Engineering Research Express

PAPER

X-ray computed tomography with seven degree of freedom robotic sample holder

Erdal Pekel[1,2], **Florian Schaff**[2,3], **Martin Dierolf**[2,3], **Franz Pfeiffer**[2,3,4] and **Tobias Lasser**[1,2]

[1] Computational Imaging and Inverse Problems, Department of Informatics, School of Computation, Information, and Technology, Technical University of Munich, Germany
[2] Munich Institute of Biomedical Engineering, Technical University of Munich, Germany
[3] Chair of Biomedical Physics, Department of Physics, School of Natural Sciences, Technical University of Munich, Germany
[4] Department of Diagnostic and Interventional Radiology, School of Medicine, and Klinikum rechts der Isar, Technical University of Munich, Germany

E-mail: erdal.pekel@tum.de

Keywords: x-ray computed tomography, robotic arm, sample holder, geometric calibration

OPEN ACCESS

RECEIVED
21 March 2022

REVISED
7 July 2022

ACCEPTED FOR PUBLICATION
18 July 2022

PUBLISHED
5 August 2022

Abstract

We present an x-ray Computed Tomography setup that integrates a seven degrees of freedom robotic arm as a sample holder within an existing laboratory x-ray computed tomography setup. We aim to provide a flexible sample holder that is able to execute non-standard and task-specific trajectories for complex samples. The robotic arm is integrated with a unified software package that allows for path planning, collision detection, geometric calibration and reconstruction of the sample. The calibration is necessary to identify the accurate pose of the sample which deviates from the expected pose due to inaccurate placement of the robotic arm. With our software the user is able to command the robotic arm to execute arbitrary trajectories for a given sample in a safe manner and output its reconstruction to the user. We present experimental results with a circular trajectory where the robotic sample holder achieves identical visual quality compared to a conventional sample holder.

1. Introduction

In this work we introduce a flexible robotic arm with seven degrees of freedom as a sample holder within a laboratory x-ray Computed Tomography (CT) setup. The arm adds flexibility to the setup as a sample holder by enabling arbitrary rotation and placement of the sample. This allows non-standard trajectories that are not restricted in their sequence, such as conventional circular or helical trajectories. In addition, the robotic sample holder can avoid occlusions on the projections that would normally be introduced by limitations of static setups where the sample is inherently mounted to non-moving parts (e.g. mounted on a plate). In the following we present our work on the integration of a robotic arm with seven degrees of freedom within a lab x-ray CT setup together with a suitable calibration mechanism. The calibration mechanism is required as a result of the insufficient placement accuracy of the robotic arm. A purpose-built sample holder with an embedded geometric structure is used to calibrate the position and orientation of the sample for later use in the reconstruction step.

The system can easily execute specific trajectories that can overcome the limitations of fixed trajectories. Arbitrary rotations can be reached with the robotic arm's seven degrees of freedom (seven rotational joints). This will enable imaging modalities that require non-standard acquisition sequences in the future, such as Anisotropic x-ray Dark-field Tomography (AXDT). AXDT is a novel imaging technique that allows the extraction of x-ray scattering and phase contrast information by employing grating interferometers [1, 2]. The robotic sample holder will enable arbitrary rotations covering the full sphere and hence expose the 3D structures of the target object by measuring the full dark-field contrast from all possible angles. Conventional sample holders pose a significant challenge when the acquisition trajectory is required to cover the full sphere of rotations, as more intervention by the user is required and it may not be possible to measure the sample from certain angles.

In the following we will provide an overview of related work on imaging with robotic arms and geometric calibration mechanisms for x-ray CT.

1.1. Related work on x-ray CT with robotic arms

Robotic arms were also used in the past in computed tomography systems [3–5]. The main difference to our work is the kind of robotic arm that is used. It offers a higher flexibility than the robotic arms that were used in related work due to its seven degrees of freedom and it has two fingers that make chained pick-and-place tasks possible without user intervention.

In [3] the source and detector are mounted on robotic arms and the sample is centered between the source and the detector by the arms The main difference to our work is that this is not a laboratory scale setup but assembly line scale and that the sample does not move but the source and detector. Therefore it is not directly comparable to our setup. The detector pixel size (100μm theirs *versus* 150 μm ours) and the voxel resolution of the reconstructions (70 μm *versus* 100 μm) is very similar to our work. The authors do not specify the exact models of the robotic arms, but from the figure it can be seen that they have four degrees of freedom (compared to seven with our robotic arm). The reduced degrees of freedom count means the robot is less flexible and it has difficulties reaching certain acquisition angles.

In [4] the authors demonstrate the advantage of non-circular CT scanning trajectories. The experiments are conducted in a simulation using a 3D model of the specimen. With a circular scanning trajectory the specimen absorbs x-rays when spheres are added to it and reconstruction quality is impacted resulting in streak artifacts. It is demonstrated that by using a simulated six DoF robotic arm a simulated non-circular trajectory that almost covers the full rotational sphere would be possible and result in a reconstruction with no artifacts.

The Siemens Healthineers *Artis zeego eco* angiography platform [5] consists of a single five DoF robotic arm which is moving the detector and source with a fixed distance between each other. The main differences to our work are that the detector and source are both mounted to the robotic arm and hence are moving parts. Also the sample in this case is a living patient. The robotic arm is positioned such that the body part of the patient which is of interest lies exactly between the source and detector.

In [6] the authors present an x-ray tomography system with two robotic arms. The x-ray source and the detector can be moved independently from each other by the two arms and the sample is mounted statically between them. There are two key differences to the system that we propose. The first is that we are moving the sample while the authors in [6] propose a system that moves the source and detector around the sample. This means that our system only requires one robotic arm instead of two. Furthermore, moving the sample instead of the x-ray source means that the system is not restricted to movable x-ray sources and detectors and hence it is more flexible. On the other hand, this means that samples in our system are more restricted in terms of size and weight, and the sample should not be deforming when being moved around. The second key difference is that our robotic arm is significantly smaller and thus fits into an existing x-ray CT setup, while the robotic arms used in [6] can reach up to 3 meters of height (compared to 1.2 meters with our arm), which might not even fit into an existing laboratory [7, 8]. Furthermore, the smaller robotic arm in our system is more affordable in comparison.

