An Algorithm for Characteristic Length Measurement from Point Cloud Data

Matthew T. Moraguez[1]
University of Florida, Gainesville, FL, 32611

The DebriSat project aims to characterize debris fragments produced by a hypervelocity impact test of a modern low Earth orbit satellite. The fragment characterization will provide data needed to update satellite break-up models. The characteristic length of a fragment, defined as the average of its three maximum orthogonal projected lengths, is used to quantify its physical size. In order to determine the characteristic length with accuracy, repeatability, and speed, an algorithm was developed to automatically extract the information from a point cloud representation of the fragment. The algorithm (i) improves measurement repeatability by eliminating human error and (ii) reduces the point cloud processing time to less than one second. Furthermore, the algorithm's output is exact for a given point cloud. The algorithm utilizes a convex hull approach to determine the diameter of the point set. Direction cosine matrices are utilized to perform the coordinate transformations needed to obtain orthogonal projections. This algorithm, coupled with a means of generating point clouds of debris fragments, allows for complete automation of the measurement process. Thus, fragment handling during measurement is minimized and the risk of fragment damage is mitigated. This paper details the characteristic length algorithm and its validation through comparison with existing NASA measurement results.

Nomenclature

Az	=	azimuth angle of **X** in $\mathcal{F}_{\mathrm{PC}}$
d_{BF}	=	number of distance computations in brute force algorithm
d_{Lc}	=	number of distance computations in characteristic length algorithm
El	=	elevation angle of **X** in $\mathcal{F}_{\mathrm{PC}}$
E_Q	=	percent error in quantity Q
$\mathcal{F}_{\mathrm{MD}}$	=	maximum dimension coordinate frame with basis vectors $\{\hat{\boldsymbol{x}}_{MD}, \hat{\boldsymbol{y}}_{MD}, \hat{\boldsymbol{z}}_{MD}\}$
$\mathcal{F}_{\mathrm{PC}}$	=	point cloud coordinate frame with basis vectors $\{\hat{\boldsymbol{x}}_{PC}, \hat{\boldsymbol{y}}_{PC}, \hat{\boldsymbol{z}}_{PC}\}$
$\mathcal{F}_{\mathrm{Xdim}}$	=	x-dimension coordinate frame with basis vectors $\{\hat{\boldsymbol{e}}_1, \hat{\boldsymbol{e}}_2, \hat{\boldsymbol{e}}_3\}$
$\mathbf{K}_{2D}$	=	vertices of the convex hull of $\mathbf{P}_{2D}$
$\mathbf{K}_{3D}$	=	vertices of the convex hull of $\mathbf{P}_{PC}$
L_c	=	characteristic length
n	=	number of data points
P_{iA}	=	index of *i*th endpoint of vector **A**
$\mathbf{P}_{2D}$	=	point cloud coordinates in $\mathcal{F}_{\mathrm{Xdim}}$ projected onto plane whose normal is **X**
$\mathbf{P}_A$	=	point cloud coordinates expressed in $\mathcal{F}_{\mathrm{A}}$
$\mathbf{R}_{A/B}$	=	the direction cosine matrix for the transformation from $\mathcal{F}_{\mathrm{B}}$ to $\mathcal{F}_{\mathrm{A}}$
t	=	computation time
θ	=	azimuth angle of **Y** in $\mathcal{F}_{\mathrm{Xdim}}$
X	=	x-dimension
X	=	vector representing the x-dimension
Y	=	y-dimension
Y	=	vector representing the y-dimension
Z	=	z-dimension
Z	=	vector representing the z-dimension

[1] Undergraduate Research Assistant, UF MAE, 939 Sweetwater Dr, Gainesville, FL 32611, Student Member

I. Introduction

THE continued growth of orbital debris poses a significant risk to satellites. In particular, fragmentation debris contributes strongly to the overall debris population. In order to gain an understanding of these fragments, ground-based hypervelocity impact tests have been conducted to simulate the conditions in a catastrophic, on-orbit collision. The debris fragments generated in the test are collected and characterized to improve the understanding and modeling of fragmentation debris. The size of fragments is of particular interest for use in satellite break-up models[1]. The size parameter used for orbital debris is characteristic length, which is the average of the object's three maximum orthogonal projected lengths. Accurate measurement of this length is required to update satellite break-up models and relate ground-based optical and radar observations of debris in orbit to their physical size[2].

Originally, fragment measurements were conducted using calipers, rulers, or graph paper. Due to the irregular shape of many of the debris fragments, hand measurements have proved difficult. The primary difficulty is the measurement of shadow dimensions, which cannot be measured directly on the object. Due to human error, the uncertainty in hand measurements is unknown. In addition, the handling of fragments to obtain measurements increases the possibility of damage[2].

To improve the quality of measurements, the NASA Orbital Debris Programs office adopted a computerized measurement approach. A handheld laser scanner was used to generate 3D models of fragments. Then, the object can be rotated and measured in a solid modeling program. This process significantly improved upon hand measurements because the results have improved accuracy and repeatability, the uncertainty can be quantified, fragment handling is reduced, and the point cloud and measurements are documented for future reference[1]. However, these computerized measurements are time consuming. The measurement process requires the user to size a bounding box to the 3D representation of the object to obtain measurements. Although this computerized approach moves closer to obtaining automated measurements, the human factor has not been completely removed from the measurement process[2].

The next step in debris fragment size characterization is to fully automate the measurement process[2]. There already exist solutions for fully automated 3D scanning of objects. However, the ability to automatically measure the characteristic length of a point cloud representation of a debris fragment has yet to be developed. The characteristic length algorithm presented in this paper will allow for complete automation of the measurement process. This advancement in debris size characterization completely removes human error from measurements. The algorithm is exact for any given point cloud. The uncertainty in the process is quantifiable and solely due to the 3D scanning of the object. Furthermore, the process is repeatable and provides documentation of how the measurements were computed.

