Literature review

Sizwe Zwane

May 2021

Abstract

In this review, we research methods that allow 3D mesh object to be converted in varies 2D versions. We explore additive manufacturing, voxelization and Weak-Morse functions. We compare the algorithm applied in terms of their computational complexity, accuracy, and applicability. We explore improves on the algorithm through hardware features such as Graphics Processing Units and parallel programming.

1 Introduction

2D images of 3D objects have been used throughout history to communicate visually. In fields such as engineering and architecture, 2D images are used describe buildings, structures and other objects. A few of the types of images such as sections, ground plans and elevations are used a lot to depict 3D objects such as houses, mosques, parks and shopping centres in these fields. But also, in these fields they have instruments for collecting 3D images of 3D object, and one of such instruments is a 3D laser scanner.

3D laser scanning has improved traditional surveying as it's able to provide much greater accuracy. Some 3D laser scanning devices are able to collect up to a million vertices per second. What is needed is a method of converting the 3D Laser scanned models of structures into various 2D versions of it. The Zamani project has required to produce a tool for generating elevations, sections, ground plans, top views, and all generic views from 3D laser scanned models.

A lot of research behind the conversion of 3D images into 2D images happened but not a lot is on very complex 3D models of structures converted into various complex 2D applications of it. These complex 2D applications have great use in the real world. The 3D models expected to work with are of a mesh structure. The use of converting these 3d mesh structure in elevations, sections, etc, is an alternative to having individuals draw the 2D images with pencil or on CAD systems. These are better due to computer processing algorithms for these conversions are much more generalised and a lot faster to compute than a human.

The structure of this review is as follows. In section 2, the methods explore for solving this are voxelization [1], multiple slicing plane [6] and mathematical level sets [4]. Additionally, a geological application. In section 3, is an analysis of comparing the various methods researched. Then its closed in section 4.

2 Past literature

2.1 Voxelization of mesh

Wang et al. (2006) [15] introduces a method slicing a voxelized model of 3D mesh pattern. The mesh pattern made up of triangles. Each of the triangles is assigned a colour.

The method for slicing works as follows. It starts off by computing the contour polygon of the intersection of the triangular face by using 3D polygon clipping to determine a necessary bounding box. Then determine the boundary voxels (figure 1) intersecting the faces and the boundary box. Then determine the contour polygon's area of the face's intersection and each boundary voxel's intersecting face. Lastly, for each triangular face, a colour must be assigned to each face then each of the voxels must be also assigned a colour based in the triangular face.

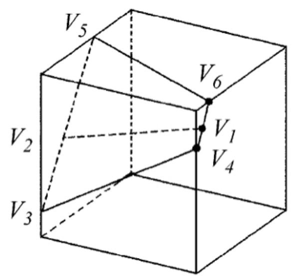


Figure 1: Two faces intersecting on a voxel

Each of the triangular faces are described by three vertices, a colour and a normal vectors. The voxelization process (figure 2) become more computationally heavy as when the size of each voxel becomes smaller. Then processing voxels to enough accuracy is still a target.

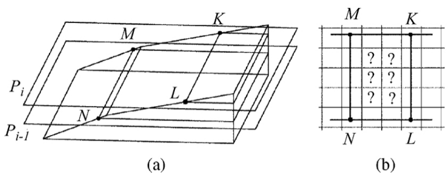


Figure 2: Voxelization between two parallel planes

The colourisation process of voxels helps simplify the identification of empty space versus the non-empty spaces. Identifying empty space is very critical for being able to project images that are further than the slicing plane.

Voxelization consume large amounts of memory. Schwars and Seidel (2010) [12] are addressing the use of octrees for storage. The accuracy of different kinds of voxelization is addressed, shown in figure 3.

