Progress In Electromagnetics Research C, Vol. 134, 93–102, 2023

Generation of Spatially-Variant Anisotropic Metamaterials in 3D Volumetric Circuits

Asad U. H. Gulib[1], Jeremie Dumas[2], Cesar L. Valle[1], Edgar Bustamante[1], Daniele Panozzo[2], and Raymond C. Rumpf[1, *]

Abstract—3D printing is revolutionizing manufacturing and is now being considered in the electronics industry. The creation of the first 3D volumetric circuit (3DVC) has created a way to make circuits smaller, lighter, into unconventional form factors and exploit physics like anisotropy more effectively than planar geometries can. While this is exciting, many problems must be solved to make 3DVCs a reality. One of these problems is electromagnetic interference and mutual coupling between components that are expected to be highly problematic in high frequency 3DVCs. Spatially-variant anisotropic metamaterials (SVAMs) could be a solution to overcome this difficulty, but research in this area is not possible without a way to generate SVAMs around multiple components. In this paper, an algorithm is integrated into CAD software that can generate SVAMs for 3D circuits which will enable future studies of SVAMs.

1. INTRODUCTION

3D printing or additive manufacturing has revolutionized the manufacturing industry and holds great promise to change how electronic devices are designed and manufactured [1]. The geometric freedom offered by 3D printing enables the idea of a 3D volumetric circuit (3DVC) where electrical components can be placed at any position and be oriented at any angle throughout the entire volume of the circuit. The first 3DVC manufactured in a single-step process was designed and 3D printed by Carranza et al. [2]. This opened a new paradigm in circuit technology because 3DVCs can be made smaller, lighter, formed into unconventional form factors, and exploit physics that are difficult in planar circuits. The greater geometric freedom component placement allows trace lengths to be shorter to improve bandwidth and power efficiency. However, to realize these benefits several challenges must be overcome including thermal management, testing, mutual coupling, design practices, and interference.

In 2012, a novel algorithm to generate spatially-variant lattices (SVLs) was introduced by Rumpf and Pazos [3]. This work followed a few dissertations from Garcia [4], Pazos [5], Berry [6], Gutierrez [7] showing the progress in spatial variance, metamaterials and photonic crystals. Other advancements in 3D printed SVLs have been reported in [8–13]. In [8], it was shown that the near field of a device can be reshaped by embedding it in a spatially variant anisotropic metamaterial (SVAM). This research demonstrated the manipulation of near fields in ways that can reduce the interference and mutual coupling between components in close proximity.

In contrast to prior work, the SVAMs discussed here are based on space stretching where the coupling between adjacent components is reduced by increasing the electrical distance between components while the physical distance remains unchanged. Transformation optics (TO) [14, 15] provides a way to determine the material properties that are needed to electrically stretch the space between components. To realize this concept in a 3DVC, an algorithm is needed to calculate the

Received 30 March 2022, Accepted 31 May 2023, Scheduled 2 July 2023
* Corresponding author: Raymond C. Rumpf (rcrumpf@utep.edu).
[1] The University of Texas at El Paso, USA. [2] New York University, USA.

geometry of the SVAM that completely infiltrates the space between all the components. This paper presents the algorithm that generates the SVAM for 3DVCs. A prototype of the algorithm was developed in [16] using the finite-difference method (FDM). However, the staircase approximation of FDM led to errors in the lattices and inefficiencies in the algorithm. In the present work, this problem was overcome by using the finite element method (FEM). Finally, the proposed algorithm was implemented and integrated into the CAD software Blender [17] using its python scripting interface. This makes it easier for a user to design 3D circuits in Blender and generate SVAMs in the same environment without having to deal with the underlying numerical methods. This will enable future research to study how to properly implement SVAMs and to evaluate their effectiveness.

2. SPACE STRETCHING WITH A NUM

There are at least two ways that SVAMs might be used to reduce coupling and interference, space stretching [18], and field sculpting [8]. The SVAMs generated in the present work operate on the principle of space stretching, where the electrical distance between components is increased without changing the physical distance. Transformation optics (TO) was used to determine the properties of the medium that can perform this function. The medium that can provide space stretching along the z axis is determined by a coordinate transformation that scales the z axis by a factor a. This coordinate transformation is applied to Maxwell's equations, the equations are rearranged to associate the math of the transform with permeability μ and permittivity ε instead of the coordinates, and the result describes the medium that performs this space stretching. The process gives

$$[\mu'] = [\varepsilon'] = \begin{bmatrix} \varepsilon a & 0 & 0 \\ 0 & \varepsilon a & 0 \\ 0 & 0 & \varepsilon/a \end{bmatrix} \tag{1}$$

