Initial Mass and Center of Mass estimation of objects from their point clouds

Sven Pegels[1]

Supervisor(s): Dr. Ranga Rao Venkatesha Prasad, Kees Kroep[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 26, 2023

Name of the student: Sven Pegels Final project course: CSE3000 Research Project
Thesis committee: Dr. Ranga Rao Venkatesha Prasad, Kees Kroep, Dr. Michael Weinmann

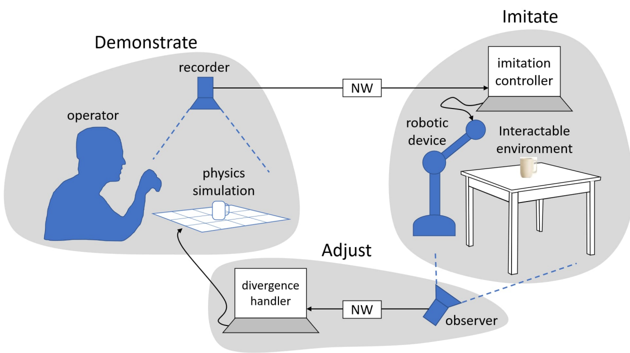


Figure 1: The proposed teleoperation setup, where the remote environment is locally simulated. This way feedback to the operator can be provided in near real-time from this simulation.

Abstract

The current state of teleoperation in Tactile Internet faces the problem of limited operating distance. To circumvent this limitation, a setup has been proposed which provides Ultra-Low Latency feedback to the operator from a local simulation of the remote environment. To be able to accurately simulate the remote environment and the objects within it, an initial estimation of the mass and Center of Mass of these objects is required. We explore three approaches to estimating these properties, using Axis-Aligned Bounding Bounding Boxes (AABB), Oriented Bounding Boxes (OBB), and convex hulls. We also explore the impact of sensor resolution and object occlusion. The performance of these methods varies and highly depends on the type of object. Sensor resolution has a high impact on the estimated mass, but not on Center of Mass. Object occlusion has a clear impact on both, suggesting the use of multiple viewpoints will improve accuracy.

1 Introduction

This section introduces the field of Tactile Internet (Ti) and our research in this field.

”The field of teleoperation coupled with force feedback will undergo a paradigm shift in the forthcoming years with the advent of Tactile Internet (TI)” [7]. Take Tactile Internet (TI) for example, which aims to allow for the perception of physical touch over the Internet. More formally, the IEEE 1918.1 “Tactile Internet” Standards Working Group defines TI as: “A network (or network of networks) for remotely accessing, perceiving, manipulating, or controlling real or virtual objects or processes in perceived real-time by humans or machines” [9, p. 258].

One of the applications of TI is teleoperation, where a remote operator is able to control a device in a remote environment. This can be extremely useful when operating in dangerous environments [13], or when people have to be separated for the sake of public health. However, this requires an incredibly low latency communication between the operator and the controlled device. Even with communication at the speed of light, the 1ms round-trip latency identified [9] for the communication to be in ”perceived real-time” limits the distance of operation to 150km. This would mean an operator stationed in the Netherlands would barely be able to operate outside of the country.

To circumvent this limitation as a whole, an alternative setup has been proposed. To be able to provide near real-time feedback to the operator, a local simulation of the remote environment is used, see Figure 1. This way the visual and haptic feedback to the operator can be directly provided from this simulation, eliminating the network latency.

However, this introduces a significant new challenge, simulating a remote environment accurately enough to provide accurate feedback. Not providing accurate feedback can lead to unwanted side effects, such as the discomfort of the operator, to more catastrophic consequences like damage to machinery. The simulation can periodically be synchronised, however, this alone is not enough. To be able to simulate the environment and the objects within it several properties of these objects are required.

Two important properties are the mass and Center of Mass (CoM) of an object. A physics simulation requires these to simulate how objects interact with each other. It might be possible to discover these properties through physical interaction with the objects, however, some initial value is still necessary. Thus, in this work, we look into the question ”What techniques can be used to make an initial estimation of the mass and Center of Mass of objects”. To answer this question we identified a number of sub-questions:

- ”How accurately can the mass and Center of Mass of an object be estimated”
- ”Are these estimates still accurate when part of the object is occluded”
- ”How does the resolution of sensors impact the accuracy of the estimations”

This work provides the following contributions:

- Insight into how accurately the volume of objects can be estimated
- Insight into how accurately the Center of Mass of objects can be estimated
- Insight into how sensor resolution impacts the estimations
- Insight into how the occlusion of objects impacts the estimations

The rest of the paper is organised as follows: Section 2 goes through research and concepts related to our research. Section 3 lays out the problems we encountered and the approaches used to solve them. Section 4 provides details on the implementation of these solutions. Section 5 explains how we tested the performance of the solutions and presents the gathered results. In Section 6 we discuss the results we gathered and relate them to previous works. Section 7 discusses the ethical challenges that were encountered. Finally, Section 8 concludes the paper as we draw conclusions from the gathered results, and identify future work.

2 Related Works

This section goes through research and concepts related to our research.

Previous research has been done into "Estimating the mass of an object from its point cloud for Tactile Internet" [2]. Here, two methods were compared under a number of necessary assumptions. Assumed was that the problem of separating different objects' point clouds and missing or occluded data were solved, as well as there being no noise in the data and it being possible to estimate the density of the object. The density of the object being known leaves only the need to calculate the object's volume. Two methods of volume estimation, Oriented Bounding Box (OBB), and surfacing the mesh with a greedy triangulation algorithm then dividing the resulting mesh into tetrahedrons, were compared against each other. The results showed that the 'surfacing and division' method generally performed better than the OBB method. However, the 'surfacing and division' method did show a large outlier, which was speculated to be caused by gaps in the mesh resulting from the surfacing algorithm [maybe include a part in results where it shows that when using an original mesh instead of a calculated one the division into tetrahedrons method is pretty much perfect]. These gaps were the result of incorrect parameters, which had to be chosen to be able to apply the same method to a wide range of objects.

This paper also identified alternative solutions to estimate the mass of an object. However, these only applied to specific types of objects, such as salmon [3], cows [12], and pigs [15]. Since a Tactile Internet (TI) application aims to deal with a large variety of objects, the aforementioned approaches are not applicable.

The estimation of the mass of an object can be separated into two individual problems, that of volume and of density estimation. A possible approach to estimating the density of an object is the use of infrared thermography [1]. This work managed to achieve a 99.1% R^2-fit for the predicted versus the actual density.

An alternative to the surface reconstruction method is a convex hull, a convex mesh which wraps around a set of n-dimensional points. While many algorithms exist applicable to x-dimensional data, we require an algorithm applicable to the 3-dimensional case. This narrows the algorithms down to QuickHull [4], a divide-and-conquer [10] method, and Chan's Algorithm [5]. While each uses a different algorithm, the resulting convex hull will be equal.

