

PROCEEDINGS OF SPIE

SPIDigitalLibrary.org/conference-proceedings-of-spie

Realtime visual-inertial SLAM integrated with radar data to triangulate targets

Joshua Girard, Dryver Huston, Dylan Burns, Tian Xia

Joshua Girard, Dryver Huston, Dylan Burns, Tian Xia, "Realtime visual-inertial SLAM integrated with radar data to triangulate targets," Proc. SPIE 12125, Virtual, Augmented, and Mixed Reality (XR) Technology for Multi-Domain Operations III, 121250G (31 May 2022); doi: 10.1117/12.2618897

SPIE.

Event: SPIE Defense + Commercial Sensing, 2022, Orlando, Florida, United States

Realtime Visual-Inertial SLAM Integrated with Radar Data to Triangulate Targets

Joshua Girard^a, Dryver Huston^b, Dylan Burns^c, and Tian Xia^d

^{a,b,c}University of Vermont Department of Mechanical Engineering, 33 Colchester Ave,
Burlington, VT

^dUniversity of Vermont Department of Electrical and Biomedical Engineering, 33 Colchester
Ave, Burlington, VT

ABSTRACT

This paper presents research concerning the use of visual-inertial Simultaneous Localization And Mapping (SLAM) algorithms to aid in Continuous Wave (CW) radar target mapping. SLAM is an established field in which radar has been used to internally contribute to the localization algorithms. Instead, the application in this case is to use SLAM outputs to localize radar data and construct three-dimensional target maps which can be viewed live in augmented reality. These methods are transferable to other types of radar units and sensors, but this paper presents the research showing how the methods can be applied to calculate depth efficiently with CW radar through triangulation using a Boolean intersection algorithm. Localization of the radar target is achieved through quaternion algebra. Due to the compact nature of the SLAM and CW devices, the radar unit can be operated entirely handheld. Targets are scanned in a free-form manner where there is no need to have a gridded scanning layout. The main advantage to this method is eliminating many hours of usage training and expertise, thereby eliminating ambiguity in the location, size and depth of buried or hidden targets. Additionally, this method grants the user the additional power, penetration and sensitivity of CW radar without the lack of range finding. Applications include pipe and buried structure location, avalanche rescue, structural health monitoring and historical site research.

Keywords: Continuous Wave radar, range finding, target mapping, SLAM, augmented reality

1. INTRODUCTION

Ground Penetrating Radar (GPR) research has been advancing rapidly and a new manifestation of this technology has made its way into the world of handheld sensing. The combination of high powered computing capabilities as well as advancements in augmented/mixed reality systems with handheld GPR devices result in a powerful new type of ‘X-ray vision’ technology. Perhaps soon construction workers and search and rescue teams will be wearing holographic lenses whilst real-time scanning through walls and underground,^{1,2} painting a 3-D map of their hidden surroundings. The goal of this paper is to lay down the framework of an algorithm that can be used as a building block for this type of technology. To begin, this research concerns a type of radar most commonly used in speed measurement applications (using the Doppler effect), which uses a continuous wave (CW) transmission. This type of radar has advantages such as higher penetration and power transmission efficiency as well as lower interference with wireless devices³ than similarly sized and powered pulsed radar systems. The glaring flaw with using a simple single frequency CW radar for ground penetrating applications is that it has no inherent mechanism that allows range finding. Pulsed radars use the time of travel for each pulse to estimate range. There is no such start and stop point for a CW scan to compare against for time of travel. Instead, this paper proposes a method where visual-inertial simultaneous localization and mapping (vSLAM) is used to triangulate targets that the CW radar has detected, similar to occupancy mapping. For every ‘positive’ scan (where the radar detects a target) the system records its pose, the position and rotation at that time. A list of poses corresponding to positive scans are the input to the intersection algorithm. This uses a set of parameters

Further author information: (Send correspondence to J.G.)

J.G.: E-mail: joshua.girard@uvm.edu, Telephone: 1 802 377 1747

Virtual, Augmented, and Mixed Reality (XR) Technology for Multi-Domain Operations III, edited by
Mark S. Dennison, David M. Krum, John (Jack) N. Sanders-Reed, Jarvis (Trey) J. Arthur III,
Proc. of SPIE Vol. 12125, 121250G · © 2022 SPIE · 0277-786X · doi: 10.1117/12.2618897

corresponding to the size of the radar's scanning influence, as well as the sensitivity to scan intersection^{4,5} to produce a 3-D model which is in turn sent to the mixed reality goggles. Using the goggles, the user can view and interact with the 3-D model in mixed reality.

