Imaging Systems Utilized in the DebriSat Fragment Size Characterization Process

Ralen Toledo[(1)], Dr. Bungo Shiotani[(1)], Dr. Norman Fitz-Coy[(1)]

[(1)] University of Florida, 939 Sweetwater Dr, Gainesville, FL USA 32603, ralent@ufl.edu

ABSTRACT

The DebriSat project was conceived to provide NASA and the DoD with an updated dataset to improve existing satellite break-up models. The DebriSat test article is a 50 kg-class satellite designed with components, materials, and processes commonly utilized in modern LEO satellites. This test article was then subjected to a laboratory hypervelocity impact (HVI) test to emulate a catastrophic on-orbit collision. In the post-HVI phase, fragments from the HVI test are being carefully characterized, where mass and size measurements are taken. Mass balances are used to measure the fragments mass and two imaging systems are utilized to measure the fragments sizes. The two imaging systems are referred to as the 3D and 2D imaging systems, depending on whether or not the fragments minimum dimension is greater than 3 mm; both imaging systems utilize point-and-shoot cameras for image acquisition. The set of parameters calculated by the imaging systems include characteristic length, average cross-sectional area, and volume.

The 3D imaging system uses six point-and-shoot cameras to obtain 126 images of the object on a turntable which are then processed using a space-carving algorithm to generate a 3D point cloud representation of the object. From the point cloud, the three largest orthogonal dimensions, volume, and average cross-sectional area are calculated. The 2D imaging system uses a single point-and-shoot camera and to generate a 2D silhouette of the fragment. The three largest orthogonal dimensions are calculated from the silhouette. Additionally, a right-angled prism mirror has been added to provide a side view of the fragment and, by extension, the fragment's height. Once the size characteristics are determined, measurement data, images, and associated metadata are archived in the database.

1 INTRODUCTION

The 2014 terrestrial breakup of the 50-kg class satellite, DebriSat, has provided a plethora of diverse fragments, a unique dataset, to rebuild NASA's standard breakup model. These fragments are current being characterized by a team of students at the University of Florida. Properties such as mass, material, color (indicating original location on the satellite), size, and shape of the fragments are recorded into a database that will allow an update to our understanding of satellite breakups. The characterization of size, in particular, has been a steep technical challenge: aiming to provide accurate models and dimensions for several hundred thousand unique fragments of varying shape and specularity.

The University of Florida is tackling this challenge by developing innovative and semi-automated imaging and characterization systems. To maximize efficiency of characterization, a requirement due to the sheer number of fragments, imaging and characterization of DebriSat's fragments has been split into separate 2D and 3D systems. Fragments that are relatively flat (defined as one orthogonal dimension being much smaller than the other two dimensions) can be characterized quickly through object detection of only two images. These fragments make up a vast majority of the DebriSat's remains. 3D fragments are processed separately and provided much more attention and time to fully characterize their complex shapes. The threshold between 2D and 3D is defined using the smallest dimension of a fragment; any fragment whose minimum dimension is below 3mm is 2D. This paper will explore the evolving process of characterizing fragment sizes per imaging system.

2 3D IMAGER

The DebriSat 3D imager is a six-camera system where each camera is evenly distributed along an arc over a turntable. The cameras' positions along the arc extend from around 8° above the horizon of the turntable to 90°, directly above the table. To collect a full image set, a fragment is first placed on the turntable. Then, each camera will capture a single image for each of 20 unique, evenly spaced azimuth positions, completing a single full rotation of the turntable. Afterward, when the turntable has returned to its original position, an additional image is taken by each camera to compare the position of the fragment before and after the turntable's motion to ensure that the

fragment has not moved during the process. Thus, a total of 126 images are taken for each fragment, with only one repeated azimuth position. The image set of the repeated azimuth position is not used after verification of the fragment's position.

