Solid Freeform Fabrication 2023: Proceedings of the 34th Annual International Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference
Reviewed Paper

VOXEL-FREE RADON TRANSFORM FOR IMPROVING SURFACE QUALITY IN COMPUTED AXIAL LITHOGRAPHY

Jennings Z. Ye, Yaxuan (X) Sun, Hayden Taylor

Department of Mechanical Engineering, University of California, Berkeley

Abstract

Computed axial lithography (CAL) is an additive manufacturing process that projects light patterns onto a rotating vial of photosensitive resin to print the desired object through the superposition of light energy. Currently, these projection patterns are generated by voxelizing a target object from an STL or another 3D file format and applying the Radon transform through an iterative optimization process. In this work, a voxel-free method to generate projection sets is proposed. Here, the Radon transform is performed directly on a non-voxelized target object by calculating intersections between incident rays of light and triangles of the original STL surface mesh. Initial simulated results show a more consistent light dose at the surface of the object, indicating improved surface quality and smoothness. Additionally, simulations show a more uniform object interior using the proposed method as well as the potential for improved speed scaling with object size.

Introduction

Computed Axial Lithography (CAL) is a volumetric additive manufacturing (VAM) process that projects a series of images through a rotating volume of photosensitive resin to print parts [1]. These projections are generated so that the accumulated light dose in the interior regions of the desired volume surpasses the threshold needed to cure the resin, while the exterior regions do not. These projection images are currently obtained by voxelizing a target geometry from a 3D object file such as an STL and applying the Radon transform to the voxelized target, resulting in a forward projection from object space to projection space.

The voxelization step results in a discretized target, which limits the part resolution and surface smoothness across different geometries. Additionally, the time and memory required by the CAL projection-generation process scale up drastically with the number of voxels in the target geometry. For a given pixel size, the memory required by a projection set with a fixed number of 2D images scales with the maximum radius (in the XY plane) of the part and the height of the part. The memory required by the voxelized target, however, scales with the square of this radius of the part as well as the height of the part. Thus, the voxelized target is a significant barrier to scaling up part sizes in CAL. In addition, the voxelization process creates additional overhead in the overall time taken to generate projection sets.

While work has been done to improve the speed of this voxelization process through complexity-based adaptive voxelization [2], this work proposes a method for bypassing the voxelization of the target altogether. Ray tracing is used to find the intersection points between incident rays of light and a surface mesh to take the Radon transform directly from an STL file.

While the Radon transform is mathematically continuous [3], most implementations of the Radon transform are discretized. This work removes the discretization in the calculation of the line integrals present in the Radon transform and is the first to do so in the VAM sphere. Note that the resulting projection images are still pixelated, with maximum resolution defined by the size and number of pixels in the projector's digital micromirror device (DMD). While the method removes the voxelization in the forward projection, back-projected reconstructions used for visual comparison are still voxelized. The algorithm for this approach is presented, and its effects on surface accuracy and speed are compared to the state-of-the-art process across multiple geometries.

Algorithm and Testing Methodology

The Radon transform takes the line integral of a function, in this case, the target object, over a set of lines defined in the 2D space. This set is defined as the perpendicular lines to a projection plane that pass through the center of each pixel, defining the value at each pixel in the image. This image is taken for multiple projection angles around the object, resulting in a projection set, also called a sinogram. This novel voxel-free approach is performed by conducting ray-triangle intersections through the surface mesh provided in an STL file. For each angle, rays are traced through the centers of pixels in a 2D grid, perpendicular to that grid. The intersection length between the ray and the object file is determined, defining the line integral directly using the STL mesh. This intersection length is then stored in that pixel location for the given angle. This process is repeated for all projection angles, obtaining a sinogram of dimensions (R, θ, Z), where R is the number of pixels spanning the maximum diameter of the part, θ is the number of projection angles in one rotation, and Z is the number of pixels spanning the height of the part. The results derived here make use of the Möller–Trumbore ray-triangle intersection [4] algorithm [5], simplified to return only the "t" values, which are the distances of the intersection points from the ray origin, rather than the actual intersection coordinates. A MATLAB implementation of the section following the definition of all ray origins and directions is provided below.

```
%origins and directions have been defined for R, theta, Z. Theta from [0, 180)
proj=zeros(rnum*angles*znum,1);
parfor i=1:length(orig)
    t = RayTriangleIntersection(orig(i,:), dir(i,:), v1, v2, v3);
    % v1, v2, and v3 are the vectorized vertex XYZ coordinates of all triangles in
the STL
    t = uniquetol(t,1e-7,'DataScale',1); %filter out duplicate intersections from
precision error, exact tolerance value may need to be adjusted
    for j=1:2:numel(t)
        proj(i)=proj(i)+t(j+1)-t(j); %every other intersection brings the ray into
the part
    end
end
proj=reshape(proj,[rnum,180,znum]);
proj=cat(proj, flip(proj,1)); %fill out angles from [180, 360)
```

The algorithm outlined above replaces the voxelization and voxelized Radon transform processes in the current CAL process with a single Radon transform process. However, any implementation of the Radon transform requires a correction filter due to the oversampling of low frequencies [6]. This filter introduces negative intensities into the projection images that cannot be

realized by a projector. As a result, several optimization algorithms have been created around this issue [1], [7]. This paper presents results using perhaps the simplest method of applying a linear offset by some percentage of the most negative value and setting all remaining negative values in the projection to zero (Eq. 1).

$$Proj = \max\left(0, Proj - C * \min(Proj)\right) \quad \text{(Eq. 1)}$$

where $Proj$ is the projection set from the algorithm with dimensions (R, θ, Z)
C is some constant between 0 and 1

While numerous window functions can be applied to the Ram-Lak (ramp) filter needed in the FBP process [8], past work indicates that of the windows built-in to the MATLAB iradon function, a Hann or Hamming window yield the best reconstruction results [6]. Accordingly, this work applied the Ram-Lak filter with a Hamming window to all projections. Results are compared for both the "perfect" projections with negative intensities included and the "clipped" projections with no negative intensities. For simplicity, the clipped projections in this work make no use of the linear offset and simply set all negative intensities resulting from the filter to zero ($C = 0$). Reconstructed objects showing the accumulated light dose were created by using the inverse Radon transform to perform back-projections from the projection space to the object space, completing the filtered back-projection (FBP) process and tomographic reconstruction of the object [9]. This was done for both clipped and perfect projections using the inverse Radon transform provided in the ASTRA toolbox [10]. Vertical and horizontal cross-sections of these objects were then visually compared. These comparisons were made for five pixel sizes (28.8 µm, 57.6 µm, 144 µm, 288 µm, and 576 µm) across three geometries: a 4 mm thick plano-convex lens with a 12.7 mm diameter (modeled after ThorLabs LA1074) on top of a 2 mm tall cylindrical holder of 10 mm diameter, a cube with a 10 mm side length 10, and a 20.2 mm tall model of Rodin's *The Thinker* (Figure 1). All sinograms were created with 360 projection angles over a full rotation.