The first two diagonal elements in Eq. (1) are called the ordinary permittivity ε_{o}, and the third is called the extraordinary permittivity ε_{e}. The values of ordinary and extraordinary permittivity can be obtained using the following equations [18, 19]

$$\begin{aligned} \varepsilon_{\mathrm{o}} &\cong f\varepsilon_1 + (1-f)\,\varepsilon_2 \\ \frac{1}{\varepsilon_{\mathrm{e}}} &\cong \frac{f}{\varepsilon_1} + \frac{1-f}{\varepsilon_2} \end{aligned} \tag{2}$$

where ε_1 and ε_2 are the permittivity values of the two materials comprising the artificial medium providing the anisotropy, and f is the volumetric fill factor for the fraction of the SVAM composed of ε_1. The NUMs are non-resonant, incredibly broadband, and typically exhibit a uniform response from DC up to a cutoff frequency where they begin to be resonant [18]. Equation (1) corresponds to a negative uniaxial medium when $a > 1$. There are no negative constitutive values in a NUM. The negative label is convention and simply refers to the fact that the ordinary permittivity is greater than the extraordinary permittivity. Both the ordinary and extraordinary permittivities are positive quantities in this work. In [18], it was shown that NUM can be formed using alternating layers of two different materials with different constitutive values. The resulting structure is called a negative uniaxial metamaterial (NUM). It was further shown that space stretching is maximized when the thicknesses of all the layers are equal and made to be less than a quarter wavelength inside the medium [18]. The concept of forming a NUM between two objects is illustrated in Figure 1.

The NUM is not a Bragg grating because the layers are much too subwavelength to produce any scattering. To demonstrate the operation of a NUM, a directional coupler was simulated using the finite-difference frequency-domain (FDFD) method [20–22]. The waveguides in a directional coupler exchange power periodically when they are brought close enough together to be electromagnetically coupled [20]. In this simulation, the NUM has increased the electrical distance between the waveguides, thus significantly weakening the directional coupling. More intuitively for some, the weakening of the directional coupling can also be explained through the mechanism of field sculpting [19]. The near-field of a device embedded in a NUM tends to develop in the directions with the highest constitutive values. In this case, the evanescent field is pushed in the transverse directions instead of in the direction connecting the waveguides. Different materials between two waveguides have been used to show that it is only the NUM that significantly weakens the directional coupling.

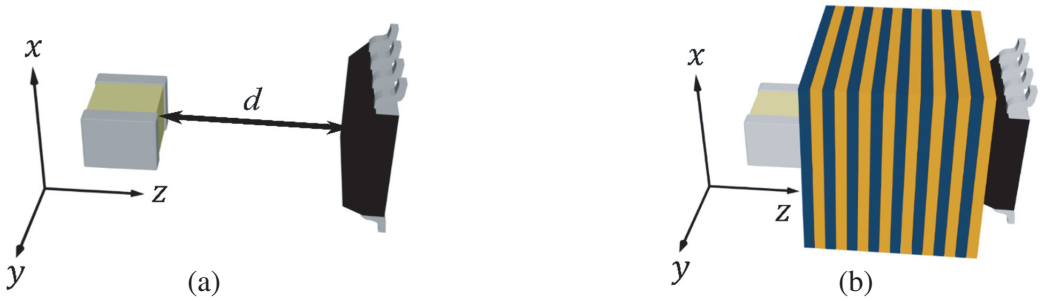


Figure 1. (a) Two objects are placed at a distance d in z direction. (b) Alternating layers of two materials form a NUM between the two objects.

The three-dimensional problem considered here was reduced to a two-dimensional simulation using the effective index method described in [20]. Four separate simulations were performed with a different homogeneous mediums placed between the same two waveguides. Figures 2(a)–(d) show various isotropic mediums placed between the waveguides. While the directional coupling is the weakest with air placed between the waveguides, directional coupling is still observed in all four cases. A NUM is placed between the waveguides in Figure 2(e), and the directional coupling is suppressed dramatically [23].