1.1 Materials and Hardware

The majority of this paper describes the algorithm at the heart of this target mapping technique. The following is a brief overview of all materials and hardware as well as software and libraries used to accomplish the data acquisition. Figure 1 shows a 3-D printed casing mounted to the back of a Subsurface brand AML Pro, CW radar. This radar emits a continuous 2.4 GHz wave and measures the reflected phase difference with comparators throughout its 4 receiver antenna array which is arranged in a planar manner, two antennas on each side of the central transmission antenna. The theory with this device is that the outgoing CW signal causes a reflection in plane with the receivers and when properly aligned with a target.³ The result is a CW device that is polarized, meaning it can detect long edges or pipes that are aligned with its antenna plane and it can do so with all of the advantages of CW radar mentioned in Section 1. Inside the casing is a Raspberry Pi 4 that handles and stores the incoming data. Additionally, there is an op-amp circuit which plugs into the AML's 3.5mm audio jack to convert the 'BEEP' sound it emits when it detects a target into a 3v3 digital signal which the GPIO pins from the Raspberry Pi can read. Finally, there is an Intel Realsense T265 tracking camera which uses vSLAM techniques such as the extended Kalman filter to estimate states and produce pose data. The T265 is an example of drift-free local positioning that is crucial to the success of this algorithm.⁶

In order to visualize scanned targets, the Microsoft HoloLens 1 will be used. It can execute spatial mapping using a depth camera, as well as pose estimation similar to the Realsense. In addition to this, it has gesture recognition to interface with 3-D models in real time.



Figure 1. The AML Pro CW radar is shown here with the attached 3-D printed enclosure for the Raspberry Pi 4, Intel Realsense T265 along with all necessary batteries and an HD screen for user interface

1.2 Software and Libraries

All of the code was written in python 3.6 both on the Raspberry Pi for data acquisition as well as on the personal computer for data processing. They are similarly structured code files as the long term goal is to integrate them both and optimize to the point that the Pi is completing all of the calculations without the need for file transfer from one machine to the next. Some python libraries of note are PyVista⁷ and the Point Processing Toolkit (PPTK).⁸ The PPTK library allowed quick and computationally light visualization of point clouds including over 10,000,000 points which was helpful for early iterations of the software before it was optimized. The PyVista library is an incredibly powerful tool for extracting the surface of a set of points in space and then using built-in algorithms to derive a mesh and save it as a solid object file (through Delaunay triangulation).⁷

2. METHODS

The AML, like any other radar, has a finite range and pattern where the signal is emitted. Figure 2 shows the approximate shape of this pattern. The AML detects objects in the frustum of a pyramid, there are four main parameters defining this region. The region is defined in the algorithm by its height (r), the angles that it expands from the antenna are θ and ϕ (shown in Figure 2) as well as the width of the antenna.

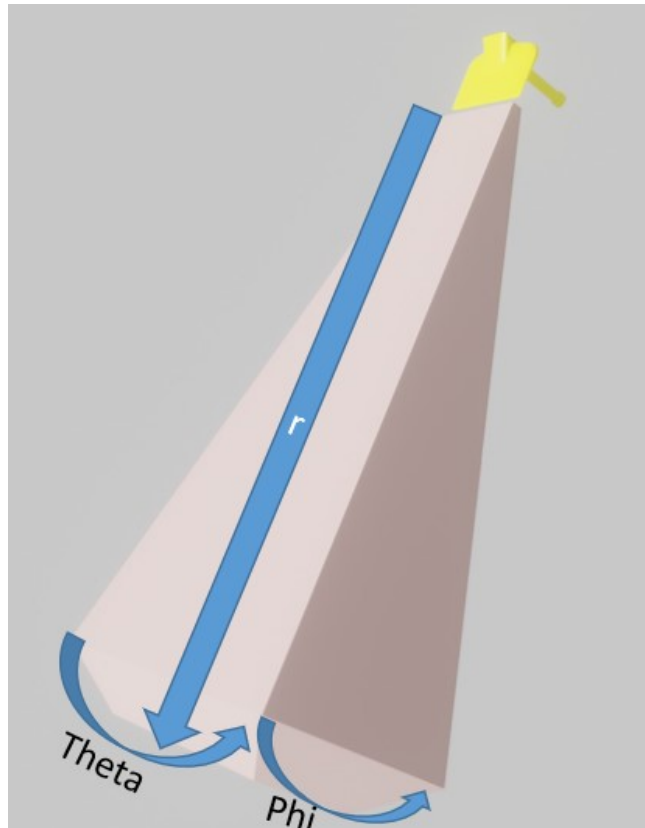


Figure 2. The light pink region represents a frustum of a pyramid which approximates the zone in which the AML is able to measure the phase difference of a reflected signal