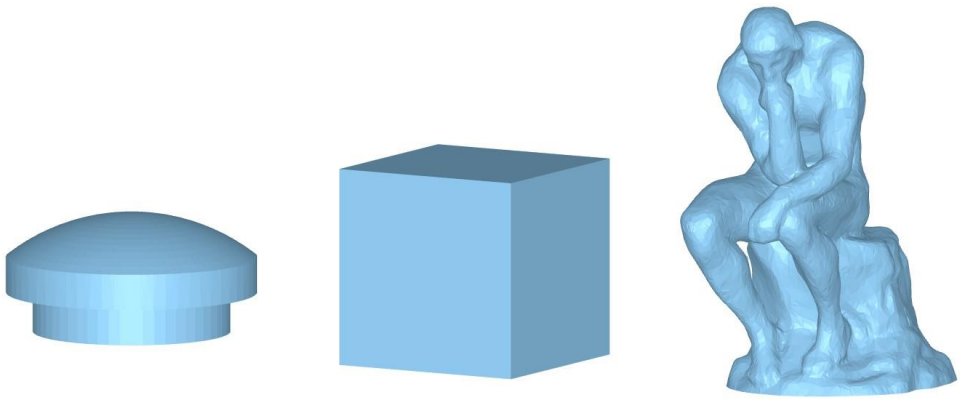

Figure 1. Tested STL Structures

Figure 1 depicts the STL representations of the geometries rather than the exact CAD model as the STL is the actual input into all algorithms tested. For the lens, both coarsely (22 µm tolerance) and finely (8 µm tolerance) meshed STLs were generated to see the effects of STL fineness on both surface accuracy and speed of both the new and current algorithms. For the cube, tests were conducted with the faces of the cube facing the primary X, Y, and Z, axes ("unrotated cube"), as well as with the cube rotated 45° in the XY plane.

Each geometry was chosen for distinct features. The lens is a geometry that requires high surface accuracy and investigates the differences between methods on an object of significant curvature. The cube tests the sensitivity of the two projection methods to rotation, as it is expected that both methods can perfectly locate the edge of the unrotated cube, but not the rotated cube due to the discretization of the pixels. *The Thinker* is used to compare performance for a complicated geometry with more variation in feature size and curvature.

In addition to reconstruction accuracy, the time to obtain the projection sets was benchmarked in MATLAB and compared across these processes for all geometries and pixel sizes tested. The new process presented above was run on 12 parallel CPU cores, while the voxelized method was run in two versions, one on 12 parallel CPU cores [1], and the other using the ASTRA toolbox [10] with GPU acceleration. While these two implementations of the voxelized transform differ slightly, maximum differences in voxel-wise comparisons of reconstructed dose for these objects are below 5% and are visually nearly identical.

Results

Vertical and horizontal cross-sections for perfect and clipped reconstructions of five STL structures (coarsely and finely meshed LA1074 lenses, unrotated and rotated cube, and *The Thinker*) were compared for both voxelized and voxel-free projection methods. Printing resolution in CAL is mostly concerned with the reconstruction accuracy at the surface of the object. Thus, visual comparisons will focus primarily on the smoothness and gradient of the dose of light delivered at the surface, although comments will also be made on differences in the interior dose. All reconstructions have been normalized so that the maximum dose is one.

Vertical cross-sections for perfect reconstructions of the delivered dose for the finely meshed lens on the highest resolution (28.8 µm pixel size) are shown below for both voxel-free and voxelized Radon transform methods (Figure 2).

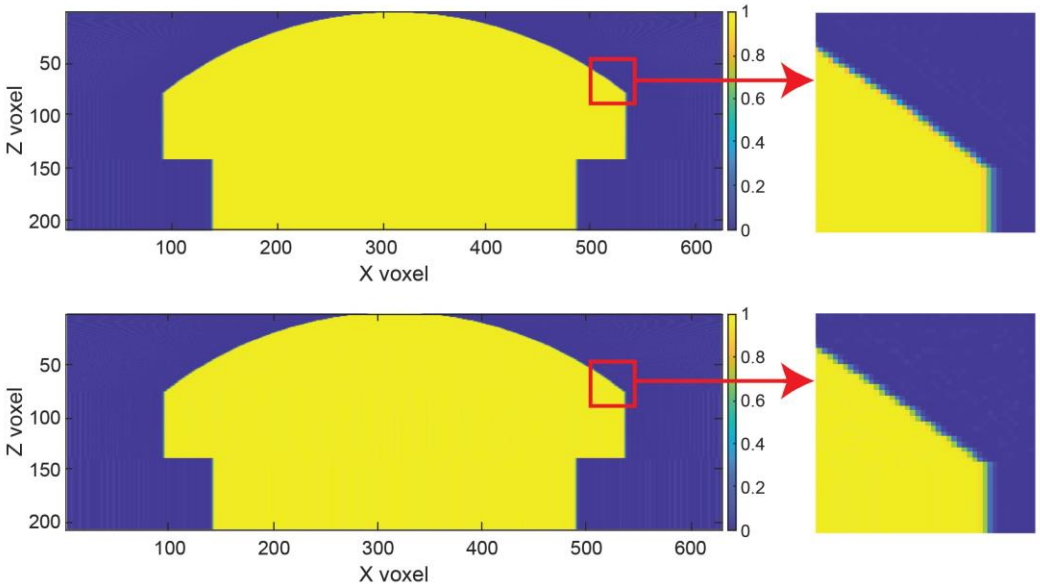

Figure 2. Vertical cross sections of "perfect" FBP reconstructions of the LA1074 lens with different forward projection methods. Top – voxel-free, bottom – voxelized

Differences in dose near the surface of the object can be seen between the two methods, although they are visually minor. We can further look at comparisons using reconstructions of the clipped projections that set negative intensities to zero. This depicts a possible configuration when printing as the perfect reconstructions cannot be simply achieved by projecting the original images due to the presence of negative intensities resulting from the Ram-Lak filter. This comparison is provided below in Figure 3.