After image capture, an object detection algorithm is employed to generate 120 binary images identifying the fragment from the background. Then, a space-carving algorithm converts those binary images, with knowledge of the camera's location and orientation on hand, into a 3D point cloud (a meshed point cloud is shown in Fig. 1). Directly from the 3D point cloud, the three max orthogonal dimensions (X, Y, Z) and characteristic length (Lc) are calculated. Afterward, the point cloud is triangulated and then volume (V) and average cross-sectional area ($ACSA$) are calculated. After calculation, all data is stored in a database. Characteristic length is defined as:

$$Lc = \frac{1}{3}(X + Y + Z) \tag{1}$$

To calculate volume, a tight boundary alpha shapes algorithm [1] is used to generate facets around the point cloud. Average cross-sectional area (ACSA) is calculated using an evenly distributed point cloud of n point-of-view vectors around the space-carved 3D model. The cross-section from each point-of-view is calculated and then the average is taken as the result:

$$ACSA = \frac{\sum_i^n \sum_j^m Area_j^* * \cos(\theta_{ij}^*)}{n} \tag{2}$$

where: θ_{ij}^* is the angle between the "i" view vector and the "j" surface normal, m is the total number of surfaces whose angle θ_{ij}^* is less than $180°$, and n is the total number of evenly distributed points-of-view. Experimentally, n was chosen to equal 1000 points-of-view.

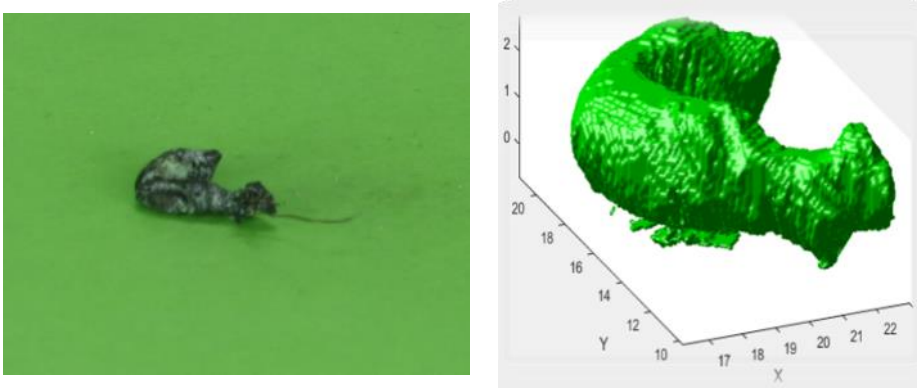

Fig. 1. An example of a space-carved object (object on the left and generated model on the right).

The DebriSat 3D imager, shown in Fig. 2, initially went into operation imaging and characterizing fragments in January of 2018. After a year of operation, however, it became clear that an upgrade was due. The ensuing upgrades impacted the hardware, software, and consistency of accuracy. The following sections elaborate on the development and design of the many aspects of the 3D imager.

Fig. 2. 3D imager in January of 2018.

2.1 3D imager hardware

The primary hardware components essential to the 3D imager are the: cameras, camera mounts, camera control system, turntable, lights, and backdrop. In putting these many components together several design considerations are essential to the system's success: long-term stability of the cameras (contributing to accurate knowledge of the cameras' position and orientation), an effective lighting and backdrop combination to reduce shadows and enable clean object detection, and automation of system.

2.1.1 Cameras

The cameras used by the 3D imager are Canon PowerShot S110. These cameras were selected because of their low cost, adjustable focus, and adjustable zoom capabilities. Additionally, an open-source software package exists for Canon cameras called CHDK (Canon Hack Development Kit) [2] that allows for near-complete control of the cameras, perfect for remote and automated photography. These cameras, powered externally, have endured over a year over frequent imaging and are still in good operation.

2.1.2 Camera mounts

After selecting the cameras, camera mounts had to be found. Accurate space-carving requires excellent knowledge of the cameras. That requirement places a burden on the camera mounts to maintain their precise position during prolonged sessions of image capture, during which the cameras command mechanical motion (e.g. extending and retracting lens, focusing, image capture, etc.). Additionally, the intended purpose of these off-the-shelf cameras was for point-and-shoot photography, not for fully constrained, stationary photography; custom mounts were needed achieve full constraint. The design of such robust camera mounts occurred in multiple phases. The first phase of design produced 3D printed plastic (polylactic acid, PLA) camera mounts, quick to manufacture and useful for testing the functionality of the system. Unfortunately, after many months of use cracks appeared in the mounts and replacement was demanded, as seen in Fig. 3.

