Solid Freeform Fabrication 2016: Proceedings of the 27th Annual International
Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference
Reviewed Paper

MANUFACTURABILITY ANALYSIS TOOL FOR ADDITIVE MANUFACTURING USING VOXEL-BASED GEOMETRIC MODELING

Saish Tedia and Christopher B. Williams
Design, Research and Education for Additive Manufacturing Systems Laboratory
Department of Mechanical Engineering, Virginia Tech, Blacksburg, VA 24061

Abstract

While Additive Manufacturing (AM) processes provide unparalleled design freedom, they still impose some constraints on the geometries that can be successfully fabricated. Thus, there exists a need for predictive analysis of part geometries' manufacturability. Existing algorithms based on surface representations require several computationally intensive manipulations. In this paper, the authors present a framework for performing manufacturability analysis of parts to be manufactured by AM using a voxel-based representations schema. The input triangular mesh is first converted into a voxel representation using Ray Casting. Through a series of simple computations on a binary three-dimensional array, the tool provides feedback on infeasible features, minimum feature size, support material, orientation and manufacturing time for different build orientations. The tool's ability to effectively analyze parts for manufacturability is evaluated against several sample geometries.

1. Introduction

1.1 Motivation

The advancement of Additive Manufacturing (AM) technologies coupled with the feasibility of producing complex geometries has led to the widespread, but erroneous, belief among users that any model that can be designed in a computer-aided design (CAD) environment can be manufactured using AM. In reality, the manufacturability of a given model is dependent on a combination of printer resolution, layer thickness, print orientation and print process parameters [1]. For example, a limited printer resolution can result in differences between the CAD model and the manufactured part, such as fine features not being manufactured properly or being damaged during post-processing [2]. Currently, there is no provision in CAD software programs to automatically identify manufacturing constraints. Hence, users have to rely upon heuristic design for AM (DfAM) guidelines that are beginning to emerge in literature (e.g., [2-5]). AM service bureaus are required to manually perform digital inspection and evaluation of hundreds of incoming .STL files for their manufacturability. Thus there exists a requirement for automated software applications that can help the users to visualize these factors for a given geometry and give automatic feedback on its manufacturability.

While pre-process planning software exist (e.g., Netfabb, Magics), they are focused in checking the validity of the STL file and performing necessary repairs. Additional functionality has been recently added to these software programs including automated generation of lattice structures, Boolean manipulations of multiple part files, and added process planning functionality such as slicing, orientation, and tool path planning. Most of the commercially available processes planning software are focused in preparing the part for printing, slicing, support generation and tool path generation. Neither these types of software programs provide manufacturability feedback based on geometric analysis. Moreover, these programs are mostly intended to give results for a specific printer or specific set of print parameters. Some of these programs provide an option to the user to automatically select a build orientation where the software generally provides orientation based on minimum z- height. Subsequently, the models are sliced and sent to a tool path generator that generates instructions to manufacture the part. Build time is estimated only after the tool path has been generated and not upfront in the design stage. These existing solutions are primarily focused on preparing the printing process, instead of evaluating the manufacturability of a part. For better Design for Additive Manufacturing (DfAM), there exists a need for part preprocessing software that also performs manufacturability analysis.

1.2. Prior Art

There have been prior attempts in developing manufacturability analysis tools for AM. Specifically, researchers have investigated schema for automating the identification of manufacturing constraints associated with feature size, support material, and manufacturing time. However, each of these tools has been presented independently; there has been no unifying approach that simultaneously computes all of these factors and uses the results to inform processing decisions. These prior research studies are presented below along with the overall goals for a manufacturability analysis tool.

1) Minimum Feature Size: The type of AM process and resolution of the machine being used limits the minimum printable feature size of a part [6]. In addition, high-aspect ratio features (e.g., thin walls) can be so fragile as to not be able to survive post-processing (e.g., [2]); hence a minimum wall thickness has to be maintained to provide sufficient strength to the part. Several automated methods have been proposed in the literature for determining the thickness of 3D features:
   - *Medial Axis Transformation*: The medial axis transform (*MAT*) is a shape model that represents an object by a set of maximal inscribed spheres and the medial axis is the loci of the centers of these spheres, and can be visualized as the skeleton of the object [7]. The thickness can be computed as distance of medial axis from boundary. Medial axis theorem is a well-established concept in Computer Graphics and Computer Vision and considerable work has been done to analyze 3D shapes using this concept [8-10]. MAT also been extended to analyze models for AM. Nelaturi et al. [11] provide a manufacturability feedback model in terms of a printability map based on resolution errors encountered in AM using techniques from mathematical morphology and medial axis theorem. However, MAT is extremely sensitive to small noise and artifacts resulting in many unwanted branches in skeleton which have to be removed. Also special care has to be taken to compute the corners. These result in extra computation and hence thickness maps for intricate shapes are difficult to compute.
   - *Distance Transform*: Distance transform is used to compute Euclidean distance of a point from the nearest point on the surface. Distance transform operation is also a commonly used technique and has been used on voxel models and other volumetric representations to assess shape of 3D objects for various applications [12-14]. This technique has been implemented for AM by Telea et al. [15], where the authors use a voxel based approach that relies on distance transform and morphological operations to perform 3D printability analysis. They classify voxels based on geometry to identify common issues in printing bridges, spikes, and holes. However, not all regions that are detected after distance transform are useful, and extra topological analysis is required to detect thin features and classify them as critical based on defined metrics, which increases computation. Also, as mentioned by Telea et al.,

this method is not fully automated. Still, distance transformation appears to be better approach than MAT as it is easier to implement than MAT. However its implementation in literature till date has been limited to low resolution voxel models.

- *Polygonal offsetting*: An offsetting operation is another way to compute feature size in a given model. Offsetting operations on a solid have been defined in literature [16]. However, algorithms that offset each site of a contour separately generate offset segments that can self-intersect each other. These techniques then rely on Boolean intersections to trim away excess offset geometry. Many cases between points, edges, boundaries, surfaces etc. need to be carefully considered in implementation. It is non-trivial to carry out all the above-mentioned operations using a polygonal model. Such techniques are computationally complex and numerically instable. Chen and co-authors have proposed a new computational method based on point-based offsetting operation of polygonal models for manufacturability analysis based on minimum feature size [17].

2) Support Material: A key feature of process planning software for all non-powder bed AM systems is generating support material for a given polygonal model. Outside conventional approaches, there have been few algorithms in literature to design effective support structures. Das et al. [18] have provided a methodology for calculating support structure volumes with the use of a point Quadtree. Strano et al. [19] developed an optimization algorithm to use pure mathematical 3D implicit functions for the design and generation of cellular support structures. Voxelizer [20], an open source commercial software developed by ZMoprh, uses voxel representation to calculate support structures; however, no literature is available in the area of generating support structures directly using voxels.

3) Build Time Estimate: For better DfAM, users must be aware of the build time estimates required to manufacture their parts in the design stage itself. In most commercial software programs, build time for AM processes is calculated based on toolpath, which is the last pre-processing step. Researchers have adopted different techniques for time estimation in various AM processes. Alexander et. al [21] provided a build time estimate model for FDM which was used to calculate cost estimate for the model. The estimates were based on area of cross-section of each layer calculated from polygonal model and road width of part and support material. Authors have also presented algorithms to develop generalized models for estimating build time based on regression, neural networks etc. [22-24].

1.3. Research Goal

The primary goal of this work is to realize a preprocessing software solution that assists designers in product realization by automating the assessment of the manufacturability of their designs. Specifically the authors aim to use a voxel-based representation schema as a means for automatically identifying (i) minimum feature size, (ii) support material consumption relative to orientation, (iii) build time estimate

Essentially a three-dimensional pixel, a voxel can be used as a base feature for volumetric representation of part models. The voxels are cuboidal in shape and aligned with the Cartesian coordinate system; hence the 3D model is represented as stack of voxel layers and each voxel can be accessed by their x, y, and z indices. As such a voxel representation of a solid body can be represented by a binary three-dimensional array in which a value of 1 means the voxel is 'on' and value of 0 means the voxel is 'off'.