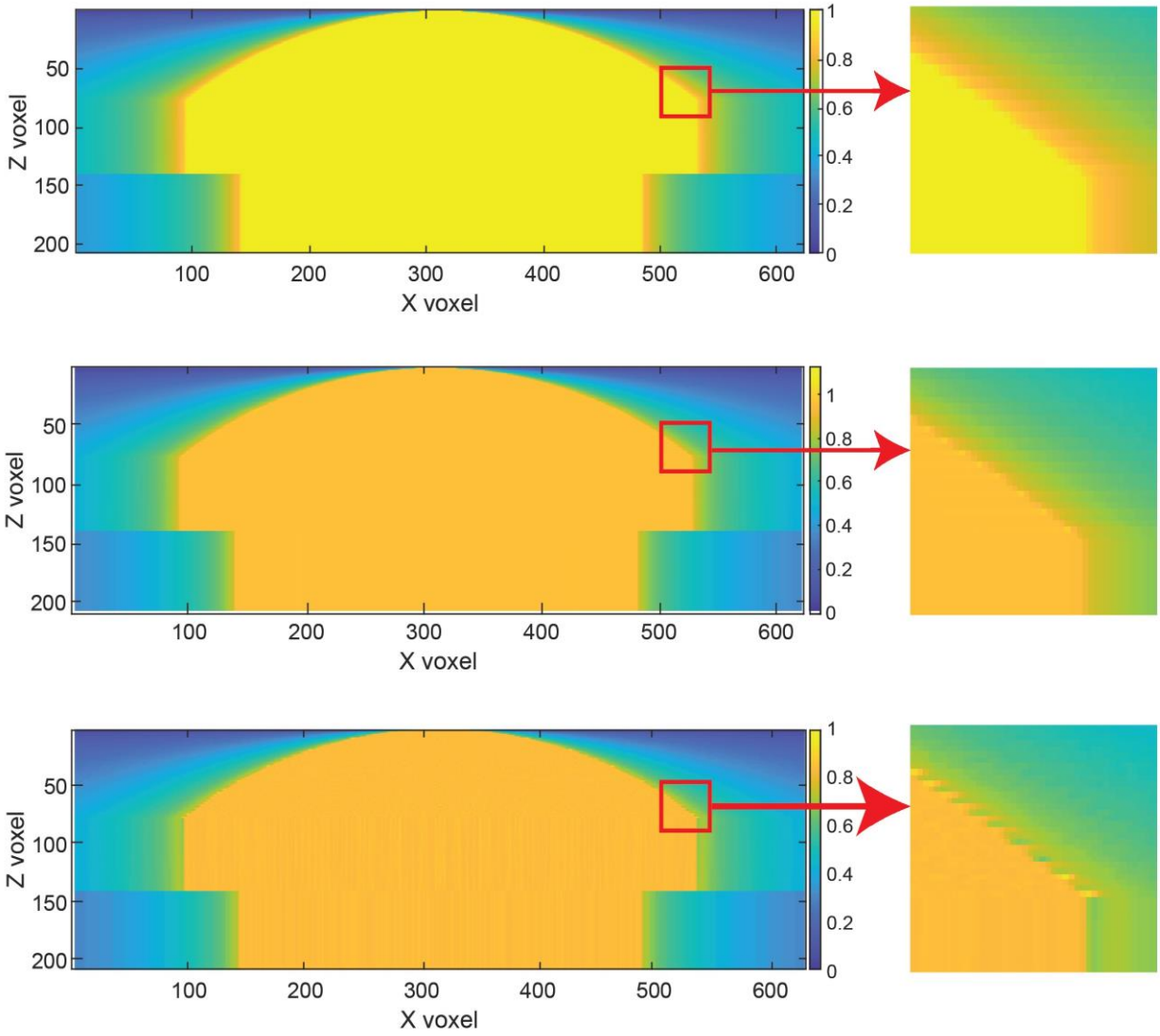

Figure 3. Vertical cross sections of "clipped" FBP reconstructions of the LA1074 lens with different forward projection methods. Top – voxel-free, middle – voxel-free, colorbar adjusted, bottom – voxelized

One immediate difference is that the voxel-free method delivers a higher relative light dose to the interior of the lens. To make comparisons of the surface easier, the colorbar for the voxel-free method was adjusted so that the interior dose better matches the result from the voxelized method. While differences in relative interior dose are not inherently a concern, the lower relative dose in the voxelized reconstruction is caused by individual "hot spots" of higher dose seen at the reconstruction surface that may cause non-uniform curing. The voxel-free transform creates a smoother dose at the surface, although the gradient between in-part and out-of-part doses is similar. The voxelized method exhibits periodic layers that are larger than desired and jut out from the bulk

volume. Some of these protrusions have particularly high doses exceeding the average dose within the object, leading to a lower relative interior dose. Additionally, the voxel-free method also produces a smoother interior dose free of the vertical banding seen in the lower parts of the lens with the voxelized method. This result can be further analyzed by looking at reconstructions of individual layers. Perfect and clipped reconstructions of the center layer of the lens are seen in Figures 4 and 5 below.

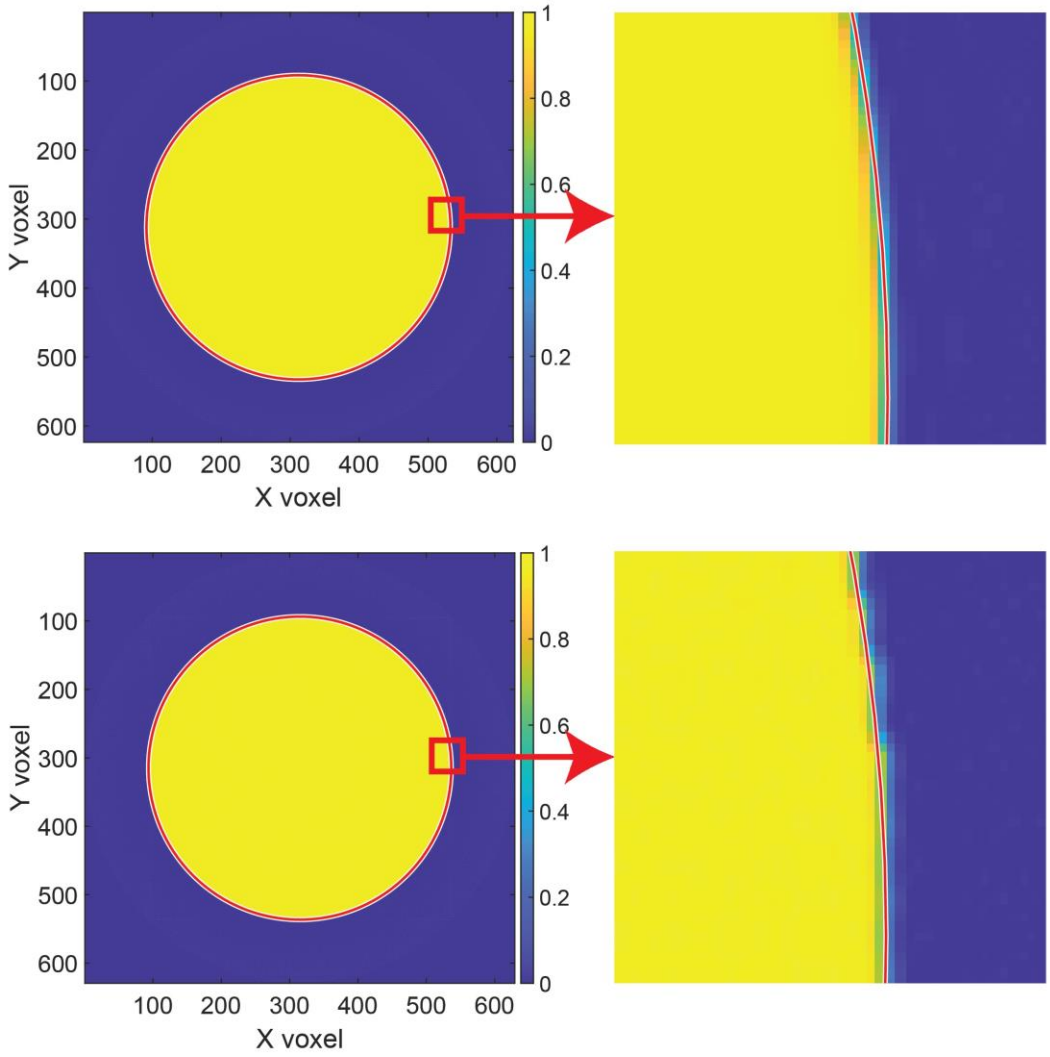

Figure 4. Horizontal cross sections of "perfect" FBP reconstructions of the LA1074 lens with different forward projection methods. Top – voxel-free, bottom – voxelized

Figure 4 depicts a layer in the wider cylindrical portion of the lens. The red line shows the desired location of the surface. This line is based on the desired CAD geometry, and not the input STL. However, the STL tolerance of 8 µm is much finer than a single pixel so comparisons to the line can still be made. The voxel-free method exhibits better conformity to the desired shape, particularly in the upper part of the zoomed-in region. As with the vertical cross sections, the differences are more noticeable in the clipped reconstruction, as seen in Figure 5 below.