As for the Center of Mass (CoM), an approach for estimating the CoM of shape-unknown objects exists [16]. However, here the interaction of a robotic arm with the objects is used, which is strictly forbidden in our case as we aim to make an estimation before any physical interaction happens.

3 Methodology

This section lays out the methods used to estimate volume and Center of Mass, and why those methods were chosen. Background information on the mentioned methods is provided in Section 2.

Making an 'initial estimation' forbids any physical interaction with the object, as we aim to make an estimation of the mass and Center of Mass (CoM) before the object is interacted with. This meant that the object and its environment could only be observed from a distance. In this case, this observation would be done with the use of RGB-D cameras. This would provide us with a regular RGB image, as well as depth information in the form of the distance from the camera sensor to where that camera ray hit some object in the environment. This depth information can be translated into a set of 3-dimensional points, each point consisting of an x, y, and z coordinate. Thus the data we were working with is a set of XYZ-points of the environment.

While estimating the mass and Center of Mass (CoM) of an object are two separate problems, similar assumptions had to be made for both. Firstly, the methods used for either problem required the data to be of only the object itself. This could however be assumed to be solved, as existing segmentation algorithms provide a way to separate the object from the rest of the data [8]. Secondly, in reality, the gathered data will likely contain noise. Again, existing algorithms allowed us to filter out noise, allowing us to work with perfect noiseless data [11]. Finally, objects were assumed to be simple objects with an equal density throughout their entire volume.

One assumption not made was the lack of missing information. While we did still explore the situation where information about the entire object is available, our focus lay on the more realistic situation where only data from a single view of the object is available. However, We still assumed our view of the target object was not blocked by any other object. Exploring both situations would allow us to say something about the difference in results, and thus the necessity of multiple viewpoints. These multiple viewpoints could be gathered either through multiple cameras or by moving the camera through the environment. Besides that we aimed to provide more consistent results without any large outliers, improving on the results attained by Thomas Baars [2].

3.1 Mass Estimation

The mass M of an object is equal to the product of its volume V and the material's density d. This splits the problem of mass estimation into two sub-problems. Assuming the density is known, left us with the need to estimate the volume of an object. While this was a strong assumption, and previous research done by Haocheng Yang [14] did not provide a working approach for mass estimation, it was necessary to assume this part of the problem had been solved or would be solved in the near future.

As this work builds upon the research by Thomas Baars [2], the two methods for volume estimation described in their work were implemented first. Doing this would allow for the performance of other methods to be directly compared against what was achieved previously, with the aim to provide improved results.

AABB and OBB bounding boxes

One of the previous approaches used the object's Oriented Bounding Box (OBB), using the volume of the box as an estimation. While being a very naive approach, it can provide an upper bound on the true volume of the object, and can quickly be calculated from a set of XYZ-points. Unlike an

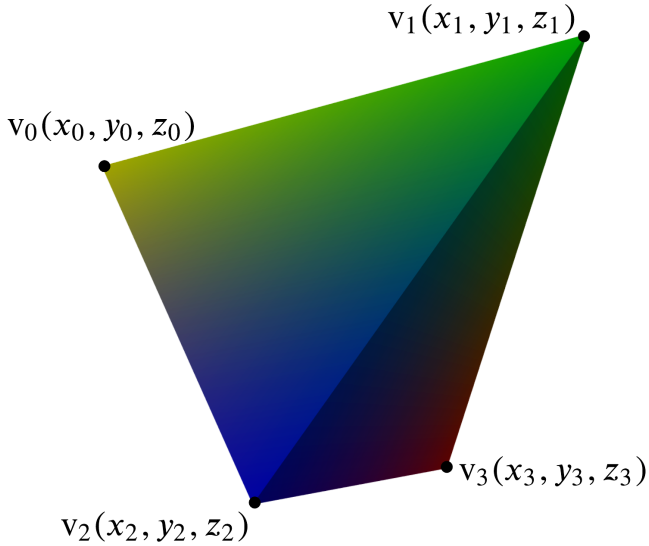


Figure 2: A tetrahedron created from the vertices v_1, v_2, and v_3 of a surface triangle, and a reference point v_0 [2].

Axis-Aligned Bounding Box (AABB), it can be oriented such that it wraps more tightly around the set of points, therefore being closer to the true volume of the object. Besides that, being able to be oriented makes the resulting volume independent of the orientation of the object.

On the other hand, while an AABB will provide a less accurate estimation, the error of this estimation might be more consistent. A consistent error could be corrected by a combination of a constant factor and offset. However, this is purely speculation, and results will have to show whether this holds any truth.

Convex Hull

While a bounding box provides a quick estimate, it is unlikely to be accurate. A more accurate approach would be to reconstruct the surface of the object, and use the volume of that reconstructed mesh as an estimation. This is the second approach explored by Thomas Baars [2]. While providing accurate estimates, it also resulted in large outliers caused by gaps in the reconstructed mesh. It was concluded that to prevent these gaps, different parameters would be needed for different objects. This however would require the use of some form of object identification.

An alternative approach we explored is the use of a convex hull, which wraps a convex mesh around the entire set of points. Unlike the mesh reconstruction method, a convex hull is guaranteed to not contain any gaps in the mesh. On the downside, it will not wrap as tightly around objects which have holes or other concave features, such as a torus (Figure 3e) or a mug (Figure 3f). Therefore it likely will not be as accurate of an estimation as the volume of a well-resurfaced mesh.

To calculate the volume of the convex hull we used the same method used for the reconstructed mesh. For this, the mesh first had to be turned into a mesh consisting of only triangles, known as triangulating a mesh. The mesh is divided into tetrahedrons, after which the volume of each individual tetrahedron is summed together. A tetrahedron is created for each triangle in the mesh, made up of the three vertices of the triangle and a reference point. Where this reference point is does not matter, as long as it is the same for each tetrahedron. The volume of a tetrahedron can be computed using the following formula:

$$\begin{aligned}&\frac{1}{6}|-(x_2-x_0)(y_3-y_0)(z_1-z_0)+(x_3-x_0)(y_2-y_0)(z_1-z_0)\\&+(x_2-x_0)(y_1-y_0)(z_3-z_0)-(x_1-x_0)(y_2-y_0)(z_3-z_0)\\&-(x_3-x_0)(y_1-y_0)(z_2-z_0)+(x_1-x_0)(y_3-y_0)(z_2-z_0)|,\end{aligned}$$

where v_1, v_2, and v_3 are the vertices of the surface triangle, and v_0 is the reference point, see Figure 2. The sign of this volume can be calculated as the inner product of the vector from v_0 to v_1, and the surface normal.

This requires the surface normals to be oriented towards the outside of the mesh. For the sake of this calculation, the surface normal of a triangle can be any vector perpendicular to its surface. This can be obtained from cross product between the vector from v_1 to v_2, and the vector from v_1 to v_3. The normal then has to be oriented towards the outside of the mesh. The mesh being a convex hull allowed us to take the average of all the vertices, and orient the normal vector away from this point, meaning, at an angle $\geq 90\,\mathrm{deg}$.

