array of pixel intensity values;

applying the first derivative on edge intensity projection to provide an edge gradient; and

determining the test subpixel edge position from the edge gradient.

10. The method of claim 9, wherein the step of determining a test subpixel edge gradient from the test subpixel edge position comprises the steps of:

identifying a set of pixel positions proximate to the test subpixel edge position;

dividing the set of pixel positions into subpixel positions;

plugging the subpixel positions into the edge gradient to obtain an array of subpixel gradient values of the test scan portion; and

determining the test subpixel edge gradient based on the array of subpixel gradient values of the test scan portion.

11. The method of any one of claims 2 to 6, wherein the step of determining a subpixel edge position comprises the steps of:

converting each pixel in the at least one array of pixels to RGB values to obtain at least one array of RGB values;

determining a normalized array of pixel intensity values from the at least one array of RGB values;

determining an edge intensity projection from the normalized array of pixel intensity values;

applying the first derivative on the edge intensity projection to provide an edge gradient; and

determining the subpixel edge position from the edge gradient.

12. The method of claim 11, wherein the step of determining a subpixel edge projection from the subpixel edge position comprises the steps of:

identifying a set of pixel positions proximate to the subpixel edge position;

dividing the set of pixel positions into subpixel positions;

plugging the subpixel positions into the edge intensity projection to obtain an array of subpixel intensity values; and

determining the subpixel edge projection based on the array of subpixel intensity values.

13. The method of any one of claims 8 to 12, wherein the step of identifying a set of pixel positions proximate to the model subpixel edge position, the test subpixel edge position or the subpixel edge position comprises:

identifying 4 pixel positions immediately preceding the model subpixel edge position, the test subpixel edge position or the subpixel edge position; and

identifying 4 pixel positions immediately following the model subpixel edge position, the test subpixel edge position or the subpixel edge position.

14. The method of any one of the preceding claims, wherein the plurality of training images comprises 32 training images.

15. A method for calibrating a controller of a XY gantry, the XY gantry comprising a frame and a mover configured to move in a X axis direction and a Y axis direction within the frame, the mover comprising a camera centrally located on the mover, the method comprising the steps of:

aligning a calibration jig with the XY gantry, the calibration jig comprising at least one calibration dot;

positioning the mover over the at least one calibration dot;

capturing with the camera a set of training images of the at least one calibration dot;

applying a set of scans to each training image in the set of training images to obtain a

plurality of scan portions, the plurality of scan portions focusing on different edges of the at least one calibration dot;

for each scan portion in the plurality of scan portions:

converting the scan portion into a first interpolated array of pixel intensity values; and

feeding as input data, the first interpolated array of pixel intensity values into a neural network model, the neural network model comprising an encoder layer and a decoder layer, the encoder layer configured to create a low-dimension representation of the input data, and the decoder layer configured to recreate the input data;

capturing with the camera a set of testing images of the at least one calibration dot;

applying the set of scans to each testing image in the set of testing images to obtain a plurality of testing scan portions, each testing scan portion in the plurality of testing scan portions having a scan position;

for each testing scan portion:

converting the testing scan portion into a second interpolated array of pixel intensity values; and

feeding as input data, the second interpolated array of pixel intensity values into the neural network model to output a modified interpolated array of pixel intensity values, the modified interpolated array of pixel intensity values representing a modified edge of the testing scan portion;

grouping the testing scan portions that have the same scan position into a group;

selecting an ideal modified edge for each group by comparing the modified edges of the testing scan portions within the group;

determining a denoised dot center of the at least one calibration dot from the ideal modified edges;

calculating a position adjustment function from the denoised dot center of the at least one calibration dot; and

applying the position adjustment function to the controller of the XY gantry.

16. The method of claim 15 wherein the neural network model further comprises a loss function; and wherein the loss function implements *MSE Loss* and *Standard deviation Loss Across Images* and is configured to help the decoder layer recreate the input data with as little variability as possible.

17. The method of claim 15 or claim 16, wherein the encoder layer comprises interleaving *linear* layers and *ReLU* layers.

18. The method of any one of claims 15 to 17, wherein the decoder layer comprises interleaving *linear* layers, *ReLU* layers and *sigmoid* layers.

19. The method of any one of claims 15 to 18, wherein the step of selecting an ideal modified edge for each group by comparing the modified edges of the testing scan portions within the group comprises the steps of:

determining a peak first derivative value of each of the modified edges of the testing scan portions within the group;

comparing the peak first derivative values of the modified edges; and

selecting as the ideal modified edge, the modified edge that exhibits the peak first derivative value which is median out of the peak first derivative values.

20. The method of any one of claims 15 to 19, wherein each scan portion comprises at least one array of pixels; and the step of converting the training scan portion into a first interpolated array of pixel intensity values comprises the steps of:

converting each pixel in the at least one array of pixels to RGB values to obtain at least one array of RGB values;

determining a normalized array of pixel intensity values from the at least one array of RGB values; and

performing interpolation on the normalized array of pixel intensity values to generate the first interpolated array of pixel intensity values.

21. The method of any one of claims 15 to 19, wherein each testing scan portion comprises at least one array of pixels; and the step of converting the testing scan portion into a second interpolated array of pixel intensity values comprises the steps of:

converting each pixel in the at least one array of pixels to RGB values to obtain at least one array of RGB values;

determining a normalized array of pixel intensity values from the at least one array of RGB values; and

performing interpolation on the normalized array of pixel intensity values to generate the second interpolated array of pixel intensity values.

22. The method of any one of claims 15 to 21, wherein the set of training images comprises more images than the set of testing images.

23. The method of any one of claims 1 to 22 wherein the at least one calibration dot is a plurality of calibration dots.