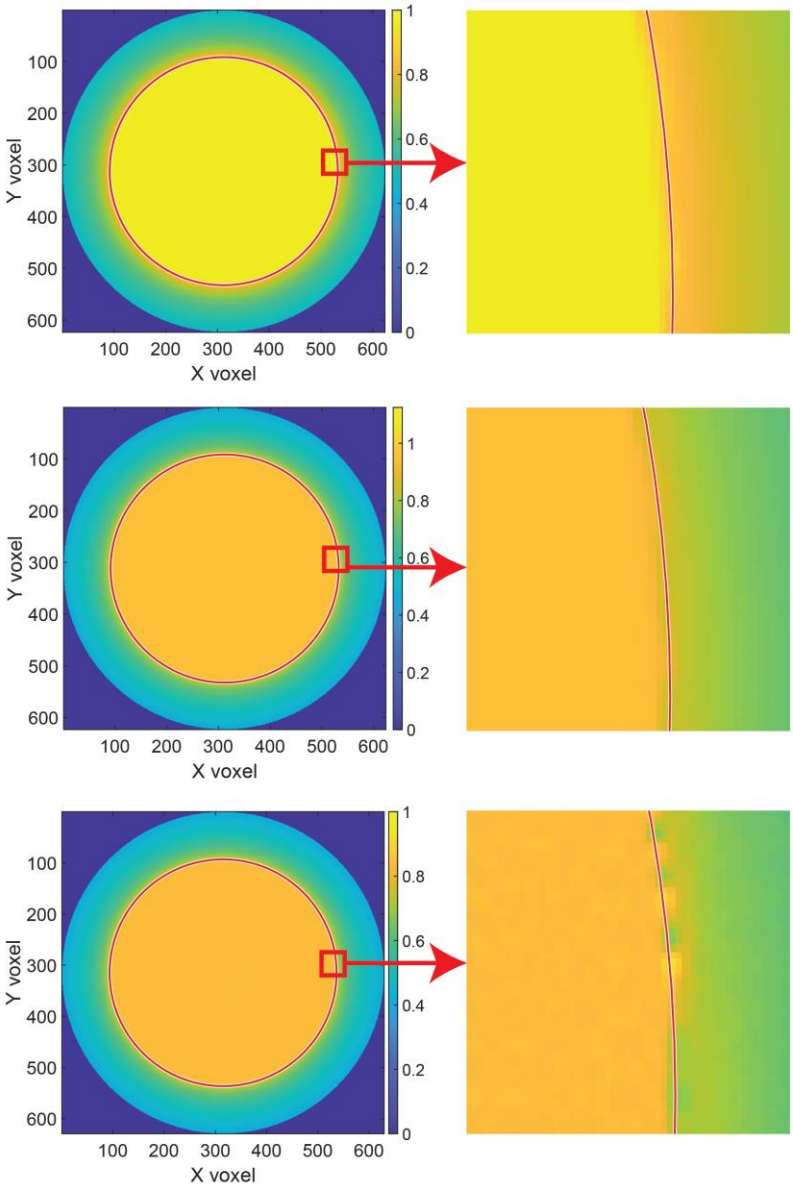

Figure 5. Horizontal cross sections of "clipped" FBP reconstructions of the LA1074 lens with different forward projection methods. Top – voxel-free, middle – voxel-free, colorbar adjusted, bottom – voxelized

As seen in the vertical cross sections, the voxel-free method produces a higher, smoother relative light dose in the interior of the object. In the clipped reconstructions, the difference in surface smoothness is apparent, with the voxelized method showing protrusions that are potentially consistent with what is seen in Figure 3. As with the vertical cross sections, the voxel-free method shows a smoother dose at the surface of the object, and a similar gradient between the in-part and out-of-part dose.

These tests were conducted for the lens across five different pixel sizes and two different STL resolutions. While Figures 2-5 presents results from the finely meshed STL at the smallest pixel size tested (28.8 µm), Figure 6 below shows reconstructions of the center layer of the lens for the coarsely meshed STL (22 µm tolerance) at the same pixel size, as well as for the finely meshed STL (8 µm tolerance) at ten times this pixel size.

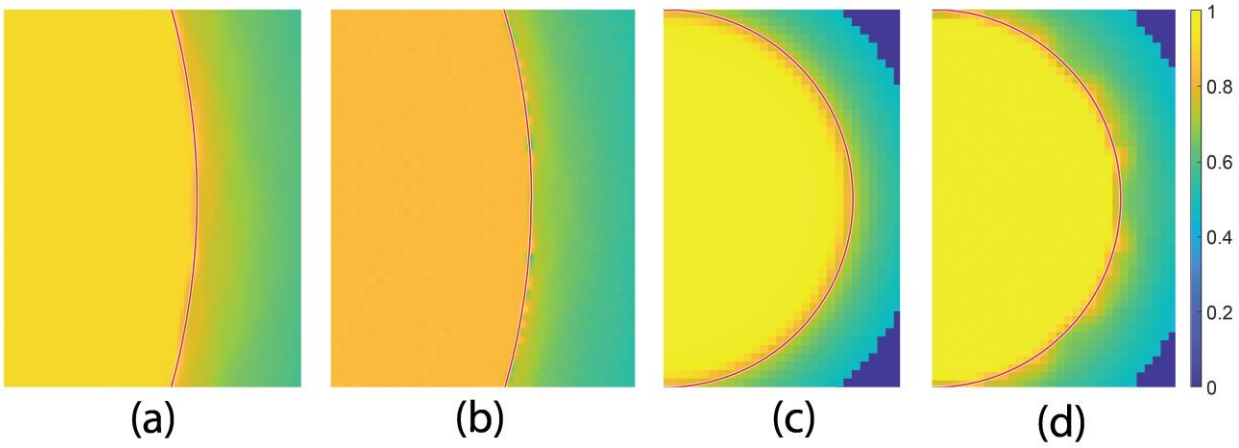

Figure 6. "Clipped" dose reconstructions of horizontal cross sections for different STL meshes and pixel sizes. (a) Voxel-free, 28.8 µm pixel size, coarse STL, (b) Voxelized, 28.8 µm pixel size, coarse STL, (c) Voxel-free, 288 µm pixel size, fine STL, (c) Voxelized, 288 µm pixel size, fine STL

Figure 6 shows that the differences seen in Figures 2-5 are still seen at other pixel sizes and STL mesh fineness, albeit to varying degrees. The smoothness and gradient of the dose at the object's surface are similar for the coarsely meshed STL in Figures 6a and 6b when compared to the earlier figures using the finely meshed STL. This is especially true for the voxelized image in 6b, possibly due to the tolerance of the coarse STL mesh still being smaller than the voxel size. Differences between the voxel-free and voxelized methods are more apparent at the larger pixel size, with the voxel-free method maintaining a smoother surface with better conformity to the desired geometry.

These comparisons were also made for two configurations of a 10 mm side-length cube, one configuration with the faces towards the primary axis, and another with the cube rotated 45° in the XY plane. This was done to investigate the sensitivity of both methods to rotation. The hypothesis is that the voxelized method can perfectly locate the surface in the first case but may struggle more with the second. Reconstructions of a horizontal cross-section for the unrotated cube are shown in Figure 7 below. As with Figures 2-5, the pixel size used was 28.8 µm.