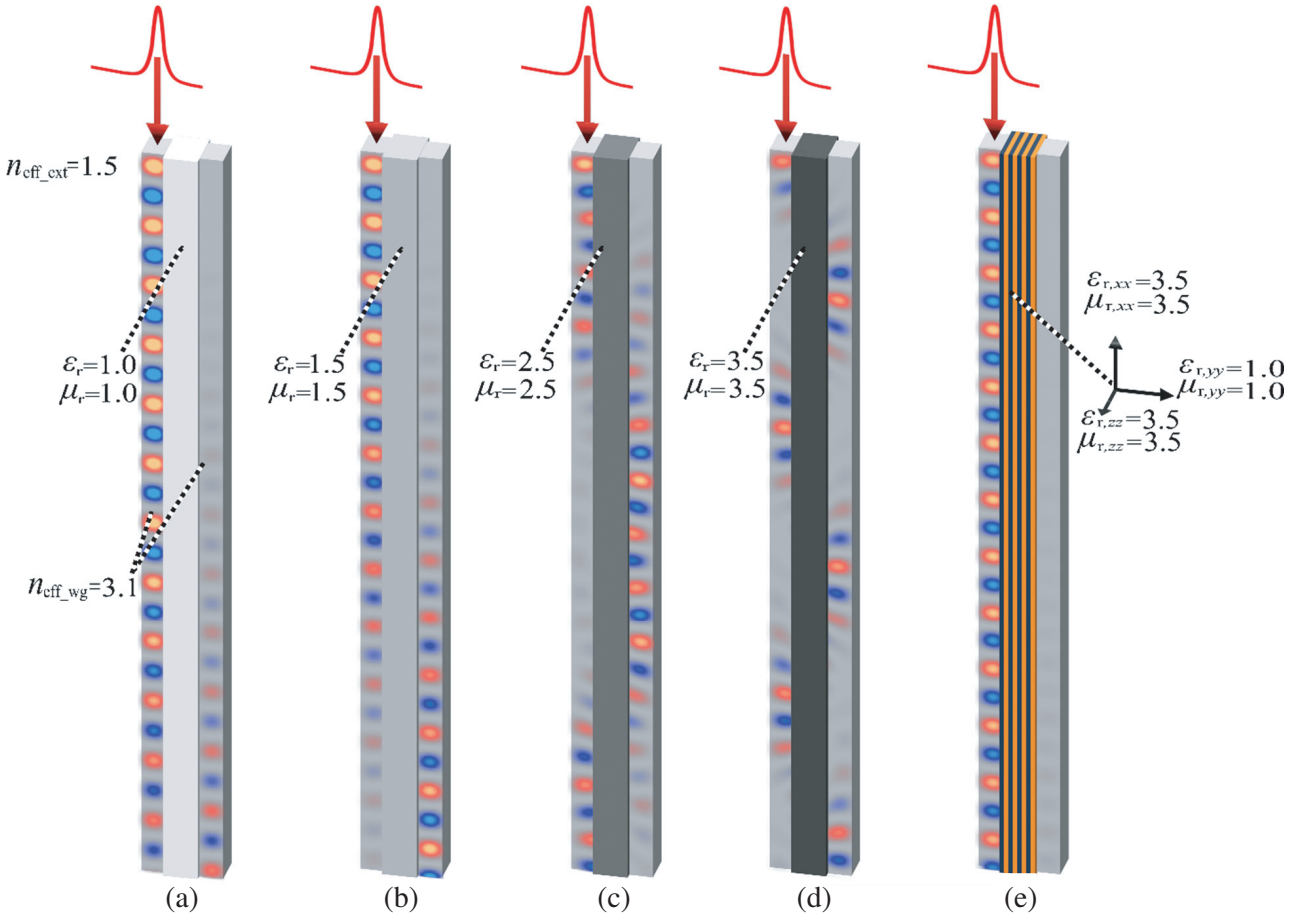


Figure 2. Simulation results for directional coupler using different materials between the waveguides. (a)–(d) Range of homogeneous materials between waveguides. (e) NUM between the waveguides.

3. ALGORITHM TO GENERATE A SVAM

It is trivial to generate a NUM between just two objects. However, it is not as clear how to place a NUM between multiple components in a 3DVC while keeping the overall structure of the NUM smooth, continuous, and free of defects. The orientation of the NUM must be spatially varied while being kept continuous and maintaining the baseline design of alternating layers between adjacent components. The resulting spatially-varied NUM is called a spatially-variant anisotropic metamaterial (SVAM). This section presents an algorithm that can generate an SVAM around any number of components in a 3DVC.

The core of the algorithm was developed using the geometry processing library libigl [24] which is written in C++. The first step of the algorithm is to import the meshes of the 3DVC components and create a bounding region that will contain the SVAM to the vicinity of the components. The second step is to apply the Voronoi algorithm to identify regions around each component where isolated sections of the SVAM will be generated and then later connected. The space allocated to each object from the Voronoi algorithm is called Voronoic cell. Third, volumetric tetrahedral meshes are generated over the boundary of each Voronoic cell. Fourth, Laplace's equation is solved in each Voronoic cell to generate a linear gradient function inside the cells that extends from 0 at the boundary of the component to 1 at the boundary of the Voronoic cell. Fifth, isocontours are created from these gradients that become the interfaces between the alternating materials comprising the SVAM. To bound the SVAM to the final shape of the circuit, a Boolean operation is performed to remove any part of the SVAM that resides outside of what is defined by the shapefile.

To illustrate the algorithm, consider the 3DVC shown in Figure 3(a) in which an SVAM will be generated. The black lines forming a box around the components define the final shape of the circuit with SVAM.

