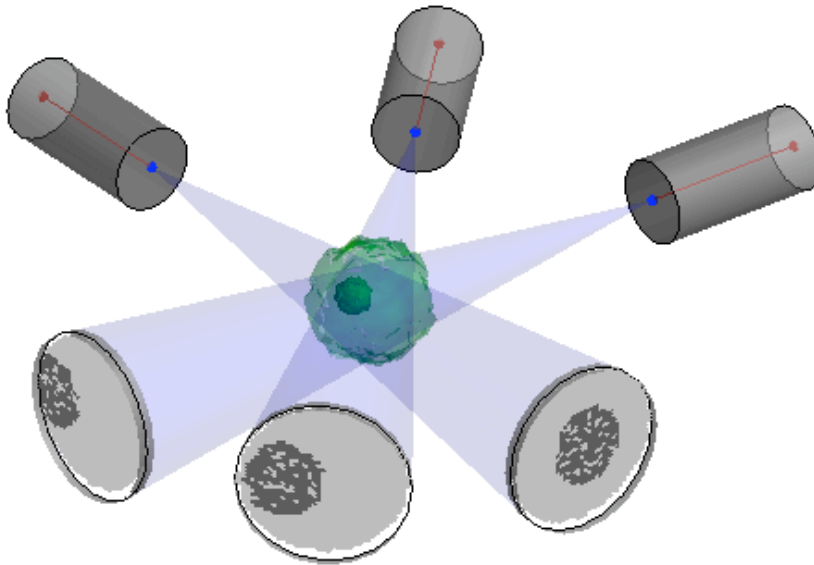


Geometric Calibration and Reconstruction on a Portable X-ray Computed Tomography Device

Samuel M. Johnston

Introduction

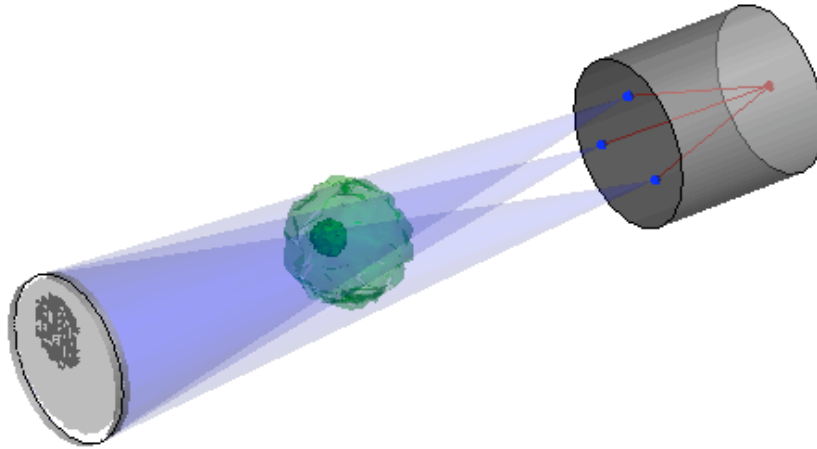
X-ray computed tomography (CT) is a popular tool for non-destructively producing cross-sectional and volumetric images of physical objects[1]. In this technology, an X-ray source and a detector are mounted on a rotating gantry, and X-ray images from many different angles are acquired and then passed to a reconstruction program to create volumetric images. Unfortunately, the gantry and the rotational geometry can impose several onerous constraints on the production and usage of such a system: the gantry is quite heavy, it can only scan objects that fit within the inner diameter, and the technological complexity of making it both fast and stable can greatly increase the total cost of the system. There are many situations in which it would be desirable to have access to X-ray CT in locations outside the clinic, such as in rural and military locations. However, the size, weight, and cost of conventional X-ray CT systems obviates this usage. Since an X-ray CT system ultimately consists of an X-ray source and detector acquiring images at multiple orientations, and these components by themselves are amenable to substantial reductions in size [2][3], it is worthwhile to consider whether X-ray CT can be achieved without a gantry.