II. Characteristic Length

The characteristic length, L_c, is used to quantify the physical size of debris fragments in satellite break-up models. Using L_c, the size of irregularly shaped objects is quantified with a single parameter. Characteristic length is defined as the average of the object's three maximum orthogonal projected lengths, X, Y, and Z [2]. In this definition, X is the longest projected length, Y is the longest projected length normal to **X**, and Z is the longest projected length normal to both **X** and **Y** [1].

$$L_C = \frac{X + Y + Z}{3} \quad \text{(Ref. 1)} \tag{1}$$

The x-dimension, X, is taken as the largest distance between any two points on the object (see Fig. 1). This can be interpreted as the maximum shadow dimension of the object. For example, X for a cube is its body diagonal[2]. The x-dimension is expressed as the magnitude of the vector **X** that connects points P_{1x} and P_{2x} in the original point cloud, $\mathbf{P}_{PC}$.

$$\mathbf{X} = \mathbf{P}_{PC}(P_{1x}) - \mathbf{P}_{PC}(P_{2x}) \tag{2}$$

$$X = |\mathbf{X}| \tag{3}$$

The y-dimension, Y, and the z-dimension, Z, both lie in the projection, or shadow, of the object onto the plane whose normal is **X** (see Fig. 2). This can be thought of as the silhouette of the object when viewed looking down the direction of **X**. Then, Y is defined as the largest distance between any two points, denoted P_{1y} and P_{2y}, in this

orthogonal projection[2]. For example, if the projection orthogonal to **X** of the object is an ellipse, then Y would be the major axis. In vector notation, Y is found as the magnitude of **Y**, the largest projected dimension orthogonal to **X**.

$$\mathbf{Y} = \hat{\mathbf{X}} \times [\mathbf{P}_{PC}(P_{1y}) - \mathbf{P}_{PC}(P_{2y})] \tag{4}.$$

$$Y = |\mathbf{Y}| \tag{5}.$$

Finally, Z is defined to be the maximum projected length orthogonal to both **X** and **Y** [2]. For example, if the projection orthogonal to **X** of the object is again an ellipse, then Z would be the minor axis. In vector notation, the expression for Z is the magnitude of **Z**, the longest projected length onto a line normal to both **X** and **Y**.

$$\mathbf{Z} = \left([\mathbf{P}_{PC}(P_{1z}) - \mathbf{P}_{PC}(P_{2z})] \cdot \hat{\mathbf{Z}}\right)\hat{\mathbf{Z}}, \quad \text{where } \hat{\mathbf{Z}} = \hat{\mathbf{X}} \times \hat{\mathbf{Y}} \tag{6}.$$

$$Z = |\mathbf{Z}| \tag{7}.$$

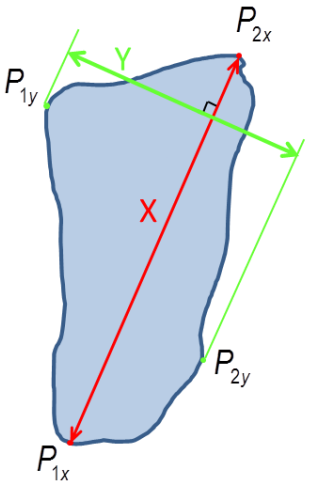


Figure 1. The x- and y-dimensions for an arbitrary shape. *The points P_{1x} and P_{2x} are the two surface points comprising X. Note that Y is the maximum shadow dimension perpendicular to* **X** *because the points P_{1y} and P_{2y} do not, in general, already lie in a plane normal to* **X**.

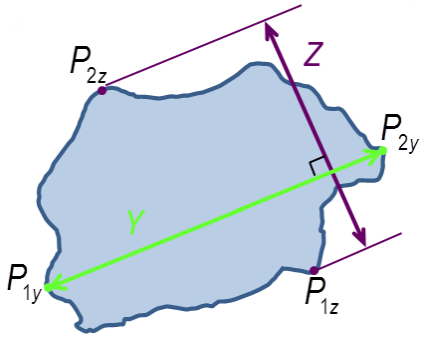


Figure 2. The y- and z-dimensions for an arbitrary shape. *The shape shown is the projection of the object onto a plane normal to the object's* **X**. *The points P_{1y} and P_{2y}, which make up the Y, are the two points farthest apart in the projection. Finally, Z is measured as the maximum shadow dimension normal to* **Y** *in the plane normal to* **X**.

III. Algorithm Implementation

The characteristic length algorithm outputs the characteristic length and maximum orthogonal projected lengths for point cloud data given in x-, y-, z- coordinates. The algorithm first computes X for the point cloud. Once X is known, the point cloud is projected onto a plane whose normal is **X**. Then, Y is determined from the 2D projection. Finally, Z is computed from the projection of the 2D data onto a line orthogonal to **Y**.

A. Computing the Convex Hull

A typical problem in the field of computational geometry is determining the diameter of a point set. The diameter is defined as the maximum distance between any two points in the set. This is the exact definition of X for the original point cloud. In addition, this is the definition of Y if the point set is considered to be the 2D point cloud projection. Thus, well-established work in the field of computational geometry can be applied in the characteristic length algorithm[3].

In computational geometry, the convex hull is a fundamental structure that is defined as the smallest convex polygon that contains all points in the original set. The convex hull for an arbitrary point set is shown in Fig. 3. Efficient algorithms, such as Graham's scan, the divide-and-conquer approach, or Chan's algorithm, have been developed that can determine the convex hull of a point set in $O(n \log n)$ running time[3]. It is known from Ref. 3 that "the pair of points determining the diameter [of a point set] are both vertices of the convex hull". For this reason, using the convex hull significantly reduces the number of points whose distances must be computed. To determine maximum dimensions, only the vertices of the convex hull of the original point cloud must be considered. The distance between each of the vertices of the convex hull is computed. Then, the maximum of these distances is selected as the diameter of the original point set. The significance of the convex hull is in reducing the number of points on which pairwise distances must be computed, which is a slow $O(n^2)$ running time operation.

