Solid Freeform Fabrication 2021: Proceedings of the 32nd Annual International
Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference
Reviewed Paper

ADAPTIVE VOXELIZATION FOR RAPID PROJECTION GENERATION IN COMPUTED AXIAL LITHOGRAPHY

Kevin Coulson, Joseph Toombs, Magnus Gu, Hayden Taylor

Department of Mechanical Engineering, University of California, Berkeley

Abstract

Computed axial lithography (CAL) is a tomographic additive manufacturing technology that offers exceptionally fast printing in a wide range of materials. CAL involves pre-computing a sequence of light patterns to be projected into a photopolymer. For a uniform spatial discretization of the target geometry, computational time scales inversely with the cube of the discretization pitch, which makes it challenging to exploit the full space-bandwidth product of available spatial light modulators. This work introduces an adaptive voxelization approach to reduce computational expense. Using one of several proposed mesh-based complexity analyses, a CAD model is recursively subdivided into stacked sub-meshes of varying voxel resolution. These complexity methods can be tailored to emphasize complexity in particular regions. Each sub-mesh is then independently voxelized before projections are generated and optimized in parallel. On a four-core CPU, this method results in a $2-6\times$ speedup with applications in high-precision CAL and other voxel-based additive manufacturing computations.

Introduction

Computed axial lithography (CAL) is a photopolymer resin-based volumetric additive manufacturing (VAM) method which pre-computes a projection sequence based on an inverse computed axial tomography algorithm [1]. During CAL printing, 2D images are projected through a rotating cylindrical volume such that the cumulative energy dose is sufficient to polymerize the target geometry. To resolve fine details and achieve a smooth surface finish, it is desirable to use high-resolution images during projection. Currently, the 2D images are pre-computed based on a uniform spatial discretization, or voxelization, of the target geometry – referred to as the ‘standard’ approach. However, the computational time of projection generation scales inversely with the cube of the discretization pitch. This limits the accessibility of high-resolution CAL printing despite improvements in the achievable optical resolution of the system.

Voxel-based methods are becoming more prominent in pre-processing for additive manufacturing. While the .STL file format can more precisely describe the surface of a given geometry, the simplicity of the voxel format - the 3D analog to a pixel image - enables the use of complex computational techniques to improve AM processes. Examples include lattice generation [2], optimized build orientation [3], energy and error optimization [4], and tomographic reconstruction itself [1]. Additionally, adaptive resolution voxelization, in which the voxel size is non-uniform, has previously been applied in feature recognition. In one application, a point cloud representation of a 3D-scanned ear is analyzed and recursively subdivided according to proximity functions which capture the nearness of every point to a set of keypoints [5]. This subdivision has the effect of emphasizing distinct surface features.

Several layer-based AM processes have begun to utilize adaptive layer heights for more efficient printing, including digital light processing [6] and fused deposition modeling [7]. These methods analyze the contour of the target geometry and calculate a set of layer heights required to maintain an error threshold based on the intrinsic surface errors introduced by a given AM process. The goal of adaptive layer height methods is to improve print speed while maintaining reasonable accuracy.

In this work, an adaptive voxelization scheme is constructed to enable efficient projection sequence generation for CAL printing. This process dramatically improves computational speed while maintaining reconstruction accuracy (Figure 1). First, several mesh-based complexity methods, which are used to identify complex features of the target geometry, are introduced and analyzed. The results of this analysis drive the recursive subdivision of a CAD model into stacked sub-meshes which are then voxelized at non-uniform resolutions. Then, the algorithms used to generate and reconstruct the final projection sequence - including a parallel processing implementation - are described. Finally, computational speed and reconstruction accuracy are evaluated with respect to the current state-of-the-art.

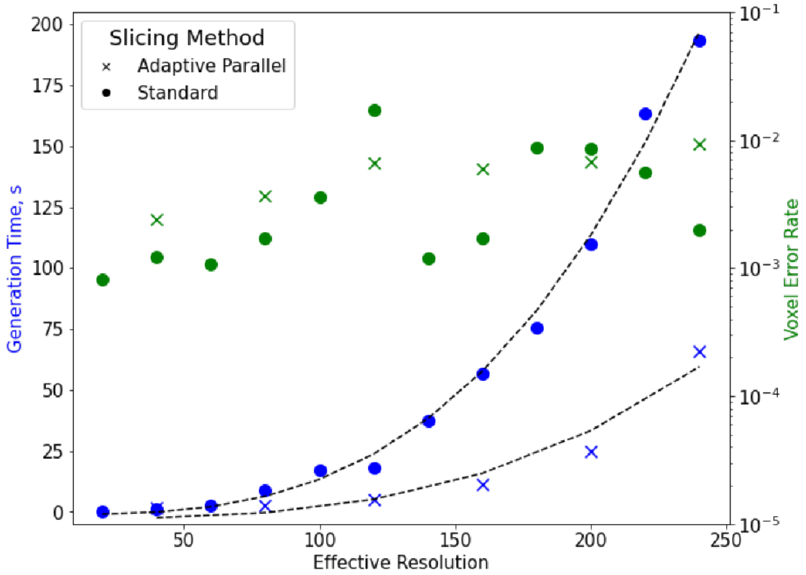
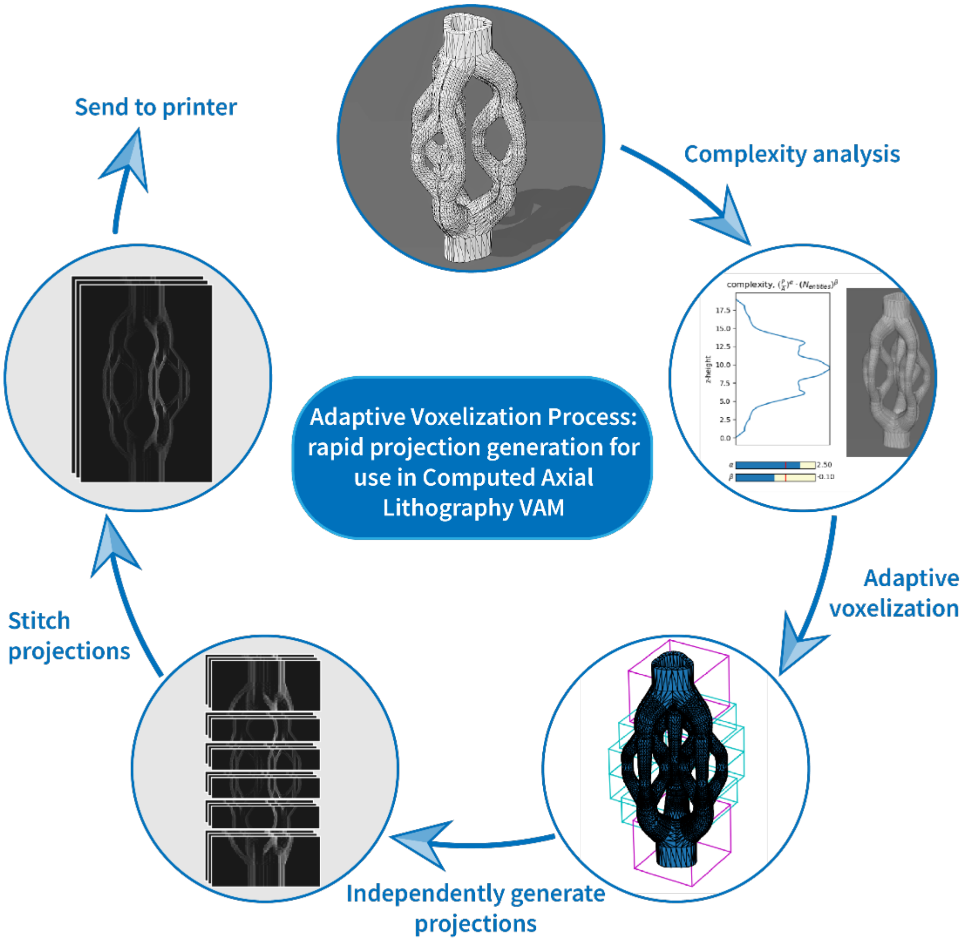