Voxel representation is used in this work as it allows for a unifying approach to easily compute all these factors simultaneously. In addition, voxel representation allows for easy computations on the part geometry using simple 2D algebraic and Boolean operations, which are usually much less computationally expensive than 3D ones required for polygonal models.

Another motivation for this approach is the voxel representation's compatibility with voxel-based AM processes such as Material Jetting, Binder Jetting and mask-projection Vat Photopolymerization techniques. These AM processes function by converting the input polygonal model to bitmap image files that are used to drive the deposition/imaging tool. Hence, a preprocessing software based on voxel representation can be used to provide manufacturability feedback, as well as serve as direct process control without the addition of any additional steps (and potential information loss) found in model translation. Some work has been done in the area of printing directly from voxels (e.g. Hiller and co. [25, 26], Doubrovski et. al [27], Brunton et. al [28]). Voxel representations can also be used to derive toolpaths for other AM processes for extrusion AM which will be implemented in the future.

In this work, a voxel-based representation is utilized to store geometric information in a discrete and efficient format; simple Boolean and algebraic operations are used to facilitate geometric computations to compute factors central to manufacturability analysis (Sections 3 and 4). The proposed manufacturability analysis tool could be integrated and automated into a CAD software package to assist a designer in creating a manufacturable product by bringing manufacturability upfront in the design process to guide better Design for Additive Manufacturing (DfAM) practices. **Figure 1** shows a schematic diagram of the various steps integrated in the proposed tool in order to evaluate the manufacturability of a given part and prepare it for printing.

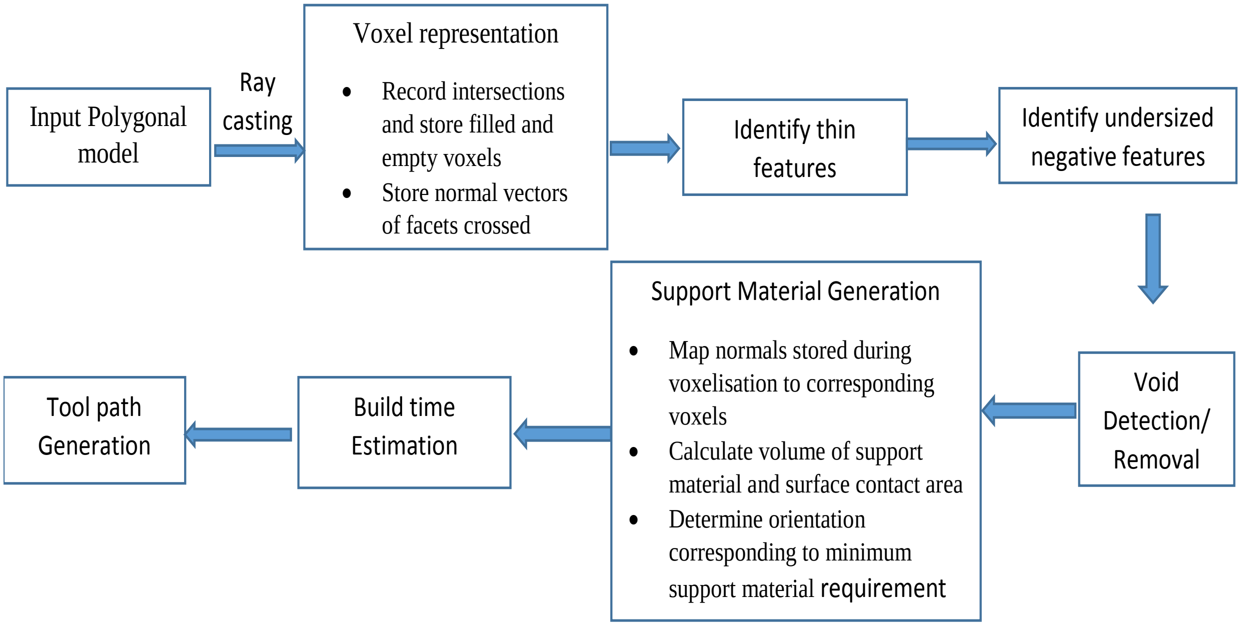

Input Polygonal model
Ray casting
Voxel representation
Record intersections and store filled and empty voxels
Store normal vectors of facets crossed
Identify thin features
Identify undersized negative features
Void Detection/ Removal
Support Material Generation
Map normals stored during voxelisation to corresponding voxels
Calculate volume of support material and surface contact area
Determine orientation corresponding to minimum support material requirement
Build time Estimation
Tool path Generation

Figure 1: Flow-chart listing the steps integrated in the proposed tool

1. **Theoretical Framework: Voxelization and Rendering**

The authors present in this paper a new manufacturability analysis approach based on ray-casting and voxel models, both of which are well established concepts in computer graphics. In this section, techniques for voxelization of polygonal models and rending the resulting voxel model for user analysis are presented.

2.1 Voxelisation of Polygonal Model

A considerable amount of work has been done in the area of using voxels as base feature for representing geometry of objects in AM (e.g. Chandru et. al [29], Lin et. al [30], Ma et. al [31, 32] etc.). As noted in Section 1.3, voxel representations of solids can be stored as 3D binary arrays. The three-dimensional grid is formed by dividing the bounding box of the input polygonal mesh. The resolution of the grid depends upon the desired voxel size, which in turn depends upon layer thickness and printer resolution. The method implemented in this work for creating a voxel representation from an input polygonal model is based on a ray intersection method similar to that described by Patil and Ravi [33].

In this work, the resulting 3D binary grid array is defined as a MATLAB gpuArray object to transfer some of the further computations on the array to the GPU, which saves computational time for rendering and calculating manufacturability parameters. However, to transfer all computations to the GPU, a custom CUDA kernel script is required, which is beyond the scope of this work. This technique only works for water tight meshes, which means that the input STL file should be free from errors.

2.2 Voxel Model Rendering

Rendering of the voxel representation is required so that users can visualize problematic features identified by the tool such as thin sections, overhangs that require support etc. and can modify their design accordingly. The voxel model is displayed as a quadrangular surface mesh. The voxels are classified into boundary voxels (voxels that lie on the surface of the object), exterior voxels (voxels that lie outside the object mesh) and interior voxels (voxels that lie inside the mesh). To increase the speed of rendering, only boundary voxels are used for display. Interior voxels are displayed only while computing manufacturability characteristics, but are reset as empty for rendering. Single stray voxels that occur due to resolution errors which will be absent in normal objects are identified and reset as empty voxels as discussed in [33]. The results of voxelization and rendering are shown in Section 4.

2. **Manufacturability Analysis Methodology**

3.1 Minimum Feature Size

In AM, there are three dimensions to consider: the two planar 2D dimensions (X and Y) and the Z dimension. Every AM system has a specific X, Y and Z resolution Since these resolution values can be different, one needs to evaluate the dimensions of the features in all three directions separately and compare them with the resolution or desired feature size in corresponding direction. We define thickness of a sample section (e.g. section PQ in Figure 2a) of the object along the ray direction as the distance from P to the intersection point of the ray with the opposite surface Q. This is calculated separately each for X, Y and Z directions. Here T_x is the thickness of section RS which would be compared with resolution/desired thickness in X direction. Similarly, T_y is the thickness of section RS which would be compared with resolution/desired thickness in Y direction.