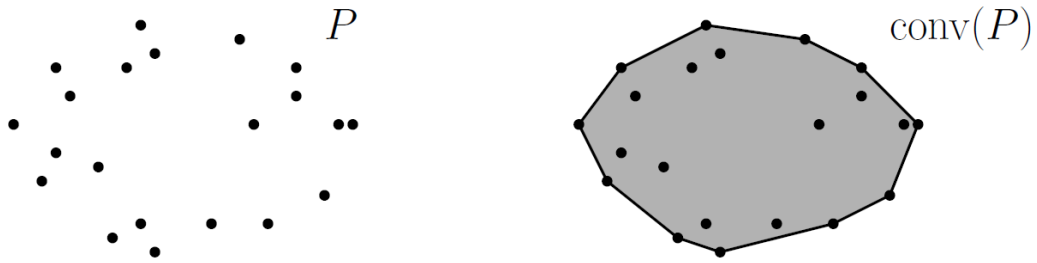


Figure 3. The convex hull of a point set. *This figure, taken from Ref. 3, shows the original point set (left) alongside the convex hull of the set (right). Note that the convex hull surrounds all points in the set. The diameter of the point set must include two points that are vertices of the convex hull.*

B. Coordinate Transformations

In the algorithm, orthogonal projections of the point cloud must be determined in order to compute Y and Z. Direction cosine matrices (DCMs) are utilized to achieve these coordinate transformations. The three coordinate frames, $\mathcal{F}_{\text{PC}}$, $\mathcal{F}_{\text{Xdim}}$, and $\mathcal{F}_{\text{MD}}$, that are used in this process are defined in Table I.

TABLE I
COORDINATE FRAMES

Coordinate Frame	Description	Basis Vectors
$\mathcal{F}_{\text{PC}}$	The point cloud frame is the frame in which the original x-,y-,z-coordinates of the point cloud are expressed.	$\hat{\boldsymbol{x}}_{PC}, \hat{\boldsymbol{y}}_{PC}, \hat{\boldsymbol{z}}_{PC}$
$\mathcal{F}_{\text{Xdim}}$	The x-dimension frame is such that the first basis vector lies in the direction of **X**.	$\hat{\boldsymbol{e}}_1 = \hat{\mathbf{X}}, \hat{\boldsymbol{e}}_2, \hat{\boldsymbol{e}}_3$
$\mathcal{F}_{\text{MD}}$	In the maximum dimension frame, the first, second, and third basis vectors lie along the direction of the object's **X**, **Y**, and **Z**, respectively.	$\hat{\boldsymbol{x}}_{MD} = \hat{\mathbf{X}}, \hat{\boldsymbol{y}}_{MD} = \hat{\mathbf{Y}}, \hat{\boldsymbol{z}}_{MD} = \hat{\mathbf{Z}}$

The coordinate transformation from $\mathcal{F}_{\text{PC}}$ to $\mathcal{F}_{\text{Xdim}}$ is obtained from a 3-2 rotation sequence through angles *Az* (azimuth) and *-El* (elevation) as shown in Fig. 4. The resulting DCM, $\mathbf{R}_{Xdim/PC}$, is:

$$\mathbf{R}_{Xdim/PC} = \begin{bmatrix} c_{El} & 0 & s_{El} \\ 0 & 1 & 0 \\ -s_{El} & 0 & c_{El} \end{bmatrix} \begin{bmatrix} c_{Az} & s_{Az} & 0 \\ -s_{Az} & c_{Az} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{El}c_{Az} & c_{El}s_{Az} & s_{El} \\ -s_{Az} & c_{Az} & 0 \\ -s_{El}c_{Az} & -s_{El}s_{Az} & c_{El} \end{bmatrix} \tag{8}$$

$$\mathbf{P}_{Xdim} = \mathbf{R}_{Xdim/PC}\mathbf{P}_{PC} \tag{9}$$

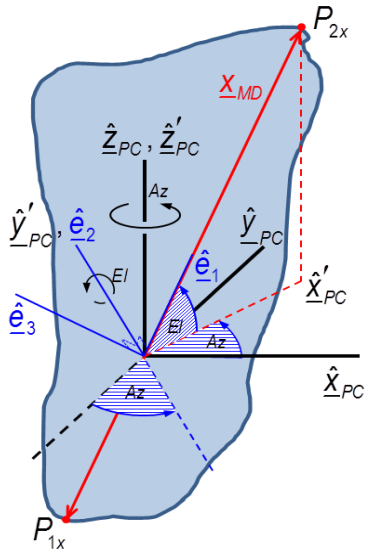


Figure 4. Coordinate transformation from $\mathcal{F}_{\mathbf{PC}}$ to $\mathcal{F}_{\mathbf{Xdim}}$. *The transformation occurs according to the azimuth and elevation angles of* **X** *in* $\mathcal{F}_{\mathrm{PC}}$. *The unit vector* $\hat{\boldsymbol{e}}_1$ *lies along* **X**. *The point cloud's Y and Z measurements both lie in the projection of the point cloud onto the* $\hat{\boldsymbol{e}}_2, \hat{\boldsymbol{e}}_3$*-plane.*

The projection of the point cloud onto the plane whose normal is **X** can be completed by simply extracting the $\hat{\boldsymbol{e}}_2$ and $\hat{\boldsymbol{e}}_3$ components of the point cloud. This 2D point cloud projection, $\mathbf{P}_{2D}$, can be used to compute Y (see Fig. 5). The azimuth of **Y** in $\mathcal{F}_{\mathrm{Xdim}}$ is computed and used for the final transformation. The coordinate transformation from $\mathcal{F}_{\mathrm{Xdim}}$ to $\mathcal{F}_{\mathrm{MD}}$ is given by the following DCM, $\mathbf{R}_{MD/Xdim}$:

$$\mathbf{R}_{MD/Xdim} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\theta & s_\theta \\ 0 & -s_\theta & c_\theta \end{bmatrix} \tag{10}.$$

$$\mathbf{P}_{MD} = \mathbf{R}_{MD/Xdim}\, \mathbf{P}_{Xdim} \tag{11}.$$

Once this transformation is completed, **X** lies along $\hat{\boldsymbol{x}}_{MD}$, **Y** lies along $\hat{\boldsymbol{y}}_{MD}$, and **Z** lies along $\hat{\boldsymbol{z}}_{MD}$. Therefore, Z is determined as the difference between the maximum and minimum $\hat{\boldsymbol{z}}_{MD}$ component of the point cloud in $\mathcal{F}_{\mathrm{MD}}$. Thus, $\mathcal{F}_{\mathrm{MD}}$ is the frame in which the point cloud can be expressed such that the maximum orthogonal projected dimensions lie along their respective basis vector directions.