Traditional X-ray CT

Tomosynthesis is an emerging technology that partially addresses some of these issues: Instead of rotation, the X-ray source is translated with respect to a stationary object and

detector[4]. This removes the spatial constraints of the gantry and reduces the complexity and cost, but it still requires a bulky mechanical system to translate the X-ray source. Since X-rays are conventionally generated by firing a high energy cathode ray at a metal anode, some X-ray CT systems, such as Inverse Geometry CT[5], avoid mechanical movement of the X-ray source by using electromagnetic deflection to sweep the cathode ray over the surface of the anode. We believe that an X-ray tube with a mobile cathode ray could be sufficient to provide the translation of the X-ray source needed to perform tomosynthetic reconstruction and produce useful images.



Tomosynthesis with a mobile cathode ray

To perform reconstruction with this system, it is necessary to have a geometric model of the system that describes the relationship of the acquired image data to the reconstructed volume data. The goal of this project is to construct this model, use it to delineate the reconstruction algorithm, and identify the parameters and measurements required to perform the algorithm.

Coordinate System

The goal of X-ray CT is to reconstruct a 3-dimensional image of an object from a series of 2-dimensional X-ray projections through an object. The image is discretized into an array of pixels, and the volume is discretized into an array of voxels.

If we ignore scatter, the trajectory of X-rays from the source through an object to a pixel on the detector may be described as a straight line. Since every point along that line contributes to the attenuation of the X-rays, the CT reconstruction process backprojects the value at each pixel down the line from detector to source, adding this value to each voxel that is intersected by the line. Consequently, the parameters that describe this line for each pixel at each detector and source location must be known before reconstruction can proceed. We will now construct the equation for this line and show what parameters are needed. While it is sometimes convenient to represent this set of positions and

transformations as vectors and matrices in a 4-dimensional affine space, we do not use that representation here because it provides no significant reduction in computation.

The coordinate system is defined with respect to a stationary detector. The origin is located at detector pixel (0, 0), the x -axis is the horizontal detector axis, the y -axis is the vertical detector axis, and the z -axis is normal to the detector plane and oriented towards the X-ray source. The source location \mathbf{s} is identified by its x , y , and z coordinates. To find the detector point \mathbf{p} that an object point \mathbf{o} will be projected to, we must construct the line from the \mathbf{s} to \mathbf{o} and intersect it with the detector plane $z = 0$:

$$\begin{aligned}\mathbf{p} &= \mathbf{s} + k(\mathbf{o} - \mathbf{s}) \\ z_p &= z_s + k(z_o - z_s) \\ 0 &= z_s + k(z_o - z_s) \\ k &= \frac{-z_s}{z_o - z_s}\end{aligned}$$

Each pixel has length and width Δp , so any pixel (u, v) has absolute coordinates $\mathbf{p} = (u\Delta p, v\Delta p, 0)$. Each voxel has length and width Δv , and the voxels are numbered consecutively from the volume origin at \mathbf{v} , so any voxel (x, y, z) has absolute coordinates $\mathbf{o} = (x\Delta v - x_v, y\Delta v - y_v, z\Delta v - z_v)$. This allows us to calculate the projected pixel for each voxel:

$$\begin{aligned}k &= \frac{-z_s}{z_o - z_s} \\ u &= \frac{x_s + k(x\Delta v - x_v - x_s)}{\Delta p} \\ v &= \frac{y_s + k(y\Delta v - y_v - y_s)}{\Delta p}\end{aligned}$$

Since the location and size of the volume are chosen based on user preferences, and reasonable default values can be constructed by setting the voxel size equal to the pixel size and the volume location adjacent to the detector, the source location is the only geometric parameter that must be calibrated before reconstruction can proceed.