The main concept behind the algorithm is that a target is likely to lie in a location observed by multiple scans,⁹ see Figure 3. Additionally, scans from very different locations that intersect also indicate a high likelihood of a target. In a real life scanning scenario, multiple scans from nearby poses may be positive. Using the angle of intersection, the algorithm can differentiate between intersections with high probability of a target and intersections with a low probability¹⁰ as shown in Figure 4. In order to take this concept and computerize it, all of the geometry must be discretized.

First, a finite resolution must be chosen. For rapid troubleshooting tests, 10 cm worked well and for accurate and final renderings, resolutions as high as 1 cm were used. The limiting factor here is the cubic time complexity, doubling the resolution results in an 8 times increase in compute time. To initialize the algorithm, a 3 dimensional array is created with dimensions just large enough to fully encapsulate the shape in Figure 2. A loop indexes through the array and logic statements determine if the current index falls within the modeled frustum. The result is a discretized version of the scanning area contained in this array. A list of position vectors pointing to each index within the frustum is also saved.

Next, for each Realsense pose estimation recorded by the Raspberry Pi, the stored quaternion is applied to each position vector in the aforementioned list. This has proven as an effective and efficient way to process freehand rotations.¹¹ Equation 1 describes how quaternion rotation q is applied to a vector, which is interpreted

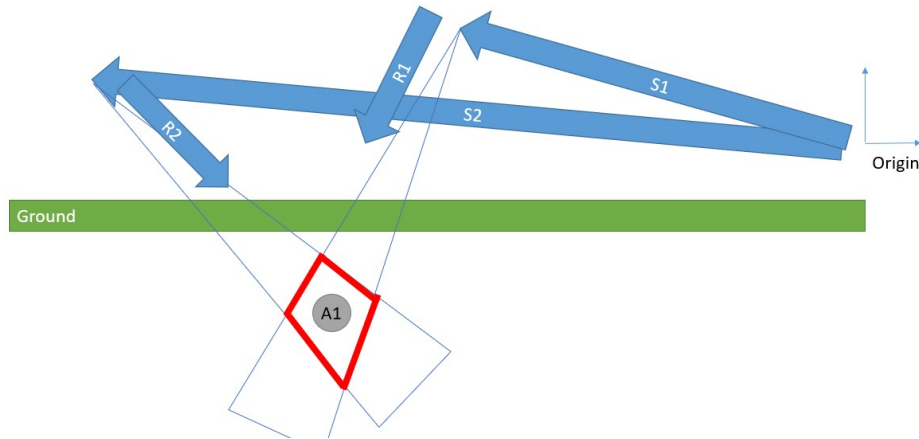


Figure 3. A 2-D visualization of the main concept behind the triangulation algorithm. S_i 's are position vectors and R_i 's are vectors representing the direction of the AML after being rotated to the orientation of the scan

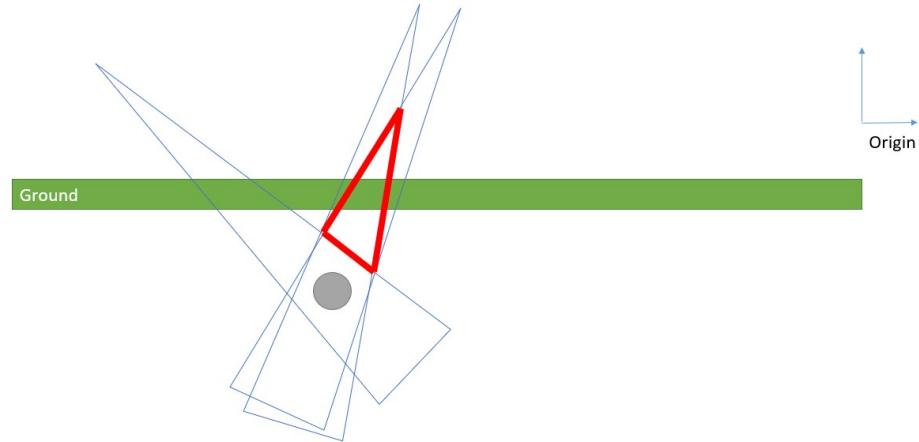


Figure 4. A depiction of the presence of an inaccurate intersection zone which is caused by a high density of scans originating from similar poses

as quaternion p with real part equal to 0. The translation can then be simply added to the resultant p' . This final vector is then used to point to a discrete spatial block dS in a main array which represents the entire area being scanned.

$$p' = qpq^{-1} \quad (1)$$

Definition: dS_{ijk} is a discrete spatial cube in absolute space with side lengths equal to the spatial resolution. The location in the grand array is indicated by subscripts ijk where i, j , and k can be multiplied by the resolution to yield dS 's location in actual space.