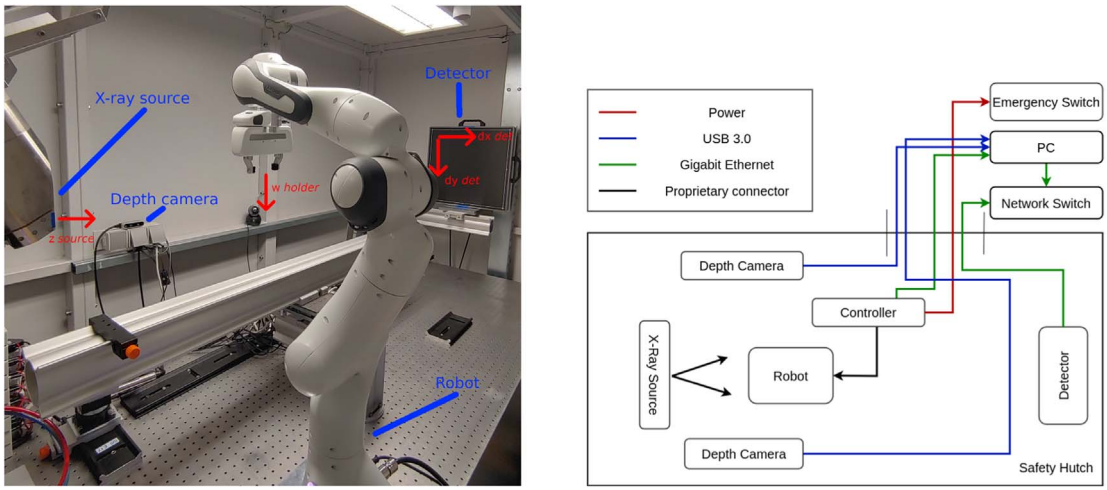
1.2. Related work on geometric calibration in x-ray CT

The seven joints of our robotic arm imply seven degrees of freedom but a higher number of joints also introduces a higher error on the placement of the sample. A calibration mechanism is needed for determining the projection parameters of the sample. The projection parameters can be split into the external and internal camera parameters. The external parameters are the rotation and position of the camera relative to the sample (or vice versa). The internal parameters concern the camera system itself. In our imaging system the internal parameters are fixed and they are determined beforehand.

In [9] the authors propose a generic calibration method for tomographic imaging systems with flat-panel detectors. A flat calibration phantom with 44 spheres in total on two parallel planes is used for calibration. Both sets of camera parameters (internal and external) are extracted from the images in a direct computation step.

In [10] the authors propose a calibration method based on a cylindrical calibration phantom similar to what we will use here. At least two sets of spheres in a circular arrangement are needed in order for this approach to work because this allows the extraction of the center of the calibration phantom's coordinate system. The geometric parameters are computed directly with a closed analytical expression for each image. The parametrization of the geometry only permits rotational acquisition trajectories, whereas our calibration method works with arbitrary placements of the sample and hence any kind of trajectory.

In [11] the authors propose a method similar to [10]. A calibration phantom with two sets of spheres that are arranged in an ellipse is used. The difference to [10] is that the geometric parameters are not calculated directly but an optimization step is introduced for computing the rotation parameters. Additionally, this method is valid



(a) Lab photo (b) Hardware schematics

Figure 1. Hardware setup. In figure 1(a) the robotic arm is mounted on a table with the source and the detector inside a safety hutch. The source to robot distance is 40 cm and robot to detector distance is 176 cm. Two depth cameras monitor the movement of the robot and send a stop signal to the robot controller when the executed trajectory interferes with obstacles. The robotic arm can also be stopped by a manual power switch that was routed to the operator table outside of the hutch (see figure 1(b)). The relevant coordinate systems are visualized in red in figure 1(a). The x and *y*-axis are determined by the right-hand rule.

for arbitrary geometries, not just rotational trajectories compared to [10]. The main difference to our method is that two ellipses are used for calibration instead of a helix.

In summary when compared to [9] our method offers more flexibility in terms of the constraints on the calibration phantom because we do not restrict the arrangement of the markers to obtain a certain 3D coverage. When compared to [10] and [11] our approach is much simpler in terms of the analytical expressions that are needed and it is decoupled from the specific geometrical structure on the calibration phantom. And in general, our approach is more suitable for robotic arms with higher degrees of freedom because it is valid for arbitrary geometries as opposed to the methods in [9] and [10] which are restricted to rotational acquisition trajectories.

2. Methods

In this section the methods for operating the robotic arm as a sample holder in a lab x-ray CT setup are discussed in detail. After introducing the hardware components and software architecture of the system more specific parts like path planning, collision detection, calibration and reconstruction are described.

2.1. Hardware setup

The hardware components of the system are displayed in figure 1 . The main difference to a conventional x-ray CT setup is the seven degrees of freedom robotic arm *Panda* from the manufacturer FRANKA EMIKA [8]. It has a maximum reach of 855 mm and a repeatability of 0.1 mm when repeatedly moved from a specific starting pose to a goal pose. It has two fingers that can move on a fixed axis and grasp objects. The maximum allowed payload is 3 kg. The robotic arm and the depth cameras are connected directly to a computer while the detector is accessible through a network interface. The robotic arm can be turned off in case of emergency from outside of the safety hutch with a power switch (see figure 1(b)).

Two *Intel Realsense D435* depth camera capture the movements of the robot and provide 3D information about the surroundings as a point cloud. The cameras are connected directly to the workstation and they are used for the collision detection mechanism described in section 2.5.

The robotic arm is mounted on a table inside a safety hutch for x-ray CT which houses the x-ray source and the detector (see figure 1(a)). The detector has a maximum resolution of 2880 $\times$ 2880 and is connected to a different workstation on the network which provides a network interface for triggering image capturing. Our workstation retrieves the raw 16-bit grayscale image over the network.

2.2. System architecture

In figure 2 a logical overview of the different system components is displayed. The pipeline consists of three main steps: sample acquisition, calibration and reconstruction. The robotic arm only needs to be active during sample acquisition; the remaining steps can also be executed in a different environment, for example on a server

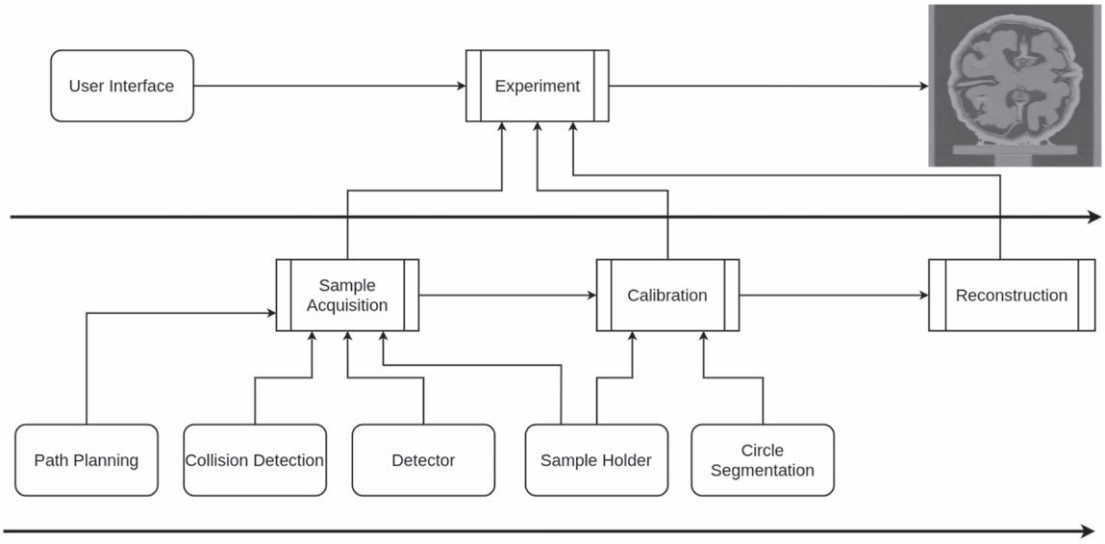


Figure 2. System Architecture. The user interacts with the system with a web browser using a web user interface. All three tasks (sample acquisition, calibration and reconstruction) can be executed independently. The preconditions for running a task are represented by arrows, e.g. the calibration can only be run when the sample acquisition is finished, information about the given sample holder is available and the circles were registered on the detector images. The final output of running all steps of the pipeline is a 3D reconstruction volume which is saved on the filesystem.