Calibration

To find the source location, we use a phantom consisting of an array of small metal pellets that strongly attenuate X-rays. The pellets are placed in a plate of material that weakly attenuates X-rays, and the plate is affixed firmly to the surface of the detector. The locations of the pellets with respect to the central pixel are known in advance, and the beads are separated from the detector surface by a nontrivial distance Δz . Each time an image is acquired with the system, local minima in intensity values are found and compared to the known attenuation properties of the metal, and these image points are mapped to the pellets.

```

known maximum pellet projection radius r
known pellet attenuation a
for each u,v
  window = projection[u-r:u+r,v-r:v+r]
  if projection[u,v] = minimum(window)
    and projection[u,v]-maximum(window) >= a
      add (u,v) to list of minima
  end
end
end

```

The numbers in the projection are not the raw values from the images, but the output from a logarithmic correction that is described later.

The pellet projections can be mapped to the physical pellet locations in a straightforward way because the projection operation preserves the x and y order of objects in the same xy plane. Given two pellets p_1 and p_2 at (x, y, z) and $(x+a, y+b, z)$,

$$k = \frac{-z_s}{z - z_s}$$

$$u_1 = \frac{x_s + k(x - x_s)}{\Delta p}$$

$$u_2 = \frac{x_s + k((x + a) - x_s)}{\Delta p} = \frac{x_s + k(x - x_s)}{\Delta p} + \frac{ka}{\Delta p} = u_1 + \frac{ka}{\Delta p}$$

$$v_1 = \frac{y_s + k(y - y_s)}{\Delta p}$$

$$v_2 = \frac{y_s + k((y + b) - y_s)}{\Delta p} = \frac{y_s + k(y - y_s)}{\Delta p} + \frac{kb}{\Delta p} = v_1 + \frac{kb}{\Delta p}$$

This preservation of order allows us to discard local minima that do not fit the known preexisting pattern. This can be accomplished by fitting lines to the minima according to known orientations, and identifying those points whose distance from the nearest line exceeds a given threshold.

We then construct lines from the pellet projections through the known physical locations of the pellets. In an ideal geometry, the lines would all intersect at the location of the source, but numerical errors may preclude true intersection. Consequently, we estimate the intersection by finding the middle point between each pair of lines and then finding the average of these middle points.

```

empty point list
for each detected pellet
  get detect1 coordinates (u1*dp,v1*dp,0)
  get object1 coordinates (x1,y1,z1)
  for each other detected pellet we haven't checked yet

```

```

    get detect2 coordinates (u2*dp,v2*dp,0)
    get object2 coordinates (x2,y2,z2)
    find point that is closest to the lines
      (detect1,object1) and (detect2,object2)
    add point to list
  end
end
source = mean(list)

```

The algorithm for finding the point closest to two lines is too tedious to write here, but it can be found at [6].

The algorithms described thus far probably do not have optimal run times, but the input size (generally less than 100 pellets) is small enough so that the time spent in this portion of the program is negligible compared to reconstruction time.

Reconstruction

The attenuation of X-rays in matter is described by the linear attenuation coefficient $\mu(x)$, which is the fractional difference between the number of photons $I(x)$ entering an incrementally thick slice of material at x and the number of photons $I(x) - dI(x)$ that exit the slice along the same path:

$$dI(x) = -I(x)\mu(x)dx$$

When this expression is integrated over the entire path from source to detector, we arrive at Beer's Law[1]:

$$I_{out} = I_{in} \exp\left(-\int_{in}^{out} \mu(x)dx\right)$$

I_{out} is the value measured at the detector, and I_{in} can be determined with radiation counters when the X-ray tube is manufactured. This allows us to find the line integral of the attenuation coefficient along the path from source to detector:

$$\int_{in}^{out} \mu(x)dx = \ln\left(\frac{I_{in}}{I_{out}}\right)$$

Dividing this quantity by the length of the path gives us the average attenuation coefficient along the line segment:

$$\bar{\mu} = \frac{\int_{in}^{out} \mu(x)dx}{\int_{in}^{out} dx} = \frac{\ln\left(\frac{I_{in}}{I_{out}}\right)}{\int_{in}^{out} dx}$$

X-ray CT proceeds by assigning to each voxel the average of the average attenuation coefficients of all line segments that pass through the voxel, for all X-ray projections. This process of following the lines back from pixels to voxels is called backprojection.