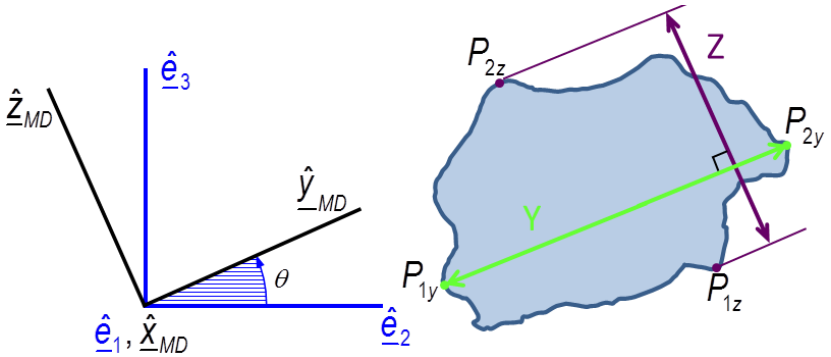


Figure 5. Coordinate transformation from $\mathcal{F}_{\mathbf{Xdim}}$ to $\mathcal{F}_{\mathbf{MD}}$. *The transformation is shown for the 2D projection into the plane whose normal is* **X**. *This transformation occurs according to the azimuth angle of* **Y** *in* $\mathcal{F}_{\mathrm{Xdim}}$.

C. Overall Algorithm Process

The characteristic length algorithm uses coordinate transformations to achieve orthogonal projections along with convex hulls for determining maximum dimensions. The algorithm is represented in the flow chart shown in Fig. 6.

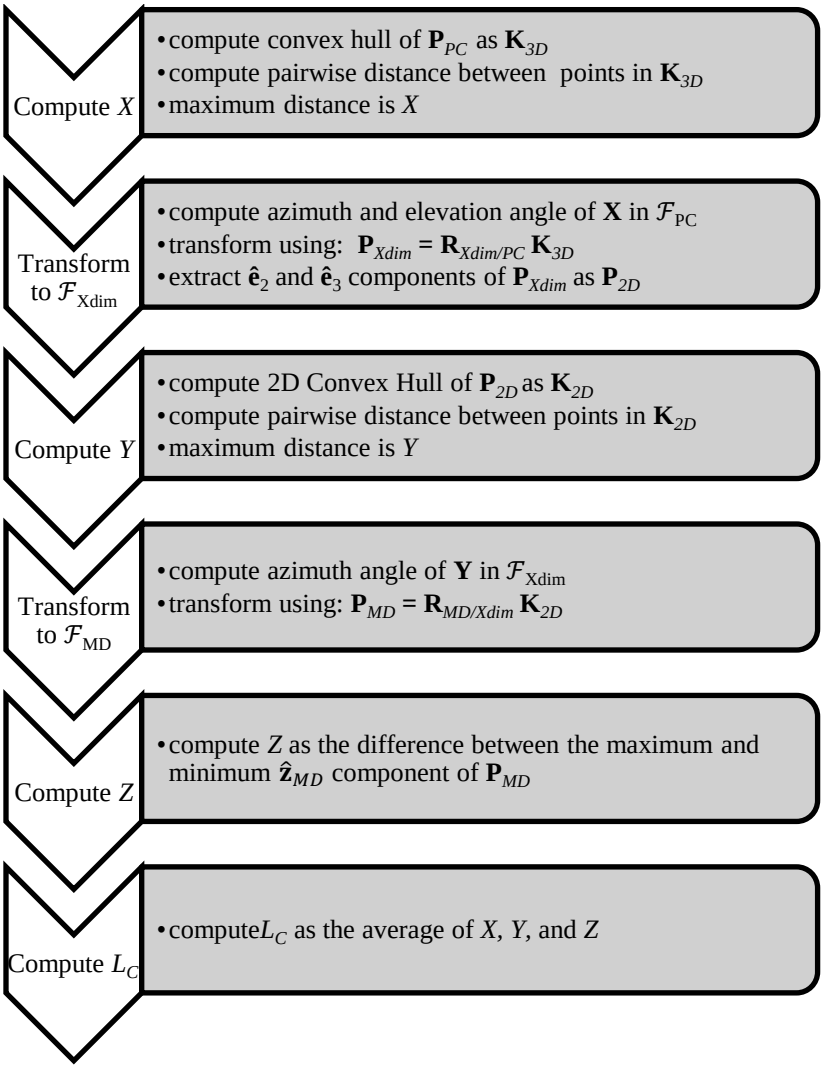


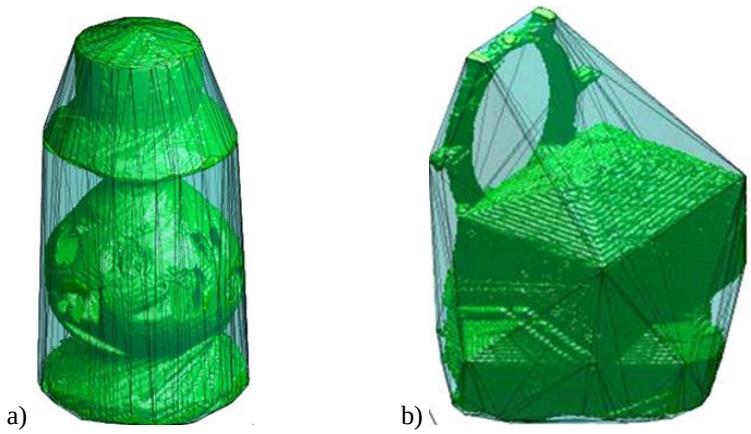
Figure 6. Characteristic length algorithm process. *The flow chart above shows the high-level process followed by the characteristic length algorithm as it computes the characteristic length from a point cloud.*

IV. Correctness of the Algorithm

The characteristic length algorithm was evaluated by verifying the accuracy of the convex hull approach and the coordinate transformations. Then, the algorithm was used to measure simple geometries with known maximum dimensions. Test objects were measured using calipers and compared to the algorithm measurement. Lastly, the algorithm results were compared to an independently developed brute force approach.