Figure 1. Cubic curves are fit to the projection generation time data for the standard and parallel processed adaptive projection computation methods. Effective resolution for the adaptive process is the base resolution for each sub-mesh multiplied by the number of sub-meshes. For the standard process, effective resolution is the same as the resolution. The voxel error rate, a measure of predicted over-exposure, shows no consistent trends. However, the errors for each approach are within an order of magnitude. This demonstrates the benefit of the adaptive voxelization approach – significant computational speed-up without sacrificing accuracy.

Methods

The process for utilizing the proposed adaptive voxelization scheme is detailed in Figure 2. The complexity analyses and projection stitching are built around the pre-existing process for projection computation developed for CAL printing.



Send to printer
Complexity analysis
complexity, $(\frac{P}{A})^{\alpha} \cdot (N_{entities})^{\beta}$
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
z-height
α
2.50
β
-0.10
Adaptive Voxelization Process: rapid projection generation for use in Computed Axial Lithography VAM
Adaptive voxelization
Stitch projections
Independently generate projections

Figure 2. A process outline for the adaptive voxelization approach.

Complexity Analysis and Subdivision

The goal of an adaptive voxelization algorithm is to make more efficient use of computational resources by using greater computed projection resolution for more complex sections of the target geometry. Therefore, methods of identifying complexity in a target geometry must be developed.

In its simplest physical representation, CAL has no interactions between voxels in the *z*-direction (the axis of rotation for the cylindrical resin volume), unlike some VAM technologies [8]. In this case, the computation associated with each horizontal row of the voxelized target is self-contained. Therefore, splitting the model using *z*-normal planes will enable independent projection sequence generation while losing no data.

Complexity analysis will ultimately drive the subdivision progression in the adaptive voxelization scheme. Figure 3 compares the layer-based complexity for several geometries using each proposed method, with the overall complexity as the integral under these curves. The simplest of the proposed complexity methods, 'n-vertices' and 'vertex density', are based on mesh properties. These methods do not require layer-based analysis and the total complexity is calculated by counting the number of vertices and calculating the vertex density of a mesh, respectively. These methods can provide a rough estimation of the complex regions of a part. Other metrics such as a bulk-volume[1]-to-volume ratio and surface-area-to-volume-ratio have been demonstrated to provide similar results to the vertices-to-volume ratio [9]. However, mesh-based definitions of complexity are not without limitations. Vertex and triangle metrics are dependent on mesh export resolution and mesh refinements. Surface area and volume metrics require watertight meshes, which can be difficult to maintain after subdivision because the new faces required to cap a sub-mesh at the split plane are not often trivial to create.

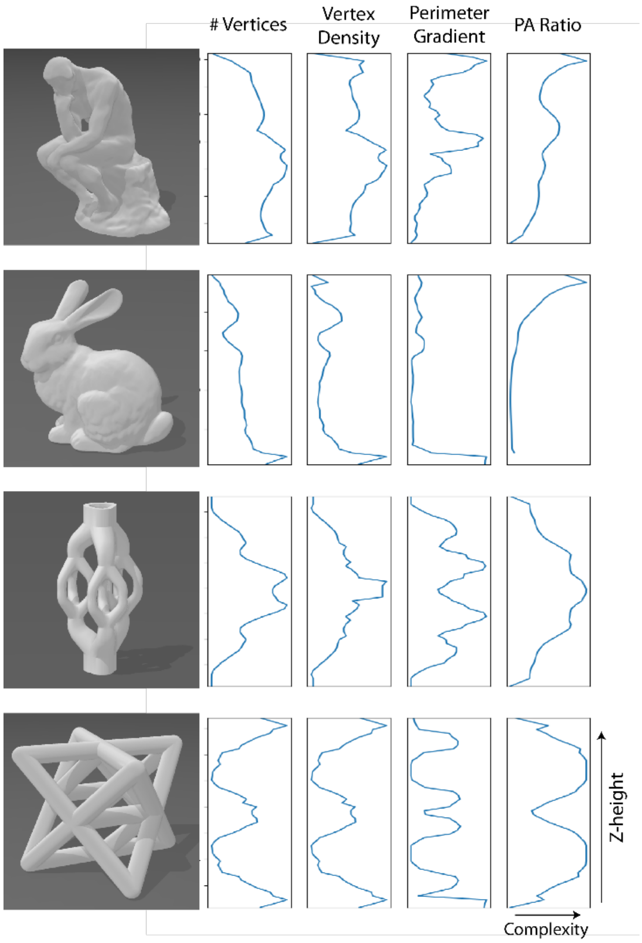


Figure 3. Complexity comparisons. Models from top to bottom: Rodin's Thinker, the Stanford Bunny, a triple vasculature for microfluidics, and an octet truss unit cell. Methods from left to right, 'n-vertices', 'vertex density', 'perimeter gradient', and 'perimeter-to-area ratio.'