Each dS_{ijk} represents the set of N vectors which are normal to each scan that intersected with this cube of space. This is shown in Figure 5. For the sake of the algorithm, dS_{ijk} has two weighting attributes, N and D where N is the number of scans and D is the minimum dot product between any two scan vectors. Scan vectors are determined by transforming the unit vector in the $-\hat{k}$ direction (same direction as the AML antenna points) by the quaternion representing that scan's orientation. Each vector has magnitude 1 meaning that all the dot products are simply equal to the cosine of the angle. Moreover, as this is mainly a ground penetrating application, there is no need to worry about angles beyond π radians as this does not make physical sense.

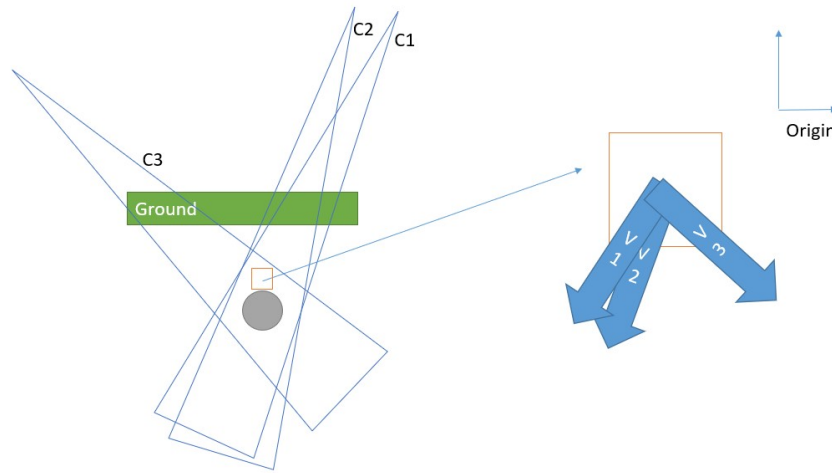


Figure 5. Investigating a discrete spatial chunk, dS . V_i 's represent the direction of each scan C_i which encapsulated this particular chunk

D is defined below in Equation 2 where v_m and v_n are vectors contained in the set at dS_{ijk} , for all combinations m and n but $m \neq n$ up to N .

$$D = \min(v_m \cdot v_n) \quad (2)$$

A spatial cube whose D value is very low indicates that some two scans intersected there at a very high angle. A spatial cube whose N is very high indicates many scans intersected there. The key to the algorithm is to realize that neither of these two conditions are sufficient to say that there is a high probability of the target being present at this spatial cube. Rather, that relatively high N and low D at the same time lead to a high probability of a target. Figure 6 helps visualize this concept at an extremely low resolution.

Algorithm: Highlight dS_{ijk} 's with N values above a certain threshold **and** D values below a certain threshold.

Note: neither of these values must necessarily be static, which leaves room for interesting future work.

2.1 Experimental Setups

Two main experimental setups were tested in order to verify different aspects of this method of computing scan intersections. First, a 2.5 cm diameter metal rod was placed on the ground behind a horizontal wooden barrier and scanned in order to verify that the algorithm was successfully differentiating between high and low angle intersections. This was designed in order to minimize geometric complexity and use the most simple possible radar target - solid metal (aluminum in this case). The second test was more complex, and it was designed to test the geometric capabilities of the system as well as to showcase the advantages of using CW radar and the AML device. This test was a section of 10 cm PVC pipe about 1 m long with a T joint at the end of it, buried under moderately fine sand (at a depth of anywhere from 5-15 cm).