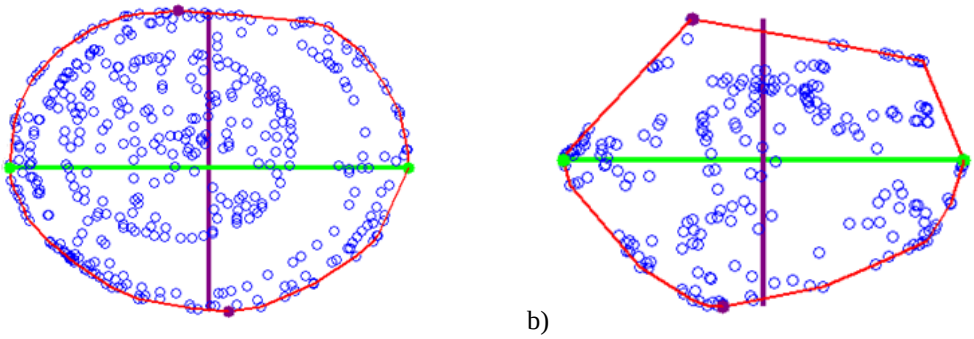
A. Computing the Convex Hull

The exactness of the convex hull approach for determining the diameter of a point set is known from computational geometry[3]. However, to ensure proper implementation of the approach, plots of 3D scanned objects along with their convex hull superimposed were developed. The two images shown in Figs. 7 and 8 show a visual representation of the data reduction that occurs when using the convex hull. Using the convex hull approach allows the algorithm to discard all internal points and concavities in the point cloud that will not contribute to the maximum dimensions. Only the vertices of the point cloud need to be considered as potential points giving the maximum dimension.



a) b)

Figure 7. Convex hull of test sample point clouds. *The green solid is the original 3-D scanned point cloud data. The light blue surface surrounding the object is the convex hull of the object. The only points that could potentially contribute to the maximum dimension are the vertices of the convex hull. The point cloud on the left originally contained ~725,000 points, yet the convex hull representation contains only 416 vertices. The point cloud on the right originally contained 1.3 million points, but the convex hull reduced this number down to 223 points whose distances from one another must be computed.*



a) b)

Figure 8. 2D convex hull of projected test sample point clouds. *The 2D projections of the two test samples from Fig. 7 are shown above. The blue circles represent the points in the projection of the convex hull,* $\mathbf{K}_{3D}$*, onto a plane whose normal is* **X**. *The red line surrounding these points represents the convex hull,* $\mathbf{K}_{2D}$*, of the projected data points. The green line represents Y and the purple line is Z. The convex hull encloses all points and reduces the number of points to measure. As expected, the maximum dimensions are measured from vertices on the convex hull.*

B. Coordinate Transformations

The accuracy of the transformations and orthogonal projections are checked by the algorithm. After the transformation to $\mathcal{F}_{\text{Xdim}}$ is conducted, the algorithm checks that the vector connecting P_{1x} and P_{2x} lies only in the $\hat{\boldsymbol{e}}_1$ direction. This means that the $\mathbf{X}$ expressed in $\mathcal{F}_{\text{Xdim}}$ is given as $[X, 0, 0]$. Similarly, once the transformation to $\mathcal{F}_{\text{MD}}$ is completed, the algorithm checks that the vector connecting $\mathbf{Y}$ lies only in the $\hat{\boldsymbol{y}}_{MD}$ direction. This means that $\mathbf{Y}$ is expressed in $\mathcal{F}_{\text{MD}}$ as $[0, Y, 0]$. Lastly, Z will be expressed in $\mathcal{F}_{\text{MD}}$ as $[0,0, Z]$.

C. Measurement of Simple Geometries

The correctness of the algorithm's measurements was first validated by measuring simple geometries whose maximum orthogonal dimensions and characteristic lengths are known. For example, the lengths X, Y, Z, and L_C for a sphere are all equal to the sphere's diameter[2]. For a cube, X is given by the body diagonal. Running the algorithm on computer-generated point clouds of these geometries confirmed that the measurement result agreed with known geometric calculations. Furthermore, the same point cloud was fed into the algorithm several different times in different initial orientations to ensure that the result was independent of initial orientation. Again, the algorithm proved to be correct in returning the same measurement result regardless of the point cloud orientation.

D. Comparison to Brute Force Approach

The algorithm measurements were also compared to the results from a brute force computational approach. The brute force approach involves calculating the distance between every pair of points in the original point cloud and selecting the largest distance as *X*. The unit vector $\hat{\boldsymbol{e}}_1$ is defined as the unit vector along the direction of **X**. Then, the cross product between $\hat{\boldsymbol{e}}_1$ and the vector between every combination of points in the point cloud is computed. The largest of these cross product magnitudes is taken as *Y*. The unit vector $\hat{\boldsymbol{z}}_{MD}$ is computed as the unit vector normal to both **X** and **Y**. Then, the dot product between $\hat{\boldsymbol{z}}_{MD}$ and the vector between every combination of points in the cloud is calculated. Finally, *Z* is found to be the maximum absolute value of these dot products. The brute force approach is summarized by the flow chart in Fig. 9.

