Limb-Based Shape Modeling: A Demonstration on Itokawa

Dahlia A. Baker* and Jay W. McMahon†

The need for computationally efficient shape modeling techniques is required for the autonomy of small body exploration missions. This limb-based method focuses on extracting edge information from images on approach and building a low-to-medium resolution shape model from multiple viewing angles about the body. This approach was developed with shape from silhouette concepts and was previously tested on virtual shapes and viewing geometries. In this paper, we aim to use the same approach on simulated images captured with Blender to replicate a mapping orbit. By choosing the asteroid Itokawa, this study tests the proposed method’s response to concavity about the surface, a variety of viewing angles and resolutions, and camera geometry. It has been observed that the density of the final solution depends on the selection of images used in the mapping algorithm, and that the capability to resolve the whole body to an acceptable threshold for navigation requires analysis along several axes of rotation.

I. Introduction

Most small-body exploration missions rely on optical data downlinked to Earth to complete precise body-relative navigation. A valuable outcome of this process is a high-resolution shape model and a record of associated landmarks, which are then used to correlate the current state of the spacecraft to the model of the body. Current industry methods implemented to build a shape model are stereophotoclinometry (shape from shading), and stereophotogrammetry (shape from motion) [1]. These approaches require data downlink and ground processing to obtain reference data for navigation solutions, and the result is a high-resolution mapping of the visible surface of body. However, for orbiting purposes, high resolution models are not necessary and cannot be easily calculated and manipulated. Low-resolution on-board shape modeling can be an effective solution to compromise between computational power of spacecraft systems and navigation requirements for deep space asteroid exploration missions, and it would further enable autonomy of the spacecraft. This is especially true for the case of proximity navigation and landing schemes, where trajectory decisions must be made in real time. Many different approaches to this task have been proposed, such as extracting a shape from the silhouette of the target body [2]. This method would sample the observed outer edge of the volume at several different camera orientations and spacecraft attitudes and build a “visual hull”, which can be represented as a point cloud model. Each viewing orientation around the target body provides a updated contour which is then incorporated into the full shape solution. The method examined in this paper completes a similar task by sampling limb profiles of Blender [3] generated images of Itokawa in optimal lighting conditions and finding where the views intersect to solve for the hull [4]. Using an asteroid with particular concavities, we can test the limb-tracing algorithm and it’s response to the unique small body environment. This method was derived from the work done on mapping shapes of small satellites using limb and terminator profiles [5]. By sampling from the full contour of identified pixel locations of the limb every 5°, we can reduce the processing load required to build a suitable shape model on-board for navigating. The limb is discovered using an active contour algorithm in Matlab, and then sampled down and considered in the three dimensional bodycentric frame. Once the limb is extended in three dimensions for every view in the set of images provided, the limb rays are trimmed down based on heuristics defined for this shape from silhouette method. The trimmed rays go on to form the final identified surface points.

II. Limb Extraction and Tracing

We begin this process with a data set of images synthetically generated using Blender software. Creating a realistic but ideal environment, we can optimize the lighting conditions, apply assumptions about our camera, and have complete knowledge of the central pixel location on the body.

The first step is to load a 50,000 facet shape model of Itokawa, sourced from the Small Body Mapping Tool [6], and position the camera at a distance away. The camera focal length varies depending on the test case being considered, as does the distance from camera to center of body. The lighting source is placed directly behind the camera to simulate a 0°phase angle with the Sun. Our camera is set to the field of view and distance we desire for the case, and

---

*Graduate Research Assistant, Colorado Center for Astrodynamics Research, University of Colorado Boulder, 3775 Discovery Drive, Boulder, CO 80303

†Assistant Professor, Colorado Center for Astrodynamics Research, University of Colorado Boulder, 3775 Discovery Drive, Boulder, CO 80303
the shape model is rotated and positioned in the central view to the target latitude and longitude.