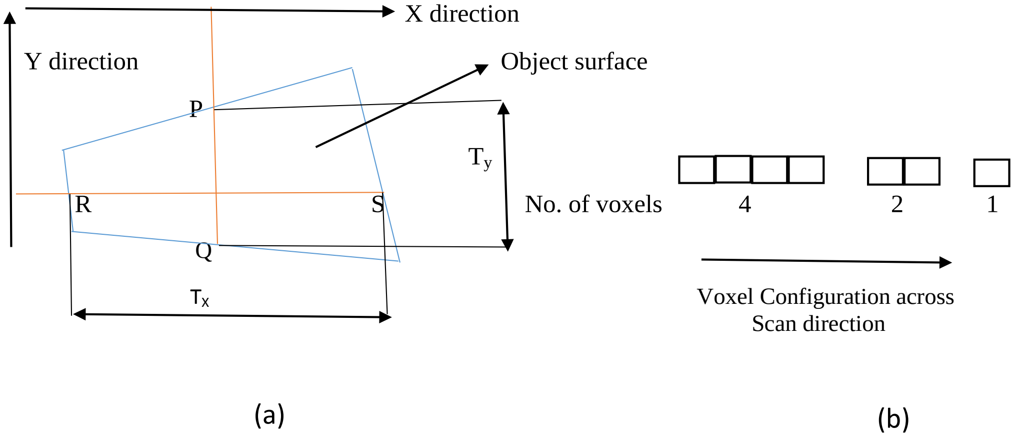


Fig. 2 Minimum feature size calculation steps in X and Y direction

The steps involved in minimum feature size calculations include:

- Rays are cast along the Z-direction incrementally to find the locations where they intersect the mesh. The first ray is passed through minimum Z coordinate (Z_{min}) of the model and the process is carried out layer-by-layer from the Z_{min} to the maximum Z coordinate (Z_{max}) of the model.
- The intersections are stored and sorted in ascending order for each ray.
- All the voxels that the ray passes through between two surfaces after crossing one facet before crossing another facet are counted and stored.
- Let n be the number of voxels between two intersections and R_i , R_{i+1} be the points of intersection of the ray at a particular section. The feature size/thickness of any section of the object in Z direction can then be calculated by the following equation:

$$t=\text{dist.}\ (R_i, R_{i+1}) = n \times d_j \tag{1}$$

where d_j is the voxel dimension in the direction of voxelization

- If this obtained value is less than the resolution or minimum desired feature thickness specified by the user in that direction, than corresponding voxels are highlighted.
- The same procedure is repeated from other two coordinate axis directions and the results are combined together.

Since, in the voxelisation process, we are also passing rays in X, Y and Z direction, this algorithm is implemented as part of voxelisation algorithm to minimize computations.

3.2 Identifying Undersized Negative features

Negative features such as holes that are under the machine's resolution might also not be manufactured correctly. In some processes, it is also required to maintain a minimum size of the holes so that support material can be removed during post-processing [2]. The process to identify negative features is similar to that for minimum feature size described in the above section using ray casting. The only difference is that now voxels between each set of even-to-odd intersection (empty voxels between two surface boundaries) are counted and stored. If the product of number of empty voxels and voxel dimension in that direction is less than machine resolution or desired negative feature size specified by the user, those voxels are highlighted.

Another technique is to classify negative features based on area. This may be helpful for circular or regular polygonal holes where a minimum area can be used to control size of the hole. Each xy, yz and zx slice is analyzed where only empty voxels are displayed. The area is measured by calculating the actual number of pixels in each of the empty regions. Hence, user can also specify a minimum area for negative features and if the area of negative feature in any plane is less than the specified value, the voxels are highlighted.

3.3 Support Material Generation

After importing the STL file, it is required to set an orientation for the part initially. If V1, V2 and V3 are the row matrices containing X, Y and Z co-ordinates of vertices of all the facets, then the rotation is performed by multiplying the transpose of these matrices by rotation matrices $R_x, R_y\ and\ R_z$ around the X- and Y and Z axes respectively to find new vertices $V1_{new}$, $V2_{new}$ and $V3_{new}$. If the part has to be rotated by θ_x in x direction, θ_y in Y direction and θ_z in Z direction, then the new vertices can be calculated by following equations (2-7):

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{bmatrix} \tag{2}$$

$$R_y = \begin{bmatrix} cos(\theta_y) & 0 & sin(\theta_y) \\ 0 & 1 & 0 \\ -sin(\theta_y) & 0 & cos(\theta_y) \end{bmatrix} \tag{3}$$

$$R_z = \begin{bmatrix} cos(\theta_z) & -sin(\theta_z) & 0 \\ sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$V1_{new} = R_z \times R_y \times R_x \times V1 \tag{5}$$

$$V2_{new} = R_z \times R_y \times R_x \times V2 \tag{6}$$

$$V3_{new} = R_z \times R_y \times R_x \times V3 \tag{7}$$

After voxelisation, subsequent rotation is simply done by rotating the 3D grid using the same $R_x, R_y\ and\ R_z$ rotation matrices. Hence, it is not required to voxelise every time the part has to be rotated.

Depending upon the AM process and machine, certain features such as some surfaces, overhangs, negative drafts and undercuts etc. which are at an angle (measured between normal and horizontal) that is less than a critical value require support material to be manufactured. For example, this limiting value is 35^0 for Selective Laser Melting [34]. The steps involved in support material calculations are discussed below:

- During voxelization, while determining the facets that are crossed by rays, the normal vectors of respective facets are also stored. If the normal vector data is not available in the STL file, then normal vectors are computed using simple cross product of the vectors forming the facet.

- For generating support, only Z voxelization data is required since the facets that are not going to be crossed during Z voxelization would essentially be parallel to Z axis (i.e. facets in XZ or YZ planes). These facets will have their normal aligned with the horizontal and hence the angle that their normals make with the horizontal will always be 0^0 and are not required for support calculation.
- For generating support, all interior voxels are removed from analysis as none of these voxels would require support material.
- Among the remaining voxels, only those voxels are considered for analyses that correspond to facets crossed in Z voxelization. This is done by traversing through each x, y voxel from Zmin to Zmax incrementally and only voxels that are at the start or end of a Z boundary are displayed.
- The corresponding normal data from Z voxelization is then mapped with the respective voxels that represent these facets.
- All remaining voxels which are supported by a voxel below them are identified and removed from analysis.
- For remaining voxels, the angle that normal makes with the horizontal is calculated.
- If this angle is greater than the specified critical angle of the machine, then the empty voxels below them are turned on and marked as support voxels.
- Voxels corresponding to rays that could not be voxelised cannot be mapped with their respective facet normal vector. The normal vectors for these voxels are computed by interpolating from the surrounding voxels.
- Finally, all the support voxels are projected down until any part voxel or base voxel is encountered. The number of support voxels are counted and multiplied by volume of one voxel to find the total volume of support material required.
- In order to compute the support contact area, voxels in contact with support voxels are checked whether they belong to the part volume or not. The surface contact area is simply the area of the faces in contact with support voxels

Subsequently, the tool also identifies optimum orientation with minimum support material. The geometry is incrementally rotated around x- and y-axes, with default resolution of 10°.The algorithm iteration loop is on until the supports for all the orientations are calculated. Once all the possible orientations are investigated, the orientation that requires minimum support volume is identified.

3.4 Build Time Estimation

In material extrusion, building a layer of material of a part has several stages. The part's external contours are drawn first and then the interior is filled with a tightly rastered pattern. Support structures are deposited, also in a raster pattern but spaced wider than that used to fill the part interior. To deposit material for the i_{th}layer, time taken can be given by sum of time taken to build external contour, time for infill and time taken to deposit support structures. While calculating time to build an external contour for i_{th}layer, only the boundary voxels are considered for analysis. The time $t_{contour}$ is given by number of voxels in the contour $n_{voxel},$ the cross-sectional area of a voxel, A_{voxel} , the contour width w and the nozzle velocity for drawing outline, $v_{outline}$, shown in equation (8):

$$t_{contour} = \frac{n_{contour} \times A_{voxel}}{w \times v_{outline}} \quad (8)$$

Next, only the internal voxels are displayed. Each y voxel is traversed from $xmin\ to\ xmax$ and total voxels at each x co-ordinate n_{xi} as well as total no. of roads $n_{roads}$are stored. Here, retraction of nozzle in going from one road to next is also considered. However, retraction of nozzle within one road or between multiple contours is not taken into account. This approach is similar to assuming that the raster angle is 90°. An approximation for the time taken to generate infill t_{infill}, is determined using the default printing velocity v, nozzle X/Y axis movement speed v_{xy} and infill percentage W_p shown in equation (9):

$$t_{infill} = \sum_{xmin}^{xmax} \left(\frac{n_{xi} \times d_y}{W_p \times v} \right) + \frac{(n_{roads} - 1) \times d_x}{W_p \times v_{xy}} \quad (9)$$

Here d_x and $d_y$are voxel x and y dimensions respectively.