[1] Bulk-volume is the sum of the volume of an object and the volume enclose by an object: e.g., the bulk volume of a hollow sphere is simply the volume of a solid sphere of the same radius.

The ‘perimeter gradient’ and ‘perimeter-to-area ratio’ are layer contour-based complexity methods. Using these metrics, the complexity is calculated by taking a sum of the complexity of all layers. Despite CAL’s departure from layer-based 3D printing, slicing the model can provide insight into the relative complexity along the z-axis of the target. The ‘perimeter gradient’ method uses the layer-to-layer difference in section perimeter to define regions of the geometry with the most significant cross-section changes as complex. The ‘perimeter-to-area ratio’ is calculated for each layer by dividing the perimeter of each layer contour by its area. In cases where there are multiple entities represented in a layer, the summed perimeter and summed area are used to compute the ratio. The perimeter-to-area ratio is simply calculated and the basis of more rigorous measures of shape complexity, like the fractal dimension index [10]. This method ultimately assigns high complexity to regions of a geometry with relatively small cross sections and, in theory, to layers with high fractal dimensions.

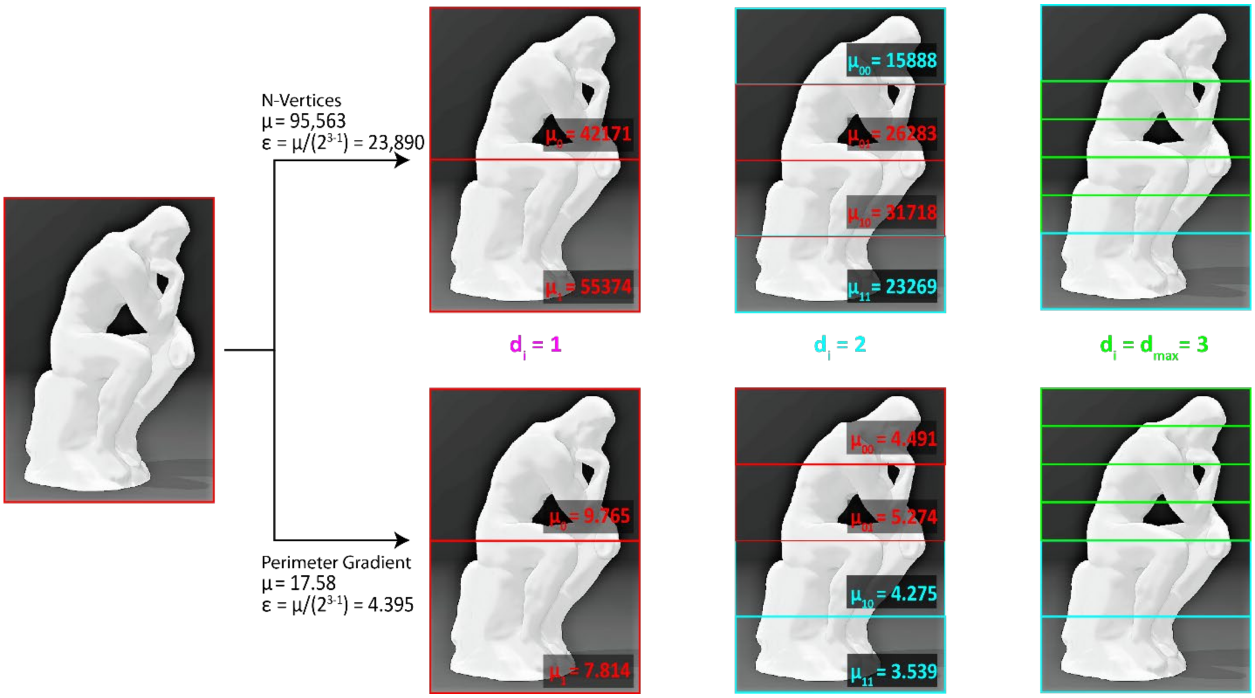


Figure 4. The subdivision process for the Thinker using the ‘n-vertices’ (top) and ‘perimeter gradient’ (bottom) methods. A binary tree labelling structure is used to track subdivisions: e.g., $i = 0$ represents the top half of the model and $i = 01$ represents the upper third quartile of the model. If μ_i, the complexity of a given sub-mesh, exceeds ε, the subdivision process continues. While each method subdivides the model into six sections, they emphasize complexity in different regions of the geometry.

The recursive subdivision algorithm (Figure 4) used is consistent across all proposed complexity analyses. First, the maximum subdivision depth, d_{max}, is defined. Then, one of several complexity methods is used to define the complexity, μ, of the full geometry and a complexity threshold, ε, is set using Eq. 1. This threshold is chosen to distribute the complexity uniformly across subdivisions. For example, with d_{max} set to 3, the maximum number of resultant sub-meshes is 8. This threshold ensures that each sub-mesh contains less than 1/4th of the total complexity.

$$\varepsilon = \frac{\mu}{2^{d_{max}-1}} \tag{1}$$

The recursive subdivision process begins, using a z-normal plane at the centroid of the mesh to split the model if μ_i, the sub-mesh complexity, exceeds ε. Subdivision terminates when all sub-meshes are below the complexity threshold or the maximum depth is reached. This process creates a binary tree structure with each leaf representing a sub-mesh.