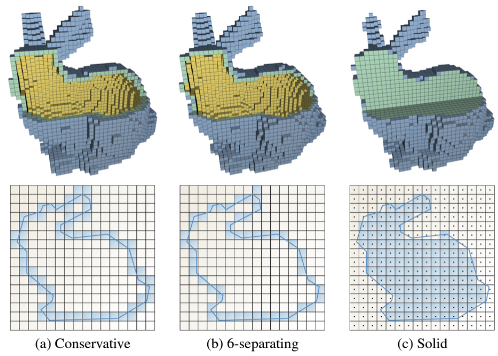


Figure 3: Types of voxelizations

2.2 Mathematical level set approach

Meha-Parra et al. [9] covers embedding a lot of the topology of manifolds in Morse theory [7] is based on twice differentiable functions, this is referred to as Strong-Morse functions in this paper. In figure 4, the kinds of manifolds produce from Strong-Morse functions. These functions are defined on continuous domains. All the critical point of s Strong-Morse function are non-degenerate.

Now with Weak-Morse functions, they are able to satisfy the Morse criterion but a finite set of slices. Weak-Morse and non-Morse function are defined equivalently in discrete domains. They are not equivalent in continuous domains. The figure 5 categorises the slicing foe Weak-Morse functions.

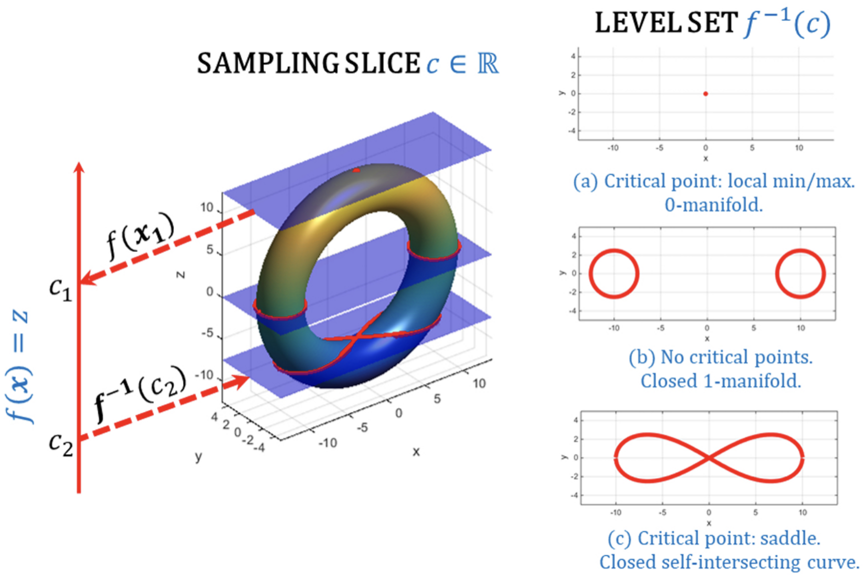


Figure 4: Variations of manifolds for a Morse

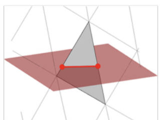
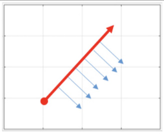
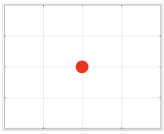
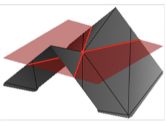
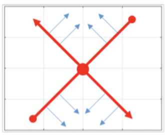
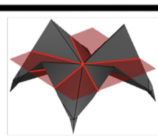
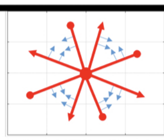
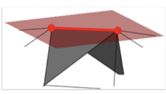
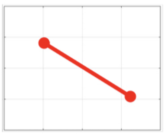
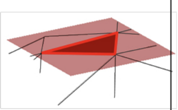
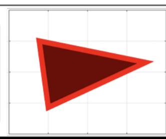
	Continous case	Discrete case	3D Slicing	2D Level Set
Morse level sets: $\nabla_{\mathcal{M}} f(x) \neq 0 \ \lor\ \det(\mathbf{H}_{\mathcal{M}} f(x)) \neq 0$	Non-critical point $x \in \mathcal{M}$: $f^{-1}(c)$: • 1-manifold • Oriented	Face/plane intersection: $f^{-1}(c),\ t_k \in T$: • p_0: edge intersection $e_i \in t_k$ • p_f: edge intersection $e_j \in t_k$		
	Local min (or max) $x \in \mathcal{M}$: $f^{-1}(c)$: • 0-manifold	Point/plane intersection: $f^{-1}(c),\ x_i \in X,\ x_j \in N_X^X(x_i)$: • $x_i^z = c$ • $x_j^z > c$ (< for local max)		
	Non-degenerate saddle $x \in \mathcal{M}$: $f^{-1}(c)$: • 1D non-manifold • Oriented, self-intersecting	Point/plane, face/plane intersection: $f^{-1}(c),\ x_i \in X,\ N_X^T(x_i)$: • $x_i^z = c$ • Adjacent faces $N_X^T(x_i)$: $num\_intersections = 4$		
Non-Morse level sets: $\nabla_{\mathcal{M}} f(x) = 0 \ \land\ \det(\mathbf{H}_{\mathcal{M}} f(x)) = 0$	Degenerate saddle $x \in \mathcal{M}$: $f^{-1}(c)$: • 1D non-manifold • Oriented, self-intersecting	Point/plane, face/plane intersection: $f^{-1}(c),\ x_i \in X,\ N_X^T(x_i)$: • $x_i^z = c$ • Adjacent faces $N_X^T(x_i)$: $num\_intersections = 2m, \quad m > 2$		