3.2 Center of Mass Estimation

Besides the mass, we also aimed to estimate the Center of Mass (CoM) of an object. A logical step was to see how the previously mentioned approaches for estimating volume could be used here.

AABB and OBB bounding boxes

As for the OBB and AABB approach, a way to estimate the CoM would be to take the center of the box. The center can be obtained by taking the average of all 8 vertices of the box, or of 2 vertices laying opposite of one another. While this is a very naive approach, the result might be promising, as in general, the CoM of an object lies somewhere near its center. This is more true for symmetric objects than it is for objects with the bulk of their material on one single side. The same hold for objects with an equal density throughout the object, compared to those consisting of materials differing significantly in density. In our case, however, we assumed density to be consistent throughout the object.

Convex Hull

The convex hull approach could be used here as well, by taking the CoM of the hull mesh as an estimate. As a convex hull is able to wrap more tightly around a set of points, the estimated CoM is expected to be closer to the true CoM. We assumed this would especially show for less symmetric objects, as the convex hull will resemble the shape of the object more than a simple OBB. To calculate the volumetric center of a mesh, a formula can be derived as follows:

$$C_3V_3 = C_1V_1 + C_2V_2, \tag{1}$$

where,

$$C = \frac{1}{V}\sum_{i=0}^{n} c_i v_i, \tag{2}$$

gives,

$$C_{\text{wanted}} = \frac{1}{V_{\text{wanted}}}\Big(\sum_{i=0}^{n_{\text{full}}} c_i v_i - \sum_{j=0}^{n_{\text{extra}}} c_j v_j\Big). \tag{3}$$

Here, Eq. (1) shows the relation between the CoM of two separate objects `1` and `2` with their CoM C_x and a volume V_x, and the object `3` which consists of objects `1` and `2`. Eq. 2 calculates the CoM C of an object consisting of n sub-objects, where the volume and CoM of an object i are v_i and c_i respectively. Combining Eq. 1 and Eq. 2 results in Eq. 3. Here, the `wanted` object is a sub-object of the `full` object, where `extra` is any sub-object of `full` not part of `wanted`.

The derived equation (3) can be used to calculate the CoM of the convex hull using the same approach as the use of division into tetrahedrons to calculate the volume of a mesh. Instead of summing signed sub-volumes, sub-CoMs weighted by the corresponding tetrahedron's volume are summed. The tetrahedrons are the sub-objects, and the sign of each tetrahedron decides whether it belongs to the `wanted` or the `extra` sub-object.

3.3 Occlusion

While having information about the entire object is useful, it is not realistic. A more realistic situation would be to only have points on the object gathered from a single point of view. This introduces occlusion of a large part of the object, whose xyz-points are now missing from the data.

Volume

When looking at an object from a single point of view, we generally can see about half of the object. In the case of a depth camera, this would result in a partial set of points. The back half of the sphere, which cannot be seen by the camera, is an exact mirror of the front half. This suggests that multiplying the volume of the front half by a factor of 2 results in the volume of the full mesh.

The bounding box and convex hull approaches used before could be used to estimate the volume from occluded data. Because a single viewpoint only sees about half of the object, a convex hull around the partial set of points would generally result in a mesh consisting of half of the object. To get an estimation of the volume of the full object, we multiplied the volume of the convex hull by a factor of 2. As for the AABB and OBB, we decided not to multiply these by any factor. Since the bounding boxes are able to wrap significantly less tightly around the points, both are likely to contain a significant amount of volume which in reality is not part of the original object. We theorised that this extra volume could compensate for the missing data.

Center of Mass

Estimating the CoM from partial data is considerably more complex, as this deals with a point in 3D space and not a single scalar value. However, the same principle of only being able to see about 50% of the object applies here as well.

As theorised before, the CoM of an object generally lies in the center of the object. Since both AABBs and OBBs take up extra volume, which might account for the missing volume not seen by the camera, the same could hold for the CoM. This especially holds for the AABB, since it orients its box with the x, y, and z axis instead of fitting as tightly around the points as possible. Depending on the viewing angle, this bounding box might still encapsulate the occluded part of the object, thus having its center near the true CoM of the full object.

Since the convex hull wraps around the partial set of points as tightly as possible, this theory was less likely thought to work. However, assuming about half of the object can be seen, the other occluded half of the object was theorised likely to resemble the front half. Mirroring this front half and combining both would result in an estimation of the full object. To then get the CoM of the full object, we averaged the CoMs of the front and mirrored half. Drawing this situation, a realisation could be made. The true CoM is simply the CoM of the front half, projected onto the mirror plane. To position the plane at the right distance from the camera, we put its origin at the point furthest away from the camera. This should put the plane at the back side of the front half.

However, to calculate the CoM of the convex hull, correct surface normals of the mesh are required. We were not able to establish a reliable method to calculate these surface normals. Therefore, instead of using the CoM, we had to use the average point of all the vertices of the hull.

4 Implementation Details

This section describes how the approaches mentioned in Section 3 were implemented. Additionally, it provides a description of the virtual depth camera used to generate partial views of objects.

The Point Cloud Library (PCL) offers most of the functionality needed to implement the methods we used to estimate the mass and Center of Mass (CoM) of an object. PCL does most of these calculations on PointCloud[1] objects, which are a set of points consisting in our case of 3D points. The 3D objects used to

To obtain the Axis-Aligned Bounding Box (AABB) and Oriented Bounding Box (OBB), PCL's MomentOfInertiaEstimation[2] class can be used. From a point cloud it provides the minimum and maximum point of the computed AABB, and the relative minimum and maximum point, the position, and the rotation of the computed OBB. Based on the min- and max-points of the bounding boxes it is possible to find the length, width and height of the boxes. From these, the

[1]https://pointclouds.org/documentation/classpcl_1_1_point_cloud.html

[2]https://pointclouds.org/documentation/classpcl_1_1_moment_of_inertia_estimation.html

volume of the box can be calculated. The center of each box can be obtained by averaging the min- and max-point.