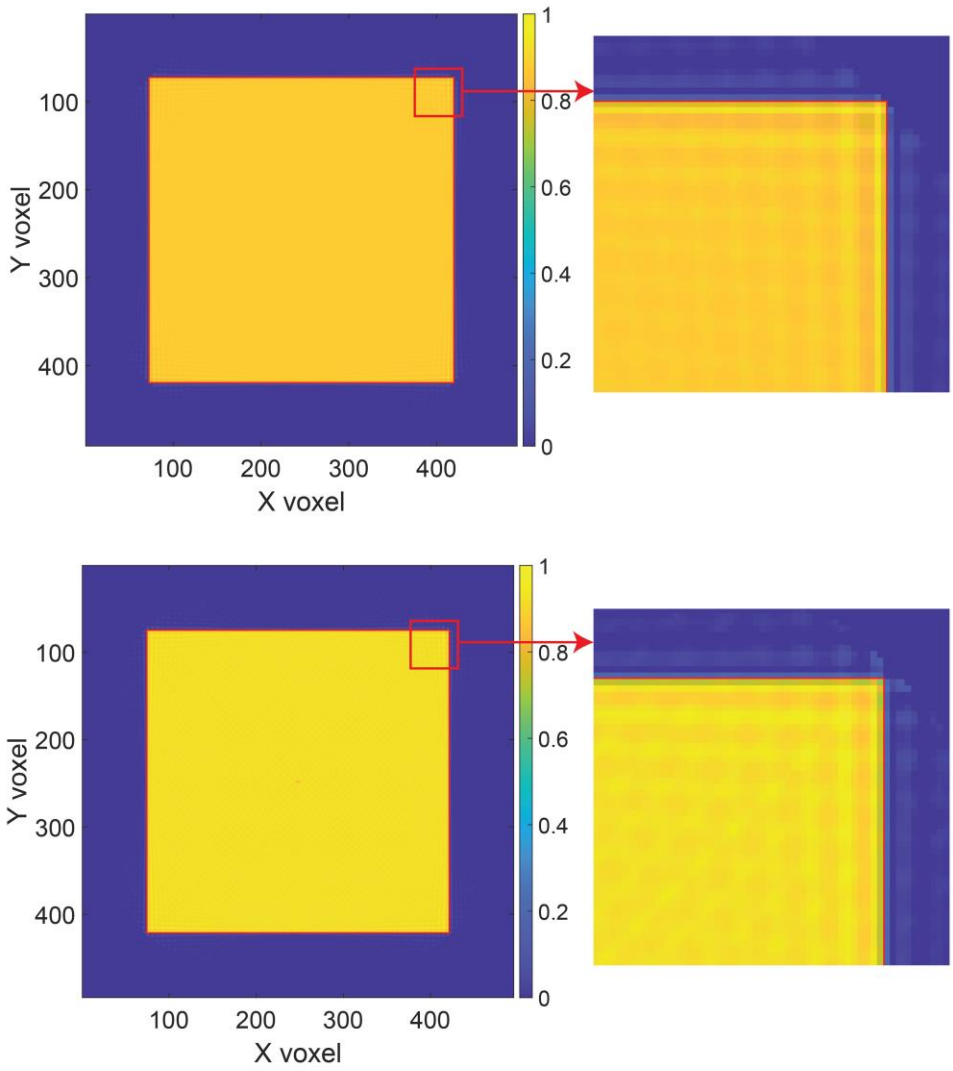

Figure 7. Horizontal cross sections of "perfect" FBP reconstructions of the unrotated cube with different forward projection methods. Top – Voxel-free, bottom – voxelized

In this case, the surface is well-defined in both methods and matches perfectly with a line of voxels. Despite differences in average dose, no individual hot spots are visible. This comparison is also made for the rotated cube in Figure 8.

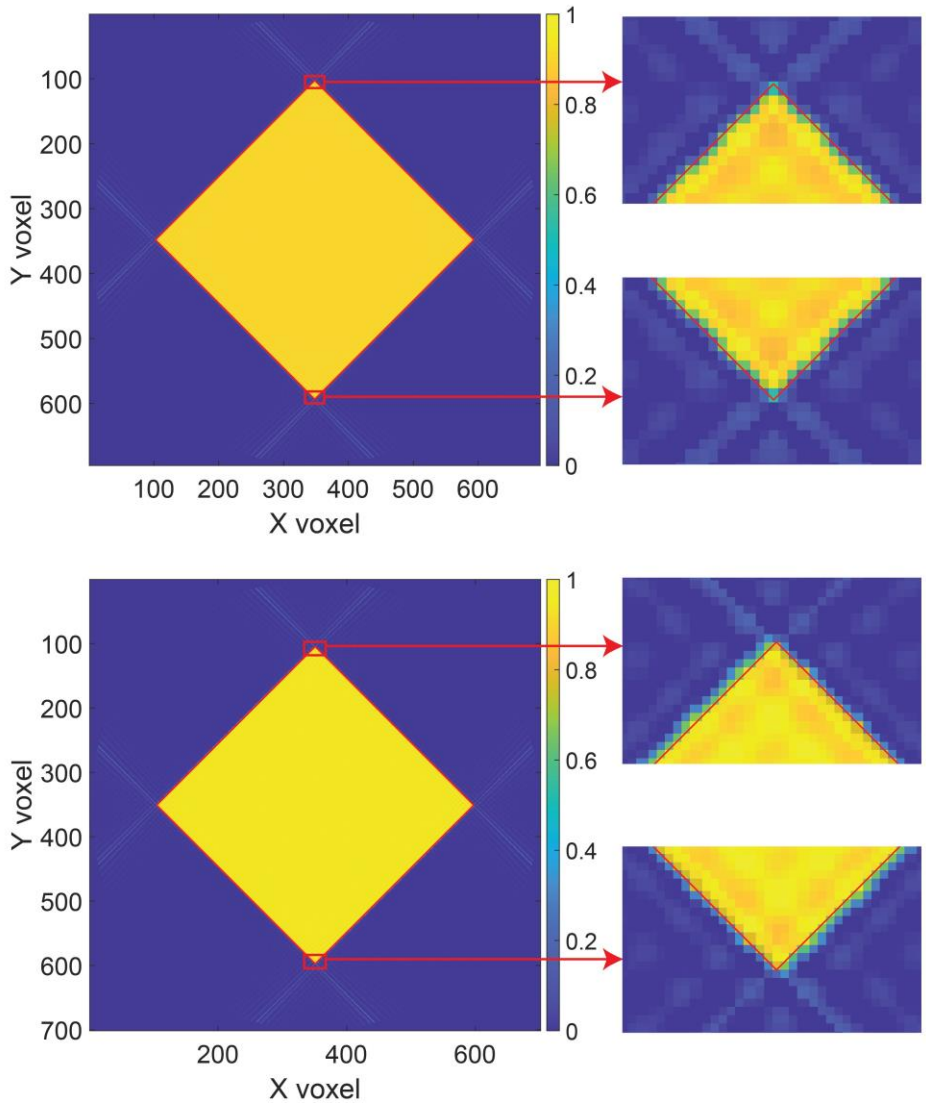

Figure 8. Horizontal cross sections of "perfect" FBP reconstructions of the rotated cube with different forward projection methods. Top – Voxel-free, bottom – voxelized

Unlike in Figure 7, the surface of the rotated cube does not align cleanly with a line of voxels in either method. Despite this effect, the voxel-free method maintains the symmetry of the cube in all directions. The voxelized method, however, creates a half-voxel shift of the vertices in opposite directions, leading to a longer-than-desired object in one direction and a shorter-than-desired object in the other. As with Figure 7, the voxel-free method creates a lower average relative dose, but no clear hot spots are visible.

Lastly, projections and reconstructions were taken for Rodin's *The Thinker* to investigate performance for more complex geometries. Both perfect and clipped reconstructions of a vertical cross-section are depicted in Figure 9. As before, the pixel size used was 28.8 µm.