In Figure 5, the reconstruction error is compared for the standard and adaptive projection generation methods. This error is based on a comparison between the dose reconstruction and the binary target. In this binary scheme, voxels contained inside and outside of the target region are noted as 'gel' and 'void' voxels, respectively. A threshold is set using the minimum dose received in a gel voxel of the target, and the error is defined as the ratio of void voxels which receive a greater dose than the threshold to the total voxel count. This error quantifies the predicted overcuring of the print. In Figure 5, none of the complexity analyses stands out in consistently predicting the reconstruction error, although the 'perimeter-to-area ratio' and 'perimeter gradient' methods show promising results. Even between the standard and adaptive approaches, the best-suited complexity method for prediction is inconsistent.

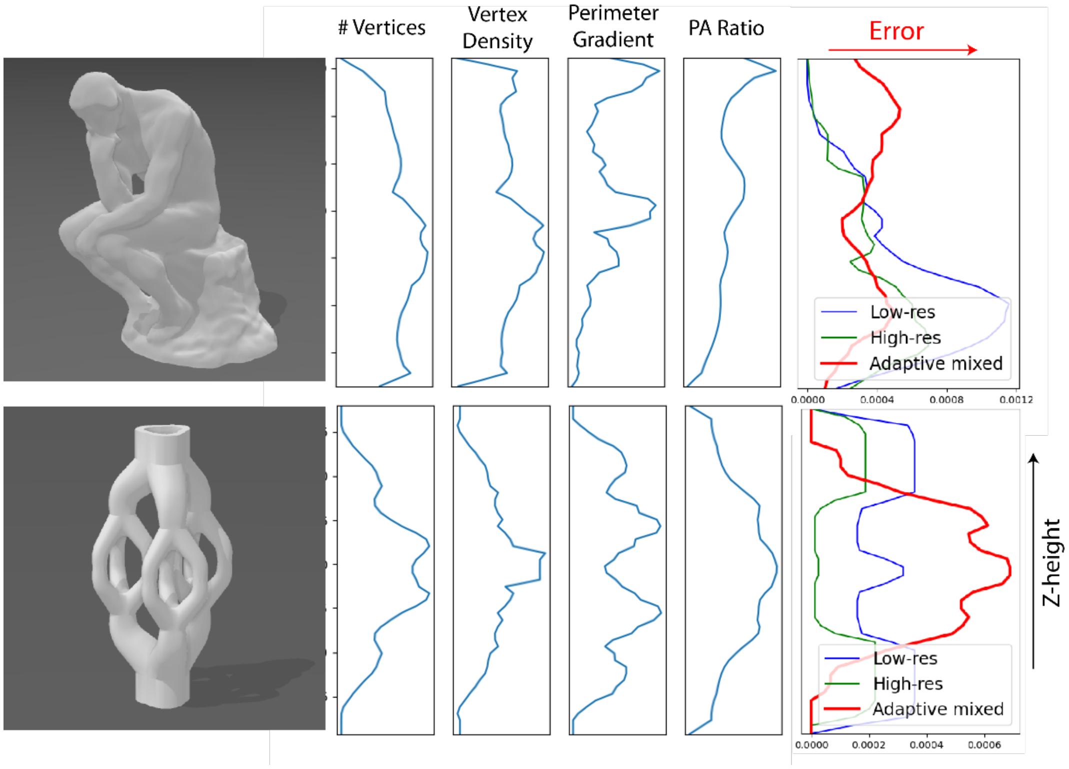


Figure 5. In the far-right column, the linear layer-based error for each model is compared for a low-resolution standard, high-resolution standard, and mixed resolution adaptive approaches. The magnitudes of errors for each are comparable. For the Thinker (top), the 'perimeter gradient' captures the adaptive error best while 'n-vertices' captures the standard error best. For the triple vasculature (bottom), the 'perimeter-area ratio' and the 'perimeter gradient' are the best predictions for the adaptive and standard errors, respectively. This inconsistency motivates the exploration of other complexity definitions.

In some cases, the results of the complexity analyses are sub-optimal: e.g., not all the complex features of a target geometry are recognized by an individual analysis method. Therefore, a user-manipulated method of defining complexity is implemented [7]. This custom definition parametrizes the perimeter-to-area ratio, P/A, and the number of distinct closed contours or entities, $N_{entities}$, in a z-slice, as in Eq. 2

$$\mu_{layer} = (P/A)^{\alpha} \cdot (N_{entities})^{\beta} \tag{2}$$

with α and β as the user-modified parameters. For any geometry, the user can adjust the parameters to match the resulting complexity curve to their own intuitive understanding of complexity. The results of this method are demonstrated in Figure 6. This method enables greater control over the complexity definition, and thus, the subdivision progression. Ultimately, these changes affect the local resolution in projection computation.