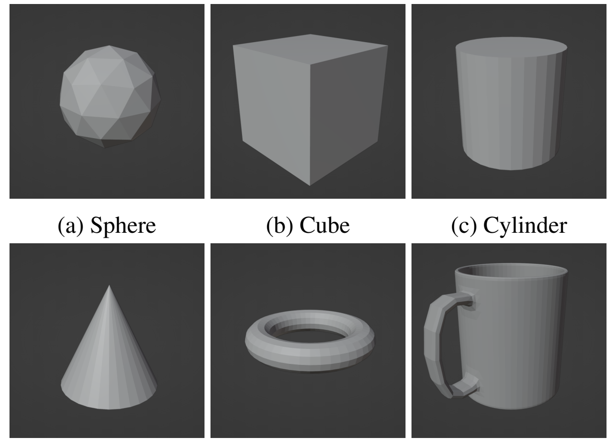
PCL's ConvexHull[3] class provides an implementation to calculate the convex hull of a point cloud. The implementation uses the QHull library's qconvex[4] class, which implements the QuickHull algorithm [4]. It provides a mesh of the calculated convex hull, consisting of a set of triangular faces those 3 vertices reference the points in the input point cloud. This method can be applied to both complete objects and partial views of objects

From this mesh, the volume can be calculated by iterating over the faces, a tetrahedron consisting of the triangular face and the reference point (0,0,0) is created. The signed volume of each tetrahedron can then be calculated and added to the total resulting volume of the mesh. A similar method can be used to calculate the CoM of the mesh. Again, iterating over the faces a tetrahedron consisting of the triangular face and the reference point (0,0,0) is created. The CoM of each tetrahedron can be calculated by taking the average of its 4 vertices, after which it is multiplied by its signed volume and added to the resulting CoM of the mesh. To obtain the actual CoM of the mesh the previous result is divided by the total volume of the mesh.

However, to obtain the sign of a tetrahedron's volume, correct surface normals are required. Since we were not able to establish a reliable method to calculate these, we resorted to using the average point of all the vertices of the mesh. PCL's CentroidPoint[5] class provides an implementation to calculate this average point.

Estimating the CoM from partial data using the convex hull approach requires a mirroring plane perpendicular to the direction of the camera. This plane can be described by a origin point on the plane, and the plane's normal. The reference point is taken as the point furthest away from the camera, and the normal is equal to the camera direction. To project the convex hull's CoM onto the plane, first, the distance d from the plane to the CoM is calculated from the plane origin o, the CoM c, and the plane normal $\overrightarrow{n}$ as $d = (c - o) \cdot \overrightarrow{n}$. Then the projected point p can be obtained with $p + d\overrightarrow{n}$. Eigen[6] provides the required Vector3 object and vector operation implementations to quickly perform the required calculations.

Yue Chen [6] implements a virtual depth camera in Unity[7], which could be used to generate partial views of objects. This depth camera shoots out rays into the scene, calculating at which xyz-coordinate each ray first intersects an object in the scene, and exports all hit points to a `.pcd` file. As input, it takes the resolution and field of view of the camera, as well as other parameters not relevant to this work. As we were required to generate partial views from multiple angles, and at multiple resolutions, some basic modifications were implemented by us.



(a) Sphere (b) Cube (c) Cylinder

(d) Cone (e) Torus (f) Mug

Figure 3: The 3D objects used in the experiments.

Object	Volume (cm^3)	AABB Dimensions (x-y-z)
Sphere	3658.7121	19 - 20 - 20
Cube	8000.0	20 - 20 - 20
Cylinder	6242.8903	20 - 20 - 20
Cone	2080.9634	20 - 20 - 20
Torus	1174.7357	25 - 5 - 25
Mug	1285.2886	27.6 - 23 - 19.9

Table 1: True volume and Axis-Aligned Bounding Box (AABB) dimensions of the models used in the experiments. See Figure 3.

5 Results

This section lays out the experimental setup and procedure used to generate results, and presents the results that were generated.

5.1 Experimental Setup

To generate our results, multiple experimental setups were used. As well as an input set consisting of 6 3D objects to perform these experiments on. These objects in Figure 3, as well as their true volume and Axis-Aligned Bounding Box (AABB) dimensions. These objects were chosen to range from high levels of symmetry, such as a Sphere, to lower levels of symmetry, such as a Cone. Besides this, a Torus and a Mug were chosen to see the impact of holes and concave features on objects. These models, except for the Mug, were created by us in Blender[8]. The Mug model was created by 'afferu' and distributed for free use on Sketchfab[9]. Using Blender the true volume and Center of Mass (CoM) were extracted, to allow for a direct comparison of the estimations against the true values.

To get a better insight into the impact of the resolution of the depth camera on both volume and CoM accuracy, a setup was created allowing us to generate partial views of objects at a range of resolutions. This setup makes use of a modified

[3] https://pointclouds.org/documentation/classpcl_1_1_convex_hull.html

[4] http://www.qhull.org/html/qconvex.htm

[5] https://pointclouds.org/documentation/classpcl$_{1_1c}entroid_point.html$

[6] https://eigen.tuxfamily.org/index.php?title=Main_Page

[7] https://unity.com

[8] https://www.blender.org/

[9] https://sketchfab.com/3d-models/blender-mug-bad3565a215a4795a119973e6816df8a

(a) Camera A (b) Camera B (c) Camera C

Figure 4: The camera angles used in the experiments. A mug object is used as an example to clarify the positions of the cameras relative to the objects.

	Camera A	Camera B	Camera C
Position	(0,0,-50)	(-15,15,-45)	(-30,30,-30)

Table 2: The XYZ-coordinates of the camera positions from which partial views of the test objects were generated.

version of the depth camera implemented by Yue Chen [6]. The resulting `.pcd` files could then be used as input to the respective volume and CoM estimation functions.

To generate volume and CoM estimations from complete objects, the models in the input set were exported to `.pcd` files. These could then be used as input to the respective volume and CoM estimation functions.

As for estimations from occluded data, a partial view of each object was generated from a single viewing angle and a set resolution. Intermediate results on volume estimation over a range of resolutions with an aspect ratio of 16:9 from '16x9' to '1920x1080' showed that generally, resolutions higher than '640x360' did not cause a significant change in results. Besides that, the time it takes to generate partial views is directly proportional to the total pixel count of the camera, ranging from $\tilde{3}$ minutes for a resolution of '640x360' to over 30 minutes for a resolution of '1920x1080'. These facts combined made us choose a resolution of '640x360' to generate the required partial views from multiple camera angles with sufficient accuracy, while not taking too much time. These results can be found in Figure 5, 6, 7, 8, 9, and 10. As for the Field of View (FoV) of the camera, a vertical FoV of $43\deg$ was chosen to match the Xbox Kinect. While this device does have a different aspect ratio, this would not have any impact on the results. Using the Kinect's 4:3 aspect ratio would only decrease the width of the camera view, which in our case does not change the results.

3 camera angles were used to generate partial views, whose views are visualised in Figure 4. The positions of these cameras can be found in Table 2. The cameras were oriented towards the origin, which generally aims them towards the center of the objects.

As the evaluation method for the performance of the volume estimations, the estimated volume as a factor of the true volume was chosen. As for evaluating the performance of the CoM estimations we chose the L2 norm, or the Euclidian distance from the true CoM to the estimated CoM.

5.2 Volume

Table 3 shows the results for volume estimation from complete data, using the Axis-Aligned Bounding Box (AABB), Oriented Bounding Box (OBB), and convex hull approach.