This method slightly underestimates the true time because the deceleration of the nozzle at corners is also not taken into account. Also the raster pattern could be at an angle which is not taken into account. Next, only the support voxels are displayed. The equation for the time to deposit the support structure for the i_{th}slice is similar to equation (9) except that the support does not fill the entire area. Supports are deposited less densely in widely spaced raster patterns compared to the closely packed raster in the part interior. Hence, support infill percentage, W_s and support infill density $\rho_{support}$ are used to calculate the time taken to deposit support structures, $t_{support}$ for ith layer using equation (10) below:

$$t_{support} = \sum_{xmin}^{xmax} \left(\frac{n_{xisupport} \times d_y}{W_s \times v \times \rho_{support}} \right) + \frac{(n_{roadssupport} - 1) \times d_x}{W_s \times v_{xy} \times \rho_{support}} \quad (10)$$

Finally, approximate time taken to move from one layer to next t_{layer} is added using Z axis movement speed v_z and layer thickness t_z using equation (11) below:

$$t_{layer} = \frac{t_z}{v_z} \quad (11)$$

Total time is given by adding all the time components for each layer as shown in equation (12) below:

$$t_{total} = \sum_{i=1}^{n} (t_{contour} + t_{infill} + t_{support} + t_{layer}) \quad (12)$$

3.5 Manufacturability Analysis: Void Detection

Unknown voids in the input model may increase the file size and can significantly decrease the strength of the part [42]. To compare two STL files: one having voids and other without voids, Boolean comparison of one facet of parent model with all the facets of other model is required.

Also, removing a void requires remeshing of the model. Hence, void detection and removal using polygonal models are computationally intensive. Voxel based representations on the other hand allows for simple Boolean operations on complex binary 3D arrays in a matter of few seconds as well as void removal by selectively turning on material at detected location. Hence this feature can be used to detect cyber threats or for quality inspection purposes during post processing [35]. Voxel representations can also be used to examine empty spaces for closed voids by checking the surrounding voxels which might trap support material, powder or resin which would be impossible to remove. This feature is in development phase and will be added in the future.

4. Manufacturability Analysis Results

Several functions have been written to perform voxelisation, rendering and analysis. All computations are performed in MATLAB R2016 on a Dell PC with 16 GB RAM and Intel i-7 3770K CPU @3.5 GHz.

4.1 Voxelisation and Rendering

The time required for voxelisation of some sample models are given in **Table 1**. The Bunny STL model had 10400 facets and the pelvis bone STL file had 41,100 facets (8,978 kB). At the resolution of 1000×1000×1000, the time taken was 50-100 seconds depending on the shape and size. **Figure 3** shows the voxelised Bunny model at two different resolutions.

Table 1 Voxelisation results

Model	No. of facets	Resolution		
		200×200×200	500×500×500	1000×1000×1000
		Time (s)	Time (s)	Time (s)
Bunny	10400	7.142	37.012	97.15
Teapot	57600	4.6	21.459	56.57

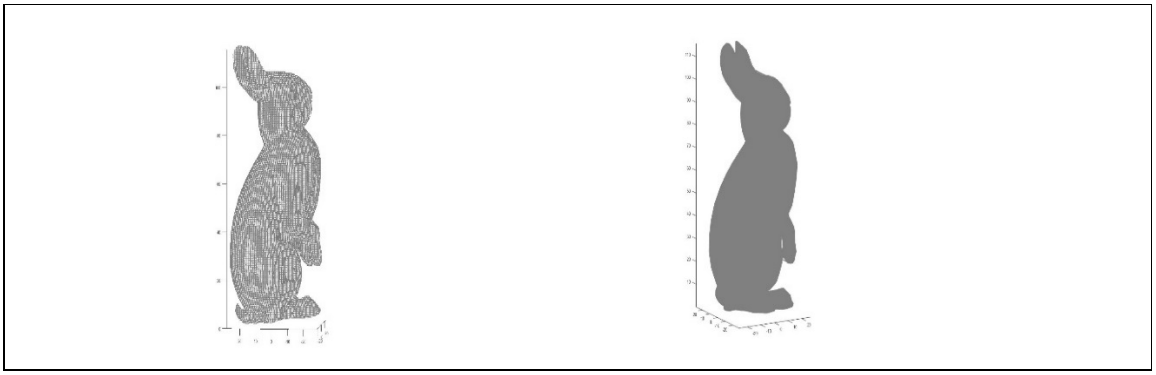


Fig. 3 Voxelised model of Bunny at Grid resolution (a) 200 ×200×200 (b) 800 ×800×800

4.2 Minimum Feature Size

The results of minimum feature size analysis for few sample models are given in **Figure 4**. For models (a)-(c), all features which have feature thickness less than 1mm in any direction have been highlighted. The sample widget shown in Fig. 4(a) has rectangular features with wall thickness 0.3 mm, 0.5 mm and 2 mm. If the minimum resolvable size in x-y plane of machine being used to manufacture this part is 1 mm, the highlighted features would not be manufactured correctly. Consider the coupler shown in Fig. 4(b) to be manufactured by Material Jetting. In Material Jetting, the minimum Survivable Feature Size for such circular features has been shown to be 1.13 mm [2]. Hence, these features mostly would not survive post- processing. Similarly, for Snowman model shown in Fig. 4(e), the sharp corners at the eyes, nose, buttons and hands have one voxel each with voxel dimension less than 0.2 mm. Hence, these sharp corners would not be manufactured on an AM machine with resolution less than 0.2 mm in all directions.