3. RESULTS

The result of the first, more basic test is shown in Figure 7. Additionally, the cartoon visual is placed next to it to show the similarity. This view is from the side of the scanning domain, it is looking along the metal rod axially, the rod goes in/out of the paper. This test was a success, low angle of intersection zones were successfully excluded and the red dots (indicating areas of high target probability) were at the proper depth. Each dot in the picture represents 2 cm and the grid lines are 20 cm. The red region (approximately 25 cm wide) is clearly bigger than the 2.5 cm pipe but it is likely due to using too high θ and ϕ in the intersection algorithm (essentially,

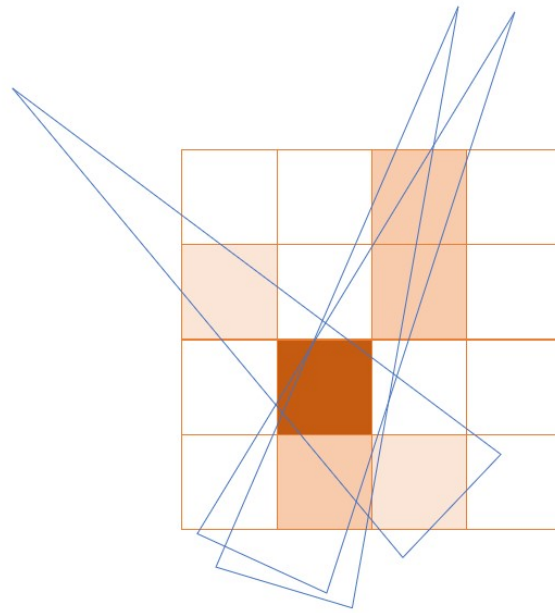


Figure 6. Drawing more discrete chunks can help visualize the algorithm, in this case the color of each square represents the probability of a target lying within it according to the algorithm, based on the three blue scans shown

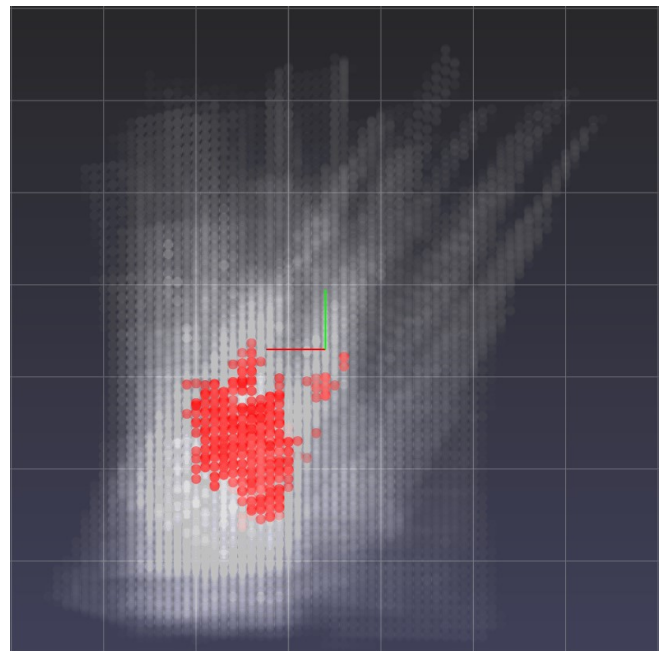
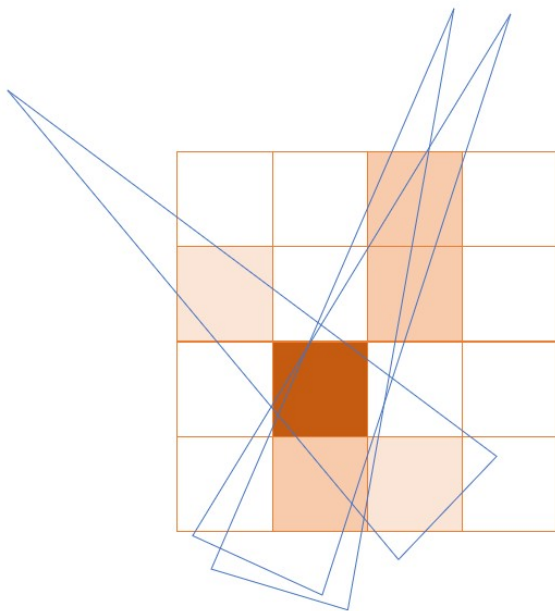


Figure 7. An encouraging result from a preliminary 3-D test, this is a side view of a batch of scans which closely resembles the cartoon example (placed side-by-side for comparison)

modeling the scans thicker than they really are). Regardless, the point was not yet to generate a perfect 2.5 cm cylinder but to instead verify that low angle zones were being properly excluded.