computer. This decoupling is achieved by saving intermediate results like images or sensor data to the filesystem and reading the data for the calibration and reconstruction independently. The components that are necessary for executing the steps in the pipeline that they are connected to are on the bottom of the drawing. For example, for sample acquisition, path planning, collision detection, and the detector interface need to be active, and the geometry of the sample holder known for execution.

The user interface is decoupled from the rest of the system into a web-page that runs on any modern device with a web-browser. The communication between the user interface and the system is carried out with predefined messages protocols. Intermediate results of the experiment like the currently acquired detector image are displayed on the interface.

2.3. Sample holder

The sample holder is a critical component of the system as it allows the robotic arm to grasp samples of arbitrary shape and is a fundamental part of the calibration process where the position and orientation of the sample is identified. The 3D models of the sample holder and the rail component are visualized in figure 3. The sample holder consists of two parts. The bottom part is where the robot's fingers can grasp the holder steadily. The upper part fulfills the actual purpose of placing a geometric structure around the sample on a cylinder. Prior to attaching the sample holder to the robotic arm's fingers the sample needs to be glued to the mounting plate which is inserted into the cylinder from the top at the intended position. There is no need to screw the mounting plate, as there is enough friction with the cylinder to hold the plate in place (see figure 3).

The cylinder is 5.6 cm tall and 3.5 cm in diameter inside. The sample holder was designed with a 3D modeling software and printed using a 3D printer with accuracy of 0.08 to 0.2 mm on all three axes. The printing accuracy is important as the local coordinates of the spheres in the 3D model are used as reference points in the calibration algorithm.

The geometric structure embedded in the sample holder is a helix which is made up of 50 embedded aluminium spheres of 0.678 mm diameter. These spheres were fixed by hand on notches that were included in the design process of the holder. The spheres appear as circles on the detector images that will be segmented during calibration.

The helix can be parametrized by the following 3D parametric curve:

$$h(\tau) = \begin{pmatrix} u(\tau) \\ v(\tau) \\ w(\tau) \end{pmatrix} = \begin{pmatrix} r * \cos(\rho * \tau + \phi) \\ r * \sin(\rho * \tau + \phi) \\ \tau \end{pmatrix} \quad \tau, r, \rho, \phi \in \mathbb{R} \tag{1}$$

τ runs between the local w coordinates of the first sphere and the last sphere of the helix: $w_{\min} < \tau < w_{\max}$ where $w_{\min}, w_{\max} \in \mathbb{R}$.

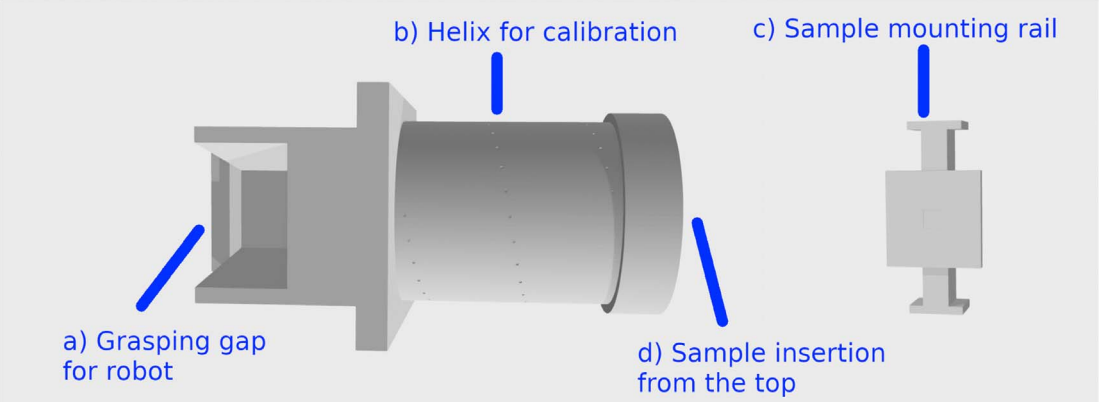


Figure 3. Sample holder. (a) The robot can grasp the sample holder with its two fingers by sliding into a conically shaped gap for easier engagement of the fingers. (b) The middle part of the holder houses a helix structure that is used in the calibration step. The aluminium spheres are glued into holes and can then be segmented in the acquired images. (c) The sample is fixed on the mounting rail which is inserted from the top of the holder into the hollow cylindrical structure (d).

The parameters r (radius), ρ (frequency) and ϕ (phase shift) parametrize the helix. They can be determined by fitting the sphere coordinates from the 3D model of the sample holder to equation (1) with a least-squares term. The source code of this process can be found in the file *helix_fitter.py* in our repository [12].

The helix can be discretized by choosing a fixed number $H \in \mathbb{N}$ of points $\{\tau_i\}_{i=1,\dots,H} \in [w_{\min}, w_{\max}]$ for the free parameter τ:

$$h_i = (u(\tau_i) \quad v(\tau_i) \quad w(\tau_i))^\top. \tag{2}$$

2.4. Path planning

In this section, we are going to outline the important aspects of controlling the robot for use as a sample holder in x-ray CT setups. The main component is task planning which includes all steps that are necessary to place the sample at the correct place.

Task planning consists of multiple steps. The first step is determining the acquisition trajectory for the given tomographic task. The acquisition trajectory consists of a set of N poses (positions and orientations) for the center of the sample holder that is currently attached to the robot. The acquisition poses are expected to lie on the intersection of the central ray of the x-ray source with the vertical operating plane of the robotic arm in order to fit the sample (holder) in the limited field-of-view of the imaging system. The poses are targeted at the center of the sample holder because our goal is to image the sample that is contained inside the sample holder. These poses can be generated from the user interface of our software package.

In the second step of task planning information on which particular link of the robotic arm should reach the given pose is added. The kinematic representation (a chain) of the robotic arm is extended by a virtual link that starts at the arm's last link and points to the center of the current sample holder. The path planning algorithm can now place the tail of this virtual link at the goal position. This will ensure that the goal pose, which serves as input to the next step, is exactly at the center of the current sample holder. Without this modification to the kinematic chain instead of the sample holder's center the robotic arm's last link would be placed at the goal pose.