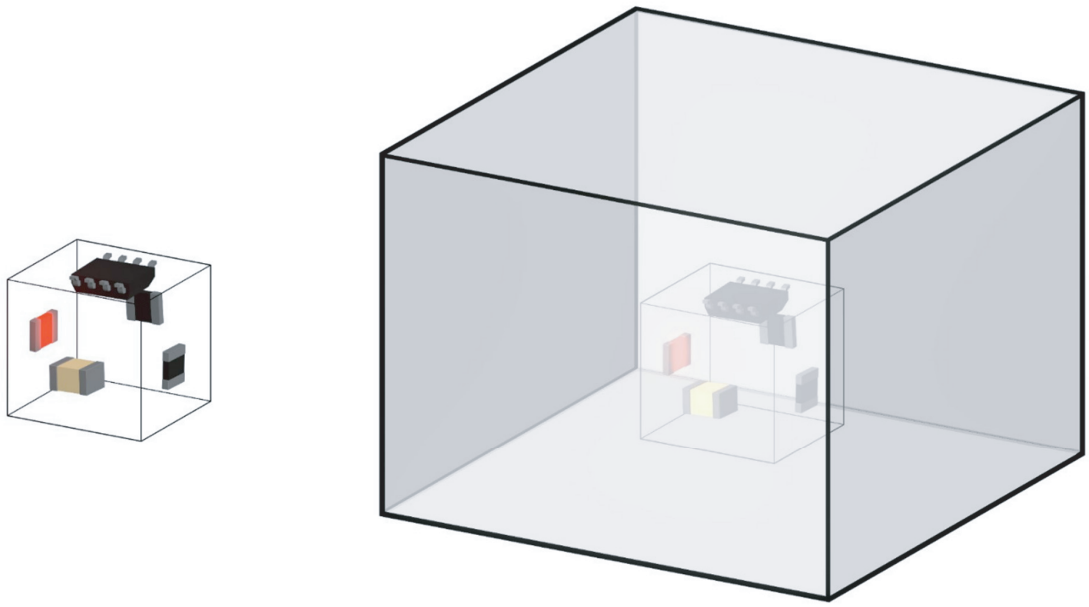


Figure 3. (a) Circuit components of a 3DVC. (b) Circuit components with bounding region.

First, the mesh data of circuit components are imported into the algorithm. The maximum distance between the components is measured, and a bounding region is created whose length is multiple times larger than the maximum distance. The bounding region should be large enough so that it will be able to perform the Boolean operation to give the desired shape of the circuit at the end of the algorithm. For the current circuit, the bounding region is a square box, and its length is five times larger than the maximum distance. The Voronoic cells will be generated inside the bounding region in following steps.

The bounding region restricts the Voronoic cells from extending out to infinity.

As the algorithm progresses, isolated regions of the SVAM will be generated around each circuit component separately inside each Voronoic cell. Later, these sub-SVAMs will be combined to form the overall SVAM. This requires separate volumes of space to be identified around each component such that when stacked together they perfectly fill the volume of the 3DVC without any overlaps or voids. The isolated regions are calculated by applying the Voronoi algorithm [25–27] on circuit components and the bounding region. The Voronoi algorithm identifies the volume of space around each component that is closer to that component than any other component. A Voronoic cell is calculated from the intersection of half-planes created by taking the perpendicular bisector of the closest two points [28]. The CAD files used in this algorithm are STL/OBJ, which are surface meshes. The vertices of these triangular meshes are the point clouds of the object. The Voronoi algorithm is applied to these point clouds. Some points at the outer surface of the circuit components do not have another point to perform Voronoi tessellation. So, these Voronoic cells reach infinity. The bounding region is created to provide additional points that restrict Voronoic cells from going to infinity. Figure 3(b) shows the bounding region with the circuit components.

Figure 4 shows the components surrounded with their respective Voronoic cells. The Voronoic cells are only surface meshes. The next step is to calculate a gradient inside the Voronoic cells. Volumetric meshes are needed to perform the gradient operation. The tetrahedral meshing library TetWild [29] was used to convert the surface meshes to volumetric tetrahedral meshes.