The second and more realistic¹² test setup results are shown in Figure 8. The scan data picture is rotated to match the orientation of the photo of the target sitting above ground, and the resolution is the same as Figure 7. Hopefully the T shape makes itself clear to the reader but following figures will show more 3-D interpretations of it. This data was collected in a few minutes by carefully moving the apparatus around above the sand, making

second passes over areas where the radar indicated a target.¹³ Then, in order to start finding intersections, second passes were made around the scanning zone at different angles to try and triangulate the depth and 3-D form of the target.

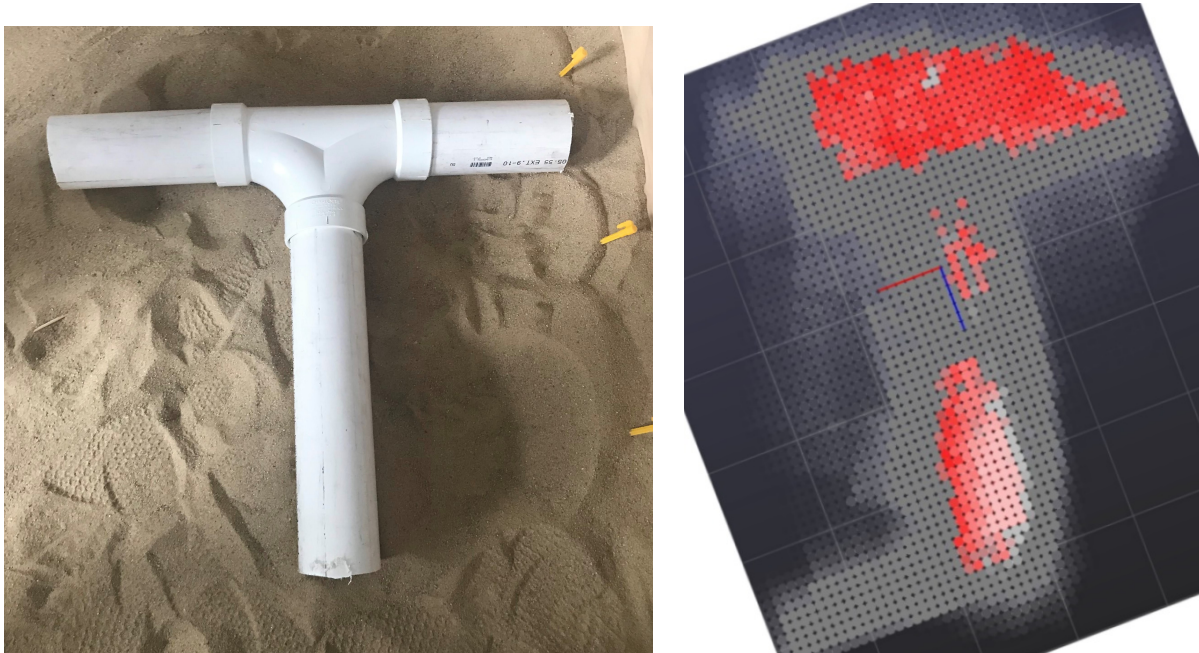


Figure 8. Side-by-side comparison of top view of a 10 cm diameter PVC pipe T joint and the 3-D model generated using the intersection algorithm. The pipe was buried in sand about 5 cm under the surface at the shallowest point before the scanning occurred

From Figure 8, the red dots form a volume in 3 dimensional space. Using the PyVista library, it is easy to extract the surface of this volume and then save it as a common solid file format, in this case STL which is later converted to FBX for Microsoft HoloLens compatibility. The 3-D surface resulting from the data in Figure 8 is shown in Figure 9. Clearly, this is an orthographic view of that T shaped PVC pipe. The scale of the 3-D model indicates that the pipe is about 20 cm in diameter when in reality it is actually 10 cm, but the important result is that the shape and general size are represented in the point cloud and the resulting surface. It is an important distinction to make that the algorithm is accurate but not very precise. It is surrounding the target by a 3-D region about 200% oversized, but indeed containing the true target. This could be another case of not properly matching the real world scan size with the model parameters ϕ and θ , or perhaps it has to do with the angular precision of the Realsense T265's IMU. After all, a 1° error at the camera's location would compound to almost 2 cm of error at 1 m distance from the radar. Then, 2 cm off in either direction in addition to the resolution of 2 cm could lead to 4 cm of error on each side of the target resulting in a 10 cm target appearing to be 18 cm in diameter, not far off from what is occurring here.