Object	AABB error	OBB error	Convex Hull error
Sphere	2.08	2.07	1.00
Cube	1.00	1.00	1.00
Cylinder	1.28	1.28	1.00
Cone	3.84	3.84	1.00
Torus	2.66	2.66	1.89
Mug	9.86	10.35	6.51

Table 3: Volume estimation results from a fully known mesh using the Oriented Bounding Box (OBB) and Convex Hull method. The error is defined as the estimated value relative to the true value. An error of 1.0 meaning a perfect estimation, and 2.0 meaning the estimated volume is twice as large as the true volume.

These results were generated from the complete mesh of the objects directly exported from Blender.

Table 4 shows the results for volume estimation from partially occluded data, using the Axis-Aligned Bounding box (AABB), Oriented Bounding Box (OBB), and convex hull approach. The partial views clouds were generated from 3 camera angles, see Figure 4, using the modified depth camera.

In both tables, the error is defined as the estimated volume as a factor of the true volume. An error of 1.0 meaning a perfect estimation, and 2.0 meaning the estimated volume is twice as large as the true volume. For readability, the values are rounded to 2 decimals.

5.3 Center of Mass

Table 3 shows the results for Center of Mass (CoM) estimation from complete data, using the Axis-Aligned Bounding box (AABB), Oriented Bounding Box (OBB), and convex hull approach. These results were generated from the complete mesh of the objects directly exported from Blender.

Table 4 shows the results for CoM estimation from partially occluded data, using the Axis-Aligned Bounding box (AABB), Oriented Bounding Box (OBB), and convex hull approach. The partial viewpoints were generated from 3 camera angles, see Figure 4, using the modified depth camera.

In both tables, the error is defined as the L2 norm. An error of 0 thus meaning a perfect estimation. For readability, the values are rounded to 2 decimals.

5.4 Resolution

Figure 5, 6, 7, 8, 9, and 10 plot the volume estimation results using the Axis-Aligned Bounding box (AABB), Oriented Bounding Box (OBB), and convex hull approach, over a range of resolutions. The partial viewpoints were generated from camera angle C (4c), over a range of 16:9 resolutions from '16x9' to '1920x1080', using the modified depth camera. The error is defined as the estimated volume as a factor of the true volume. An error of 1.0 meaning a perfect estimation, and 2.0 meaning the estimated volume is twice as large as the true volume.

Figure 5, 6, 7, 8, 9, and 10 plot the volume estimation results using the Axis-Aligned Bounding box (AABB), Oriented Bounding Box (OBB), and convex hull approach, over a range of resolutions. The partial viewpoints were generated

Object	Camera	AABB error	OBB error	Convex Hull error
Sphere	A	1.01	1.07	0.97
	B	1.37	1.06	0.93
	C	1.46	1.11	0.94
Cube	A	0.00	0.00	0.00
	B	1.00	2.65	1.62
	C	1.00	2.68	1.65
Cylinder	A	0.46	0.46	0.67
	B	1.26	1.42	1.37
	C	1.28	1.51	1.40
Cone	A	1.83	1.46	0.72
	B	2.49	2.00	1.10
	C	3.17	2.30	1.24
Torus	A	0.95	0.95	1.22
	B	2.15	2.25	2.90
	C	2.23	2.36	2.76
Mug	A	5.67	5.00	5.98
	B	9.59	10.62	8.98
	C	9.64	16.49	9.98

Table 4: Volume estimation results from a partial view of an object. Partial views were generated from 3 camera angles. Specifics about the camera angles can be found in Table 2. A visualisation of these camera angles can be found in Figure 4. The error is defined as the estimated value relative to the true value. An error of 1.0 meaning a perfect estimation, and 2.0 meaning the estimated volume is twice as large as the true volume.

from camera angle C (4c), over a range of 16:9 resolutions from '16x9' to '1920x1080', using the modified depth camera. The error is defined as the estimated volume as a factor of the true volume. An error of 0 thus meaning a perfect estimation.

6 Discussion

This section discusses the results presented in Section 5, and relates these results to previous works.

6.1 Volume

Looking at Table 3, where the accuracy of three approaches to estimating volume from complete data are presented, we see that the convex hull approach performs extremely well for the first four objects. For the other 2 objects, however, a large error of 1.89 and 6.51 is shown. This can be explained by the fact that the Torus contains a large hole through the middle, and the Mug features a thin handle containing a hole hole protruding from the side, and a large concave dent. As the name suggests, a convex hull is unable to wrap tightly onto these concave features.

Comparing this to the results presented by Thomas Baars [2], the convex hull approach presented in this work provides a promising improvement for objects without concave features. The use of a convex hull instead of reconstructing the mesh using a greedy triangulation algorithm seems to solve the issue with gaps in the resulting mesh.

As for volume estimations from a partial view of the objects, Table 4 shows a similar pattern. While not as accurate, the estimated volume of the first four objects is still within a factor of 2 of the true volume. Our theory about being able to see about half of the object from a single view, thus having to double the convex hull's volume does seem to hold some truth. For the Sphere, the error is only 6%. For the other objects, it can be observed that the more sharp angles the object has, the greater the error becomes, going from 0.94 to 1.24, to 1.40, to 1.65 for the Sphere, Cone, Cylinder, and Cube respectively.

For the other two objects, however, the errors become even larger than those estimated from complete data. While the convex hull approach does not look viable for objects with concave features, comparing the two tables does suggest that gathering multiple viewing angles of an object is likely to result in a more accurate estimation.

Object	AABB error (cm)	OBB error (cm)	Convex Hull error (cm)
Sphere	0.00	0.00	0.00
Cube	0.00	0.00	0.00
Cylinder	0.00	0.00	0.00
Cone	5.00	5.00	4.39
Torus	0.00	0.00	0.00
Mug	2.73	2.47	2.27

Table 5: Center of Mass (CoM) estimation results from a fully known mesh using the Oriented Bounding Box (OBB) and Convex Hull method. The error is defined as the L2 norm.

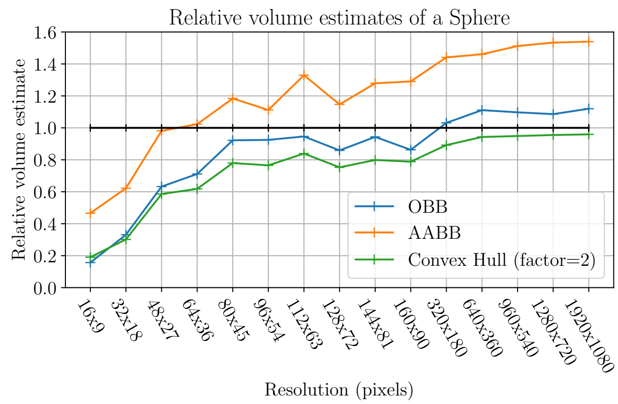


Figure 5: Volume estimation results from the partial view of a Sphere (3a), generated using a virtual depth camera from camera angle C (4c) over a range of 16:9 resolutions.