	Critical 1D set $E \subset \mathcal{M}$: $f^{-1}(c)$: • 1D non-manifold • Non-Oriented, self-intersecting	Edge/plane intersection: $f^{-1}(c),\ (x_i, x_j) \in E$: • $x_i^z = c,\ x_j^z = c$ • $(x_{opp}^z(x_i, x_j) - c) \cdot (x_{opp}^z(x_j, x_i) - c) > 0$		
	Critical 2D set $F \subset \mathcal{M}$: $f^{-1}(c)$: • 2-manifold	Face/plane intersection: $f^{-1}(c),\ (x_i, x_j, x_k) \in T$: • $x_i^z = c,\ x_j^z = c,\ x_k^z = c$		

Figure 5: Weak-Morse function in continuous and discrete domains, and appropriate slicing

2.3 Multiple slicing planes

Gregori et al. (2014) [5] focuses on the efficiency of triangular mesh slicing algorithms. This is from the perspective of Additive Manufacturing, which is a process of building 3D objects based on layering flat slices of the object. This process is famously known as 3D printing. The triangular mesh is sliced into 2.5-D contours (flat surfaces always have a 3D form in the real world) of the object.

Since this paper focuses on efficiency, algorithmic complexities are uses for measure so, a few parameters must be defined. n and k are the number on triangles and slicing planes respectively, also $\overline{n}$ and $\overline{k}$ are the average number of triangles and slicing planes respectively.

The simplest method of slicing is the trivial slicing by testing each mesh triangle along all the slice planes. The other algorithms compared are the Sweep Plane Slicing [8] and Triangle Grouping [13]

The proposed optimal algorithm is called the Incremental Slicing Algorithm. Intervals are formed by taking inclusive bounds of the highest and lowest z-coordinates of each triangle. An interval tree [14] are used to the intervals, and for a given plane value z, we use the stabbing problem [11] to retrieve all interval containing the plane.

The comparison of the various algorithms in tabulated in figure 6

Algorithm	Memory Construction	Worst case slicing	Contour assembly	Worst case total (assuming $k = \Omega(\log n)$)
Trivial	$\mathcal{O}(n)$	$\mathcal{O}(nk)$	$\mathcal{O}(\overline{n}\log\overline{n})$	$\mathcal{O}(nk + \overline{n}\log\overline{n}k)$
Triangle Grouping	$\mathcal{O}(n\log n)$	$\mathcal{O}(n^2 + n\log n\overline{k})$	$\mathcal{O}(\overline{n})$	$\mathcal{O}(n^2 + n\log n\overline{k})$
Sweep Plane	$\mathcal{O}(n\log n)$	$\mathcal{O}(k^2 + n^2)$	$\mathcal{O}(\overline{n})$	$\mathcal{O}(n^2 + k^2)$
Incremental Slicing (our method)	$\mathcal{O}(n\log n)$	$\mathcal{O}(n\overline{k})$	$\mathcal{O}(\overline{n})$	$O(n\overline{k})$