3.1 Augmented Reality Integration

The output FBX file can simply be uploaded to the HoloLens and viewed in mixed reality. Currently, the 3-D model must be manually aligned with the environment using some anchor points which correspond to the bounding box of the original scanning area.¹⁴ These could come in the form of cones, stakes, room corners or walls etc. Figure 10 (left) shows the view through the HoloLens with the 3-D model of the buried T-pipe displayed. It is important to note; the HoloLens uses a depth camera to compute a 3-D spatial map of the surroundings and by default, clips any 3-D holograms in view where they touch the spatial map, so for visibility, the model is actually placed a few inches above its true location so it can be seen clearly above ground. As the AR segment of this project is still under development, full underground transparency will be added soon.

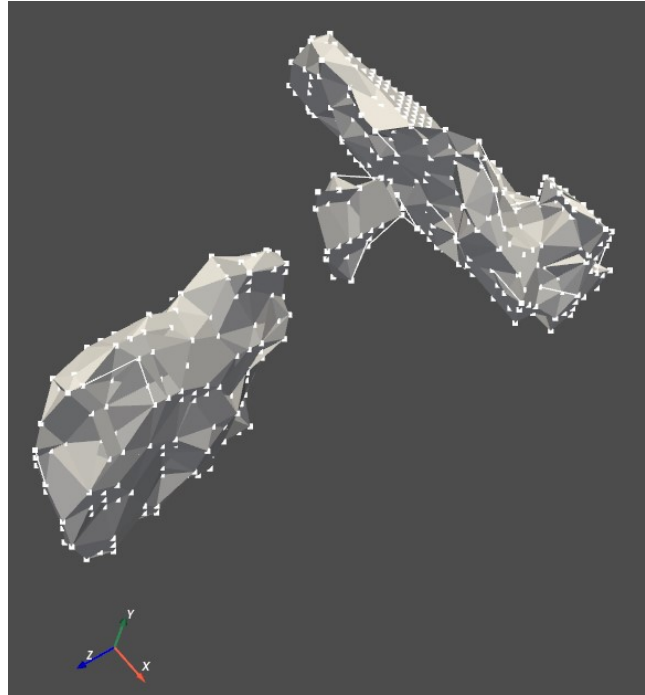


Figure 9. An orthographic view of the red dots from Figure 8 after being processed through the PyVista library and converted to a mesh

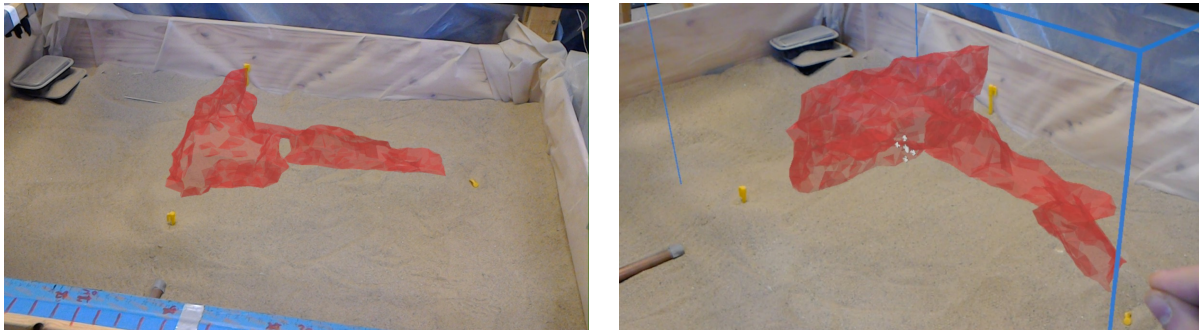


Figure 10. The view through the HoloLens 1, looking at the buried T-pipe, the yellow stakes indicate where the pipe is buried, the model is then moved with user gestures

For better visibility, and to demonstrate the gesture based interaction, Figure 10 (right) shows the user ‘pinching’ the model to raise it out of the ground to have a better look.