6.2 Center of Mass

For the estimation of the Center of Mass (CoM), Table 5 shows promising results for the estimation from complete data. Four of the objects resulted in a perfect estimation using all three approaches. While the two other objects, the Cone and the Mug, show an error greater than 0, this error is consistent for each object. This suggests that all three methods are comparatively viable approaches. However, out of the three, the convex hull approach still performed best. Besides that,

Object	Camera	AABB error (cm)	OBB error (cm)	Convex Hull error (cm)
Sphere	A	5.03	5.03	0.18
	B	2.79	4.62	0.69
	C	1.87	4.29	1.07
Cube	A	10.00	10.00	10.04
	B	0.00	2.10	9.42
	C	0.00	5.79	5.85
Cylinder	A	6.28	6.28	2.58
	B	0.10	1.68	6.38
	C	0.03	4.94	4.77
Cone	A	7.02	4.24	1.48
	B	5.88	3.39	2.10
	C	5.10	3.09	4.52
Torus	A	7.85	7.85	3.22
	B	1.32	1.31	9.10
	C	0.91	0.67	8.64
Mug	A	6.90	7.03	2.30
	B	5.59	4.80	5.01
	C	5.61	4.74	4.98

Table 6: Center of Mass (CoM) estimation results from a partial view of an object. Partial views were generated from 3 camera angles. Specifics about the camera angles can be found in Table 2. A visualisation of these camera angles can be found in Figure 4. The error is defined as the L2 norm. An error of 0 thus meaning a perfect estimation.

comparing the results to the dimensions of the objects in Table 1, the estimated CoMs appear to still lay well within the bounds of the objects.

In Table 6 the results for estimating the CoM from a partial view of the object are presented. For each object, it shows a larger error compared to when the full data is available. This again suggests that gathering views of the objects from multiple angles will improve the accuracy of the estimated CoM. For all the approaches, the error lies well within the bounds of the object. The only big outlier being the Mug. Interestingly, camera angle A (4a) seems to perform the best. The opposite was hypothesised, since this angle generally shows the least of the objects' features. Camera angle C (4c) on the other hand shows more features of the objects, but performs the worst.

The CoM estimations from both complete and partial data for objects with concave features can likely be improved with the use of a concave hull, since it is able to capture these concave features much better.

6.3 Resolution

As for the impact of the resolution of the camera, the volume estimation error in Figures 5, 6, 7, 8, 9, and 10 clearly stabilise at a resolution of '640x360' or higher. Thus it can be said that a resolution higher than this will not bring significant improvements. However, these results are highly dependent on other factors. These factors being the size of the objects, the distance from the camera to the objects, and the field of view of the camera. Therefore comparing the accuracy of the volume estimates against the number of points on the object would be a better metric.

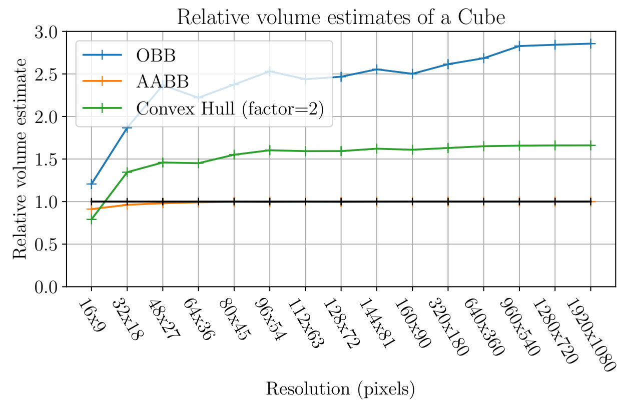


Figure 6: Volume estimation results from the partial view of a Cube (3b), generated using a virtual depth camera from camera angle C (4c) over a range of 16:9 resolutions.

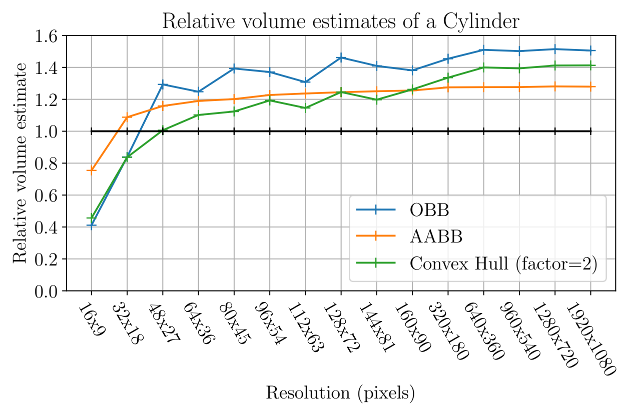


Figure 7: Volume estimation results from the partial view of a Cylinder (3c), generated using a virtual depth camera from camera angle C (4c) over a range of 16:9 resolutions.

The results for estimating the CoM over a range of resolutions in Figures 5, 6, 7, 8, 9, and 10, show a much different result. The estimated CoM using the convex hull approach is highly inconsistent, jumping back and forth between high and relatively low errors, and not having a clear resolution at which the result stabilises. On the other hand, both the AABB and OBB approaches show fairly promising results with an estimated CoM well within the bounds of the objects. The naive AABB approach generally performs the best. However, this seems to depend on the shape of the objects, as cube-like objects show a clearly more accurate result. As for the impact of the resolution of the camera, this looks to be very small. With the exception of the Sphere, the estimated CoMs do not differ by more than 3 throughout the range of resolutions. However, higher resolutions do result in a higher accuracy.

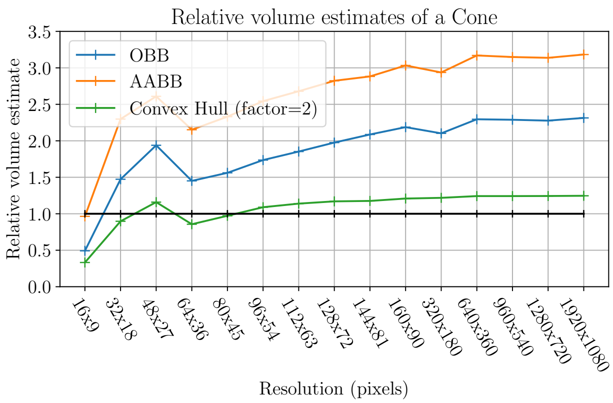


Figure 8: Volume estimation results from the partial view of a Cone (3d), generated using a virtual depth camera from camera angle C (4c) over a range of 16:9 resolutions.

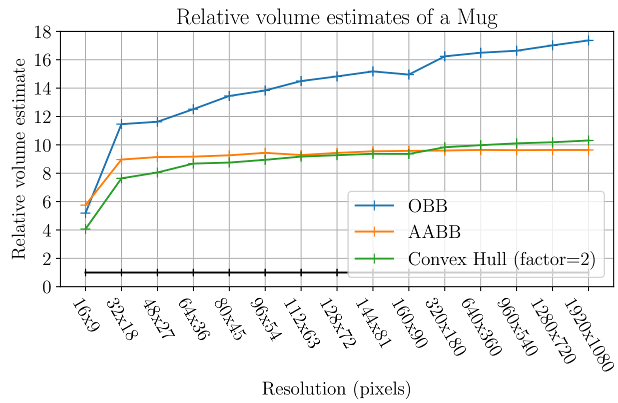


Figure 10: Volume estimation results from the partial view of a Mug (3f), generated using a virtual depth camera from camera angle C (4c) over a range of 16:9 resolutions.