The points are converted to three dimensions using the pre-set or known camera distance. This conversion relies on the assumption that all points are the same distance away from the camera. For this calculation, the field of view of the image must be known in order to translate image distances in pixels to actual distances in kilometers. After processing the first image, each subsequent view is considered and rotated into the frame of the original view. This is to standardize the frame for the ray extensions defined in the next step of the algorithm so that their intersections can be calculated. Also, note that once edge points are found, the ray is constructed by extending the location in the positive and negative z direction, which is along \( \hat{u} \), the camera to body pointing vector. The coordinates of the ends of these lines are saved and processed with a shape fitting algorithm proposed by McMahon. [4]

![Fig. 1 1, 20, 200, and 750 iterations of a shrinking fit method for edge contour detection](image1)

![Fig. 2 Blender Image and Hayabusa Image](image2)

Beginning with knowledge of the location on the body of the point central to the camera view, our distance from camera to body, and the field of view in degrees of our camera, we can begin extracting the limb from the image. Limb refers to the most external edge of the body in question, or the radius of the body in a two-dimensional projection. In this study, we are testing our methods on idealized Blender generated images and will further this work to consider the actual Hayabusa data set on approach to Itokawa. As seen in Fig. 2, the blender images are comparable to images captured by the AMICA camera onboard Hayabusa, however, we avoid observing a terminator, and have access to the full limb profile from any angle of view.

As demonstrated in the sequence of images in Fig. 1, the active contour capability in Matlab was leveraged to identify the foreground or edge of our target body. This algorithm is an iterative shrinking fit method used to extract the desired edge without self-shadowing errors. This method begins with an estimate of a bounding rectangle and reduces the size until it finds the bright foreground against the background of space. An advantage of this approach is that shadows and locations of high contrast on the surface of the body aren’t mistaken for limb points and you can be insured that every sampled point is the outer-most limb. The result is a continuous, radially closed edge shape. This edge is then sampled at every 5 degrees and the point locations in (u,v) image space are recalculated with reference to the center of the image rather than the inherent image grid frame. Now that points are organized in the camera frame with the center of the body at the origin, we rotate them into the body-frame initialized with the first image’s geometry.

The points are converted to three dimensions using the pre-set or known camera distance. This conversion relies on the assumption that all points are the same distance away from the camera. For this calculation, the field of view of the image must be known in order to translate image distances in pixels to actual distances in kilometers. After processing the first image, each subsequent view is considered and rotated into the frame of the original view. This is to standardize the frame for the ray extensions defined in the next step of the algorithm so that their intersections can be calculated. Also, note that once edge points are found, the ray is constructed by extending the location in the positive and negative z direction, which is along \( \hat{u} \), the camera to body pointing vector. The coordinates of the ends of these lines are saved and processed with a shape fitting algorithm proposed by McMahon. [4]

![Fig. 3 Edge contour sampled every 5°](image3)
Fig. 4  Two adjacent planes formed from ray lines

---

III. Solving for Shape

The limb tracing method has found tangent rays in all of the provided views of the asteroid. Now it is necessary to describe the equation of each ray as a line in the $\hat{u}$ direction. We are assuming all views are far enough away from the target body that each ray is parallel. The equation of the lines is as follows for $i$ number of lines $l$ in each view:

$$ l_i = l_{i,0} + \eta L \hat{u} $$

where $l_{i,0}$ is the initial point of the ray, $\eta$ is a scale factor from 0 to 1 describing how far along the ray we have trimmed, and $L$ is the length of the pre-trimmed ray. For the sake of this investigation, this length $L$ has been standardized to 1km.

With a sample of $N_p$ edge points, we can describe $N_p$ number of planar quadrilaterals by connecting each adjacent edge point as seen in Fig. 4, and describing it’s surface normal direction with the following equation:

$$ \hat{n}_k = \frac{\hat{u} \times (l_{i+1,0} - l_{i,0})}{|\hat{u} \times (l_{i+1,0} - l_{i,0})|} $$

This surface normal definition becomes crucial for determining which intersection points lie on the body and which lie off the body being resolved. These planes become what we refer to as a "patch". We describe a set of patches for each view so that we have an equal number of edge points and number of patches per view. Now, comparing each pair of views, we search for intersections of the planes that contain each of the patches. As shown in Fig. 5, there are many intersections and this number increases with the number of views sampled. If the equations planes intersect, then we look to see if the finite patch intersects as well. If an intersection is discovered, we record it’s $\eta$ value, or percentage of how far along the ray it hits, along with the relationship between the normal vectors of the two intersecting planes. Evaluating the sign of the dot product of the two normal vectors will determine which intersections fall on the surface and which are external to the surface that we wish to resolve. Rays are trimmed to stretch between their two most central intersection points.

---

![Fig. 5 Itokawa sampled along equator: every 10°](image)

Once all intersections are calculated and rays are trimmed, we sample the remainder of the segments to get a population of points that are on the body. Every point is given an associated normal as an average of the normals from the planes that it’s parent ray was an edge to. These points and normals can then be used in a Poisson surface reconstruction algorithm to fit a final shape.