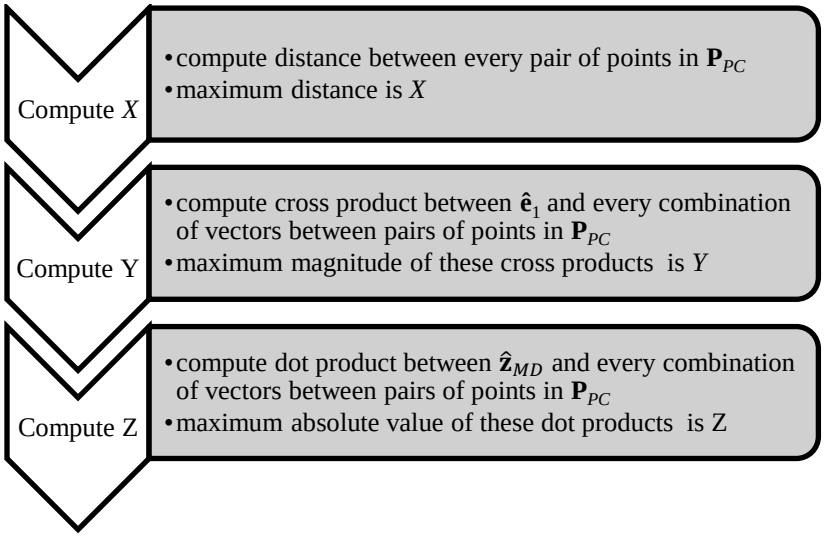


Figure 9. Brute force approach process. *The flow chart above shows the brute force approach used to validate the characteristic length algorithm. The brute force approach does not rely on convex hulls or coordinate transformations to measure the point cloud.*

The brute force approach does not rely on the computation of convex hulls or coordinate transformations. Although the brute force approach is computationally inefficient, it is used to validate the algorithm's results because the approach is derived directly from the definitions of *X*, *Y*, and *Z*. Thus, an agreement in the results of the brute force approach and the characteristic length algorithm indicates the accuracy of the latter. The brute force approach confirmed the accuracy of the characteristic length algorithm for all objects processed, such as the geometric shapes, the 3D scanned test samples, and previously scanned NASA debris fragments. Furthermore, calipers were used to measure the physical dimensions of every test object to confirm the algorithm and brute force result.

V. Comparison of Results with NASA Measurements

The algorithm was used to process fourteen point clouds of actual hypervelocity impact test debris fragments that NASA had previously measured. To compare the results, the algorithm was used to produce three orthographic views of the point cloud, with the maximum dimensions labeled, for visual inspection of the results. These plots show that the measurements exist in the original point cloud data, that the maximum dimensions are taken as orthogonal projections, and that the measurements appear intuitively correct. The indices of the points in the original point cloud, $\mathbf{P}_{PC}$, that gave the maximum dimensions were also reported from the algorithm. By reporting P_{1x}, P_{2x}, P_{1y}, P_{2y}, P_{1z}, and P_{2z}, the measurement result can be confirmed by simply evaluating Eq. (3), Eq. (5), and Eq. (7) to give *X*, *Y*, and *Z* directly from the original data.

The measurement results from the brute force approach were taken as the actual measurement for the point cloud. Then, the measurement percent errors were computed with respect to the brute force result. A positive percent error indicates that the brute force result is greater than the NASA result. The characteristic length algorithm exactly matches the brute force result. For this reason, the percent error of the algorithm was zero for all measurements for all objects tested. To validate the result, the agreement between NASA and brute force measurements were computed and are shown in Table II.

TABLE II
MEASUREMENT RESULT COMPARISON

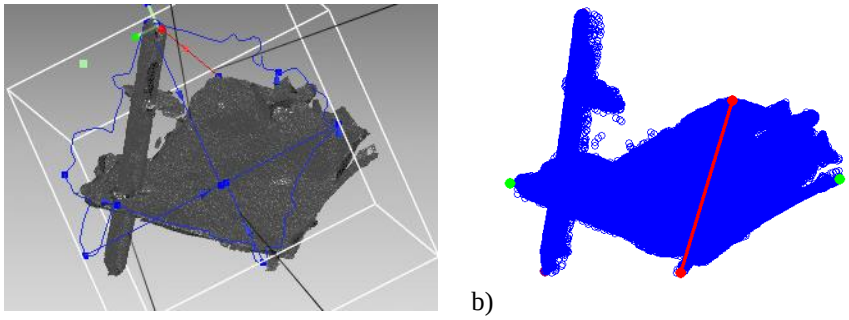
Object	X (mm)	Y (mm)	Z (mm)	L_c (mm)	E_X (%)	E_Y (%)	E_Z (%)	E_{Lc} (%)
Shuttle MG	129.00	111.06	60.39	100.15	0.3	0.5	3.3	1.0
Mallard	263.41	139.08	102.89	168.46	0.5	2.7	7.9	2.6
SOCIT69	303.67	82.66	58.73	148.35	0.8	0.7	5.0	1.3
SOCIT8	296.37	157.66	16.91	156.98	0.9	-1.8	7.0	0.2
Shot 3 frag 7 Lc	144.66	131.07	57.98	111.24	1.4	-6.4	0.8	-1.7
Large nugget	20.13	18.62	14.85	17.87	-1.5	0.5	0.8	-0.2
Shot 3 frag 30	65.38	30.90	18.73	38.33	1.7	-2.6	5.1	1.1
96	81.49	69.06	38.76	63.10	2.0	1.6	-2.0	1.0
Esoc2	95.06	39.76	39.20	58.01	2.2	-17.0	23.4	2.6
Shot 3 frag 20	119.69	64.04	31.65	71.79	2.5	3.7	18.3	5.1
Shot 3 frag 28	76.09	43.16	33.16	50.80	3.1	-2.0	-1.0	0.7
Hyperv23	65.87	60.17	23.02	49.69	4.5	1.3	12.7	4.5
Lowv12final	117.46	97.47	85.03	99.99	12.7	9.2	2.7	8.7

The reported X, Y, Z, and L_c were computed using the brute force approach. Measurements are reported to two decimal places based on practical limitations of the accuracy of the original point cloud. The percent errors shown in the table refer to that of the NASA measurements. The percent error for the algorithm is not reported because it was zero for all measurements on all objects. A positive percent error indicates that the brute force result is greater than the NASA result. The table is sorted by increasing X percent error (E_X).