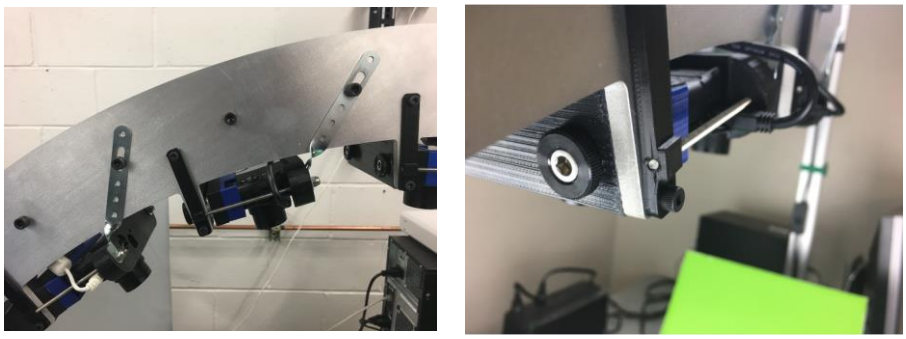

Fig. 3. 3D printed camera mounts (left) and developed cracks (right).

The initial solution to the cracking plastic was to migrate from 3D printed plastic to aluminum. The process of reassembling the new mounts, however, was halted by significant difficultly in aligning the cameras. This system uses a camera calibration technique (a single camera model developed by CalTech [3]) that requires consistent alignment of the cameras. And, the plastic of the original mounts was unknowingly warped to achieve alignment during the system's initial deployment. This unfortunate truth did not come to light, however, until the attempted alignment of the aluminum mounts in which the cameras would not maintain their position. A redesign was necessary and focused on robust constraint of the cameras and enabling alignment of the cameras. The product was aluminum brackets that allow three degrees of rotation for alignment: pitch, roll, and yaw, shown in Fig. 4.

Fig. 4. Final aluminum camera mounts.

2.1.3 Camera control system

The hardware side of the camera control system includes the mechanical control of the cameras and the devices that make up the communication channel of the cameras. CHDK allows near full control of Canon cameras however, while capable of many things including the ability to turn off the cameras, the software package cannot power the cameras on. Thus, to achieve the goal of automation, solenoids were added to the cameras during the upgrade of the camera mounts. These solenoids, controlled by the computer via an Arduino Uno and activated using power relays, complete the control loop and allow the cameras to be automatically turned on.

All six cameras are connected to and controlled by a Raspberry Pi 3B which is in turn connected to the local area network. The Raspberry Pi communicates with the primary workstation and image processor via the local area network, the whole process of which is shown in Fig. 5. The primary purpose of this setup is to enable two-way communication with the cameras and to enable the parallel processing of image capture (through the Raspberry Pi) and control of the graphical user interface (GUI) on the primary workstation.

Fig. 5. Path of camera control.

2.1.4 Turntable

The turntable is an RT-3 rotary stepper motor by Newmark Systems. The turntable provides the only movement of the fragment during the image capture process. Two separate turntable tops have been manufactured to sit atop this turntable: a calibration plate and space-carving plate, shown in Fig. 6. These plates attach to the turntable magnetically with a custom adapter piece, allowing the replacement of plates to be quick and simple. Fragments are placed upon the space-carving plate for imaging. This plate has on it a thin sheet of green paper, part of the backdrop, for identifying and isolating the fragment upon the turntable.

Calibration requires a known checkerboard grid to be seen by all cameras. The position of this grid defines the shared origin and horizontal plane, atop which fragments will sit for imaging. Originally, the grid was printed on paper and manually taped on top of the space-carving plate. However, the placement of the grid upon the table was inherently unrepeatable with this approach. As a result, a separate calibration plate was designed, and the checkerboard pattern was laser etched upon its surface to guarantee consistent location for each calibration. After calibration, that plate is replaced with the space-carving plate for fragment imaging.

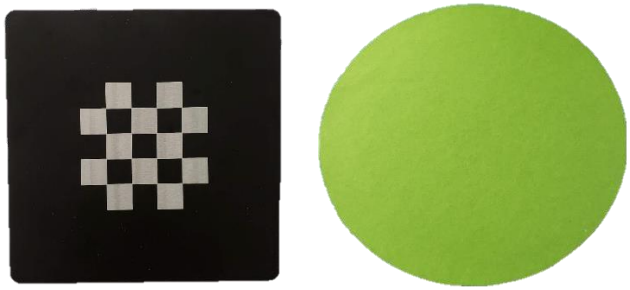

Fig. 6. Calibration turntable plate (left) and space-carving turntable plate (right).