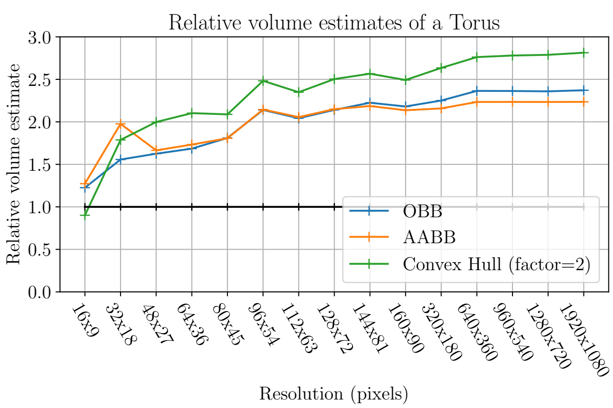


Figure 9: Volume estimation results from the partial view of a Torus (3e), generated using a virtual depth camera from camera angle C (4c) over a range of 16:9 resolutions.

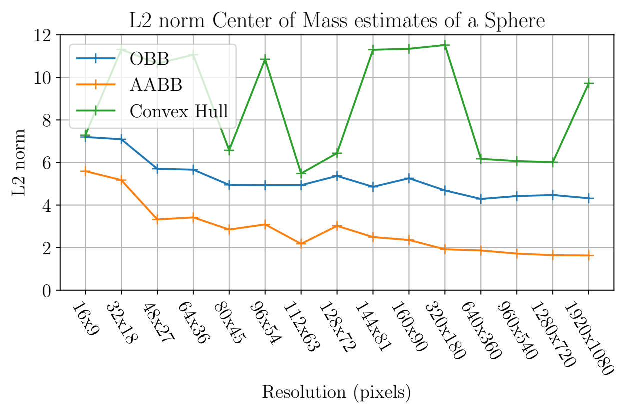


Figure 11: Center of Mass estimation results from the partial view of a Sphere (3a), generated using a virtual depth camera from camera angle C (4c) over a range of 16:9 resolutions.

7 Responsible Research

This section discusses the ethical challenges that were encountered.

Doing research in a responsible manner means proving the ability for others to reproduce our experiments, both to verify our results and compare results generated at a later stage using alternative approaches. Therefore, we aimed to provide sufficient details on the implementation of our approaches to both volume and Center of Mass estimation. Both through providing step-by-step instructions on our own implementation, and references to any outside libraries we used.

As for the use of the work of others, with the exception of the Mug (3f), all data in the form of 3D objects were created by us. The mug we used was acquired from Sketchfab[10], which was made available under the Attribution-ShareAlike 4.0 International[11] licence. This license allows for the free use of this object.

[10]https://sketchfab.com/3d-models/blender-mug-bad3565a215a4795a119973e6816df8a

[11]https://creativecommons.org/licenses/by-sa/4.0/legalcode

8 Conclusions and Future Work

This section lays out the conclusions drawn from the results and whether they provide answers to the questions we stated in Section 1, as well as listing possible future improvements.

In Section 1 we stated a number of questions which we aimed to answer through this work.

On the question "What techniques can be used to estimate the mass of an object", while not being able to provide a definitive answer, it can be concluded that the approach using a convex hull provides a clear improvement over the approaches explored previously [2]. However, on objects with concave features further improvements are necessary. These improvements might be achieved with the use of a concave hull.

As for the question "What techniques can be used to estimate the Center of Mass of an object", all three approaches work well for the range of objects which we experimented on, providing a viable method to estimate the Center of Mass

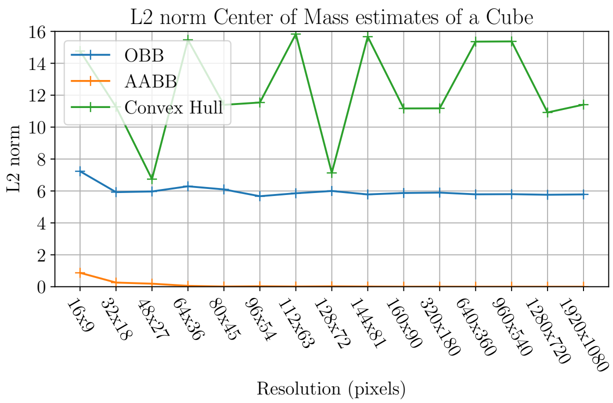


Figure 12: Volume estimation results from the partial view of a Cube (3b), generated using a virtual depth camera from camera angle C (4c) over a range of 16:9 resolutions.

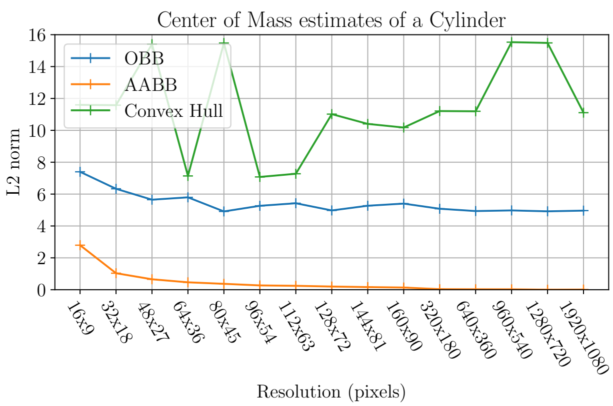


Figure 13: Volume estimation results from the partial view of a Cylinder (3c), generated using a virtual depth camera from camera angle C (4c) over a range of 16:9 resolutions.

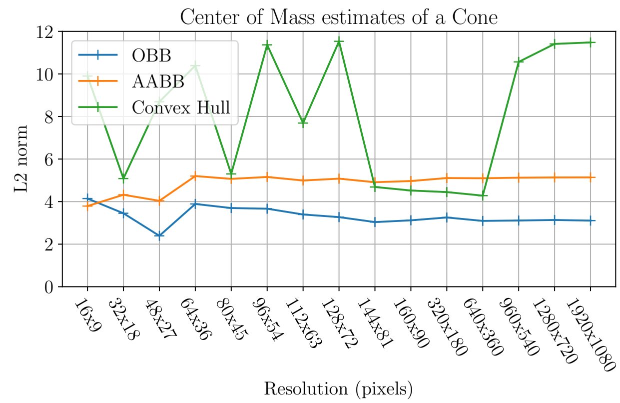


Figure 14: Volume estimation results from the partial view of a Cone (3d), generated using a virtual depth camera from camera angle C (4c) over a range of 16:9 resolutions.