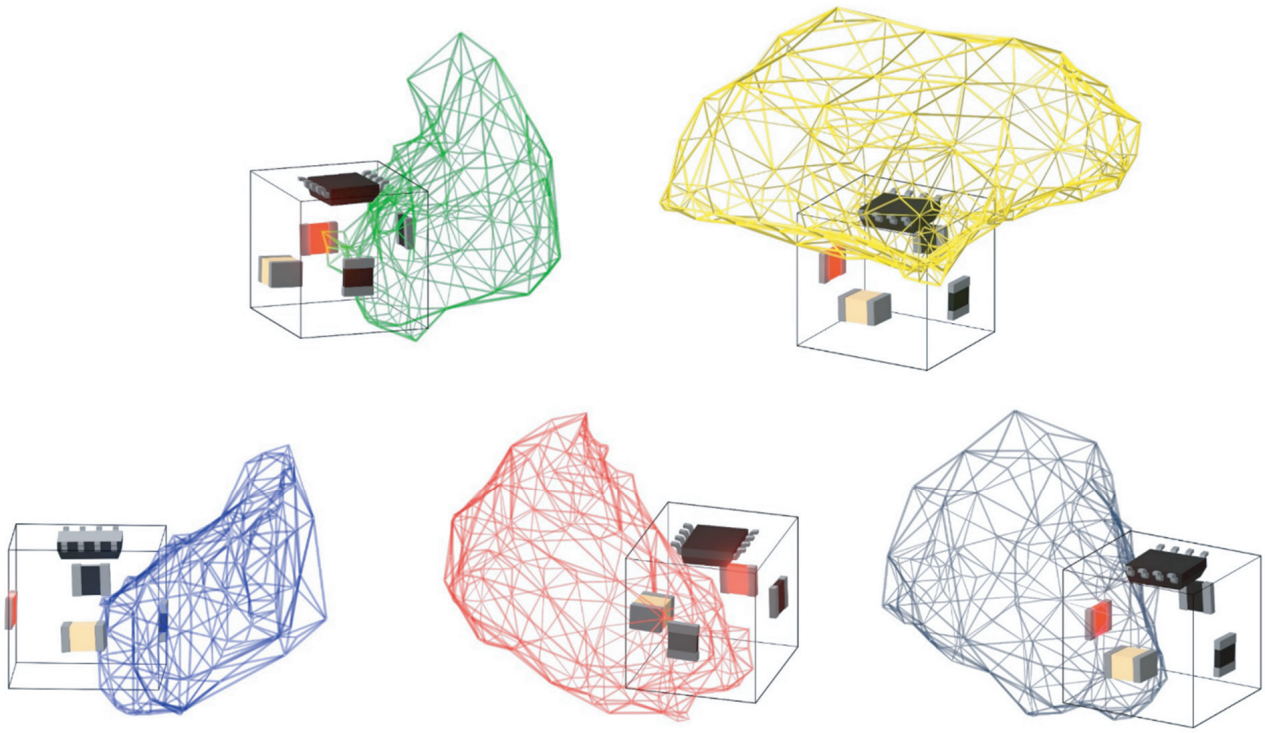


Figure 4. 3D circuit components with their corresponding Voronoic cells.

Each Voronoic cell has an outer surface created from the Voronoi algorithm. It also has a hollow inner surface defined by the outer boundary of the component it encloses. The boundary conditions are applied to both inner and outer surfaces. The hollow surface inside the Voronoic cell is set to 0, and the outer surface of the Voronoic cell is set to 1. Then a gradient is generated in the volume mesh inside the Voronoic cell by solving Laplace's equation. Laplace's equation for a scalar function f in a 3D space is

$$\nabla^2 f = 0 \tag{3}$$

Laplace's equation fills the meshes with a smooth gradient from 0 to 1 between two surfaces throughout

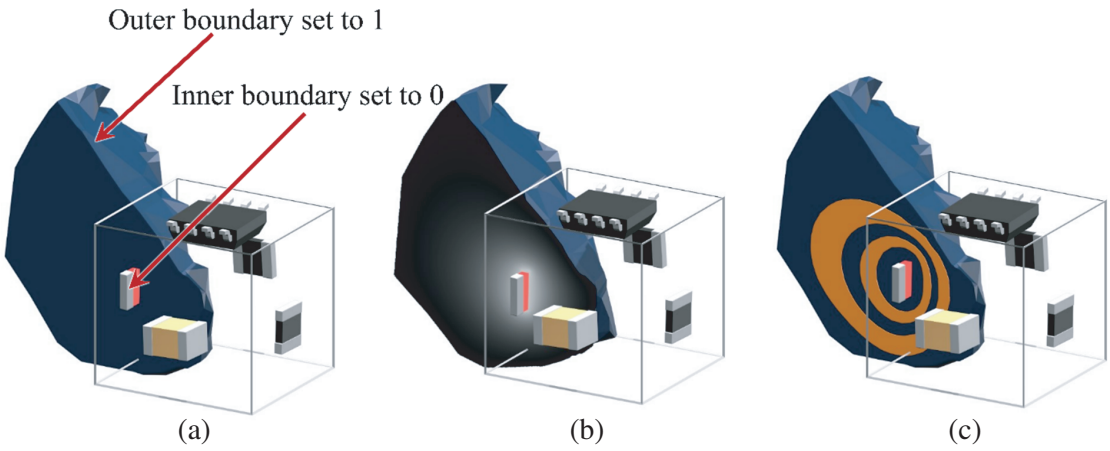


Figure 5. (a) Boundary conditions to solve Laplace's equation. (b) Gradient inside the Voronoic cell. (c) Rings formed inside Voronoic cell from the isocontours of the gradient.

the entire Voronoic cell. Figure 5 shows the cross-section of a Voronoic cell. Figure 5(a) shows the boundary conditions used to solve Laplace's equation. Figure 5(b) shows the gradient inside the Voronoic cell after solving the Laplace equation.