2.1.5 Backdrop

The backdrop is defined as all surfaces that appear in the image alongside the targeted fragment. A necessity of the backdrop is to be easily identifiable and thus easy to isolate from the fragment during image processing and object detection. That goal is achieved by applying a consistent green layer to all surfaces within the field of view of the cameras. Green was selected because it is a color not expected to appear on any of the fragments recovered from DebriSat. A couple materials were tested to cover the carry the green color: Vinyl and construction paper. Vinyl has a smooth, consistent surface but is ultimately too reflective. Construction paper has a matte appearance in the images and its inherent surface roughness has not been a source of any difficulty in the object detection process. An image of the backdrop with relation to the turntable and plate is shown in Fig. 7.

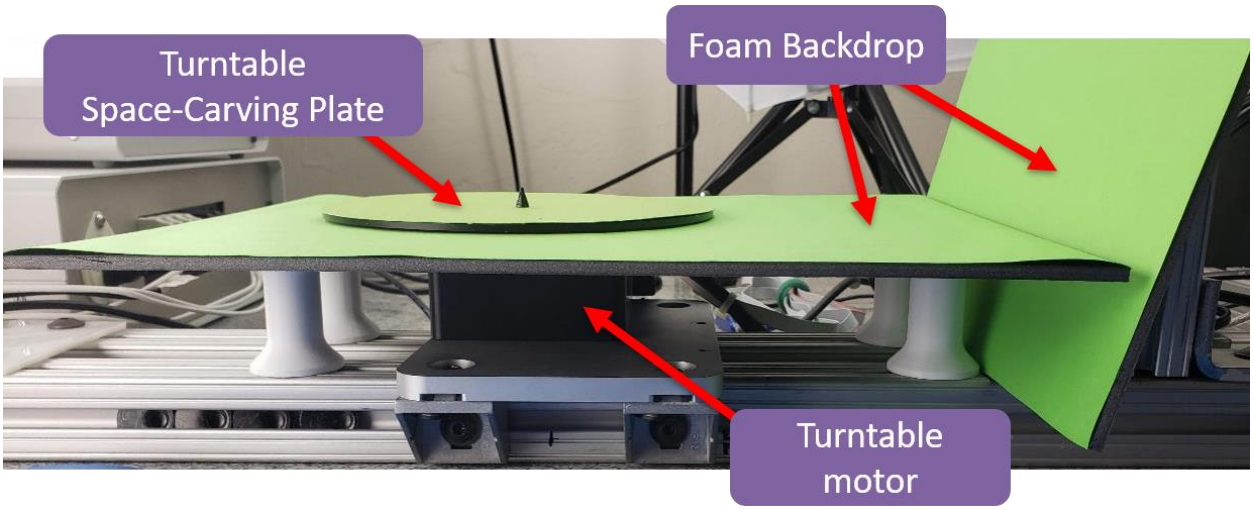

Fig. 7. Turntable with backdrop.

2.2 3D imager software

Since the initial development of the 3D imager, most of its software has undergone significant updates, namely: the GUI, calibration, object detection, and space-carving. The overall goal of these updates is to make the system easier to use, provide more accurate results, and automate the imaging process. These goals have ultimately been achieved through the collective update of the imager's software.

2.2.1 GUI

To interface with and command the 3D imager, a GUI was developed through MATLAB's GUIDE and App Designer tools. The primary objectives of the GUI are full control of the system, ease of use, and to keep a low learning curve. The initial GUI was unable to achieve all these goals. For one, it did not include the calibration process, a separate GUI was called for that. Additionally, control of the cameras was limited. The system did not have two-way communication with the cameras, commands were simply sent but nothing was received so camera status was always unknown until images were either present or missing. This gap in control forced a manual reboot of the cameras over any communication or status issue with the cameras. The addition of solenoids automated the reboot process but that was still a blind approach. Thus, one of the major focuses during the update was to enable two-way communication with the camera and since the update, the need to reboot the cameras has almost disappeared, replaced with strategic, automated troubleshooting.

In addition to improved camera control, the calibration process was also incorporated into the GUI, alongside image capture and space-carving. Operator guidance was another new feature of the GUI. Prompts with annotated images appear as the operator works to keep the learning curve down and reduce mistakes, shown in Fig. 8.