In the third and last step, the inverse kinematics for the given poses are calculated. It calculates the angles at the joints that are required to reach the given goal pose and calculates a series of angles from the given starting position to reach the goal. The resulting inverse kinematics is a series of angles and timestamps (also called *trajectory* in the robotics literature, different than the *acquisition trajectory* from step one) where the first set of angles matches the current state of the robot.

We swapped the default inverse kinematics backend for this task in the *franka_ros* package provided by the manufacturer with the *TRAC-IK* library which has a higher solving rate and a shorter runtime for inverse kinematics tasks on robotic arms with high degrees of freedom like ours [13].

2.5. Collision detection

The inverse kinematics trajectories from section 2.4 are executed one after another with the robot by sending the individual joint angles to the controller in figure 1(b). This is a dangerous task as the robot could crash with its surroundings during trajectory execution. For this reason two collision detection mechanisms are employed: passive and active.

2.5.1. Passive collision detection

The passive collision detection reads user-defined configuration files containing information about nearby objects from the filesystem at system startup and forwards them to the manipulation framework. The framework passes these objects on to the inverse kinematics such that they are considered as obstacles in the *configuration space* when planning the trajectories.

2.5.2. Active collision detection

The active collision detection is a process that retrieves the currently executed trajectory and a point cloud from the depth cameras (see figure 1(b)). It then checks if there are potential collisions in the retrieved point cloud with the current trajectory of the robotic arm. If collisions are detected, the arm is stopped immediately.

A similar collision detection mechanism for robotic arms was already implemented in [14]. While the concepts are similar to our implementation we decided to implement our own algorithm for easier integration with our robotic arm and our software package.

The process that handles the collision detection loads the 3D structure of the robotic arm's links from the filesystem which will be used together with the current joint angles of the arm to calculate the approximate position of the robotic arm's links.

The active collision detection mechanism is triggered when a trajectory is received and the robotic arm starts to move. The trajectory consists of a set of joint angles for each of the seven joints of the robotic arm and is transferred to a GPU together with the point cloud from the depth cameras and the 3D structure information of the robot's links.

All points from the point cloud that resemble the robotic arm are removed otherwise the robot itself would be registered as a colliding object. This is done by a self filter which calculates the distance of each point in the point cloud to individual points on each link of the robotic arm and removes all points from the point cloud that fall below a certain threshold. Afterwards the arm's movement in 3D space is calculated while executing the current trajectory. This movement profile is compared to the current point cloud input from the depth cameras and checked for collisions.

In an additional verification step the algorithm checks if the reported collision points from the point cloud are not the result of a noisy measurement by checking that the 27 neighbours in discretized 3D space were also reported as collision points. Finally, the points that fulfill this criterion are reported as actual collision points.

If there are collision points after the noise filtering step the trajectory execution by the robotic arm is stopped.

The source code can be found in the files *CollisionDetector.cpp*, *detection.cl* and *verification.cl* in our repository [12].

2.6. Calibration

The calibration procedure tackles the issue that the robotic arm does not sufficiently accurately place the sample at the desired position due to inaccurate path planning and inaccurate electrical motors at its joints. Reading the sensors of the robotic arm and deducing the samples current position is also insufficient to determine the correct position as inaccurate values are reported. However the exact position of the sample at each view is required for the reconstruction. With the calibration procedure we are able to identify the actual positions and orientations of the sample for the reconstruction step. For the calibration a sample holder with an embedded geometric structure that can be detected on the detector images is necessary. A suitable sample holder was introduced in section 2.3.

The calibration is implemented in multiple steps (see figure 4). The first step is the post-processing of the detector image. Its contrast is enhanced and a median filter with kernel size 5 is applied to reduce noise and improve the segmentation results. The calibration circles on the image are detected in the next step with the circle Hough transform algorithm [15]. The result is a set of 2D circle center coordinates $\hat{m}_j = \begin{pmatrix} d_{x,j} & d_{y,j} \end{pmatrix}^\top$ on the detector.

Equation (1) and the current position of the robotic arm are now used to project a set of helix points h_i (equation (2)) onto the detector image for comparison with the segmented points $\hat{m}_j$ and determining the geometry of the sample.

For this projection the intrinsic camera matrix K and the external parameters R and t are needed. K is fixed for the current x-ray CT setup and R, t are determined by the robotic arm's current position.

In the following we will provide an overview of the equations that will lead to the final least-squares term that includes the aforementioned comparison algorithm. This least-squares term is used for determining the actual pose of the sample by utilizing an optimisation algorithm.

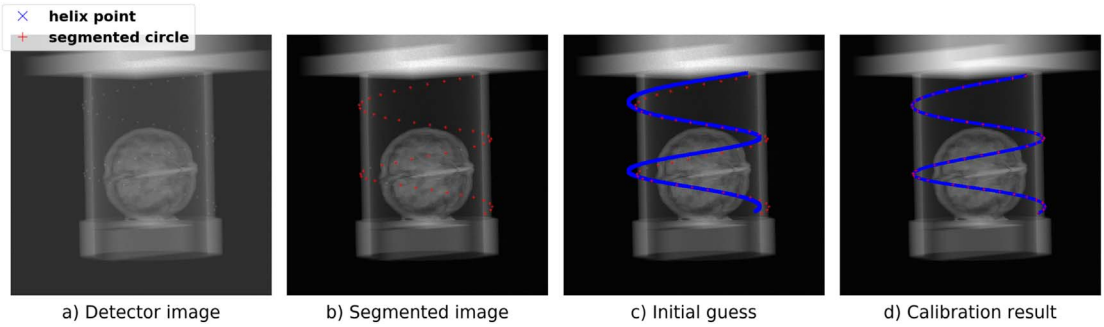


Figure 4. Calibration procedure. In (a) the flat-field corrected detector image is displayed. This image is contrast-enhanced and subsequently a circle detection algorithm is executed. The resulting image where the detected circle centers are marked with red crosses is displayed in (b). Given the geometry of the sample holder and the robot's sensor readings when acquiring the image, an initial guess of the helix location (blue crosses) is projected onto the image plane (c). The parameters that define the rotation and translation of the helix are optimized in a least-squares problem in the 2D image domain. The resulting parameters are used to project the helix again to the image domain to display the final outcome of the calibration (d).

There are three critical coordinate systems in our setup (see figure 1(a)). The first is fixed to the x-ray source with x, y and z-axis. The second is fixed to the center of the sample holder with u,v and w-axis and moves with the robotic arm as it is attached to the arm's fingers. The third is fixed to the detector with d_x and d_y axis.