The marching tetrahedra method [30] is applied for isocontouring the gradient solution. The idea of this method is to contour the isosurface passing through each tetrahedron. The method iterates over all tetrahedra in the mesh and stitches together the final mesh [24]. This creates rings inside the Voronoic cell. Figure 5(c) shows the rings formed inside the Voronoic cells. These isocontours form the interfaces between alternating layers of the SVAM. Figure 6 shows the rings generated in the space between multiple circuit components. This figure also shows that the isocontours created in each Voronoic cell have been stitched together to form a single continuous SVAM.

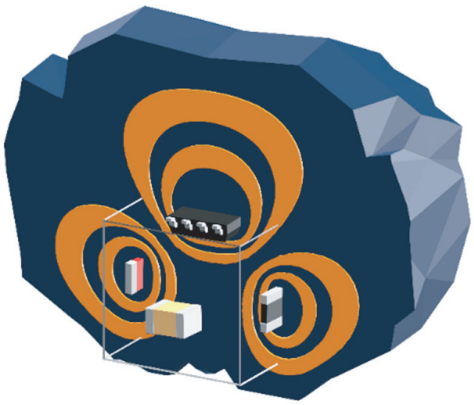


Figure 6. Cross-section of Voronoic cells showing rings formed around the multiple components.

The desired shape of the final circuit needs to be provided in the form of a CAD file. The file containing the final shape of the circuit will be called the shapefile. This is shown in Figure 6 by the white box outline. A Boolean operation is performed with the volumetric isocontours and the shapefile. It should be noted that the length of the bounding region should be large enough to perform a Boolean operation to result in the desired shape. Figures 7(a) and (b) show two alternating layers of SVAM after the Boolean operation. These two layers represent two different materials that form the SVAM. Figure 7(c) shows the two alternating layers together forming the final SVAM.

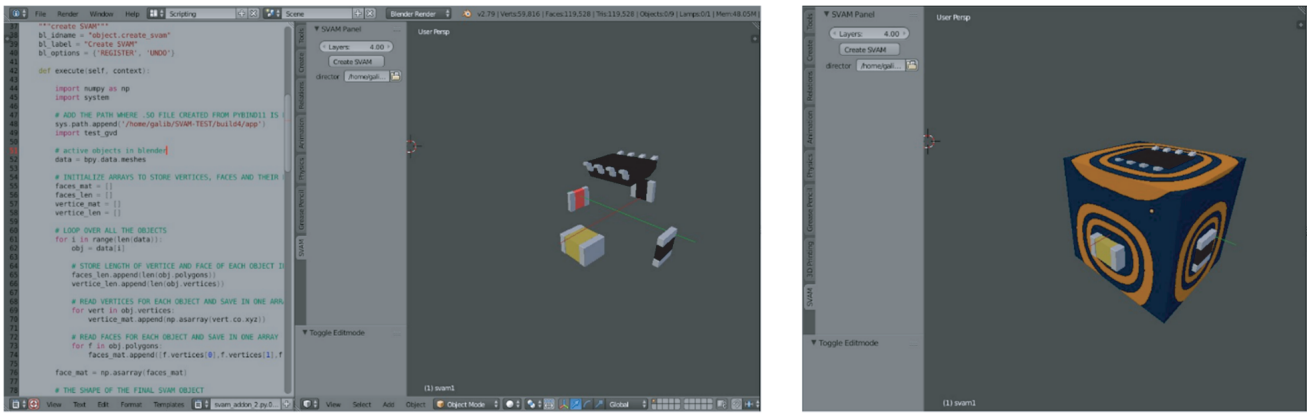
(a) (b) (c)

Figure 7. (a) Material 1 of the SVAM. (b) Material 2 of the SVAM. (c) Both materials together form the overall SVAM.

4. INTEGRATING THE SVAM ALGORITHM INTO BLENDER

The SVAM algorithm was integrated into the CAD software Blender to give it a graphical user interface (GUI) and make it more convenient to use. A user can design or import a 3DVC in Blender and use this algorithm to fill the space between components with an SVAM. Blender is a popular, free, and versatile CAD software that allows python scripting to incorporate custom add-ons. The algorithm was written in C++, and Blender uses python scripting. To merge the two, pybind11 [31] was used to connect the C++ SVAM algorithm to Python in Blender. Screenshots of the SVAM algorithm in Blender are shown in Figure 8. The left window in Figure 8(a) shows part of the python code that creates the SVAM panel and links the SVAM algorithm with Blender. The middle window is the SVAM panel that appears after running the python code. The right window is the Blender viewport showing circuit components where the SVAM will be formed around.

After importing or designing the circuit components in Blender, mesh data of the components are extracted using python, and these data are sent to the algorithm written in C++ using pybind11. The algorithm creates a bounding region, performs Voronoi tessellation, solves the Laplace equation, creates isocontours around each object, and performs the Boolean operation for the final SVAM. The C++ code returns two mesh files for two different materials which are visualized in the Blender 3D viewport. The two mesh files making the SVAM can be seen in blue and orange colors in Figure 8(b).