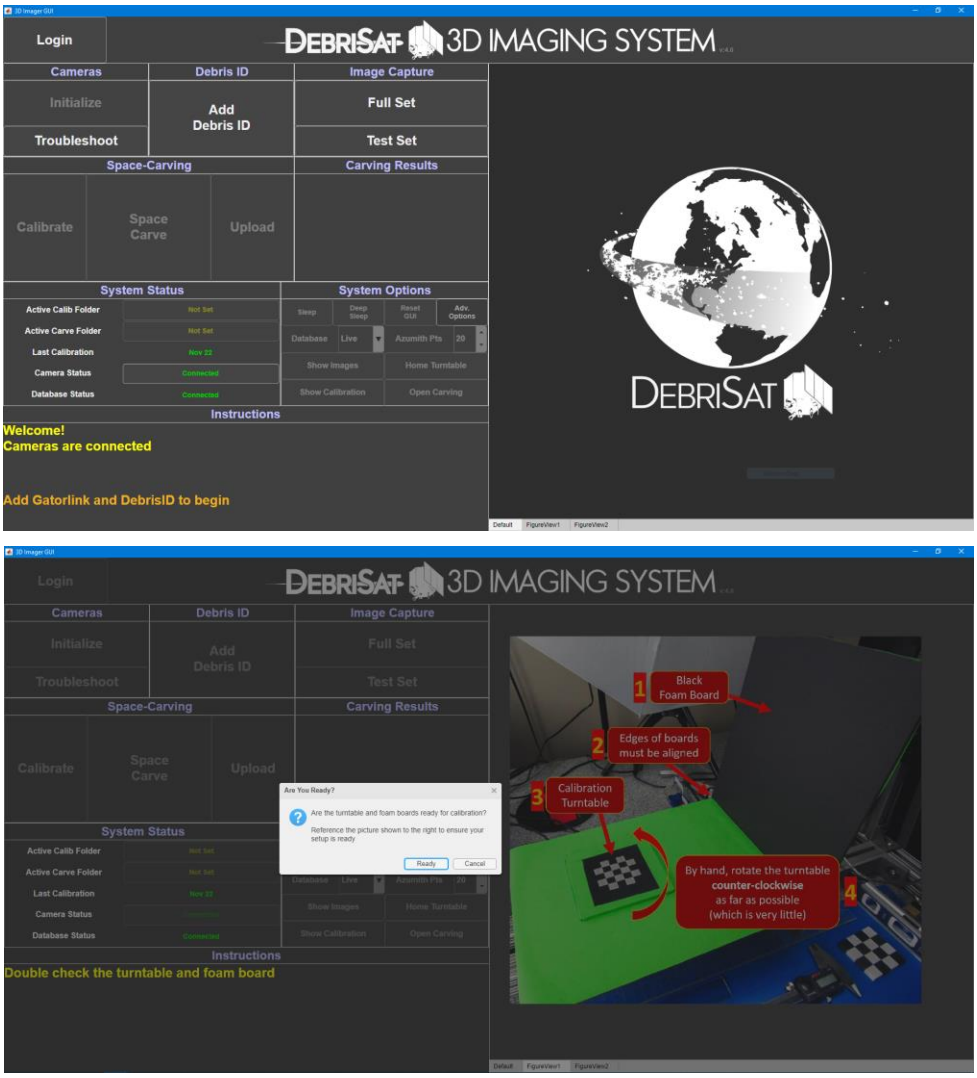

Fig. 8. 3D Imager GUI homepage (top) and guided prompts (bottom).

2.2.2 Calibration

Calibration is an essential part of space-carving as it provides comprehensive knowledge on the camera(s) for each image captured and thus allows for an accurate reconstruction of all cameras in the 3D space around the fragment. A manual calibration process, along with a dedicated GUI, was developed by CalTech [3] and used in the initial deployment of the 3D imager. The algorithm, using a single camera model, required a checkboard grid to be imaged by each camera and at each of the 20 turntable azimuth positions used for space-carving. After image capture, an operator would go through each image and manually click of the corners of the grid. The output of calibration are unique camera intrinsic and extrinsic parameters for each camera and each azimuth position. However, this process is extremely tedious and difficult to reproduce. It would take operators a couple hours to go through an entire image set and the results would vary greatly between operators working on the same image set. The inconsistent results lead directly to inconsistent accuracies in the resulting space-carved model. To resolve this issue, the corner detection process was automated with an innovative approach. The algorithm works as follows:

1. The calibration grid is identified from an image using the visually unique characteristics of the grid relative to the backdrop; the grid is white and contrasts strongly against a black background as seen using the intensity channel in the HSV colorspace
2. The result of the grid's identification is a binary image, set to true only where the grid exists; from this, a quadrilateral is best fit around the grid (Fig. 9 - left)
3. Using the known number of squares in the grid, the corners are predicted as evenly spaced points within the quadrilateral (Fig. 9 - left)
4. The image is then stretched to normalize the dimensions of the grid, transforming the grid to appear as a square regardless of the angle of the camera (a small change for cameras directly above the grid, but significant for cameras whose field of view is near perpendicular with the plane of the calibration grid); the predicted points are transformed accordingly
5. The previous step provides a clearer image of the grid corners, particularly for cameras whose field of view is near perpendicular with the plane of the calibration grid; now, an AXDA corner detection algorithm [4] is used along with the transformed corner predictions to detect the corners in the grid (Fig. 9 - center)
6. The detected corner points are then transformed back to the original dimensions of the image (Fig. 9 - center)
7. This process is repeated for each image and each camera
8. Using knowledge of the turntable's rotation, corner tracking is performed to maintain corner labels/identity through the incremental azimuth positions (Fig. 9 - right)

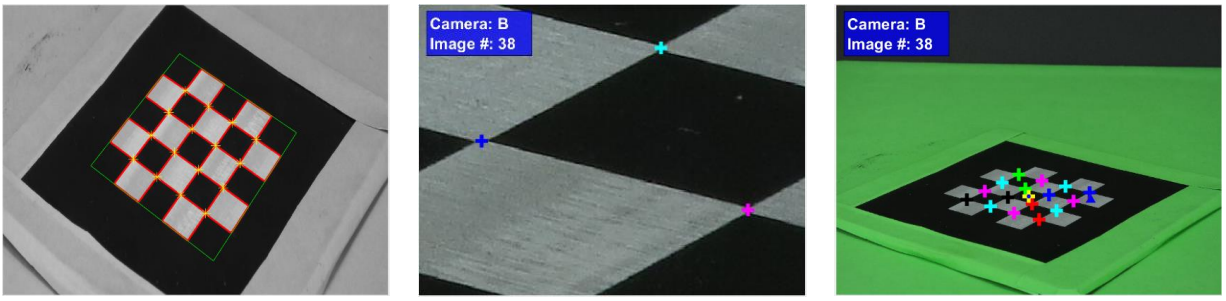

Fig. 9. Quadrilateral fit and corner prediction (left), corner detection (center), corner tracking (right).

After calibration the camera extrinsic parameters are visualized are verified by operators before being saved and used for space-carving.

2.2.3 Object detection

Object detection is a critical step before space carving in which binary images are produced, identifying the fragment from the background. The initial approach to object detection was a simple green mask that worked as follows:

1. Calculate the greenness of the image using the following formula, where G is the green channel, R is the red channel, and B is the blue channel in an RGB colorspace:

$$Greenness = G * (G - R) * (G - B) \quad (3)$$

2. Binarize the image by thresholding based on individual pixel value compared to the mean greenness multiplied by a constant modifier
3. Dilate the object(s) remaining in the binary image

This algorithm worked for many fragments but failed on fragments that with a specular surface, a category into which many of DebriSat's fragments fall. Thus, an update was required for this algorithm too. A great deal of experimental work gave way to the following different approach:

1. Determine magnitude of directional gradients using a Sobel filter (edge detection)
2. Calculate convex hull of the object using the edge detection
3. Identify the background as the area outside the object's convex hull and then remove it
4. Identify and remove potential holes in object using knowledge of object background
5. Strengthen object edges using edge detection
   - Helps with weak color edges (i.e. faces of the object that look similar to the background due to material reflectivity and color cast)
6. Binarize using shape of image histogram

This approach provided acceptable results for many fragments that had minor specularity and color cast but does not work for every fragment. The output and resulting model using this object detection technique is shown in Fig. 10 and Fig. 11. Research is still ongoing to find a more universal approach that can identify objects despite the color or specularity.

Fig. 10. Cropped original image of a pyramid (left), result of object detection (center), space-carved model (right).