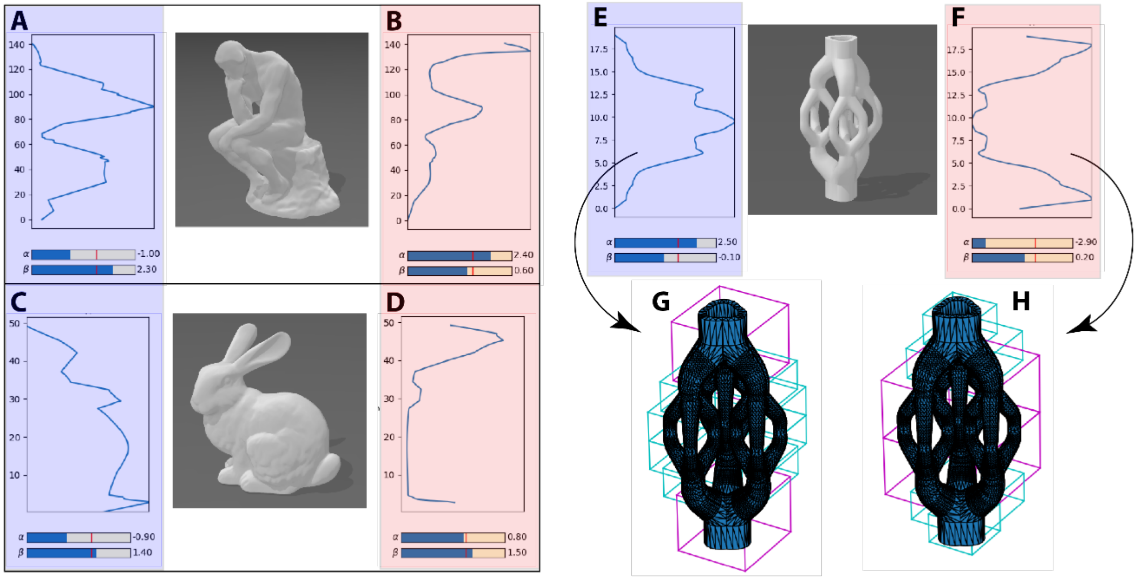


Figure 6. For the Thinker, (A) and (B) represent two possible complexity definitions by changing the α and β parameters. For the Stanford Bunny, (C) and (D) serve the same purpose. For the Triple Vasculature, complexity definition (E) results in the subdivision scheme (G), while complexity definition (F) results in subdivision scheme (H). A flexible, modular complexity definition is useful for two reasons. First, it broadens the scope of applications beyond CAL-specific complexity. Second, the CAL process is still being studied to find which features are difficult to print – this method doesn't tie down complexity to certain feature types.

Projection Computation and Stitching

After recursive subdivision, each sub-mesh leaf has a 3D target geometry computed on a Cartesian voxel basis. The discretization pitch is dependent on a base resolution, r_{base}, and scales with the inverse of 2^{d_i}. This approach implements increased resolution for sections of greater complexity (i.e., sections at a greater subdivision depth).

We can now move onto projection initialization and optimization. However, the targets for each leaf are centered around their own centroid, rather than the global centroid. The resulting tomographic reconstructions would thus be inconsistent between leaves. To account for this offset in projection initialization, the ASTRA toolbox [11] enables the use of a rotational axis offset. This offset is calculated in x and y according to Eq. 3, Eq. 4.

$$\Delta x = (2^{d_i} \cdot r_{base})/\Delta z \cdot \delta x_i \tag{3}$$
$$\Delta y = (2^{d_i} \cdot r_{base})/\Delta z \cdot \delta y_i \tag{4}$$

where δx and δy are the leaf offsets from the global axis of rotation and width and Δz is the global height. Further explanation of these equations is found in Figure 7 below.

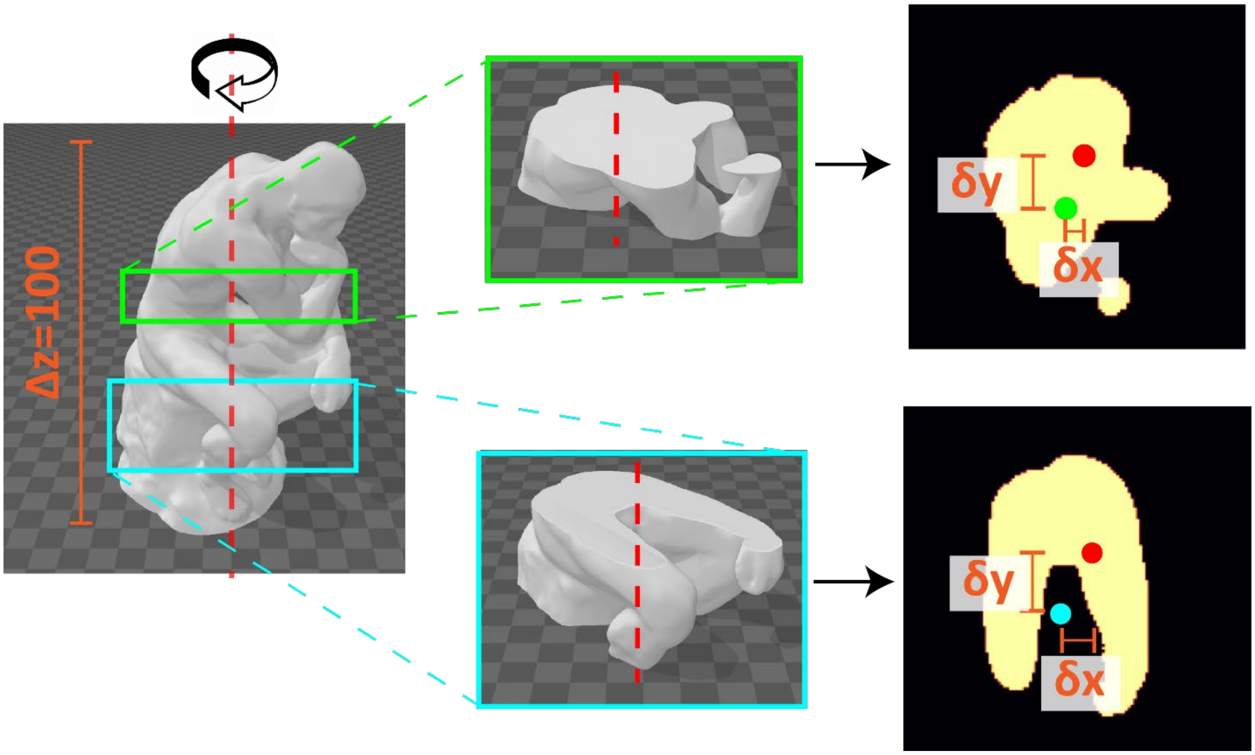
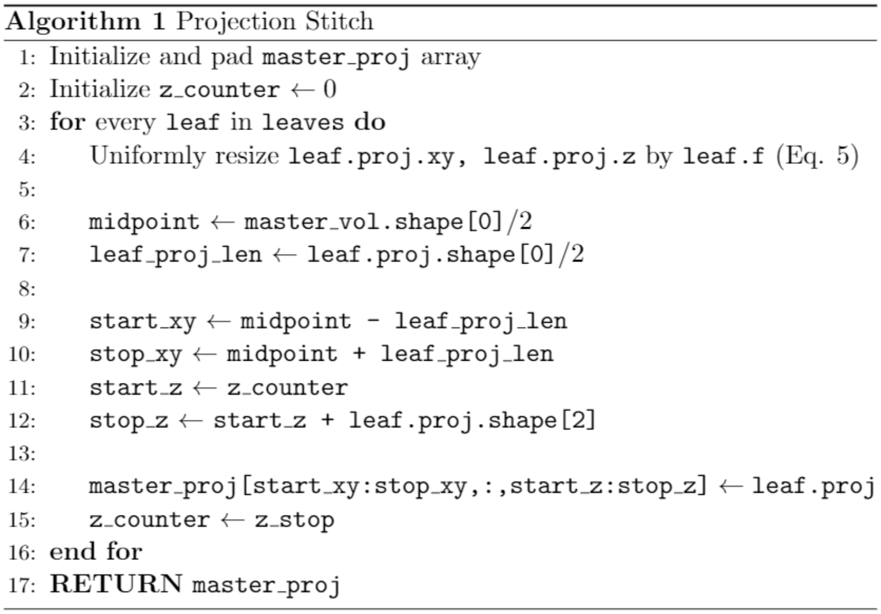