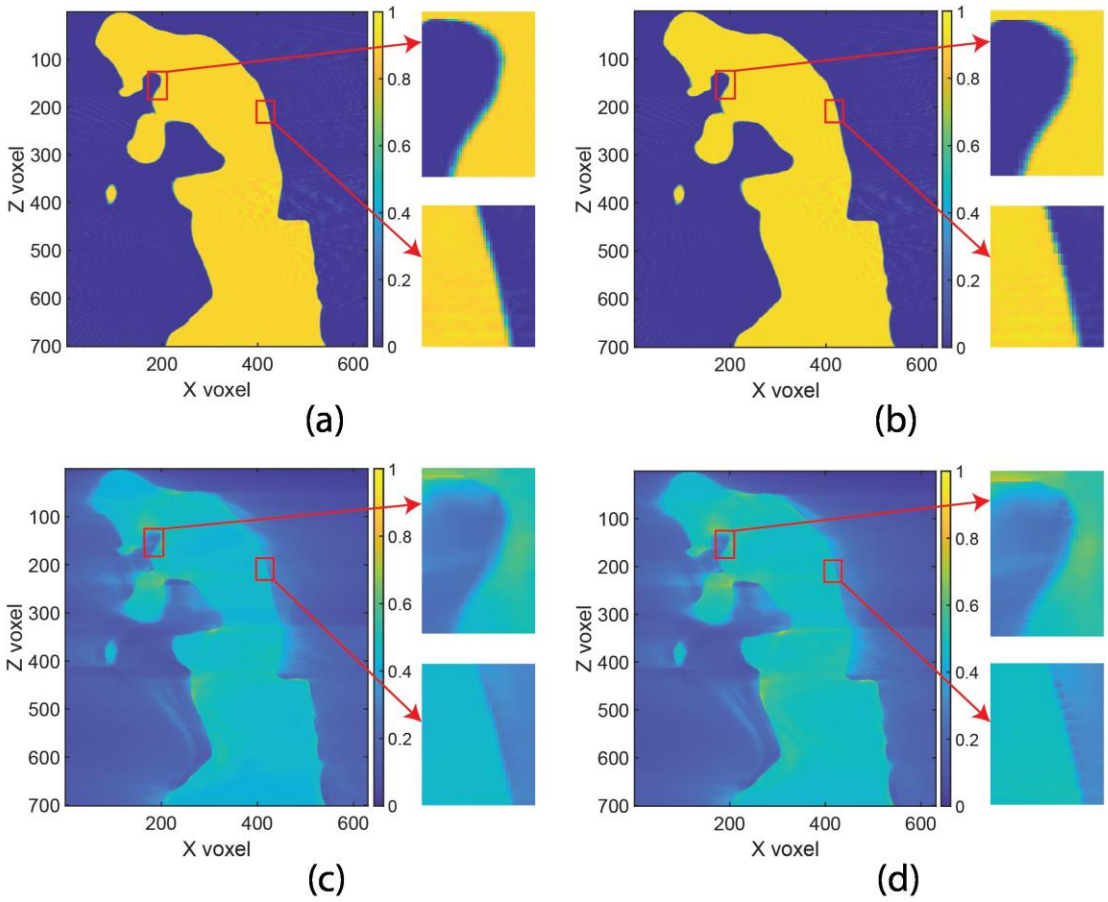

Figure 9. Vertical cross sections of FBP reconstructions of The Thinker with different forward projection methods (a) Voxel-free perfect reconstruction (b) Voxelized perfect reconstruction (c) Voxel-free clipped reconstruction (d) Voxelized clipped reconstruction

Comparing the voxel-free and voxelized reconstructions for *The Thinker*, the voxel-free reconstruction shows smoother surfaces for both perfect and clipped reconstructions. The gradient of the dose at the surface is similar for the two methods. The interior dose is slightly lower for the voxel-free method in this geometry, but as with the cube, no unique hot spots are visible. Compared to the lens, the interior of the object is less uniform in dose across both methods. As with the lens, we can also look at reconstructions of a horizontal layer, shown in Figure 10.

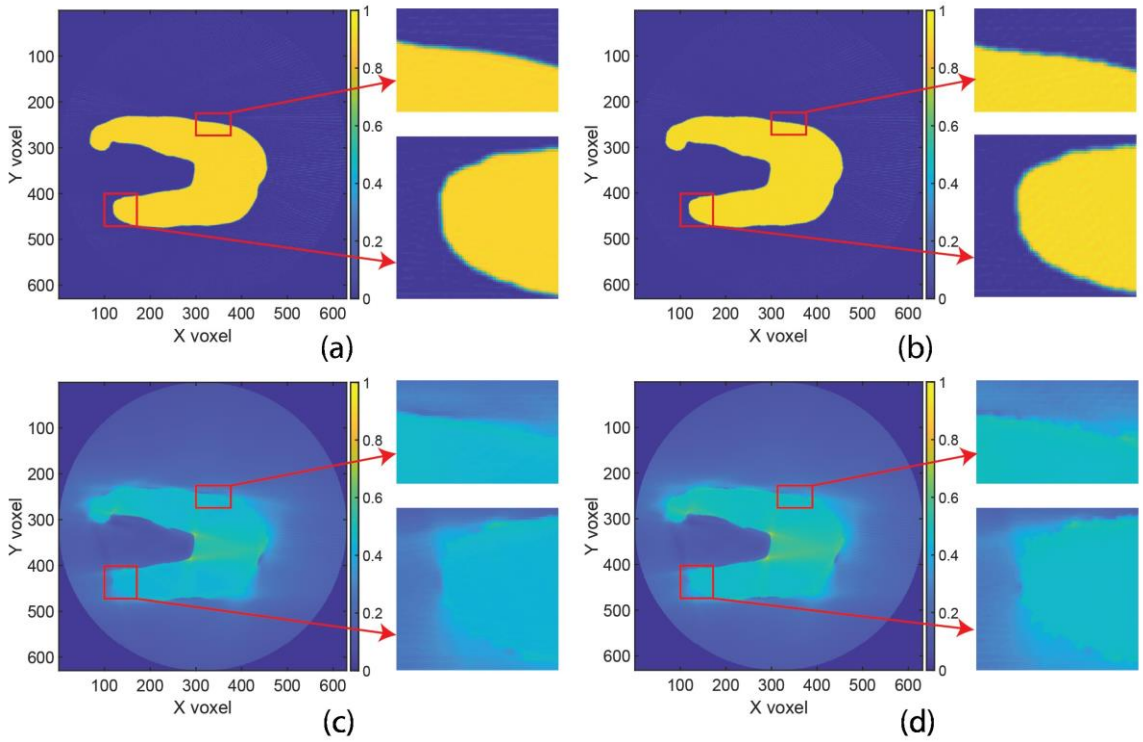

Figure 10. Horizontal cross-sections of FBP reconstructions of The Thinker with different forward projection methods (a) Voxel-free perfect reconstruction (b) Voxelized perfect reconstruction (c) Voxel-free clipped reconstruction (d) Voxelized clipped reconstruction

The differences seen in Figure 9 remain when looking at the horizontal cross-sections, with the voxel-free method producing a noticeably smoother result.

In addition to reconstruction accuracy comparisons, the time taken for each method was compared for all five STLs and five pixel sizes. All code was run in MATLAB and two methods for generating the voxelized results were included, one using the CPU, and the other using GPU acceleration. A log-log plot of the time to generate the projection sets is plotted against 1/(voxel side length). The x-axis in this case is nearly proportional to the number of pixels in each direction of the projections, as well as the number of voxels in each direction for the reconstructed object. These results are shown below in Figure 11.