Figure 6:

Minetto et al. (2017) [10] proposes a method for slicing unstructured mesh model using a series of parallel planes. This is common among a lot of the additive layered manufacturing literature. The contour construction is done using constant distance between the parallel planes (uniform slicing) or variable distance between parallel planes (adaptive slicing). Adaptive Slicing is critical for meshes where the manifolds produced by uniform are the same in successive planes. This is demonstrated in figure 7.

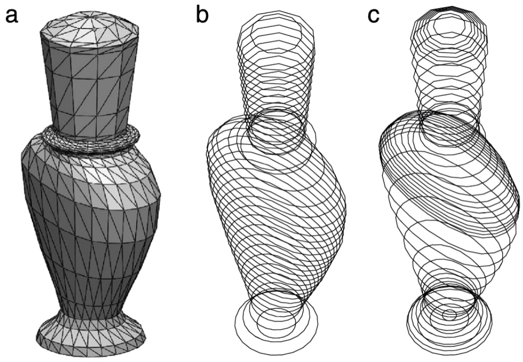


Figure 7: Triangle mesh model (a) and its variation of uniform slicing (b) and adaptive slicing (c)

These algorithms have complexity of $O(n + k + m)$ and $O(nlogk + k + m)$ for uniform slicing and adaptive slicing respectively where m is the number of triangle-to-plane intersections.

With all the improvements to the optimisation of algorithms, Dant (2016) [2] proposes, since additive manufacturing expensive computationally then parallel processing hardware can be utilised such as graphics processing units.

2.4 Geological application approach

Elsheikh et al. (2014) [3] proposes an algorithm for manifold surface meshes. It's very effective in handling triangle-to-triangle intersection in degenerate cases. The main idea is using large sets of triangle-to-edge cases together with a method of tracking curve intersections. The geological component is very important to this project as the structure that are being studied are embedded in some geology. This adds a reliability level as geological structures are very complex. The figure 8 is applying a triangular mesh imitating geological structures after imprinting intersection curves and a step for optimising the surface mesh

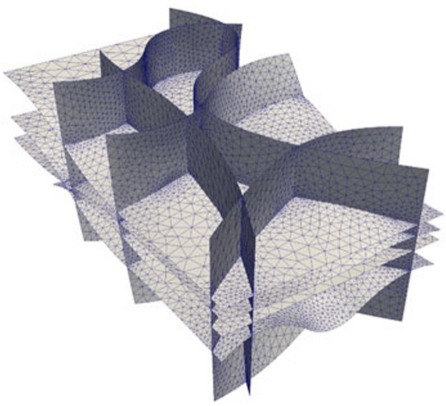


Figure 8: Geological meshing

3 Analysis

The accuracy of each approach is very important. Accuracy can be hindered by practical factors such as storage and speed. In heavily theoretical approaches, correctness matters more than accuracy. But when applying these approaches, they should be correct enough within the bounds of practical factors. The level set approach is a very theoretical approach as it based on a lot of mathematically computable functions. It does not suffer practically whereas the others do. The accuracy of voxelization, multiple slicing planes, etc. are hindered by memory and speed. We can't have an infinity number of triangles, slicing planes or voxels. The storage predicament is handled by using data structures such as octrees, graphs and binary trees as an alternative to traditional lists. And the speed predicament is handled by trying to optimise the algorithm where necessary. This is done by comparing algorithms with their complexities.

With the type of structure studied in this project, there is varying applicability with the methods used. The structures studied should be expected to have some inconsistencies such as noise from the data captured imaging has limits. The approaches from the literature have varying applicability. The level set approach won't be able to handle as much noise as voxelization. The more rounded objects can be proceed much better using level set approach than multiple slicing planes or voxelization.