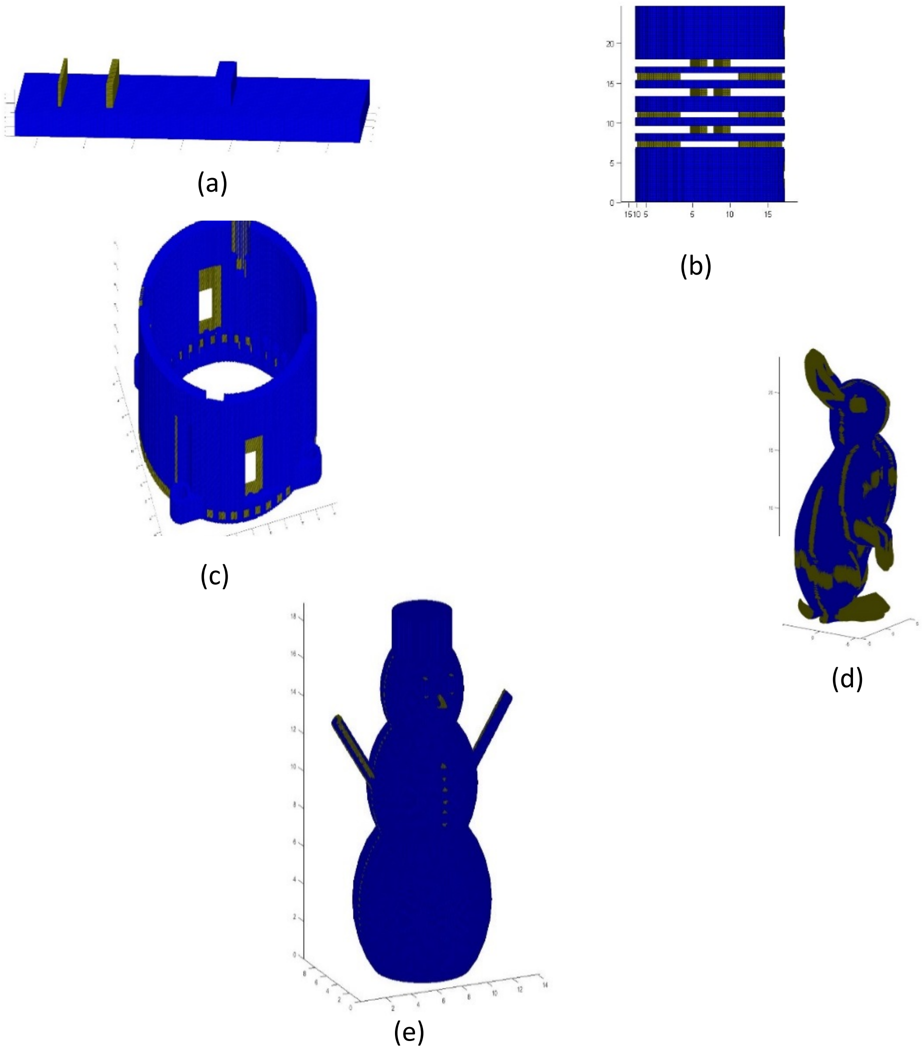
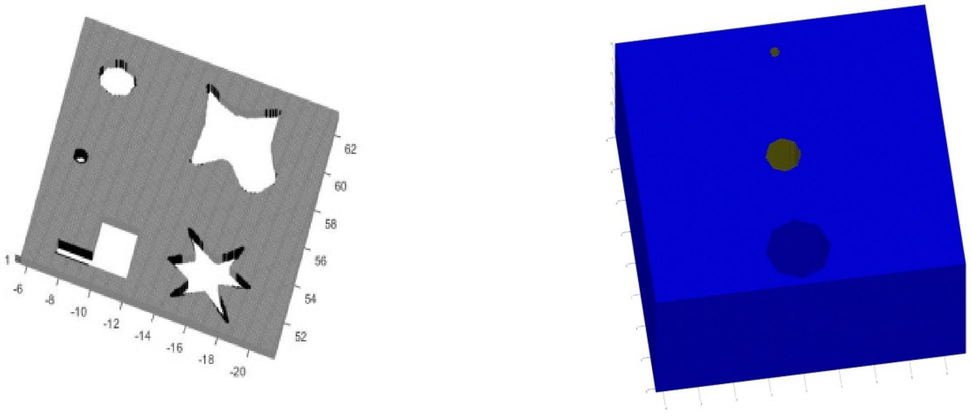


Fig. 4 Highlighted Feature sizes, t <=1 mm for a) Sample widget b) Coupler c) Gear Housing, t < 2 mm for d) Bunny and t <0.2 mm for e) Snowman models in any direction

The results of negative feature size analysis for two sample widgets at resolution 500×500×500 are shown in **Figure 5**. The problematic areas which would not be manufactured correctly have been highlighted. Consider a sample widget with a star shaped hole as shown in Fig. 5a. The extreme corners of the star shaped hole have one voxel inside it. The size of this voxel is less than resolution of the AM machine in that plane, and hence the corner would not be manufactured correctly. Here, the machine resolution or minimum desired thickness of negative space has been set to 0.2 mm in all directions. Consider another sample widget with circular holes as shown in Fig. 5b to be manufactured by Material Jetting. In Material Jetting, for efficient removal of support material, circular negative features in Vero WhitePlus specimens must have cross-sectional area less than or equal to $50\ mm^2$ [2]. The highlighted features in Fig. 5b have cross-section area less than $50\ mm^2$ and hence would not survive post-processing. Hence, the area feature can be useful to control the size of regular shaped negative features.



a)

b)

Fig.5 Negative feature analysis at resolution 500×500×500 a) Features highlighted with t=0.2 mm between boundary voxels in any direction b) Features highlighted using minimum area criteria, $A < 50\ mm^2$

4.2.1 Sample Case Study for Feature Size

Figure 6 shows CAD model of a sample widget to be manufactured using Material Jetting which has three circular holes (1, 2 and 3), one cylindrical feature (4), two overhangs (5 and 6) and a star shaped hole (7). The part is printed in the orientation shown in Fig. 6 using Objet Connex 350 machine.

The results of minimum feature size analysis are shown in **Figure 7**. The voxel size in Objet Connex is 0.09×0.06×0.03 mm. Features with thickness less than the voxel size in respective directions would not be manufactured properly. Moreover, features with thickness less than 1.13 mm in any direction are expected to break away either during printing or post-processing. The results show that features 4 and 5 would either not be manufactured correctly or break during post processing. The results of negative feature size analysis is shown in **Figure 8 (a)** and **(b)**. Circular negative features 1 and 2 are highlighted as they have cross sectional area less than to $50\ mm^2$and hence are expected to have support material trapped inside them. Also, according to the analysis, the corners in feature 7 i.e. star shaped hole will not be manufactured correctly.