There are two approaches for ordering these assignments. In the voxel-based approach, each of the voxels are updated in the order that they are stored, and at each voxel the corresponding pixel coordinates are obtained and the corresponding average attenuation coefficient is assigned:

```
choose volume coordinates
choose pixel and voxel sizes
for each projection
    perform logarithmic correction on projection
    get source coordinates from calibration
    for each x, y, z
        k = -source.z / (z - source.z)
        u = (source.x +
            k * (x * dv - volume.x - source.x)) / dp
        v = (source.y +
            k * (y * dv - volume.y - source.y)) / dp
        volume[x, y, z] += projection[u, v]
    end
end
for each x, y, z
    volume[x, y, z] /= #projections
end
```

In the ray-based approach, the line from the source to detector is constructed for each pixel, and the program moves in steps along the line, assigning the average attenuation coefficient at every step:

```
choose volume coordinates
choose pixel and voxel sizes
choose step increment ds
for each projection
    perform logarithmic correction on projection
    get source coordinates from calibration
    for each u, v
        position = source
        detector = (u * dp, v * dp, 0)
        direction = detector - source
        step = ds * normalize(direction)
        while position.z > 0
            position += step
            x = (position.x + volume.x) / dv
            y = (position.y + volume.y) / dv
            z = (position.z + volume.z) / dv
        end
    end
end
```

```

        volume[x,y,z] += projection[u,v]
        counter[x,y,z] ++
    end
end
end
for each x, y, z
    volume[x,y,z] /= counter[x,y,z]
end

```

The voxel-based technique tends to be faster, although the ray-based technique can be tuned with the parameter Δs to take less time or produce cleaner reconstructions. Both techniques can be modified to consider the varying sizes of the intersections of the rays with the voxels.

Because each voxel is assigned values that are averages of all the voxels along the line segments, the resulting raw reconstructions usually appear blurry. One approach to this problem is to perform high-pass filtering on the projection images before reconstruction. This approach is typically used for most X-ray CT systems in which the projections are acquired by rotating the source and the detector at regular intervals around the object, since that arrangement leads to a representation of the reconstruction process in the Fourier domain that allows for filtration to be performed on the projection images with a simple ramp filter. However, for more arbitrary acquisition sequences, such as those that we consider in this work, the Fourier approach requires 3-dimensional filters that must be constructed with methods similar to the reconstruction process.

Another approach to dealing with imperfect reconstruction is to simulate the X-ray projection process through the reconstructed volume, and compare the synthetic projection images with the original projection images. An error image can be calculated by subtracting the synthetic from the original, and the error can then be backprojected. In this way the reconstruction may be iteratively refined.

The simulated reprojection algorithm is practically identical to the backprojection algorithms: in both cases, the voxels and pixels must be traversed in some fashion and associated with each other. Whereas backprojection assigns the pixel value to the voxel, reprojection does the opposite. So the code

```

volume[x,y,z] += projection[u,v]

```

is replaced by

```

projection[u,v] += volume[x,y,z]

```

The ray-based method tends to work substantially better than the voxel-based method for reprojection, because in the latter there tends to be large variation in the number of voxels that are assigned to each pixel. Furthermore, since the number of voxels in each volume is substantially greater than the number of pixels in each image, parallel implementations

of the voxel-based technique suffer from excessive memory conflicts caused by multiple voxels writing to the same pixel.

The refinement is typically performed by reprojecting and backprojecting one image at a time, and this approach seems to promise faster convergence[7]. In our experience, this method can lead to severe asymmetries in the reconstruction, so we instead backproject all the error images to a separate error volume before adding this volume to the reconstruction volume. This produces cleaner reconstructions at the cost of slower convergence.