4. DISCUSSION

This method of computing intersections weighted by angular components was successful in locating targets. There are however some flaws, the thresholds of N and D values are currently static, meaning in a scanning session if a certain feature is not scanned equally as many times as the surrounding ones, it might not successfully pass above the chosen threshold. This could lead to ‘dead zones’ such as in Figure 8 along the vertical pipe where there are actually no red dots. This is because the user did not attempt to scan this area as many times as they attempted to scan the other areas (which is inherently hard to keep track of). The issue here is that this region does contain a target, and the radar was indicating that, but the algorithm overlooked this fact since the thresholds for counting a zone as ‘containing a target’ are static throughout the whole computation. Perhaps in the future they will be adjusted on a per-scan basis as the whole algorithm is implemented in real-time. Additionally, there are quality of life aspects of this entire system that need to be improved. The speed of

the intersection algorithm could be improved and uploaded to the Pi, along with a wireless network to upload FBX files to the HoloLens along with automatic anchoring would make the whole system entirely real-time and automatic. Also, it is worth noting that in more precise radar target algorithms, Snells law is used to correct for refraction that happens to the radar transmission as it passes through regions where the speed of light is different. In this case, particularly between the ground and air there is likely a significant deflection to the direction of the radar scan. Correcting for this could lead to higher precision of target estimation for a fixed spatial resolution. Nonetheless, the uncorrected results shown here indicate that this method of interpreting vSLAM enhanced radar scan data is actually promising and can help users visualize more clearly the hidden, buried or obscured structures that they are scanning. This method could be implemented for constructing radar based 3-D models through fog, barriers, underground, in the dark, or other visual hindrances.

ACKNOWLEDGMENTS

This material is based upon work supported by the Broad Agency Announcement Program and Cold Regions Research and Engineering Laboratory (ERDCCRREL) under Contract No. W913E521C0003, Department of Navy award N00014-21-L-2326 issued by the Office of Naval Research, and the U.S. National Science Foundation grant 1647095.

REFERENCES

- [1] Xu, X., Xia, T., Venkatachalam, A., and Huston, D., "Development of high-speed ultrawideband ground-penetrating radar for rebar detection," *Journal of Engineering Mechanics* **139**(3), 272–285 (2013).
- [2] Jiao, L., Ye, Q., Cao, X., Huston, D., and Xia, T., "Identifying concrete structure defects in gpr image," *Measurement* **160**, 107839 (2020).
- [3] Cloutier, P. and Oehme, D., "Apparatus and method for detecting and locating hidden objects," *United States Patent No. 7,898,456 B2* (2011).
- [4] Pereira, M., Zhang, Y., Orfeo, D., Burns, D., Huston, D., and Xia, T., "3d tomographic image reconstruction for multistatic ground penetrating radar," *2019 IEEE Radar Conference (RadarConf)*, 1–6, IEEE (2019).
- [5] Pereira, M., Burns, D., Orfeo, D., Zhang, Y., Jiao, L., Huston, D., and Xia, T., "3-d multistatic ground penetrating radar imaging for augmented reality visualization," *IEEE Transactions on Geoscience and Remote Sensing* **58**(8), 5666–5675 (2020).
- [6] Schall, G. and Wagner, D., "Global pose estimation using multi-sensor fusion for outdoor augmented reality," *8th IEEE International Symposium on Mixed and Augmented Reality* (2009).
- [7] Sullivan, C. B. and Kaszynski, A. A., "Pyvista: 3d plotting and mesh analysis through a streamlined interface for the visualization toolkit (vtk)," *The Journal of Open Sourced Software* (2019).
- [8] Lu, V., "Point processing tool kit," *MIT* (2018).
- [9] Lindstrom, C., "Sensitivity of back-projection algorithm (bpa) synthetic aperture radar (sar) image formation to initial position, velocity, and attitude navigation errors," *IET Radar Sonar and Navigation* (2021).
- [10] Kocur, D., "Experimental analyses of mutual shadowing effect for multiple target tracking by uwb radar," *IEEE 7th International Symposium on Intelligent Signal Processing* (2011).
- [11] Girard, J., Burns, D., Huston, D., and Xia, T., "Penetrating radar combined with 3-d imaging for real-time augmented reality sensing and classification," *Virtual, Augmented, and Mixed Reality (XR) Technology for Multi-Domain Operations II* **11759**, 117590N, International Society for Optics and Photonics (2021).
- [12] Orfeo, D., Zhang, Y., Burns, D., Miller, J. S., Huston, D. R., and Xia, T., "Bistatic antenna configurations for air-launched ground penetrating radar," *Journal of Applied Remote Sensing* **13**(2), 027501 (2019).
- [13] Kanno, A., "Imu-enabled nondestructive imaging system based on millimeter-wave radar," *IEEE Conference on Antenna Measurements Applications (CAMA)* (2019).
- [14] Mitsuno, D., "Effective application of mixed reality device hololens: Simple manual alignment of surgical field and holograms," *National Library of Medicine* (2019).