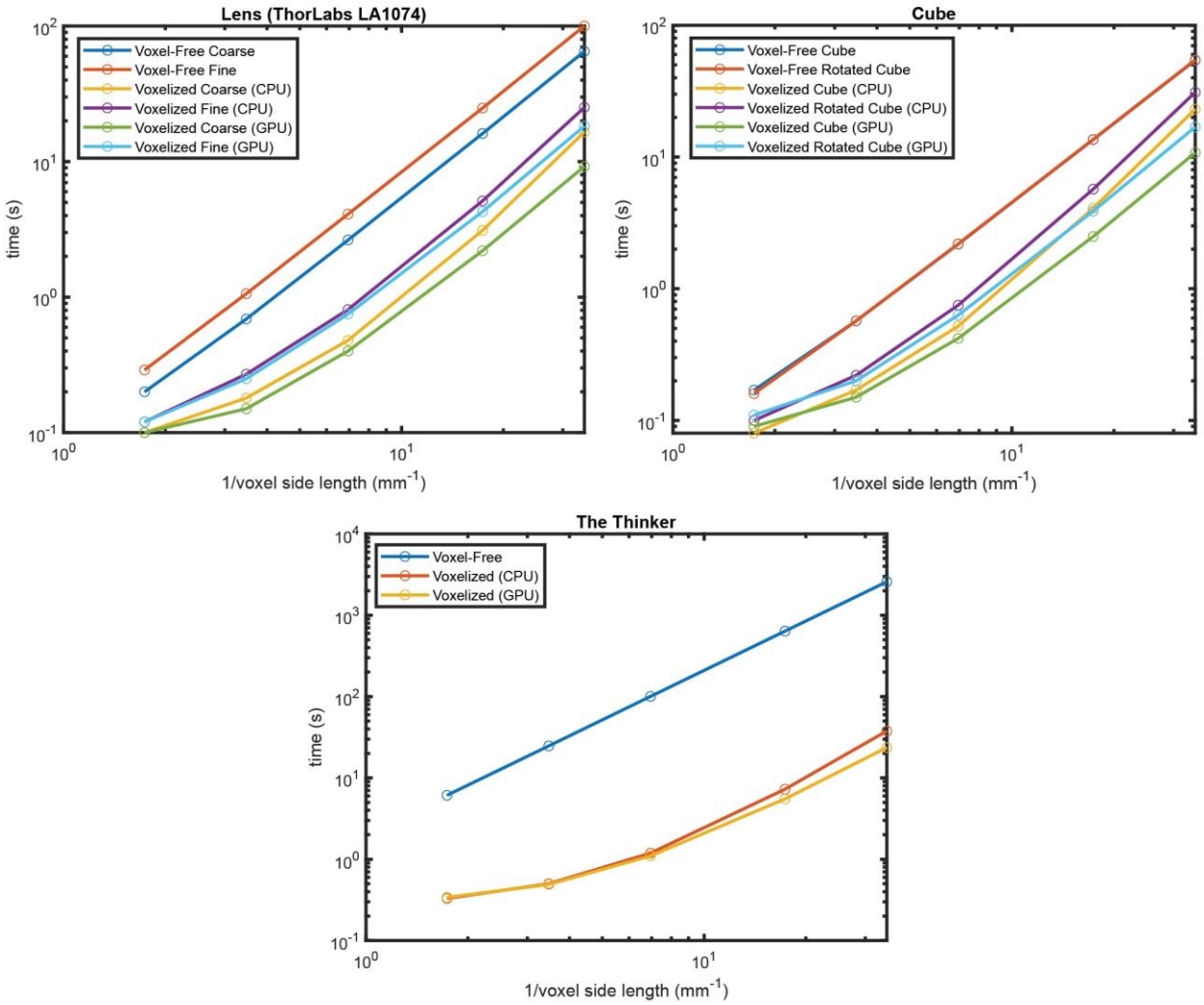

Figure 11. Time vs 1/side length comparisons

In its current implementation, the absolute speed of the voxel-free method is slower in all cases tested. However, the slope of the log-log plot is lower in all cases for the voxel-free method once getting to smaller voxel side lengths. The CPU method for the voxelized transform is slower than the GPU one, especially as voxel side lengths decrease. Differences in computation times are proportionally similar for both methods when comparing coarse and fine STLs of the LA1074 lens. For the cube, the rotated cube takes longer than the unrotated one in the voxelized methods but is effectively the same for the voxel-free method. The overall difference in time between voxel-free and voxelized methods is the greatest for *The Thinker*.

Discussion

Successful printing in CAL is often concerned with accurately controlling the light dose at the surface of the object. The interior of a printed object often cures before the surface due to the greater competition for free-radical quenching oxygen that occurs when all surrounding material is receiving a substantial light dose. The reconstructions presented in this work does not account for this effect, so we expect the surface of the object to solidify last. Ideal characteristics of reconstructed objects include a smooth surface with a high spatial dose gradient so that the surface of the part is well-formed without any over-curing or under-curing in the vicinity and with limited sensitivity to processing variations. While the focus is largely on the surface of the object, interior dose uniformity may lead to more simultaneous curing of the part, potentially reducing striations and deformities, although this needs to be tested. Note that since all reconstructions are normalized so the highest dose is one, a higher relative dose may not result in a higher absolute dose. Further testing can be done to investigate difference in absolute dose, which affects the print time of objects.

Across all geometries, the voxel-free method yielded a smoother surface with better conformity to the desired geometry, while the gradient of dose at the surface was similar in all cases for the two methods. In the lens, the interior dose was smoother in the clipped reconstructions with the voxel-free method, although this was not consistently shown in other geometries. Individual hot spots can be seen at the surface of the lens for the voxelized method. This may cause uneven curing when printing. A similar effect is seen in clipped reconstructions for both methods in *The Thinker*, although the higher dose areas are largely still in the interior of the object.

The voxelized method also produced a slightly larger reconstruction in some areas of the lens, as well as an offset result in the cube. This may be explained by voxelization precision. Horizontal cross sections of a voxelized target for the lens and rotated cube are shown in Figure 12 below. For ease of visualization, a higher voxel size of 288 µm has been used.

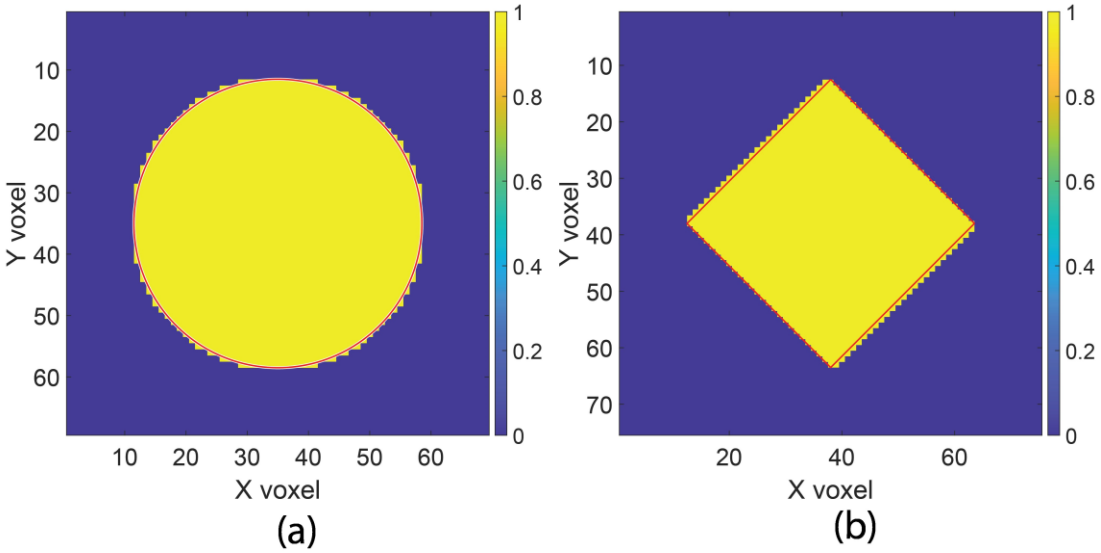

Figure 12. Voxelized target horizontal cross sections. (a) LA1074 lens (b) rotated cube. 288 µm pixel size