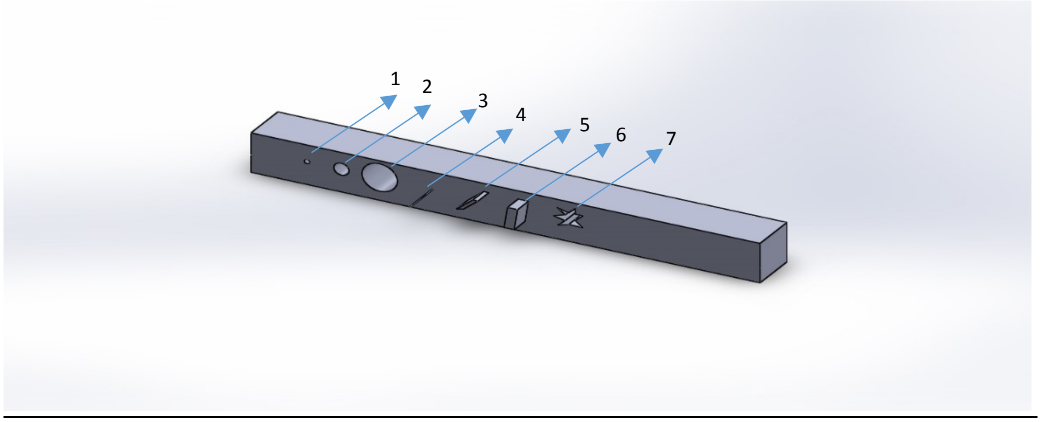


Fig. 6: CAD model of sample widget for Case Study

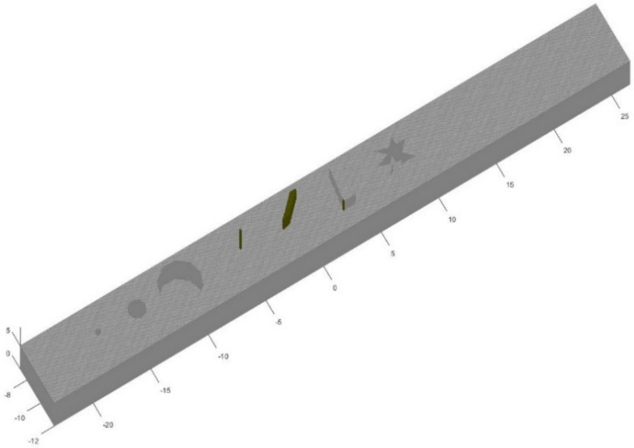
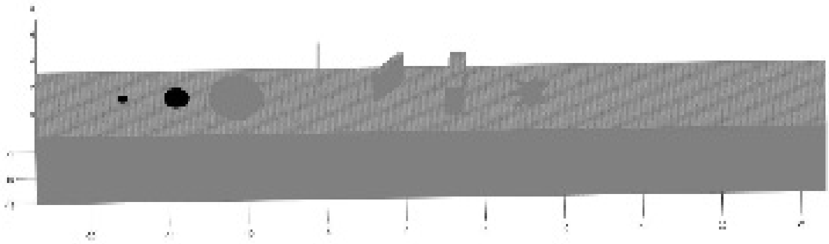
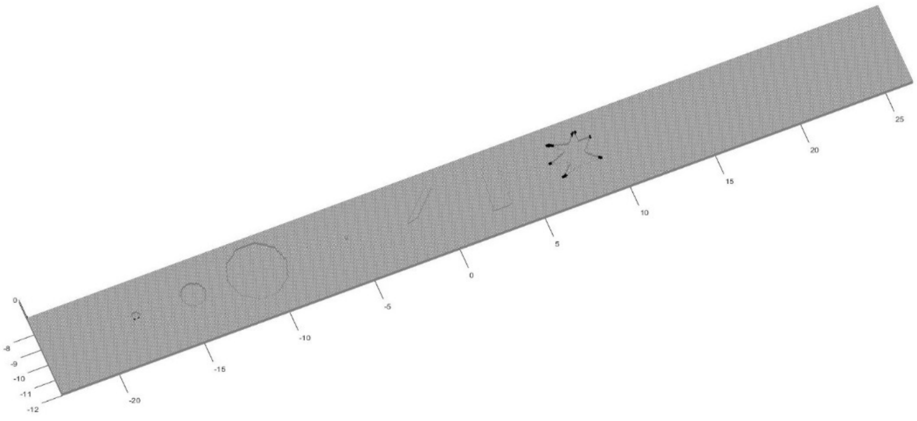


Fig. 7: Highlighted Feature sizes, t <=1.13 mm of sample widget for Case Study



(a)



(b)

Fig. 8: Negative feature analysis a) Features highlighted using minimum area criteria, $A < 50\, mm^2$ b) Features highlighted with $\mathrm{t_x} = 0.09\ \mathrm{mm}$, $\mathrm{t_y} = 0.06\, mm$, $t_z = 0.03\, mm$ between boundary voxels in any direction.

Figure 9(a) shows the image of the printed part after removing support material. It can be seen that features 4 and 5 have not been manufactured. While feature 4 failed to print during the printing process, feature 5 broke while cleaning support material with water jet. **Figure 9 (b)** shows that features 1 and 2 have support material trapped inside them as predicted during feature size analysis. Finally, **Figure 9 (c)** shows that the corners of star-shaped negative feature are nor manufactured correctly and have been approximated. These observations are consistent with the results of analysis.

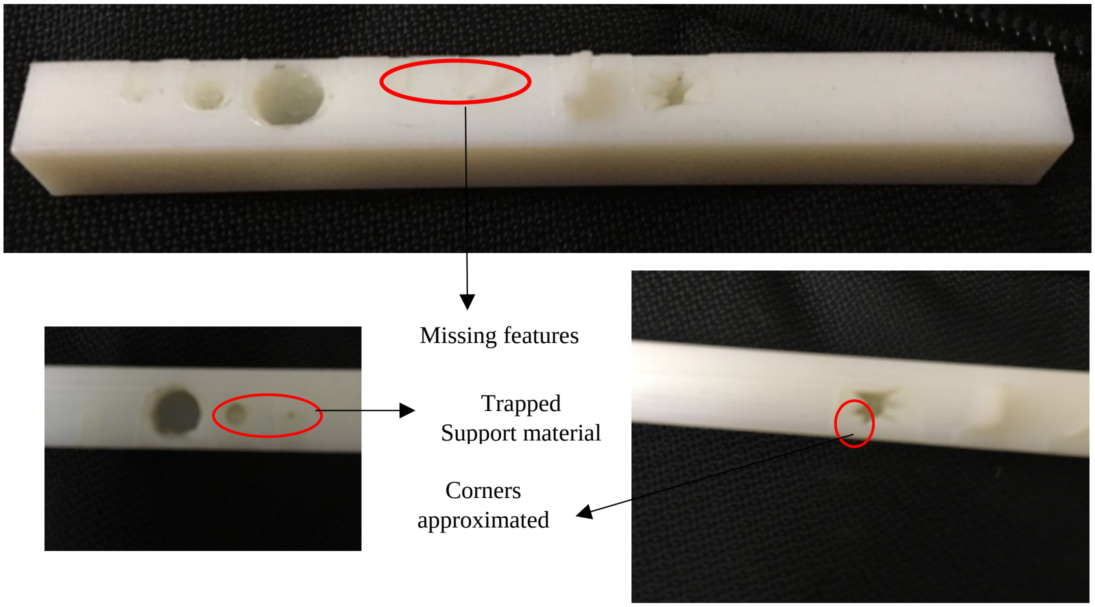
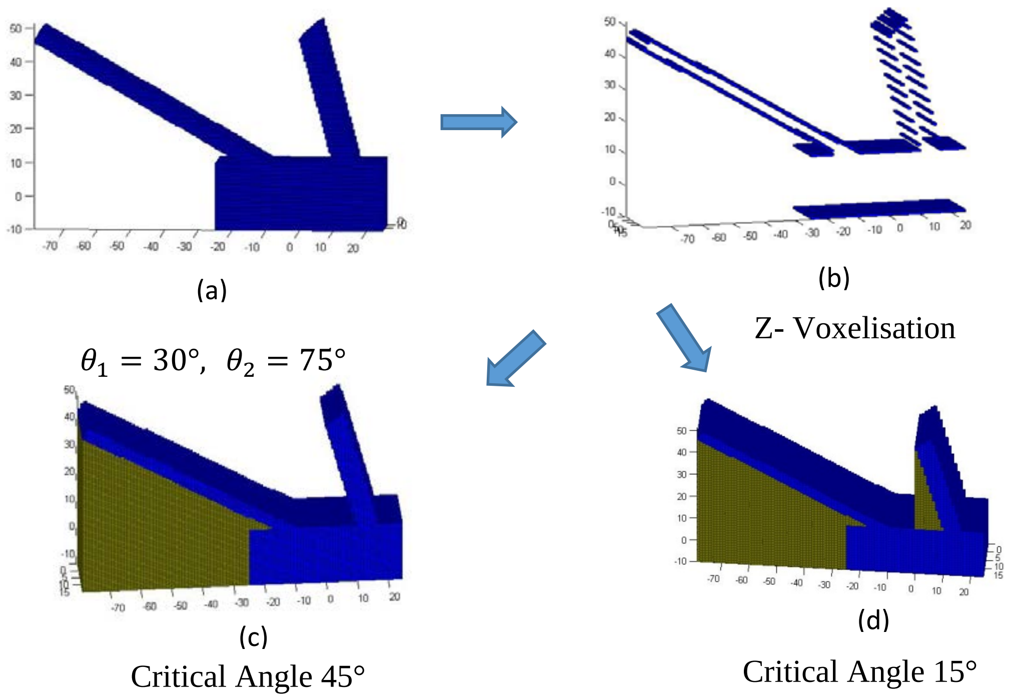


Fig. 9: a) Sample widget manufactured using Material Jetting b) enlarged image of printed part showing trapped support c) enlarged image of printed part showing star shaped negative feature

4.3 Support Material Generation

Figure 10 shows steps required to generate support material for a sample part with two overhangs at $\theta_1 = 30° \; and \;\; \theta_2 = 75°$ respectively. Fig. 10(a) shows the voxelised model of the part. At this stage, all normal vectors for surface voxels are calculated. Next, only unsupported boundary voxels at start or end of a Z boundary are considered for further analysis. This is depicted in Fig. 10b). These voxels are mapped with normal vectors of their corresponding facets. Based on these mapped normals, support is generated critical angles 45° and 15° which can be seen in Fig. 10(c) and 10(d) respectively.