The rotation R of the sample holder relative to the source can be parametrized w.l.o.g. by consecutive rotations about the z, y and x-axis:

$$R(\alpha,\ \beta,\ \gamma) = R_z(\alpha)R_y(\beta)R_x(\gamma) \tag{3}$$

t is the offset of the source center to the sample holder's center:

$$t = (x \quad y \quad z) \tag{4}$$

K is fixed and can be calculated with the parameters sdd (source to detector distance), $d_{x,p}$, $d_{y,p}$ (principal points on d_x and d_y-axis) and d_w, d_h (detector pixel width and height):

$$K = \begin{pmatrix} \frac{sdd}{d_w} & 0 & d_{x,p} \\ 0 & \frac{sdd}{d_h} & d_{y,p} \\ 0 & 0 & 1 \end{pmatrix} \tag{5}$$

These parameters are fixed for the current setup and can be determined beforehand.

We introduce the short notation $\zeta = (\alpha, \beta, \gamma, x, y, z)$ for the free parameters. The camera projection matrix P can now be calculated:

$$P(\zeta) = K(R(\alpha,\ \beta,\ \gamma) \mid t) \in \mathbb{R}^{3\times 4}. \tag{6}$$

The projection matrix is now used to project a set of $H \in \mathbb{N}$ fixed points $h_i \in \mathbb{R}^4$ on the discretized helix from equation (2) onto the detector:

$$\begin{pmatrix} d'_{x,i}(\zeta) \\ d'_{y,i}(\zeta) \\ d'_{z,i}(\zeta) \end{pmatrix} = P(\zeta)h_i \tag{7}$$

$$m_i(\zeta) = \begin{pmatrix} d_{x,i}(\zeta) \\ d_{y,i}(\zeta) \end{pmatrix} = \begin{pmatrix} d'_{x,i}(\zeta)/d'_{z,i}(\zeta) \\ d'_{y,i}(\zeta)/d'_{z,i}(\zeta) \end{pmatrix} \tag{8}$$

d'_i are the homogeneous detector pixel coordinates and m_i are the projected analytical helix points on the detector. These points resemble the expected position of the helix structure and they will be used for constructing an error term in the 2D detector image domain.

An appropriate cost function for comparing the error between the current and expected position of a measured circle center $\hat{m}_j$ and a projected point on the helix m_i is the *reprojection error*:

$$E(\zeta,\ \hat{m}_j,\ m_i) = \hat{m}_j - m_i(\zeta) \in \mathbb{R}^2 \tag{9}$$

Equation (9) will only measure the error for a specific pair of points. In our case there are

- c (detected) circles on the current image,
- s spheres glued onto the holder and
- H projected points on the helix from equation (1).

It is important to note that $c \leqslant s$ because the segmentation algorithm might fail to detect all circles.

We now compare each of the c detected circle centers $\hat{m}_j$ to all H sampled and projected points m_i and choose the pair with the smallest distance (see algorithm 1).

Algorithm 1. Calibration algorithm: cost function

Require: ζ_{step}, *circles*, *helix_points*
Ensure: residuals_min

```
residuals_min←[]
for j ← 1 to size(circles) do
  m̂_j ← circles[j]
  residuals ← []

  for i ← 1 to size(helix_points) do
    m_i ← helix_points[i]
    residuals[i] ← E(ζ_step, m̂_j, m_i)
  end for

  residuals_min[j] ← min(residuals)
end for
```

We can formulate this algorithm as a least-squares problem:

$$\text{argmin}_{\zeta=(\alpha,\beta,\gamma,x,y,z)} \sum_{j=1}^{c} \min_{1 \leqslant i \leqslant H} E(\zeta, \hat{m}_j, m_i) \tag{10}$$

The optimization problem is nonlinear due to the sine and cosine terms in the rotation parametrization. In our implementation we use the *Levenberg Marquardt* algorithm. The *Jacobian* matrix with the partial derivatives of the cost function with respect to the free geometry parameters is not computed directly, but the *2-point* finite difference scheme is used for numerical estimation.

The resulting parameters $\alpha, \beta, \gamma, x, y, z$ can be used for the reconstruction as the geometry of the given acquisition.

2.7. Reconstruction

For tomographic reconstruction, the sinogram contained 1000 equidistant x-ray projections along a circular trajectory sized 720×720 pixels with a spacing of $600\mu m$. The reconstruction volume was sized $720 \times 720 \times 720$ with isotropic voxel spacing of $100\mu m$. Using our C++ reconstruction framework *elsa* [16], reconstruction was performed using an iterative conjugate gradient solver run for 50 iterations on a Tikhonov regularized weighted least squares problem, with the Josephs method for x-ray transform discretization and parallel beam geometry. Further iterations showed no improvement on the cost function.

2.8. Software stack

The central part of our software stack is the *Robot Operating System* (*ROS*) [17] which is a middleware for the communication of independent processes across a network. Robot manipulation is accomplished with the *MoveIt!*framework [18, 19] and the *franka_ros* configuration package [20]. For image processing tasks and the circle segmentation we use *OpenCV* [21], for multithreading on the CPU *OpenMP* [22] and on the GPU *OpenCL* [23] and for the tomographic reconstruction *elsa* [16]. The scientific calculations in section 2.6 are implemented with *scipy* [24]. The 3D mesh files of the robotic arm are read with OpenMesh [25].

3. Experiments and results

Before running the main experiments, an experiment for determining the accuracy of the robotic arm was conducted. The sample holder from section 2.3 was used for all experiments for the purpose of geometric calibration. The collision detection algorithm was running in the background throughout these experiments.