The backprojection and reprojection algorithms consist of simple arithmetic and memory operations performed independently at different parts of the volume and the image. In the voxel-based approach, the independent elements are the voxels, and in the ray-based approach, the independent elements are the rays. Consequently, these algorithms naturally lend themselves to parallelization. A variety of hardware configurations to handle specialized parallel operations have become available to general users in recent years. One notable hardware class is the graphical processing unit (GPU), which is specialized for image processing operations. GPU's have been propelled by the ongoing development of computer entertainment to outpace central processing units in the growth of processing power and memory, and are available at prices that are lower than other comparable kinds of specialized computational hardware. Researchers in a variety of fields have recognized the potential to exploit GPU's to handle computational problems beyond traditional image processing, including CT reconstruction[8].

To address this interest, GPU manufacturers have developed application programming interfaces that allow general-purpose programs to be programmed for the GPU. For this purpose, Nvidia has released the Compute Unified Device Architecture (CUDA) [9], a set of tools based on the C programming language that make their GPU's accessible for general use. We have implemented in CUDA the backprojection and reprojection algorithms in both the voxel-based and ray-based forms, and have used these to build an iterative CT reconstruction program.

Multiple Detector Locations

The tomosynthesis approach is successful at reconstructing cleanly along the axis from the center of the X-ray source anode to the detector, and for most medical applications the doctor only needs to examine cross sections on a single axis. However, when multiple axes are needed, or when the volumes being reconstructed are to be used for processing by downstream analysis programs, it may be necessary to perform scans at more than one source-detector orientation. To accomplish this, we use a small, rotationally asymmetric phantom consisting of irregularly spaced holes in a weakly-attenuating medium. We place the phantom securely on the object being scanned, move the X-ray tube and detector to multiple orientations around the object, and for each scan we perform the reconstruction separately. In each reconstruction we then locate the phantom and the positions of the holes using a local minima finder similar to the one described earlier for finding pellet projections.

One scan is used as a reference scan, and then a rigid transformation is constructed for each subsequent scan that maps it back to the reference scan. Since the scale of the volume is controlled by the user, the transformation consists only of translation and rotation. The translation is found by computing the mean points of the reference and subsequent sets of points, and constructing the vector between the two. We apply the translation to the subsequent set, and then use a least squares method to find the rotation.

Rotation may occur about all three axes, and can be represented as a series of rotation matrices constructed from Euler angles α , β , and γ that take a point (x, y, z) in the reference system to a point (x', y', z') in the subsequent system:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & \sin\gamma \\ 0 & -\sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

These can be multiplied into a single 3x3 matrix:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

We can then rewrite this system of linear equations so that the matrix and vector are swapped:

$$\begin{aligned}
a_{11}x + a_{12}y + a_{13}z &= x' \\
a_{21}x + a_{22}y + a_{23}z &= y' \\
a_{31}x + a_{32}y + a_{33}z &= z'
\end{aligned}$$

$$\begin{aligned}
a_{11}x + a_{12}y + a_{13}z + a_{21}0 + a_{22}0 + a_{23}0 + a_{31}0 + a_{32}0 + a_{33}0 &= x' \\
a_{11}0 + a_{12}0 + a_{13}0 + a_{21}x + a_{22}y + a_{23}z + a_{31}0 + a_{32}0 + a_{33}0 &= y' \\
a_{11}0 + a_{12}0 + a_{13}0 + a_{21}0 + a_{22}0 + a_{23}0 + a_{31}x + a_{32}y + a_{33}z &= z'
\end{aligned}$$

$$\begin{bmatrix} x & y & z & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x & y & z \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

If we measure n points in the reference and subsequent systems, we can add them as rows to the equation:

$$\begin{bmatrix} x_1 & y_1 & z_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & z_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 \\ & & & & & \vdots & & & \\ x_n & y_n & z_n & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_n & y_n & z_n & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_n & y_n & z_n \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ z'_1 \\ \vdots \\ x'_n \\ y'_n \\ z'_n \end{bmatrix}$$