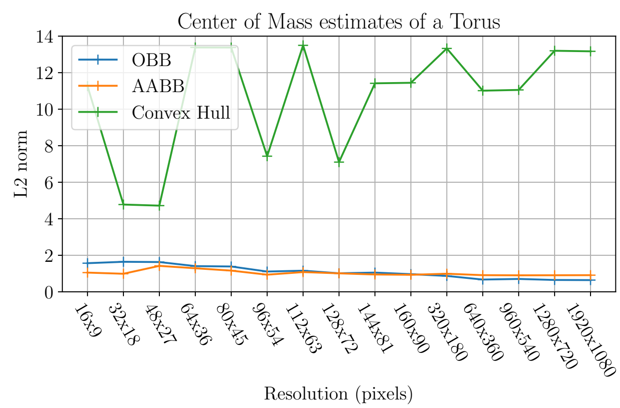


Figure 15: Volume estimation results from the partial view of a Torus (3e), generated using a virtual depth camera from camera angle C (4c) over a range of 16:9 resolutions.

(CoM) of an object when a point cloud of the entire object is available. However, here as well, even more accurate estimations may be achieved with the use of convex hulls.

We also asked, ”Are these estimates still accurate when part of the object is occluded”. We explored the situation where only a single view of the object was available. Results on CoM estimation showed that, while the presented approaches still provided viable results, the error did increase along the board. We thus conclude that, while not absolutely necessary to have knowledge of the complete object, gathering views from multiple angles will very likely increase the accuracy of the estimated CoM. The impact of occlusion on the estimated volume turned out to be significantly higher. While somewhat viable for objects with a fairly high level of symmetry, other objects with very little symmetry as well as concave features resulted in large errors. Gathering views from multiple angles therefore will improve the result, however, a larger issue lies with how to deal with holes and other concave features in objects.

Finally, we asked ”How does the resolution of sensors impact the accuracy of the estimations”, which we explored by estimating both the volume and CoM of objects from partial views generated at a range of resolutions. From the results gathered we can conclude that for volume estimation, while a resolution of ’640x360’ is sufficient, higher resolutions do provide slightly more accurate estimates. For CoM estimation using the Axis-Aligned and Oriented Bounding box approach, the resolution seemed to have an even smaller impact, where even lower resolution depth cameras would suffice. The convex hull approach on the other hand gave very inconsistent estimates which did not stabilise even at higher resolutions. Experiments using resolutions higher than ’1920x1080’ will be necessary to see how this approach behaves.

We identified two main items on which further research is required. Firstly, we were unable to use the true CoM of the convex hull, and had to resort to using the average of the mesh’s vertices. It would be worth investigating whether using the actual CoM results in more accurate CoM estimates. Secondly, we identified that the convex hull approach is sig-

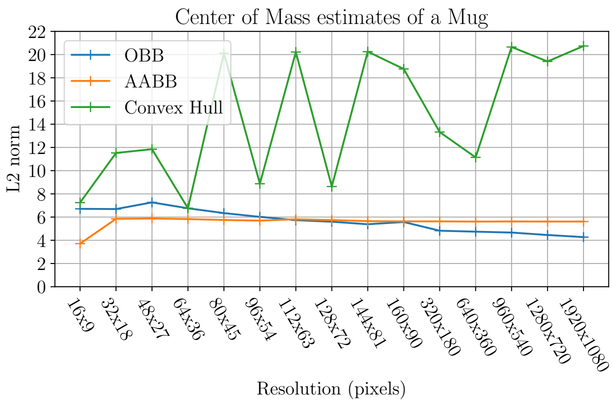


Figure 16: Volume estimation results from the partial view of a Mug (3f), generated using a virtual depth camera from camera angle C (4c) over a range of 16:9 resolutions.

nificantly less accurate for objects with holes and other concave features. Thus we find the use of a concave hull to be worth researching as well.

References

[1] Tamas Aujeszky, Georgios Korres, Mohamad Eid, and Farshad Khorrami. Estimating weight of unknown objects using active thermography. *Robotics*, 8(4):92, 2019.

[2] Thomas Baars. Estimating the mass of an object from its point cloud for tactile internet. 2022.

[3] Murat O Balaban, Gülgün F Ünal Şengör, Mario Gil Soriano, and Elena Guillén Ruiz. Using image analysis to predict the weight of alaskan salmon of different species. *Journal of food science*, 75(3):E157–E162, 2010.

[4] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22(4):469–483, dec 1996.

[5] Timothy M Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete & Computational Geometry*, 16(4):361–368, 1996.

[6] Yue Chen. Tracking physics: A virtual platform for 3d object tracking in tactile internet applications. Pre-print, 2023.

[7] Vineet Gokhale, Kees Kroep, Vijay S. Rao, Joseph Verburg, and Ramesh Yechangunja. Tixt: An extensible testbed for tactile internet communication. *IEEE Internet of Things Magazine*, 3(1):32–37, 2020.

[8] Eleonora Grilli, Fabio Menna, and Fabio Remondino. A review of point clouds segmentation and classification algorithms. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:339, 2017.

[9] Oliver Holland, Eckehard Steinbach, R. Venkatesha Prasad, Qian Liu, Zaher Dawy, Adnan Aijaz, Nikolaos Pappas, Kishor Chandra, Vijay S. Rao, Sharief Oteafy, Mohamad Eid, Mark Luden, Amit Bhardwaj, Xun Liu, Joachim Sachs, and José Araújo. The ieee 1918.1 "tactile internet" standards working group and its standards. *Proceedings of the IEEE*, 107(2):256–279, 2019.

[10] F. P. Preparata and S. J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Commun. ACM*, 20(2):87–93, feb 1977.

[11] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *2011 IEEE international conference on robotics and automation*, pages 1–4. IEEE, 2011.

[12] Şakir Taşdemir, Murat Yakar, Abdullah Ürkmez, and Şeref İnal. Determination of body measurements of a cow by image analysis. In *Proceedings of the 9th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing*, pages V–8, 2008.

[13] J.P. Verburg, H.J.C. Kroep, V. Gokhale, R. Venkatesha Prasad, and V. Rao. Setting the yardstick: A quantitative metric for effectively measuring tactile internet. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 1937–1946, 2020.

[14] Haocheng Yang. Acquiring material properties of objects for tactile simulation through point cloud scans. 2022.

[15] Yan Yang and Guanghui Teng. Estimating pig weight from 2d images. In *Computer And Computing Technologies In Agriculture, Volume II: First IFIP TC 12 International Conference on Computer and Computing Technologies in Agriculture (CCTA 2007), Wuyishan, China, August 18-20, 2007 1*, pages 1471–1474. Springer, 2008.

[16] Yong Yu, Kenro Fukuda, and Showzow Tsujio. Estimation of mass and center of mass of graspless and shape-unknown object. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 4, pages 2893–2898. IEEE, 1999.