The support material feature gives satisfactory results even for models with resolution as low as 200×200×200 and gives accurate estimates of volume of support material required. At low resolutions, the user can get very quick idea of features which would require support material and modify his design accordingly. The accuracy of the estimate increases with increase in resolution. However, if the goal is to prepare the file for printing directly from voxel representation, then the voxelisation should be done at printer resolution. **Figure 11** shows support material generated for a few sample models at resolution 500×500×500 for critical angle $\theta_{cr} = 60°$. The amount of support material required and support surface contact area for these models is given in **Table 2**. The material density value used for support material calculations is 1.25 g/cm^3.



(a)

(b)

Z- Voxelisation

$\theta_1 = 30°, \;\; \theta_2 = 75°$

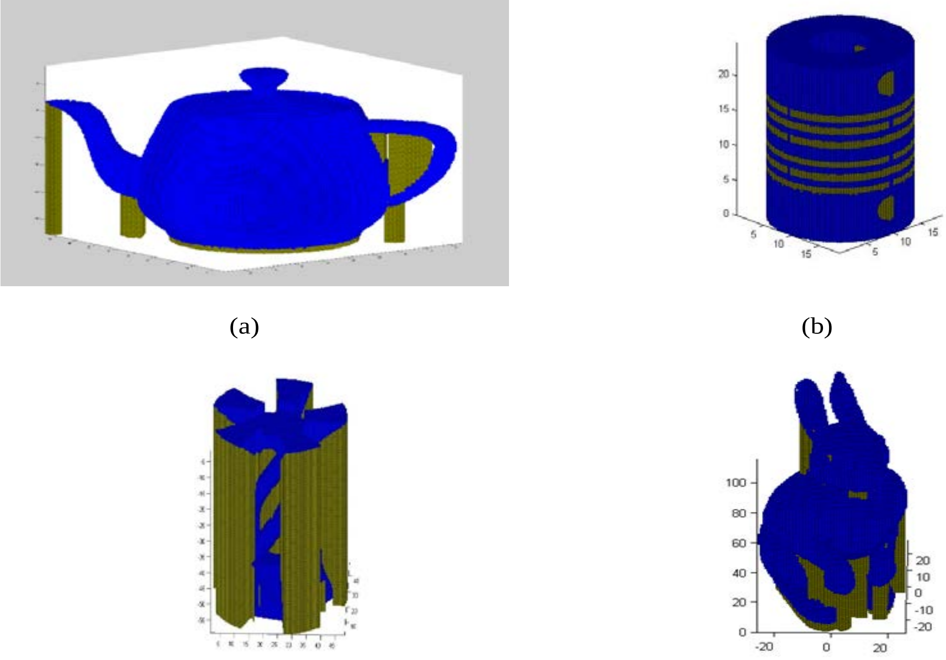
(c)

Critical Angle 45°

(d)

Critical Angle 15°

Fig. 10: Support structure generation process for a sample part to be manufactured using Extrusion



(a)

(b)

(c)

(d)

Fig. 11: Required support Structures for manufacturing using the extrusion process, $\theta_{cr} = 60°$ for a) Teapot b) Coupler c) Candle holder d) Bunny models

Model	Volume of Support Material (grams)	Support Contact Area (mm^3)
Teapot	0.839	567.3592
Coupler	1.6347	1349.4
Bunny	7.155	1378.7

Table 2: Volume of Support Material and Surface Contact Area for given models

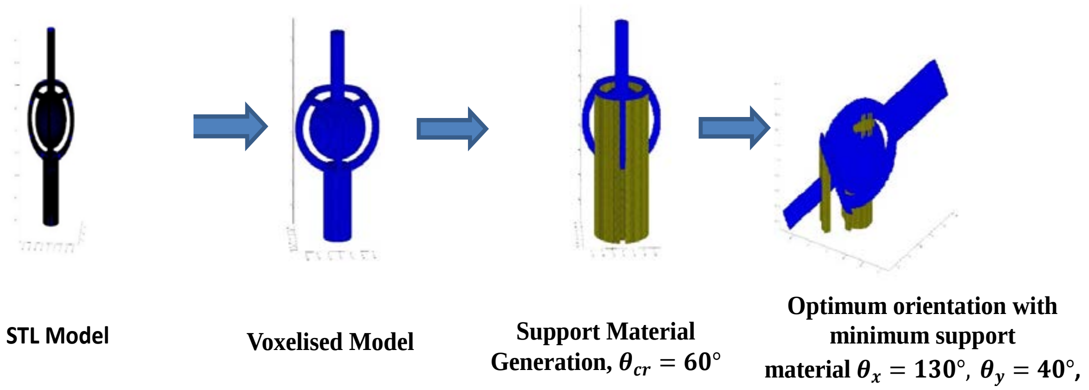

STL Model
Voxelised Model
Support Material Generation, $\theta_{cr} = 60°$
Optimum orientation with minimum support material $\theta_x = 130°$, $\theta_y = 40°$,

Fig. 12: Identification of optimum orientation with minimum support for a sample ball joint

Figure 12 shows identification of optimum orientation with minimum support for a sample ball joint model. Starting from original orientation, the tool computed supports for subsequent orientations at an interval of 10^0 and optimum orientation was identified to be $\theta_x = 130^0 and\ \theta_y = 40^0$. This result was manually verified by generating supports at various orientations using Simplify 3D software.

4.4 Build time estimation

After generating support material for above sample parts, build times for these were estimated for manufacturing using Material Extrusion. The build times were estimated using layer thickness 0.2 mm, printing speed 3600 mm/min, outline underspeed 50%, X/Y axis movement speed 9000 mm/min, Z axis movement speed 4800 mm/min, interior and support infill percentage 40% and support density 70%. The estimated build times for some sample models and their actual build times estimated using Simplify 3D software are listed in **Table 3**. The build times are found to very close to the actual build times estimated by Simplify 3D software.

Model	**Estimated Build Time**	**Build time estimated using Simplify 3D software**
Teapot	51 mins 32 secs	58 mins
Coupler	31 mins	36 mins
Bunny	144 mins 10 secs	172 mins

Table 3: Comparison of estimated build time with actual build time using Simplify 3D software

5. Closure and Future Work

In this paper, a tool has been presented for automated manufacturability analysis of shapes to be manufactured using Additive Manufacturing. The input polygonal model is first converted to a voxel model using Ray Casting Algorithm. Various simple and easy to implement algorithms have been presented to evaluate various factors governing manufacturability. The quick feedback provided by the tool can be easily used as an initial assessment of manufacturability and gives an idea to the user about features that might not be manufactured correctly, estimation of support material, support contact area and build time.

The major problem associated with voxel representation has always been large memory requirement which limits voxelisation resolution [1]. The main focus of this research is manufacturability analysis. However, this problem has been partly taken up in this work. Most of the manufacturability calculations are done during voxelisation process to keep the computations to a minimum. Some of the operations which require large computation time have been shifted to GPU using MATLAB parallel computing. However, rendering of the volume is still slower than polygonal models. This issue can be overcome in the future by using GPU based visualization technique illustrated in [36].

The authors aim to extend the reported work by validating the proposed tool and also providing additional features. For this realization following steps will be undertaken.

- Void detection feature to detect voids which may trap powder, resin or support material will be incorporated in the tool.
- A full case study will be implemented using multiple complex parts to validate the manufacturability analysis results given by the software where the results would be compared with an actual printed part.
- This tool will also be implemented in a classroom setting to evaluate the effectiveness of the tool in making DfAM decisions.
- Efforts will also be made to manufacture sample parts directly from voxel representation using Material Jetting and Material Extrusion. Toolpath generation feature for Material Extrusion will be added.
- Final research investigation will include implementing an Octree based voxelisation technique to optimize memory requirements.

6. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1546985. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

7. References

[1] W. Gao, Y. Zhang, D. Ramanujan, K. Ramani, Y. Chen, C. B. Williams, C. C. L. Wang, Y. C. Shin, S. Zhang, and P. D. Zavattieri, "The status, challenges, and future of additive manufacturing in engineering," *Comput. Des.*, vol. 69, pp. 65–89, 2015.