(a) (b)

Figure 8. (a) Screenshot from left to right of Blender showing the python scripting interface, the SVAM panel and the 3D circuit objects. (b) Screenshot of Blender showing the SVAM panel and the SVAM formed around the 3D circuit components.

The SVAM panel (middle window of Figure 8(a) and left window of Figure 8(b)) has three attributes. The first attribute at the top is 'Layers' which allows the user to select the number of alternating layers that will be generated around each component. The attribute 'Directory' allows users to browse the computer directory to upload the shapefile. After defining the number of layers and importing the shapefile, the 'Create SVAM' button is clicked to run the python and C++ code that generates the SVAM. After calculation, the final SVAM is shown along with the circuit in the 3D viewport of Blender. The right window in Figure 8(b) shows the created SVAM layers around the circuit components.

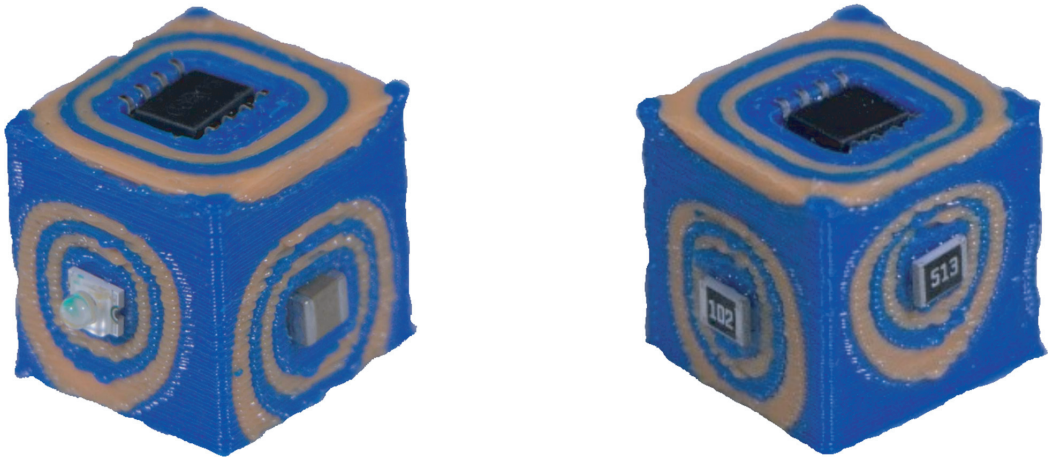


Figure 9. 3D printed 3DVC with SVAM.

5. DISCUSSION AND CONCLUSION

The circuit and SVAM depicted in Figure 7(c) was 3D printed using dielectric materials, and the manufactured device is shown in Figure 9. This shows that the algorithm can spatially vary the layers of two materials to form an SVAM, and the mesh has sufficient quality to be 3D printed. This algorithm will enable exploration of incorporating SVAMs into many types of 3DVCs. Given this algorithm, research can be performed that will determine how to best design and implement SVAMs. Topics that should be explored for SVAMs include materials, where to place the SVAM, what components to conform around, the number of layers required, and more. The future research associated with the SVAM algorithm is to design a functional circuit, manufacture and test the designed circuit.

REFERENCES

1. Gibson, I., D. W. Rosen, and B. Stucker, *Additive Manufacturing Technologies: Rapid Prototyping*, Springer, 2010.
2. Carranza, G. T., U. Robles, C. L. Valle, J. J. Gutierrez, and A. R. C. Rumpf, "Design and hybrid additive manufacturing of 3-D/volumetric electrical circuits," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, Vol. 9, 1176–1183, 2016.
3. Rumpf, R. and J. Pazos, "Synthesis of spatially variant lattices," *Optics Express*, Vol. 20, No. 14, 15263–15274, 2012.
4. Garcia, C., "3D printed spatially variant anisotropic metamaterials," Ph.D. Dissertation, University of Texas at El Paso, El Paso, 2014.
5. Pazos, J. J., "Digitally manufactured spatially variant photonic crystals," Ph.D. Dissertation, University of Texas at El Paso, El Paso, 2014.
6. Berry, E. A., "A spatially variant metamaterial design process for transformation electromagnetic devices," Ph.D. Dissertation, University of Texas at El Paso, El Paso, December 2016.