This allows us to find a least-squares solution for the elements of the rotation matrix through the pseudoinverse, constructed with the singular value decomposition:

$$\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ z'_1 \\ \vdots \\ x'_n \\ y'_n \\ z'_n \end{bmatrix} \begin{bmatrix} x_1 & y_1 & z_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & z_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 \\ & & & & & \vdots & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_n & y_n & z_n & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_n & y_n & z_n \end{bmatrix}^+$$

Once we have the transformation, we can either transform the subsequent reconstructions and add them to the reference reconstruction, or include the transformation inside the reconstruction algorithm.

To transform a reconstruction, we construct a grid based on the reference system, and for each point in the grid we find the transformed value by transforming the grid coordinates, and finding the corresponding values in the subsequent grid through linear interpolation.

To include the transformation inside the reconstruction algorithm, we apply the transformation to all position vectors when reconstructing with the images from the subsequent system:

```

source = transform(source)
volume = transform(volume)
detector = transform(detector)
position = transform(position)

```

The direction vector in the ray-based method is derived from the other vectors, and will experience rotation but not translation.

Since the composition of the phantom is known in advance, once its location and orientation are known *a priori*, the voxels it occupies can be forced to the known values during the refinement process.

Conclusion

We have illustrated the method with which a set of X-ray projection images acquired from a mobile X-ray source and detector can be used to reconstruct a volume: After we acquire the images, we perform a logarithmic correction, locate the metal pellet projections, match them to known pellet locations, construct the rays through these points and estimate their intersection in order to find the location of the X-ray source, and then step through the volume while using the source location to match the data in pixels to the corresponding voxels. The reprojection and backprojection operations are repeated as necessary to improve the reconstruction, and reconstructions from acquisitions at

different orientations can be mapped into the same space by identifying the same set of points in the different geometries.

This work is based on our previous work in developing the calibration and reconstruction for an X-ray micro-CT system with two tubes and two detectors[10], but the details of the coordinate system, the source calibration and the multiple orientation calibration methods, and the incorporation of these details into the backprojection and reconstruction functions are unique to this project.

References

- [1] Jerrold T. Bushberg, J. Anthony Seibert, Edwin M. Leidholdt Jr., John M. Boone, "The Essential Physics of Medical Imaging," Lippincott Williams & Wilkins, 2 ed. (2001).
- [2] "Handheld and Portable X-ray," <http://www.aribex.com/>
- [3] "X-Ray Detectors – Digital from Perkin Elmer," <http://optoelectronics.perkinelmer.com/Catalog/Category.aspx?CategoryName=Digital+Imaging>
- [4] James T. Dobbins III and Devon J. Godfrey, "Digital x-ray tomosynthesis: current state of the art and clinical potential," Phys. Med. Biol. 48, pp. R65–R106 (2003).
- [5] "An inverse-geometry volumetric CT system with a large-area scanned source: A feasibility study," Med. Phys. 31(9), pp. 2623-2627 (2004).
- [6] Paul Bourke, "The shortest line between 2 lines in 3D," <http://local.wasp.uwa.edu.au/~pbourke/geometry/lineline3d/>
- [7] Avinash C. Kak and Malcom Slaney, "Principles of Computerized Tomographic Imaging," IEEE Press, 1988.
- [8] Fang Xu and Klaus Mueller, "Real-Time 3D Computed Tomographic Reconstruction Using Commodity Graphics Hardware," Phys. Med. Biol. 52, pp. 3405–3419 (2007).
- [9] "CUDA Zone – The resource for CUDA developer," <http://www.nvidia.com/cuda>
- [10] Samuel M. Johnston, G. Allan Johnson, and Cristian T. Badea, "Geometric calibration for a dual tube/detector micro-CT system," Med. Phys. 35(5), pp. 1820-1829 (2008).