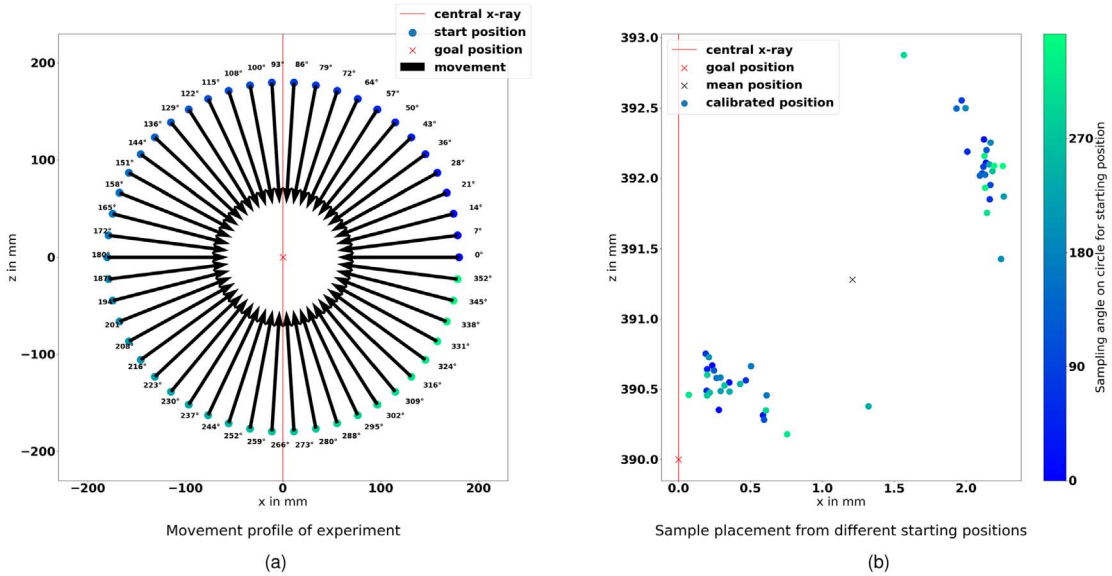


Figure 5. Robot calibration experiment. (a) Robot placement precision from different starting positions from the top perspective. Starting positions sampled on a circle parallel to the base of the robot with radius of 18 cm and the default goal position for the sample holder as center point. The goal position (cross) and orientation was identical for all measurements. The resulting images were calibrated with the calibration procedure in figure 4 . (b) The centers of the calibrated helix structures are plotted with circles. The z-axis is omitted for illustration purposes. The expected location of the circles is outlined by the red cross, which lies on the center ray of the source. The distance of the actual position of the calibration structure (circles) to the central ray (red cross) demonstrates that a calibration procedure is needed.

3.1. Robot placement error

The robotic arm does not accurately place the sample at the desired goal position and it is also not able to report the current position of the sample accurately. This hinders the reconstruction of the sample. In the following we quantitatively demonstrated the need for a calibration mechanism.

We moved the robot from different starting positions to a predefined identical target pose. The starting positions were sampled on a horizontal circle with radius 180 mm and the center at our default acquisition goal position along the central x-ray (see figure 5(a)). The resulting positions were determined by running the calibration algorithm on the acquired images. Deviations from the desired goal position therefore demonstrate the need for a calibration mechanism.

3.2. Calibration parameters

In this section we are going to state our choices for the three parameters of the calibration method in section (2.6):

- a rejection threshold on the reprojection error
- the number of spheres on the holder
- the sampling rate on the helix structure.

The threshold on the reprojection error determines when a calibration result for a given image is not accepted as valid and hence not used for the reconstruction of the sample. We chose 2.5 pixels distance between each detected circle $\hat{m_j}$ to smallest m_i as threshold. With this threshold the rejection rate of the calibration algorithm was 0.8% with 8 out of 1000 for the example walnut dataset (with $H = 10.000$). All rejections originated from false positives in the segmentation step when random points on the image were detected as circles.

The remaining parameters for the calibration algorithm are the number of spheres and the sampling rate on the helix structure. We have altered one of these parameters while fixing the other one for our walnut dataset and analysed the change in the reprojection error. Our choice for the helix sampling rate ranged from 500 to 30.000. For the number of spheres we sampled a different set of n random spheres from the detected spheres without replacement for each individual image, where $10 \leqslant n \leqslant 50$. The results are plotted in figure 6 .

For our experiments we fixed H to 10.000.

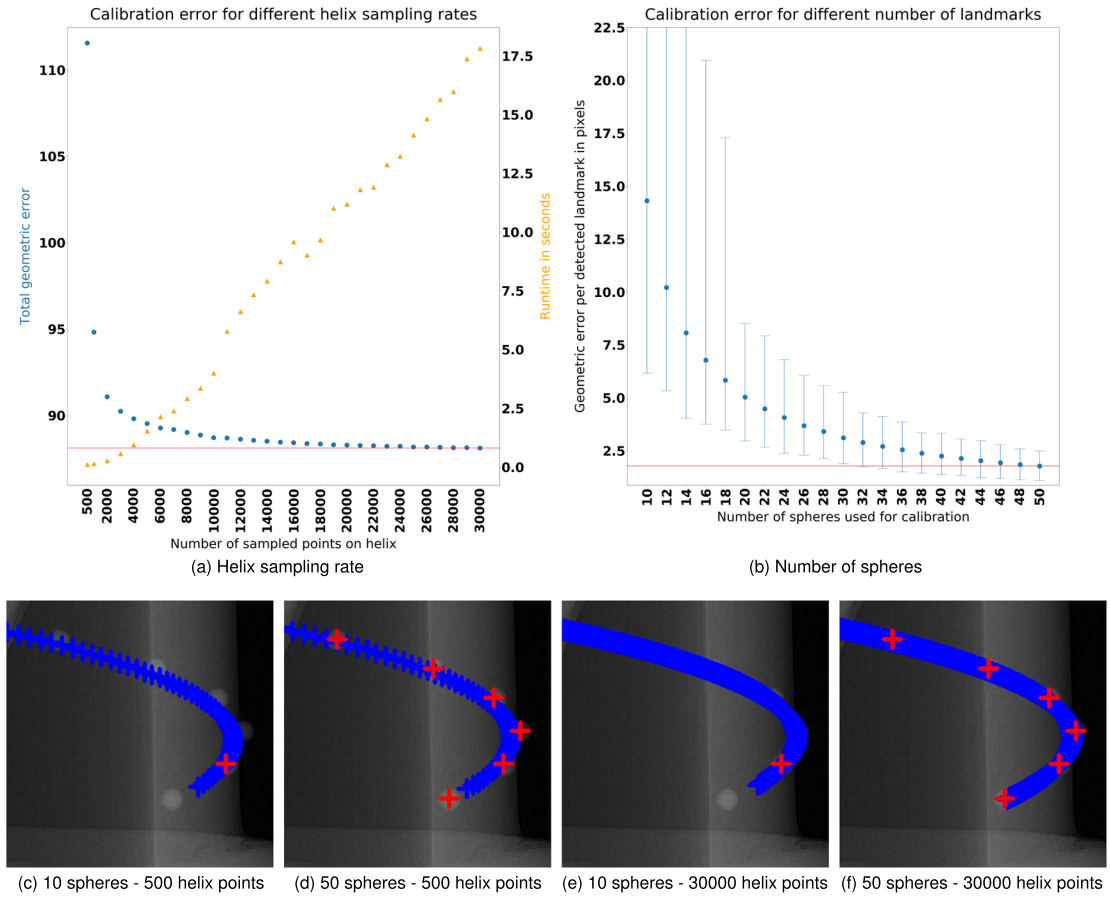


Figure 6. Calibration parameters. The calibration is influenced by two parameters: The number of points that are sampled on the helix structure for the distance measurements and the number of spheres on the sample holder. Calibration results were evaluated based on different choices for the two different parameters. The *geometric error* between the sampled helix structure and the detected spheres was used as error metric. A higher sampling rate on the continuous helix structure leads to a lower error but increases the runtime linearly (a). A higher number of spheres decreases the error but physical constraint do not allow to increase this number as there is limited space on the sample holder (b). Calibration results for different combinations of the parameters are displayed for a region of the helix structure (c) to (f). Decreasing the number of spheres severely affects the calibration results.