Figure 7. Rotation axis (green and blue circles) offsets from the true centroid (red) are calculated according to Eq. 3 and Eq. 4. The ASTRA toolbox accepts pixel values for the offsets, so we first transform from the model coordinate system to the voxel coordinate system. This is achieved by calculating the z-fraction that the sub-mesh accounts for: $(2^{d_i} \cdot r_{base})/\Delta z$*. Then this fraction is multiplied by the offset in the model coordinate system, leaving us with the axes offsets in pixels. Using* $r_{base} = 20$ *in this example, the bottom section (blue) will have an x-offset of* $(2^2 \cdot 20)/100 \cdot \delta x = 0.8 \cdot \delta x$ *while the top section (green) will have an x-offset of* $(2^3 \cdot 20)/100 \cdot \delta x = 1.6 \cdot \delta x$*. This accounts for the doubled resolution of the green section.*

The projection initialization and gradient descent optimization approaches used for CAL printing [1] are then used to generate individual projection sequences. Because each leaf can be handled independently, this method is well suited for parallel processing. Parallel processing has been implemented in tomographic reconstruction in previous work [12], but to our knowledge, this approach of segmenting a target is novel.

There are several steps in order to make the individual projection sets and dose reconstructions compatible. First their relative intensities need to be scaled. Second, they need to be resized to account for the different voxel resolutions. And last, they need to be placed into a master array to account for the x and y offsets. In optimization, a dose threshold, γ_i, which attempts to divide 'gel' and 'void' doses is calculated. To ensure uniform 3D dose dynamic range between leaves of the target and avoid dose discontinuities along the z-axis, the optimized projection of each leaf is scaled by γ_i/γ_0 with γ_0 being an arbitrary threshold. This intensity scaling method has been experimentally validated by printing on the CAL system.



Algorithm 1 Projection Stitch

```
Initialize and pad master_proj array
Initialize z_counter $\leftarrow 0$
for every leaf in leaves do
    Uniformly resize leaf.proj.xy, leaf.proj.z by leaf.f (Eq. 5)

    midpoint $\leftarrow$ master_vol.shape[0]/2
    leaf_proj_len $\leftarrow$ leaf.proj.shape[0]/2

    start_xy $\leftarrow$ midpoint - leaf_proj_len
    stop_xy $\leftarrow$ midpoint + leaf_proj_len
    start_z $\leftarrow$ z_counter
    stop_z $\leftarrow$ start_z + leaf.proj.shape[2]

    master_proj[start_xy:stop_xy,:,start_z:stop_z] $\leftarrow$ leaf.proj
    z_counter $\leftarrow$ z_stop
end for
RETURN master_proj
```

The projection stitching procedure (Algorithm 1) begins by initializing a master projection set to match the highest depth voxel resolution. Projection sets are shaped as $(n_{x,y}, n_{\theta}, n_z)$ where $n_{x,y}$ is the square voxel grid length, n_{θ} is the count of projection angles, and n_z is the voxel grid height. Therefore, to match the scaling of each set, $n_{x,y}^i$ and n_z^i are scaled by the scaling factor f (Eq. 5).

$$f = 2^{d_{max} - d_i} \tag{5}$$

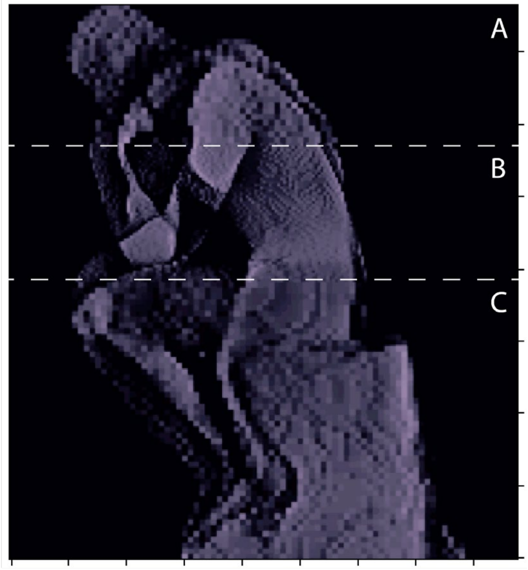


Figure 8. An adaptive resolution projection image for the Thinker with $r_{base} = 20$. From top to bottom: (A) low-resolution of 80, (B) high-resolution of 160, (C) low-resolution of 80. Despite the increased effective resolution from 80 to 100, this projection set is generated more quickly than computing the projection set at the low resolution using the standard approach.

Each scaled projection set is then placed relative to the center of the master projection set. This algorithm is modified slightly to perform dose reconstruction stitching, accounting for the x (Eq.

3) and y (Eq. 4) offsets – both scaled by f – in the placement into the master array. The result is a seamless projection set calculated and resolved using non-uniform resolution (Figure 8).