Based on the results presented in Table II, the NASA measurements appear to typically underestimate the x-dimension, and thus, the characteristic length. However, it could equally be proposed that the characteristic length algorithm and brute force approach overestimate these measurements. The discrepancy in *X* can be resolved by considering that *X* is defined to be the largest dimension of the part. Thus, if the distance between any two points in the point cloud is found to be greater than another proposed *X*, the former dimension can unambiguously be stated to be more accurate than the latter. Based on the brute force approach validation, it is known that the algorithm's results are, in fact, exactly accurate for a given point cloud. Thus, the algorithm's reported *X* is, at the very least, more accurate than the NASA approach for the given point cloud.

Now that the algorithm's *X* measurement has been validated, the discrepancy in *Y* and *Z* can be resolved by considering the definition of those maximum dimensions. Both *Y* and *Z* are defined to lie in the projection of the point cloud onto a plane normal to the direction of *X*. Clearly, a discrepancy in *X* will propagate into a discrepancy in *Y* and *Z* due to the measurement being taken in a different projection plane.

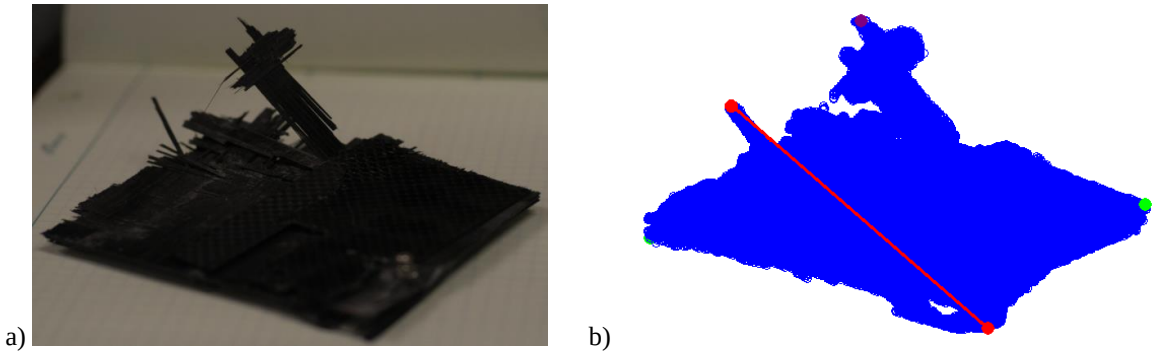
The largest discrepancy in *X* and L_c was seen in the object lowv12final. Since the algorithm's results are exact for a given set of point cloud data, this discrepancy is attributed to human error in the NASA measurement. The algorithm's processing of this object serves as a prime example of the benefit of automated measurement to remove human error. The NASA measurement shown in Fig. 10 is clearly measured between different points on the object than the algorithm measurement. It is common to find objects with complex geometries, such as lowv12final, in hypervelocity impact debris. When measuring these objects manually, it is extremely difficult to obtain an accurate, repeatable result. However, the algorithm measurement is accurate even for this complicated geometry.



a) b)

Figure 10. Measurement comparison of lowv12final. *The images above show the measurement for the object lowv12final as computed by NASA (left) and the algorithm (right). In both images, the red line indicates X. Note that the NASA X differs from the X computed by the algorithm. In fact, the algorithm found a larger dimension in the point cloud. Thus, the largest discrepancy in Table II is resolved. This error in NASA's X propagated into the error in Y and Z because the direction of* **X** *is different for NASA and algorithm measurements.*

The characteristic length algorithm measurement for X was compared to the NASA measurement using the values in Table II. A visual representation of this comparison provides further confirmation of the accuracy of the algorithm. The X measurement appears intuitively accurate for the object shot 3 frag 7 Lc (see Fig. 11), whose X measurement was consistent between NASA and the algorithm.



a) b)

Figure 11. Measurement comparison of shot 3 frag 7 Lc. *The object shot 3 frag 7 Lc (left) is shown side-by-side with the point cloud representation (right) that was provided to the characteristic length algorithm. The x-dimension is shown as a red line. The quality of the point cloud is attributed to the 3D scanning procedure and not the algorithm itself. The algorithm can only obtain measurements as accurate as the point cloud it is measuring.*

Using the algorithm, complex fragment geometries can be accurately and precisely measured. These complex geometries are commonly collected in hypervelocity impact tests. The object esoc2 (see Fig. 12) was an actual debris fragment provided by NASA. The object is representative of geometries to be measured with the algorithm.