3.3. CT measurements

We conducted four experiments in total: two samples (walnut and pistachio) were each measured with the robotic arm and a conventional rotational stage.

For each CT measurement, 1000 images were acquired with a source voltage of 30 kV, source power of 1445μ A, and exposure time of 1s.

In figure 7(a) the reconstruction of the walnut with the rotational stage is compared to the robotic arm as the sample holder. The two volumes were registered manually as we found automatic registration of the two discretized volumes to be unreliable. The slices were chosen manually for illustration purposes. The center slices of the volume from the top and the front view were extracted and cropped to the region of interest.

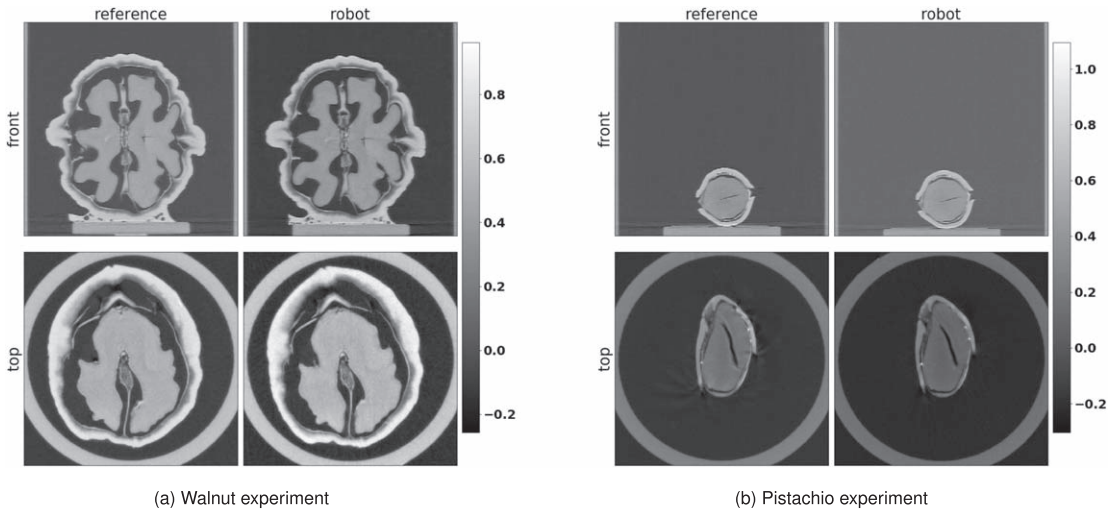
4. Discussion

4.1. Robot placement error

We can see in figure 5(b) that there are two issues: The mean of all points is shifted by more than 1 mm in both directions and the final positions vary significantly from the mean position.

The goal position lies on the intersection of the central ray of the x-ray source with the vertical operating plane of the robotic arm. This means that the goal position depends on the relative distance between the x-ray source and the robotic arm's base. From the shifted mean in figure 5(b) we can conclude that our assumption of the goal position is not correct. This systematic error likely is the result of an inaccurate measure of the relative distance between the source and robotic arm.

The second issue arises from the inaccuracy of the inverse kinematics and the limitations of the electrical motors at the joints. In the ideal case all of the colored points would lie on the same spot. However, the final



(a) Walnut experiment (b) Pistachio experiment

Figure 7. Experimental results. A walnut and a pistachio were measured and reconstructed in order to compare the conventional rotational stage (*reference*) with the robotic sample holder (*robot*). The reconstruction volumes were registered and aligned but small differences are still visible. The detector images were binned with $4 * 4$ and the reconstruction volume has dimensions 720^3. The front slice is from the perspective of the x-ray source. The top slice is from the bird's eye view. A zoom factor of 2x was applied to the slices to crop the region of interest. Our observation is that the reconstruction quality is identical despite the fact that the volumes are not aligned perfectly and hence the contrast does not match.

positions for all measurements form two clusters, without a correlation to the starting point. From the random distribution of the colored points we cannot determine a systematic pattern and hence no clear reason for these positions on the graph can be deduced for the starting angles. There is also no obvious reason for the partitioning of the points in two clusters. From these observations we conclude that a calibration procedure is necessary for determining the position and orientation of the sample and subsequently reconstructing the sample accurately.

4.2. Calibration parameters

In section 3.2 we have stated our choices for the calibration parameters used in section 2.6. For the rejection threshold on the reprojection error of the calibrated images we observed in figure 6(b) that the lowest mean error per circle is 1.77. This value could serve as a lower bound for the threshold that we are trying to determine because for the given number of spheres this is the lowest achievable reprojection error on the given dataset. However, when applied to the walnut dataset the 1.77 threshold resulted in a rejection rate of 62.3%. For this reason a higher threshold with a lower rejection rate on the walnut dataset is preferred. We have chosen 2.5 pixels as a threshold because when applied to the walnut dataset the rejection rate is very low with 0.8% and the reconstruction results are visually identical to the results of the experiment with the rotational stage as discussed in section 4.4.

For the helix sampling rate H (figure 6(a)) we observed that from 500 to 10.000 the improvement of 111.6 to 88.7 in the mean reprojection error is significant, while the runtime of the calibration per image increases from 0.15 to 4.0 seconds.

Simulating a reduction in the number of spheres on the sample holder (by random sampling without replacement) has the expected consequence of worse reprojection error (see figure 6(b)). There is no significant difference between 46, 48 and 50 circles because on most images between 46 and 50 out of 50 circles are detected by the segmentation algorithm.

In figure 6 in the bottom row the calibration algorithm was run with different combinations of both hyperparameters. Blue crosses resemble the calibration result of the helix and red crosses are the detected circles. We can observe that reducing the number of spheres to 10 makes the calibration unusable and increasing it to more than 50 was physically not possible as the circles would start to overlap on the images, especially on the curves of the helix.

4.3. Calibration

A calibration procedure example is displayed in figure 4. Figure 4(a) is post-processed and the circles are detected. Three circles were not detected. In (c) the current estimate for the external parameters were used to project $H = 10.000$ helix points (blue crosses) onto the image. To improve its overlap with the detected circles (red crosses) the parameters are optimized with the above problem statement. Finally the helix is projected onto

the image with the improved parameters (see figure 4 (d)). We can see that the overlap has improved substantially.

4.4. CT measurements

Our observation from figure 7 is that there is no qualitative difference between the results of the robotic sample holder and a conventional rotational stage as a reference. As the volumes could only be registered manually, we perform only a qualitative assessment. Two samples of different size and internal structure were measured, a walnut and a pistachio. We can see in figure 7(a) that the internal structure of the walnut was mapped as accurately for the reconstruction of the measurement with the robotic arm as with the reference experiment. The pistachio in figure 7(b) has a simpler structure compared to the walnut, but we can assess the slit in the kernel and the sharpness of the shell. When comparing the reconstruction of the measurement with the robotic arm to the reference reconstruction, we can see that there is no loss in the visual quality of the slit in the pistachio's kernel and its shell in both the top- and front-view.