Results

The computational time savings using the adaptive voxelization approach are summarized in Figure 9. The projection generation time ratio, Π, is calculated according to Eq. 6, accounting for the cubic time complexity of calculation. This lets us use collected data at all resolutions to calculate efficiency. A low Π indicates more efficient computation. All the represented data were collected using a 4-core CPU.

$$\Pi = \frac{\mathrm{T_{gen}}}{(R_{\mathrm{effective}})^3} \qquad (6)$$

The adaptive parallel approach ultimately results in a $2 - 6 \times$ speedup. Even without parallel processing, computation time can be reduced by up to half. The result of this approach is adaptive resolution projection computation that can be faster than using the standard approach to compute projections for the coarsest represented resolution of the adaptively voxelized target.

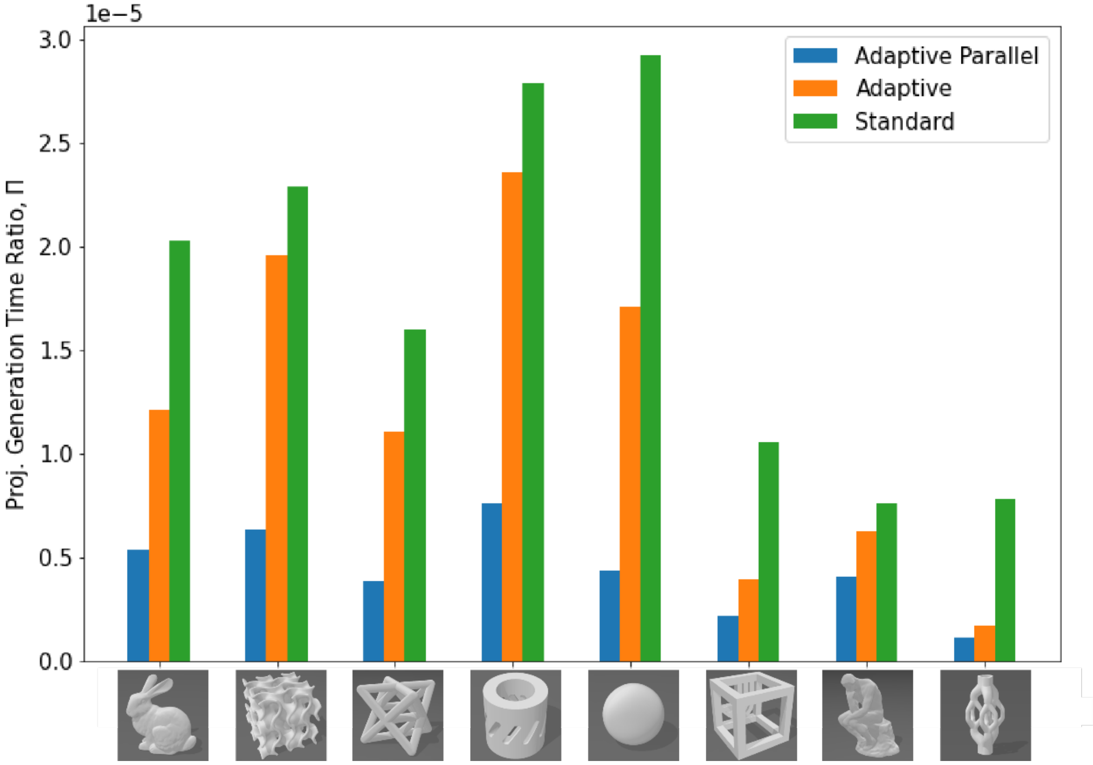
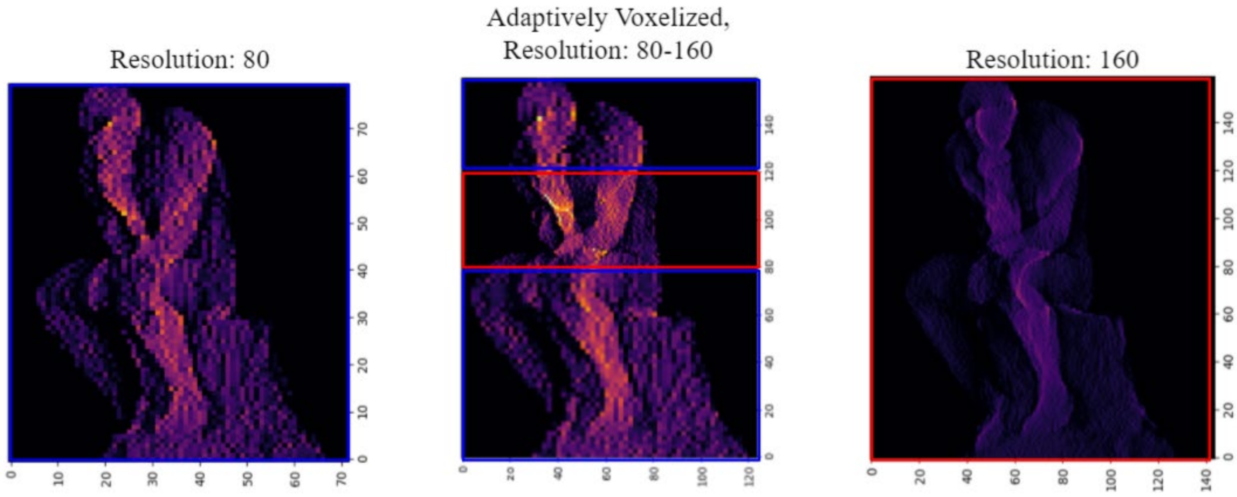


Figure 9. Models from left to right: Stanford Bunny, Gyroid, Octet Truss, Slot Cylinder, Sphere, Test Box, Thinker, and Triple Vasculature. With no exceptions, the adaptive method outperforms the standard approach. The addition of parallel processing to the adaptive approach further improves performance.