7. Gutierrez, J. J., "Independent and simultaneous control of electromagnetic wave properties in self-collimating photonic crystals using spatial variance," Ph.D. Dissertation, University of Texas at El Paso, El Paso, 2020.
8. Rumpf, R. C., C. Garcia, H. Tsang, J. Padilla, and M. Irwin, "Electromagnetic isolation of a microstrip by embedding in a spatially variant anisotropic metamaterial," *Progress In Electromagnetics Research*, Vol. 142, 243–260, 2013.
9. Rumpf, R. C., J. Pazos, C. R. Garcia, L. Ochoa, and R. Wicker, "3D printed lattices with spatially variant self-collimation," *Progress In Electromagnetics Research*, Vol. 139, 1–14, 2013.
10. Digaum, J. L., J. J. Pazos, J. Chiles, J. D'Archangel, G. Padilla, A. Tatulian, R. C. Rumpf, S. Fathpour, G. D. Boreman, and A. S. M. Kuebler, "Tight control of light beams in photonic crystals with spatially-variant lattice orientation," *Optics Express*, Vol. 22, No. 21, 25788–25804, 2014.
11. Kuebler, S. M., J. L. Digaum, J. Pazos, J. Chiles, G. Padilla, A. Tatulian, R. C. Rumpf, and S. Fathpour, "Controlling light using three-dimensional spatially variant self-collimating photonic crystals," Optical Society of America, 2014.
12. Digaum, J. L., R. Sharma, J. J. Pazos, R. C. Rumpf, and S. M. Kuebler, "Tight control of light beams in photonic crystals with spatially-variant unit cells," Optical Society of America, 2015.
13. Rumpf, R. C., J. J. Pazos, J. L. Digaum, and S. M. Kuebler, "Spatially-variant periodic structures in electromagnetics," *Phil. Trans. R. Soc. A*, Vol. 373, 2015.
14. Leonhardt, U. and T. Philbin, *Geometry and Light: The Science of Invisibility*, Courier Corporation, 2012.
15. Pendry, J. B., D. Schurig, and D. R. Smith, "Controlling electromagnetic fields," *Science*, Vol. 312, 1780–1782, 2006.
16. Gulib, A. U. H., "Numerical calculation of spatially variant anisotropic metamaterials," Masters Thesis, University of Texas at El Paso, El Paso, 2016.
17. Community, B. O., "Blender-free and open 3D creation software," Blender Foundation, [Online], Available: https://www.blender.org/, [Accessed April 2020].
18. Avila, J. A., C. L. Valle, E. Bustamante, and R. C. Rumpf, "Optimization and characterization of negative uniaxial metamaterials," *Progress In Electromagnetics Research C*, Vol. 74, 111–121, 2017.
19. Rumpf, R. C., "Chapter three — Engineering the dispersion and anisotropy of periodic electromagnetic structures," *Solid State Physics*, 213–300, 2015.
20. Rumpf, R. C., *Electromagnetic and Photonic Simulation for the Beginner: Finite-Difference Frequency-Domain in MATLAB*, Artech House, Boston, MA USA, 2022.
21. Rumpf, R. C., "Simple implementation of arbitrarily shaped total-field/scattered-field regions in finite-difference frequency-domain," *Progress In Electromagnetics Research B*, Vol. 36, 221–248, 2012.
22. Berry, E. A., J. Gutierrez, and R. C. Rumpf, "Design and simulation of arbitrarily-shaped transformation optic devices using a simple finite-difference method," *Progress In Electromagnetics Research B*, Vol. 68, 1–16, 2016.
23. Gulib, A. U. H., "Algorithms for exploration of advanced electromagnetic concepts," Ph.D. Dissertation, University of Texas at El Paso, El Paso, August 2022.
24. Jacobson, A., D. Panozzo, and Others, "Libigl — Simple C++ geometry processing library," [Online], Available: https://libigl.github.io/.
25. Dirichlet, G. L., "Uber die Reduktion der positiven quadratischen Formen mit," *J. Reine Angew. Math.*, Vol. 40, 209–227, 1850.
26. Voronoi, G., "Deuxième mémoire: Recherches sur les paralléloedres primitifs," *J. Reine Angew. Math.*, Vol. 136, 67–181, 1909.
27. Voronoi, G., "Nouvelles applications des paramètres continus à la théorie des formes quadratiques, deuxième Mémoire: Recherches sur les paralléllo'edres primitifs," *J. Reine Angew. Math.*, Vol. 134, 198–287, 1908.

28. Souvaine, D., M. Horn, and J. Weber, 2005, [Online], Available: http://www.cs.tufts.edu/comp/163/notes05/voronoi_handout.pdf, [Accessed 2021].
29. Hu, Y., Q. Zhou, X. Gao, A. Jacobson, D. Zorin, and D. Panozzo, "Tetrahedral meshing in the wild," *ACM Trans. Graph.*, Vol. 37, 60:1–60:14, August 2018.
30. Treece, G., R. Prager, and A. H. Gee, "Regularised marching tetrahedra: Improved iso-surface extraction," *Computers & Graphics*, Vol. 23, No. 4, 583–598, 1999.
31. Moldovan, W. J., J. Rhinelander, and Dean, "Pybind11 — Seamless operability between C++11 and Python," 2017, [Online], Available: https://github.com/pybind/pybind11.