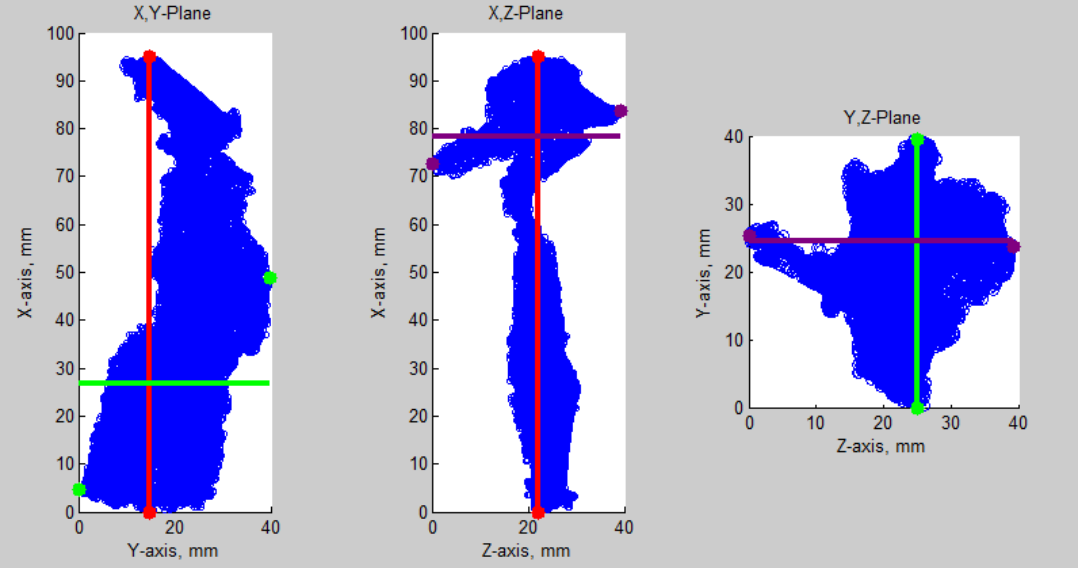


Figure 12. Orthogonal projections of esoc2 algorithm measurement. *Three orthographic projections of esoc2 expressed in $\mathcal{F}_{\mathrm{MD}}$ were produced using the algorithm. Even for this complex geometry, the algorithm measurement is repeatable. Visual inspection of the dimension lengths matches the value reported in Table II.*

VI. Computational Efficiency of Algorithm

The performance of the algorithm was also investigated using the NASA point clouds. The computational speed of the algorithm was of particular interest due to the large quantity of debris fragments requiring size characterization after a hypervelocity impact test. The speed of the algorithm arises due to the convex hull approach and efficient computation of transformations using DCMs. Using the convex hull allows for a significant reduction of the number of data points whose pairwise distances must be computed (see Table III). The data point reduction is accompanied by an even greater reduction in the number of computational operations. The number of required distance calculations, d_{BF} and d_{Lc}, was computed using the binomial coefficient to find the number of combinations

of pairs of points in the data whose distances were computed during processing. On average, the algorithm reduces the number of distance calculations by about five orders of magnitude relative to the brute force approach. This translates into a computational time of less than half of a second for all point clouds processed.

TABLE III
ALGORITHM PERFORMANCE

Object	$n(\mathbf{P}_{PC})$	$n(\mathbf{K}_{3D})$	$n(\mathbf{K}_{2D})$	d_{BF}	d_{Lc}	t (s)
Shuttle MG	146,675	1608	80	2.2E10	1.3E6	0.40
Mallard	54,818	1575	67	3.0E9	1.2E6	0.25
SOCIT69	32,634	141	18	1.0E9	1.0E4	0.02
SOCIT8	26,456	165	20	7.0E8	1.4E4	0.02
Shot 3 frag 7 Lc	148,690	127	20	2.2E10	8.2E3	0.02
Large nugget	148,339	349	46	2.2E10	6.2E4	0.11
Shot 3 frag 30	47,996	179	19	2.2E9	1.6E4	0.02
96	20,116	135	24	4.0E8	9.3E3	0.02
Esoc2	15,082	98	18	2.2E8	4.9E3	0.01
Shot 3 frag 20	17,180	106	19	2.8E8	5.7E3	0.01
Shot 3 frag 28	32,015	93	25	1.0E9	4.6E3	0.01
Hyperv23	96,187	133	21	9.2E9	9.0E3	0.02
Lowv12final	72,165	127	14	5.2E9	8.1E3	0.01

The columns $n(\mathbf{P}_{PC})$, $n(\mathbf{K}_{3D})$, and, $n(\mathbf{K}_{2D})$ represent the number of points in the original point cloud, the 3D convex hull, and the 2D convex hull, respectively. The columns d_{BF} and d_{Lc} give the number of distance computations required by the brute force approach and the algorithm, respectively. The column t gives the computational time required to measure the point cloud. The reported time was achieved on the author's personal laptop.

VII. Conclusion

The characteristic length algorithm is an accurate, repeatable, and fast method for characterizing the size of debris fragments. The algorithm returns the exact measurements for the given point cloud without any human error. The algorithm measurements are repeatable to machine precision. This repeatability is a significant improvement over caliper measurements or even NASA's computerized measurements. The limiting factor for measurement accuracy is the quality of the point cloud representation of the object. The speed of the algorithm is due to the data reduction of the convex hull approach and the computation of orthogonal projections using DCMs. Any point cloud representation of a debris fragment can typically be measured in less than one second. This computational speed is critical for characterizing the numerous debris fragments generated by a hypervelocity impact test.

This algorithm makes it possible to obtain fully automated measurements of debris fragments. These measurements are based on point clouds that can be generated by non-contact 3D scanners. Thus, potential contamination and damage to fragments during measurement can be mitigated. An imaging system to produce point clouds of debris fragments is currently being developed to provide the algorithm with quality point clouds for characteristic length measurement. This imaging system, coupled with the characteristic length algorithm, provides an end-to-end solution for automated size characterization of debris fragments.

The end goal of debris fragment size characterization is to improve the understanding of orbital debris. With the improved accuracy of measurements provided by the characteristic length algorithm, fragment size will be interpreted more accurately from optical and radar observations of debris. Furthermore, the new measurement results will improve the quality of satellite breakup models.

Acknowledgments

The author thanks Dr. Fitz-Coy for presenting this research topic for study and for his help and support throughout the process. He thanks Dr. J.-C. Liou for funding the DebriSat project for which this research was conducted. Special thanks is given to Dr. Cowardin for helping to validate the algorithm's accuracy with NASA samples and measurements. The author thanks Mr. Patankar for his guidance during the algorithm's development.

References

[1]Hill, N. M., "Measurement of Satellite Impact Test Fragments for Modeling Orbital Debris," *AIAA Technical Symposium*, 2009, pp. 5-21.

[2]Hill, N. M., and Stevens, A., "Measurement of Satellite Impact Fragments," *NASA Orbital Debris Quarterly News*, Vol. 12, No. 1, Jan. 2008, pp. 9, 10.

[3]Mount, D. M., "Computational Geometry, " *CMSC 754, University of Maryland*, 2012, pp. 10-21.