For the table below, the index is a number between 1 and 3 where 1 is poor, 2 is fair and 3 is good.

Cite	Method	Main findings	Applicability index	Accuracy index
[2]	Rapid slicing	Hardware speed up for parallel executions using GPUs	2	3
[3]	Surface manifold detection	algorithm able to deal with geological models	3	2
[5]	Additive manufacturing	optimises adaptive slicing	2	2
[8]	Rapid slicing	Implements sweep plane slice	3	3
[9]	Morse manifolds	Slicing depends on degeneracy	1	2
[10]	Unstructured triangular mesh slicing	Sweeping plane optimisation for unstructured triangular mesh	2	3
[12]	Voxelization	Octree stored voxels	2	3

4 Conclusion

In this review, methods that could help solve the problem of producing section, elevations and other views for 3D mesh structures are explored. A lot of the literature around 3D slicing is focused on additive manufacturing or rapid slicing. That's why a lot of its literature has a lot of optimization methods

We learn the voxelization is very computationally expensive and that better data structures are used typically for their storage. And the detail at which this is processed is very important because heavily pixelated imaging is not desired.

Although all these methods are very useful, not a lot of the literature explores these methods in a real-world complexity such as figure 9.

Figure 9: Mountain Museum

References

[1] Cohen-Or, D., and Kaufman, A. Fundamentals of surface voxelization. *Graphical models and image processing 57*, 6 (1995), 453–461.

[2] Dant, C. Design and implementation of asymptotically optimal mesh slicing algorithms using parallel processing.

[3] Elsheikh, A. H., and Elsheikh, M. A reliable triangular mesh intersection algorithm and its application in geological modelling. *Engineering with Computers 30*, 1 (2014), 143–157.

[4] Giaquinta, M., and Modica, G. *Mathematical analysis: An introduction to functions of several variables.* Springer Science & Business Media, 2010.

[5] Gregori, R. M., Volpato, N., Minetto, R., and Da Silva, M. V. Slicing triangle meshes: An asymptotically optimal algorithm. In *2014 14th International Conference on Computational Science and Its Applications* (2014), IEEE, pp. 252–255.

[6] Kai, C. C., and Fai, L. K. Rapid prototyping. *Nanyang technological university* (1997).

[7] Matsumoto, Y. *An introduction to Morse theory*, vol. 208. American Mathematical Soc., 2002.

[8] McMains, S., and Séquin, C. A coherent sweep plane slicer for layered manufacturing. In *Proceedings of the fifth ACM symposium on Solid modeling and applications* (1999), pp. 285–295.

[9] Mejia-Parra, D., Ruiz-Salguero, O., Cadavid, C., Moreno, A., and Posada, J. Level sets of weak-morse functions for triangular mesh slicing. *Mathematics 8*, 9 (2020), 1624.

[10] Minetto, R., Volpato, N., Stolfi, J., Gregori, R. M., and Da Silva, M. V. An optimal algorithm for 3d triangle mesh slicing. *Computer-Aided Design 92* (2017), 1–10.

[11] Schmidt, J. M. Interval stabbing problems in small integer ranges. In *International Symposium on Algorithms and Computation* (2009), Springer, pp. 163–172.

[12] Schwarz, M., and Seidel, H.-P. Fast parallel surface and solid voxelization on gpus. *ACM transactions on graphics (TOG) 29*, 6 (2010), 1–10.

[13] Tata, K., Fadel, G., Bagchi, A., and Aziz, N. Efficient slicing for layered manufacturing. *Rapid Prototyping Journal* (1998).

[14] Van Kreveld, M., Schwarzkopf, O., de Berg, M., and Overmars, M. *Computational geometry: Algorithms and Applications*, 3rd ed. Springer-Verlos TELOS, 2008.

[15] Wang, D.-X., Guo, D.-M., Jia, Z.-Y., and Leng, H.-W. Slicing of cad models in color stl format. *Computers in industry 57*, 1 (2006), 3–10.