[2] N. Meisel and C. B Williams, "An Investigation of Key Design for Additive Manufacturing Constraints in Multimaterial Three-Dimensional Printing," *J. Mech. Des*, 137(11), 111406, 2015.

[3] "Designing mechanical parts for 3D printing." [Online]. Available: http://www.shapeways.com/tutorials/designing_mechanical_parts_for_3d_printing/.

[4] C. C. Seepersad, T. Govett, K. Kim, M. Lundin, and D. Pinero, "A Designer's Guide for Dimensioning and Tolerancing SLS parts," *23rd Annu. Int. Solid Free. Fabr. Symp.*, pp. 921–931, 2012.

[5] Fernandez-Vicente, M., Canyada, M. and Conejero, A., "Identifying limitations for design for manufacturing with desktop FFF 3D printers" *International Journal of Rapid Manufacturing*, *5*(1), pp.116-128, 2015.

[6] S. Moylan, J. Slotwinski, A. Cooke, K. Jurrens, M. Alkan Donmez, "*Proposal for a Standardized Test Artifact for Additive Manufacturing Machines and Processes*," International Solid Freeform Fabrication Symposium, Austin, Tx, 2013.

[7] R. Tam and W. Heidrich, "Shape Simplification Based on the Medial Axis Transform," *Proc. IEEE Vis. Conf.*, pp. 481–488, 2003.

[8] C. W. Niblack, D. W. Capson, and P. B. Gibbons, “Generating skeletons and centerlines from the medial axis transform,” *[1990] Proceedings. 10th Int. Conf. Pattern Recognit.*, pp. 881–885, 1990.

[9] J. Damon, “Determining the geometry of boundaries of objects from medial data,” *Int. J. Comput. Vis.*, vol. 63, no. 1, pp. 45–64, 2005.

[10] N. Iyer, Y. Kalyanaraman, K. Lou, S. Jayanti, and K. Ramani, “A Reconfigurable 3D Engineering Shape Search System Part I: Shape Representation,” *Proc. DETC’ 03*, pp. 1–10, 2003.

[11] S. Nelaturi, W. Kim, and T. Kurtoglu, “Manufacturability feedback and model correction for additive manufacturing,” *J. Manuf. Sci. Eng.*, vol. 137, no. 2, p. 21015, 2015.

[12] Lu, S.C., Rebello, A.B., Miller, R.A., Kinzel, G.L. and Yagel, R., “A simple visualization tool to support concurrent engineering design,” *Computer-Aided Design*, vol. 29, no. 10, pp. 727–735, 1997.

[13] Jones, M.W. and Satherley, R.A., “Shape Representation using Space Filled Sub-Voxel Distance Fields,” *International Conference on Shape Modeling and Applications*, 316-325, 2001.

[14] F. S. Nooruddin and G. Turk, “Simplification and repair of polygonal models using volumetric techniques,” *IEEE Trans. Vis. Comput. Graph.*, vol. 9, no. 2, pp. 191–205, 2003.

[15] Telea, A. and Jalba, A., "Voxel-based assessment of printability of 3D shapes", *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, Springer Berlin Heidelberg, pp. 393-404, 2011.

[16] J. R. Rossignac and A. A. G. Requicha, “Offsetting operations in solid modelling,” *Comput. Aided Geom. Des.*, vol. 3, no. 2, pp. 129–148, 1986.

[17] Y. Chen and X. Xu, “Manufactruability analysis of infeasible features in polygonal models for web-based rapid prototyping,” *Proc. - 2010 Int. Conf. Manuf. Autom. ICMA 2010*, pp. 120–127, 2010.

[18] P. Das, R. Chandran, R. Samant, and S. Anand, “Optimum Part Build Orientation in Additive Manufacturing for Minimizing Part Errors and Support Structures,” *Procedia Manuf.*, vol. 1, pp. 343–354, 2015.

[19] G. Strano, L. Hao, R. M. Everson, and K. E. Evans, “A new approach to the design and optimisation of support structures in additive manufacturing,” *Int. J. Adv. Manuf. Technol.*, vol. 66, no. 9–12, pp. 1247–1254, 2013.

[20] “Voxelizer Software.” [Online]. Available: http://voxelizer.com/.

[21] P. Alexander, S. Allen, and D. Dutta, “Part orientation and build cost determination in layered manufacturing,” *Comput. Des.*, vol. 30, no. 97, pp. 343–356, 1998.

[22] Campbell, I., Combrinck, J., De Beer, D., & Barnard, L., "Stereolithography Build Time Estimation Based On Volumetric Calculations," *Rapid Prototyping Journal*, 14(5), 271-279. Doi: 10.1108/13552540810907938, 2008.

[23] Hollis, R. L., Geometry-Based Estimation for Additive Fabrication. (Doctor of Philosophy), Alabama in Huntsville, Huntsville, Alabama, 2001.

[24] Di Angelo, L., & Di Stefano, P., "A neural network-based build time estimator for layer manufactured objects," *International Journal of Advanced Manufacturing Technology*, 57(1-4), 215-224. doi: 10.1007/s00170-011-3284-8, 2011

[25] J. D. Hiller and H. Lipson, "Fully Recyclable Multi-Material Printing," Solid Free. Fabr. Symp., pp. 98–106, 2009.

[26] J. Hiller and H. Lipson, "Design and analysis of digital materials for physical 3D voxel printing," *Rapid Prototyp. J.*, vol. 15, no. 2, pp. 137–149, 2009.

[27] E. L. Doubrovski, E. Y. Tsai, D. Dikovsky, J. M. P. Geraedts, H. Herr, and N. Oxman, "Voxel-based fabrication through material property mapping: A design method for bitmap printing," *CAD Comput. Aided Des*., vol. 60, pp. 3–13, 2015.

[28] A. Brunton, C. A. Arikan, and P. Urban, "Pushing the Limits of 3D Color Printing: Error Diffusion with Translucent Materials," *Siggraph*, vol. 1, no. 212, pp. 1–12, 2014.

[29] V. Chandru, S. Manohar, and C. E. Prakash, "Voxel-based modeling for layered manufacturing," IEEE Comput. Graph. Appl., vol. 15, no. 6, pp. 42–47, 1995.

[30] Lin, F., Seah, H.S., Wu, Z. and Ma, D., "Voxelization and fabrication of freeform models", *Virtual and Physical Prototyping*, 2(2), pp.65-73, 2007.

[31] D. Ma, F. Lin, and C. K. Chua, "Rapid prototyping applications in medicine. Part 1: NURBS-based volume modelling," *Int. J. Adv. Manuf. Technol*., vol. 18, no. 2, pp. 103–117, 2001.

[32] D. Ma, F. Lin, and C. K. Chua, "Rapid prototyping applications in medicine. Part 2: STL file generation and case studies," *Int. J. Adv. Manuf. Technol*., vol. 18, no. 2, pp. 118–127, 2001.

[33] Forstmann, S. and Jun, O.H.Y.A., "Efficient, high-quality, GPU-based visualization of voxelized surface data with fine and complicated structures," *IEICE TRANSACTIONS on Information and Systems*, 93(11), pp.3088-3099, 2010.

[34] S. Patil and B. Ravi, "Voxel-based representation, display and thickness analysis of intricate shapes," *Proc. - Ninth Int. Conf. Comput. Aided Des. Comput. Graph. CAD/CG 2005*, vol. 2005, pp. 415–420, 2005.

[35] L. D. Sturm, C. B. Williams, J. A. Camelio, J. White, and R. Parker, "Cyber-Physical Vunerabilities In Additive Manufacturing Systems," *Int. Solid Free. Fabr. Symp. Proc.*, pp. 951–963, 2014.

[36] M. Cloots, A. B. Spierings, and K. Wegener, "Assessing new support minimizing strategies for the additive manufacturing technology SLM"," *Proc. Solid Free. Fabr. Symp.*, pp. 131–139, 2013.