---

<table>
<thead>
<tr>
<th>Test Case</th>
<th># of Points</th>
<th>x (m)</th>
<th>y (m)</th>
<th>z (m)</th>
<th>Volume ($m^3$)</th>
<th>Mean Error (m)</th>
<th>Variance (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30°</td>
<td>4250</td>
<td>583.5</td>
<td>290.0</td>
<td>334.8</td>
<td>$2.393 \times 10^7$</td>
<td>21.82</td>
<td>.212</td>
</tr>
<tr>
<td>10°</td>
<td>6596</td>
<td>583.5</td>
<td>258.8</td>
<td>341.2</td>
<td>$2.045 \times 10^7$</td>
<td>19.03</td>
<td>.176</td>
</tr>
<tr>
<td>Truth Model</td>
<td>25,350</td>
<td>561.7</td>
<td>305.1</td>
<td>243.4</td>
<td>$1.779 \times 10^7$</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 1  Comparison of Shape from Silhouette Model Results to Truth Model
IV. Test Cases: Sampling Along Equator

A. Every 30°

This case uses data from an image span that sampled along a latitude of 0 degrees and longitudes from 0-180 degrees, sampled every 30 degrees apart. See Fig. 6.

B. Every 10°

This case uses data from an image span that sampled along a latitude of 0 degrees and longitudes from 0-180 degrees, sampled every 10 degrees apart. See Fig. 7.

The point cloud is the final result of the shape from limbs method proposed in this paper. All other steps taken to solve for a surface were implemented via Meshlab software methods and serve to give a preliminary picture of how accurate and reliable our result could be if used as a navigation solution. The product of applying Poisson surface reconstruction to the resulting point clouds serves to aid visually in identifying bias in our modeling algorithm. The mean error and variance described in Table 1 refers to the result of finding the Hausdorff distance between the resulting reconstructed surface and the comparison truth model. These distances represent the heat map overlaid on the mesh presented in the appendix.

V. Discussion

This investigation has discovered two results from the test cases processed. As seen in the previous section, each case resulted in a point cloud with a more resolved shape approximation along the X-Z plane than the X-Y plane. The prediction is that the concavity of the body was only resolved for along one axis of rotation, the variable longitude. By examining a constant latitude, we restricted the model’s consideration of the concavities along the other
dimensions. Also, by only sampling one hemisphere, we shielded the intersections available on the opposite half of the shape. This would explain the planar appearance of the errant extension on the lower section of the shape model.

When sampling along the same axis, it would be fair to assume that sampling closer views would result in intersection points closer to the body surface and a more accurate shape approximation than comparing views taken further apart. In this case, the difference between resolution in the 10° and 30° models is not immediately apparent. The 10° model gives a point cloud with 6596 points. Compared to the 4250 points in the 30° result, this model could be interpreted as better resolved and therefore more reliable as a mission data product. However, the aim of this work is to seek a low-resolution method. Seeing as the Hausdorff distances reported in the Appendix for both models are similar in error shape and magnitude, it may be more fruitful to investigate image spans in the realm of 30°.

VI. Conclusion

The methods proposed in this work have shown that we can develop a shape from silhouette approach that considers optical input and can resolve for some concavity in the target when provided with a sufficient number of views. However, we have constrained ourselves to the optimal problem of viewing the entire limb of a small body with the most desirable lighting conditions which result in a fully observable contour of the edge. In future work, we would like to utilize limb information regardless of full observability. This data, along with a visible terminator, has been shown to enhance the navigation solution on approach to a body [10]. As recognized in Gaskell et. al., additional data sources could also serve to enhance the shape solution. In the case of analyzing data from the Hayabusa mission, one possible route could be to integrate LiDAR ranging data into our model, ensuring that we estimate our camera-to-body distance as accurately as possible [1]. Overall, we wish to expand this work to several more test cases and further to incorporate real data which requires our method to be robust to the uncertain lighting conditions and position solutions that are associated with this approach.

References


A. Poission Surface Reconstruction

Reported in this section are the colorized Hausdorff distance between the reconstructed shape model and the comparison shape model used to generate the initial images. The distances are reported in kilometers in the histogram on the left of the model. The left image is the standalone mesh, and the right image is the mesh overlaid with the truth model for comparison. The maximum error occurs in the most concave sections of the model. It can be inferred that more observations at varied latitudes are required to resolve this section. Otherwise, the two models are similar in error magnitude and location and further analysis needs to be completed to evaluate their utility.

A. 30° span

B. 10° span