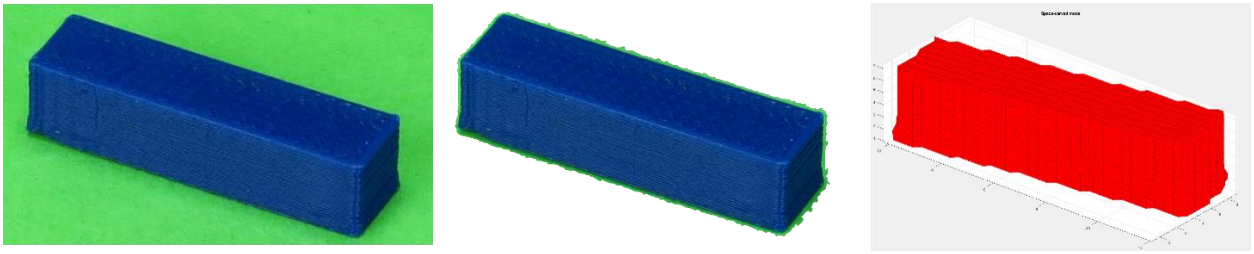

Fig. 11. Cropped original image of a prism (left), result of object detection (center), space-carved model (right).

3 2D IMAGER

The 2D imaging systems operates similarly to the 3D system. The same Canon camera and control system is used in both systems although the camera mounting brackets and structure are unique between the systems different. For the 2D imager, a single camera captures two images per object: a front-lit image and a back-lit image. Object detection generates silhouettes and 2D point clouds which are then characterized. Within each image, the fragment is captured alongside a calibration ring. The calibration ring, with known dimensions is used to scale the size of the fragment. Since the 2D imager's use began in 2016, few major updates have been implemented. A notable update is the

addition of a right-angled prism placed beside the fragment. Captured in the same images as before, the prism allows visibility and, thus, characterization of the 2D fragment's height, again scaled by the calibration ring. Additionally, a new 2D imager is currently under development to capture and characterize larger fragments.

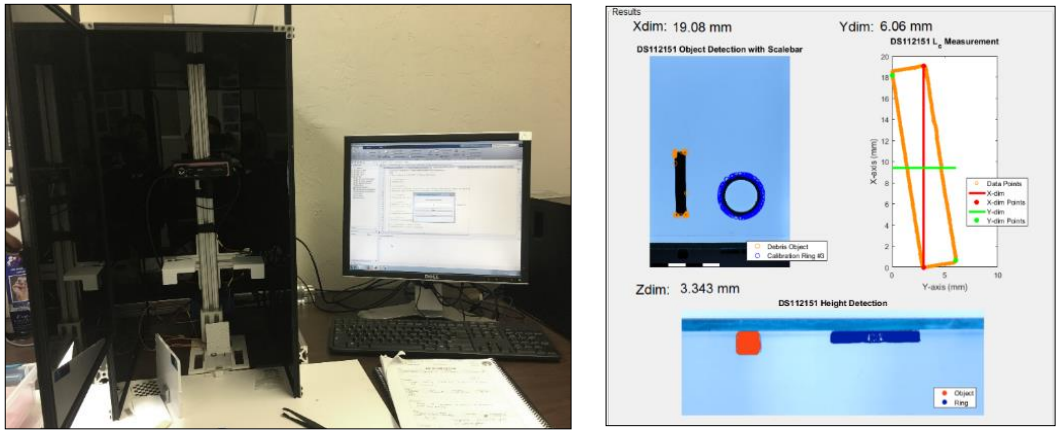

Fig. 12. 2D imager hardware (left) and GUI screenshot after fragment characterization (right).

4 REFERENCES

1. Mathworks.com. (2019). Boundary of a set of points in 2-D or 3-D - MATLAB boundary. [online] Available at: https://www.mathworks.com/help/matlab/ref/boundary.html [Accessed 23 Nov. 2019].
2. reyalp (2019). Chdkptp Project. [online] App.assembla.com. Available at: https://app.assembla.com/wiki/show/chdkptp [Accessed 23 Nov. 2019].
3. Bougue, J. (2019). Camera Calibration Toolbox for Matlab. [online] Vision.caltech.edu. Available at: http://www.vision.caltech.edu/bouguetj/calib_doc/ [Accessed 23 Nov. 2019].
4. F. Zhao, C. Wei, J. Wang, and J. Tang, "An Automated X-corner Detection Algorithm(AXDA)," Journal of Software, vol. 6, no. 5, May 2011.