4.5. Future work

In future work the system can be improved in several ways.

The sample holder could be more flexible. Its size currently limits the size of the sample but this can be tackled in another design iteration by embedding the geometrical calibration structure into the base of the holder when it is positioned upwards of the base and compressed in its height. The cylindrical envelope surrounding the sample could be removed and as a consequence, the sample also doesn't strictly need to be inserted from the top. It could be positioned right were the geometric structure would end. Depending on the mounting mechanism of the sample, samples of arbitrary sizes could be measured assuming the maximum payload of 3 kilograms is not exceeded [8].

Moreover, experiments with non-standard trajectories are subject of future work.

Finally, the accuracy of the calibration algorithm could be improved by improving the circle detection algorithm that is run on the acquired images. Currently, we are using the circle Hough transform algorithm which could be replaced by a more precise algorithm with sub-pixel segmentation accuracy.

5. Conclusion

In this work we have demonstrated the use of a seven degrees of freedom robot as a sample holder for x-ray computed tomography using our complete software package with path planning, collision detection and calibration. Our findings have confirmed that this kind of robot can be used for computed tomography with consistent results when compared to more conventional sample holders. A suitably sized sample holder with a geometric structure that can be used for calibration must be provided.

Acknowledgments

We acknowledge financial support through the Center for Advanced Laser Applications (CALA).

ORCID iDs

Erdal Pekel https://orcid.org/0000-0002-5001-6735
Florian Schaff https://orcid.org/0000-0003-2144-851X
Martin Dierolf https://orcid.org/0000-0001-5150-2949
Franz Pfeiffer https://orcid.org/0000-0001-6665-4363
Tobias Lasser https://orcid.org/0000-0001-5669-920X

References

[1] Wieczorek M, Schaff F, Pfeiffer F and Lasser T 2016 Anisotropic x-ray dark-field tomography: a continuous model and its discretization *Phys. Rev. Lett.* **117** 158101
[2] Wieczorek M, Schaff F, Jud C, Pfeiffer D, Pfeiffer F and Lasser T 2018 Brain connectivity exposed by anisotropic x-ray dark-field tomography *Sci. Rep.* **8** 1–6
[3] Ziertmann A, Jahnke P, Kerscher S, Koch M and Holub W 2020 Robot guided computed tomography—production monitoring in automotive industry 4.0 *Journal of the Japan Society for Precision Engineering* **86** 316–22
[4] Landstorfer P, Herl G and Hiller J 2019 Investigation of non-circular scanning trajectories in robot-based industrial x-ray computed tomography of multi-material objects *ICINCO 2019—Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics* **2** 518–22 (https://www.scitepress.org/Papers/2019/79664/79664.pdf)

[5] Healthineers S 2021 Siemens artis zeego eco [Online] Available: (https://www.siemens-healthineers.com/refurbished-systems-medical-imaging-and-therapy/ecoline-refurbished-systems/angiography-ecoline/artis-zeego-eco)
[6] Herl G, Hiller J, Thies M, Zaech J-N, Unberath M and Maier A 2021 Task-specific trajectory optimisation for twin-robotic x-ray tomography *IEEE Transactions on Computational Imaging* **7** 894–907
[7] KUKA 2021 Kuka kr 90 r3100 extra ha datasheet [Online]. Available: (https://www.kuka.com/-/media/kuka-downloads/imported/6b77eecacfe542d3b736af377562ecaa/0000208694_de.pdf)
[8] EMIKA F 2020 Panda datasheet [Online]. Available: (https://s3-eu-central-1.amazonaws.com/franka-de-uploads/uploads/2019/04/Datasheet.pdf)
[9] Li X, Zhang D and Liu B 2010 A generic geometric calibration method for tomographic imaging systems with flat-panel detectors–a detailed implementation guide *Med. Phys.* **37** 3844–54
[10] Cho Y, Moseley D J, Siewerdsen J H and Jaffray D A 2005 Accurate technique for complete geometric calibration of cone-beam computed tomography systems *Med. Phys.* **32** 968–83
[11] Robert N, Watt K N, Wang X and Mainprize J G 2009 The geometric calibration of cone-beam systems with arbitrary geometry *Phys. Med. Biol.* **54** 7239–61
[12] Pekel E 2021 Robotic sample holder (https://gitlab.lrz.de/IP/robotic-sample-holder/robotic-sample-holder)
[13] Beeson P and Ames B 2015 TRAC-IK: an open-source library for improved solving of generic inverse kinematics *Proceedings of the IEEE RAS Humanoids Conference (Seoul, Korea, 03-05 November 2015)* (Manhattan, New York, U.S.: IEEE) (https://ieeexplore.ieee.org/abstract/document/7363472)978-1-4799-6885-5 (https://doi.org/10.1109/HUMANOIDS.2015.7363472)
[14] Hermann A, Drews F, Bauer J, Klemm S, Roennau A and Dillmann R 2014 Unified gpu voxel collision detection for mobile manipulation planning *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* 4154–60
[15] Ballard D H 1981 Generalizing the Hough transform to detect arbitrary shapes *Pattern Recognit.* **13** 111–22
[16] Lasser T, Hornung M and Frank D 2019 Elsa—an elegant framework for tomographic reconstruction *15th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine* ed S Matej and S D Metzler 11072 (Belingham, Washington: International Society for Optics and Photonics. SPIE) 570–3
[17] Quigley M, Conley K, Gerkey B, Faust J, Foote T, Leibs J, Wheeler R and Ng A Y 2009 Ros: an open-source robot operating system *ICRA Workshop on Open Source Software* **3** 5 Kobe, Japan
[18] Coleman D, Sucan I, Chitta S and Correll N 2014 Reducing the barrier to entry of complex robotic software: a moveit! case study *Journal of Software Engineering in Robotics* **5** 3
[19] Sucan I A and Chitta S MoveIt: [online] (http://moveit.ros.org/)
[20] FRANKA EMIKA 2021 (https://github.com/frankaemika/franka_ros)
[21] Bradski G 2000 The openCV library *Dr. Dobb's J. Softw. Tools* **1** 1 (https://opencv.org)
[22] Dagum L and Menon R 1998 Openmp: an industry standard api for shared-memory programming *IEEE Computational Science and Engineering* **5** 46–55
[23] Stone J E, Gohara D and Shi G 2010 Opencl: a parallel programming standard for heterogeneous computing systems *Computing in Science Engineering* **12** 66–73
[24] Virtanen P *et al* 2020 SciPy 1.0: fundamental Algorithms for Scientific Computing in Python *Nat. Methods* **17** 261–72
[25] Botsch M, Steinberg S, Bischoff S and Kobbelt L 2002 Openmesh-a generic and efficient polygon mesh data structure *CiteSeerX* (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.3900) 10.1.1.16.3900 2002, 2021 31st of July