The above targets explain some of the errors seen in the voxelized method. The voxelizer used counts a voxel as in-part if any part of the voxel is in the interior of the desired geometry. Thus, voxels with even a small section in part are set to one in the target, resulting in a slightly

larger target in some areas, as seen in Figure 12a. While the voxel-free Radon transform still results in a pixelated image set and voxelized reconstruction, it can perfectly calculate the intersection length through one line instead of being discretized to a voxel length. For the cube, the offset result seen in Figure 8 is again shown in Figure 12b.

Comparing the speed of the methods, the voxel-free method is currently slower in all tested cases. However, the slope of the log-log plot is lower for the voxel-free method when getting to smaller and smaller voxel sizes across all STLs. The time required for the voxel-free method scales closely with the inverse square of the voxel size as the number of rays traced through the object is proportional to the radius of the object and the height of the object in voxel counts. Note that reducing pixel size has the same effect on time as proportionately increasing the object size. This is because the targets and projection sets have an equivalent number of pixels/voxels in both cases. This was also verified by obtaining projection sets for a half-size Thinker at 288 μm and 144 μm pixel sizes, which took the same amount of time as the full-size Thinker at 576 μm and 288 μm pixel sizes respectively. Thus, this scaling aspect is important as the size of objects is increased. Additionally, the voxel-free method is invariant in speed with different object rotations in the XY plane as it uses the maximum vector norm in the XY plane of the STL as its radius, while the voxelized method uses the maximum dimensions in the X and Y directions.

While the voxel-free method uses a ray-triangle intersection routine that calculates the intersections of one ray to all triangles in the STL at once, it still scales less favorably with the number of triangles in the STL than the voxelized method. This is less of a concern for scaling up sizes in CAL since the minimum feature sizes of an object scale proportionally to the object size. Thus, the STL tolerance may be scaled up accordingly, maintaining a similar number of triangles in the STL for a given geometry. However, as the complexity of geometries increases, increasing the number of triangles, the voxel-free method will lag behind the voxelized method more in its current implementation.

Although formal tests of memory were not conducted in this work, the voxel-free method may be beneficial in this regard. The voxelized target has the same dimensions as the reconstructed object, that is (R, R, Z), while the projection sets are in dimension (R, θ, Z). Thus, if the number of projections remains the same, the size of the voxelized target quickly outgrows that of the projection sets as object sizes increase. Even the lens, which is the smallest structure tested, has target dimensions of (623, 623, 208) and sinogram dimensions of (623, 360, 208) at the 28.8 μm pixel size. Thus, the lack of a voxelized target may result in a large reduction in memory required. However, the voxel-free method, in its current form, does use two other matrices of size (R, θ/2, Z) for the origin and direction matrices of the rays. In addition, θ may also need to be increased as the size of the object increases to maintain the desired resolution. Irrespective of the size of the object, the voxel-free method also uses the maximum vector norm in the XY plane as the radius instead of calculating from the maximum dimensions in the X and Y directions, decreasing the value of R by up to $2^{1/2}$-1, or 41%. In the results presented here, the projections from the voxel-free method were padded to match the dimensions of the voxelized method for comparison.

Conclusion/Future Work

The voxel-free Radon transform presented in this paper produces projections that create smoother reconstructions with better conformity to the desired geometry. Additionally, this method potentially scales better with object size when it comes to speed. Since the voxel-free method is slower in the tested cases than traditional methods, work may also be done to improve the execution time of the code, possibly using GPU acceleration or an adaptive grid fineness. As accuracy comparisons were made visually in this work, future work can attempt to quantify the critical parameters of average dose, dose gradient, and standard deviation of dose at the surface of the object in a non-voxelized manner, such as at the vertices and centroids of the input STL file. Additionally, comparisons here used the simplest form of clipping the negative intensities to zero. Future work may experiment with different linear offsets in addition to this clipping, as well as a new field of voxel-free optimization methods.

References

[1] B. E. Kelly, I. Bhattacharya, H. Heidari, M. Shusteff, C. M. Spadaccini, and H. K. Taylor, "Volumetric additive manufacturing via tomographic reconstruction," *Science*, vol. 363, no. 6431, pp. 1075–1079, Mar. 2019, doi: 10.1126/science.aau7114.

[2] K. Coulson, J. Toombs, M. Gu, and H. Taylor, "Adaptive Voxelization for Rapid Projection Generation in Computed Axial Lithography," University of Texas at Austin, 2021. doi: 10.26153/tsw/17642.

[3] R. Roopkumar, "Generalized Radon Transform," *Rocky Mt. J. Math.*, vol. 36, no. 4, pp. 1375–1390, 2006.

[4] T. Möller and B. Trumbore, "Fast, minimum storage ray/triangle intersection," in *ACM SIGGRAPH 2005 Courses*, in SIGGRAPH '05. New York, NY, USA: Association for Computing Machinery, Jul. 2005, pp. 7-es. doi: 10.1145/1198555.1198746.

[5] J. Tuszynski, "Triangle/Ray Intersection," *MATLAB Central File Exchange*, Jun. 19, 2023. https://www.mathworks.com/matlabcentral/fileexchange/33073-triangle-ray-intersection (accessed Jun. 19, 2023).

[6] S. A. Qureshi, S. M. Mirza, and M. Arif, "Inverse Radon Transform-Based Image Reconstruction using Various Frequency Domain Filters in Parallel Beam Transmission Tomography," in *2005 Student Conference on Engineering Sciences and Technology*, Aug. 2005, pp. 1–8. doi: 10.1109/SCONEST.2005.4382887.

[7] C. M. Rackson *et al.*, "Object-space optimization of tomographic reconstructions for additive manufacturing," *Addit. Manuf.*, vol. 48, p. 102367, Dec. 2021, doi: 10.1016/j.addma.2021.102367.

[8] "On the use of windows for harmonic analysis with the discrete Fourier transform | IEEE Journals & Magazine | IEEE Xplore." https://ieeexplore.ieee.org/abstract/document/1455106?casa_token=eA0TlC9_r0UAAAAA:vRkaDdO-knsOM_-IxzaW2EgPu0pdJQHsdhGhBmBp4H_Aq6gKnVSF5JTy8G4T-TkX1R1SZq1pSg (accessed Jul. 14, 2023).

[9] R. Clackdoyle and M. Defrise, "Tomographic Reconstruction in the 21st Century," *IEEE Signal Process. Mag.*, vol. 27, no. 4, pp. 60–80, Jul. 2010, doi: 10.1109/MSP.2010.936743.

[10] W. J. Palenstijn, K. J. Batenburg, and J. Sijbers, "Performance improvements for iterative electron tomography reconstruction using graphics processing units (GPUs)," *J. Struct. Biol.*, vol. 176, no. 2, pp. 250–253, Nov. 2011, doi: 10.1016/j.jsb.2011.07.017.