Discussion

When comparing the proposed complexity analysis methods, the layer contour-based metrics align well with an intuitive understanding of complexity. While mesh property-based metrics are simple, they provide less insight into the how the geometry of a target evolves along its z-axis. Regardless, neither the 'perimeter gradient' nor 'perimeter-to-area ratio' methods can capture all forms of complexity. However, the user-manipulated complexity demonstrates that parametrized combinations of layer contour-based metrics can be tuned to detect complex regions of a target geometry.

The complexity analysis dictates the subdivision scheme of the mesh, generating a binary tree of sub-mesh leaves. The proposed method of normalizing, scaling, and inserting independent projections and dose reconstructions from each leaf into master arrays enables the use of parallel processing while maintaining the compatibility of the projection sets. This approach is built around the existing approach for projection computation for CAL and provides a simple and robust solution.



6.8	4.82	32.16

T_{gen},seconds

Figure 10. Optimization time improvements using the non-parallel adaptive process. The adaptively voxelized and generated projection set has five leaves with $r_{base} = 20$, and thus an effective resolution of 100.

When compared to the standard approach, the optimization speed-up is significant for the adaptive approach. As noted above, in some cases adaptive resolution projection computation can be faster than using the standard approach to compute projection for even the coarsest represented resolution of the adaptively voxelized target (Figure 10). This result is surprising and counterintuitive, and can be explained by differences in the optimization procedure. After projections are generated, they are optimized using a gradient descent algorithm which utilizes a user-defined error threshold exit condition. Because the range of z-positions over which the gradient descent optimization algorithm operates is reduced when looking at smaller sections of the part, the solution converges – and thus falls below the error threshold – much more quickly for independent leaves than for the target as a whole. Another possible explanation is the tighter bounding boxes that can be created for independent leaves. Rather than back projecting each slice using the $n_{x,y}$ – the maximum cross section size – of the entire geometry, sections with smaller $n_{x,y}^{i}$ are calculated over a more economic domain. Additionally, parallel computing can be used to optimize the solution for each leaf independently. This particular implementation of parallel processing for tomographic reconstruction is novel and provides a simple method to subdivide calculations using a parallel beam projector.

Conclusion

The adaptive voxelization approach demonstrated in this work can support rapid generation of high-fidelity projection sequences for the CAL 3D printing system without sacrificing accuracy. Computational resources can be allocated towards regions of the target geometry which are identified as complex by using a non-uniform discretization pitch. The complexity analyses, subdivision method, and adaptive approach developed could be applied to other AM processes that use a Cartesian voxel basis in pre-computing. Future work could include an in-depth investigation into variations of the proposed complexity analysis methods, expanding on the hybrid modular approach presented in this work and potentially including the use of the fractal dimension index as a metric. Additionally, although this method enables the use of higher voxel resolutions, the voxel representation referenced as the 'target geometry' throughout this work does not truly represent the mesh. More future research into a mesh-based, ray trace-driven reconstruction algorithm could further improve the precision of projections used for CAL printing. Lastly, the development of a 3D octree subdivision structure, rather than the 1D z-slice method discussed in this work, could lead to more detailed feature recognition and subdivision, and even greater efficiency in computational resource allocation.

References

[1] B. E. Kelly, I. Bhattacharya, H. Heidari, M. Shusteff, C. M. Spadaccini and H. K. Taylor, "Volumetric additive manufacturing via tomographic reconstruction," *Sciene,* pp. 1075-1079, 2019.

[2] A. O. Aremu, J. J. Brennan-Craddock, A. Panesar, I. A. Ashcroft, R. M. Hague, R. D. Wildman and C. Truck, "A voxel-based method of constructing and skinning conformal and functionally graded lattice structures suitable for additive manufacturing," *Additive Manufacturing,* vol. 13, pp. 1-13, 2017.

[3] P. Das, K. Mhapsekar, S. Chowdhury, R. Sman and S. Anand, "Selection of build orientation for optimal support structures and minimum part errors in additive manufacturing," *Computer-Aided Design and Applications,* pp. 1-13, 2017.

[4] R. Paul and S. Anand, "A combined energy and error optimization method for metal powder based additive manufacturing processes," *Rapid Prototyping Journal,* vol. 21, no. 3, pp. 301-312, 2015.

[5] S. Cadavid, S. Fathy, J. Zhou and M. Abdel-Mottaleb, "An adaptive resolution voxelization framework for 3D ear recognition," *2011 International Joint Conference on Biometrics (IJCB),* pp. 1-6, 2011.

[6] J. S. Nielsen, D. B. Pederson, Y. Zhang and H. N. Hansen, "Adaptive layer height during DLP materials processing," *Proceedings of the 2nd International Conference on Progress in Additive Manufacturing,* pp. 246-251, 2016.

[7] F. Wasserfakk, N. Hendrich and J. Zhang, "Adaptive slicing for the FDM process revisited," *13th IEEE Conference on Automation Science and Engineering (CASE),* pp. 49-54, 2017.

[8] M. Shusteff, A. E. Browar, B. E. Kelley, J. Henriksson, T. H. Weisgraber, R. M. Panas, N. X. Fang and C. M. Spadaccini, "One-step volumetric additive manufacturing of complex polymer structures," *Science Advances,* vol. 3, no. 12, 2017.

[9] B. Valentan, T. Brajlih, D. Igor and J. Balic, "Evaluation of shape complexity based on STL data," *Journal of Achievements of Materials and Manufacturing Engineering,* vol. 17, pp. 1-2, 2006.

[10] A. R. Backes, J. B. Florindo and O. M. Bruno, "Shape analysis using fractal dimension: A curvature based approach," *Chaos,* vol. 22, no. 4, 2012.

[11] W. Aarle, W. J. Palenstijn, J. Cant, E. Janssens, F. Bleichrodt, A. Dabravolski, J. D. Beenhouwer, K. J. Batenburg and J. Sijbers, "Fast and flexible X-ray tomography using the ASTRA toolbox," *Optics Express,* vol. 24, no. 22, pp. 25129-25147, 2016.

[12] R. P. Rao, R. D. Kriz, A. L. Abbott and C. J. Ribbens, "Parallel Implementation of the Filtered Back Projection Algorithm for Tomographic Imaging," Virginia Polytechnic Institue and State University, Blacksburg, 1995.