LIMB-BASED OPTICAL NAVIGATION FOR IRREGULAR BODIES

Andrew J. Liounis

Limb-based optical navigation refers to the process of extracting relative information from the illuminated limb of an observed body. These observations generally take the form of center-of-figure line-of-sight measurements and can be generated for both regular (well modeled by a tri-axial ellipsoid) and irregular bodies. Recent work has also provided the ability to extract a full three degree-of-freedom relative position estimate when the target is a regular body, but no work has been performed on extracting a full three degree-of-freedom relative position estimate for irregular bodies. In this paper, we develop a new technique for extracting the relative position between a camera and both regular and irregular bodies based on the process of limb scanning. The resulting algorithm is described in full, and its capabilities are demonstrated by processing synthetic images for a variety of body types.

INTRODUCTION

Optical navigation (OpNav) refers to the process of extracting relative state information between a camera and a target body from images. There are a number of common techniques that are used in OpNav with varying degrees of complexity and accuracy ranging from simple moment algorithms to surface feature tracking. One particularly popular grouping of techniques is known as limb-based OpNav, where only the illuminated horizon (or limb) of a body in an image is considered. These techniques are popular due to their strong blend of accuracy and low computational cost.

Modern limb-based navigation generally takes two distinct forms. One is limb-scanning, where a shape model of a target body is combined with an a priori estimate of the relative position between a camera and the body to render predicted scan lines of intensity. These predicted scan lines are then correlated with one dimensional scan lines extracted from the image to determine the observed location of the limb in the image. The location of the center-of-figure of the body in the image is then iteratively moved to minimize the residuals in a least squares sense.

The other limb-based OpNav form is ellipse matching, where the target body must be well described by a tri-axial ellipsoid. In this technique, limb points are extracted from an image using modern image processing sub-pixel edge detection methods. These edge points are then matched with the tri-axial ellipsoid model of the body and used to solve for a full three degree-of-freedom (3DOF) relative position estimate between the camera and the target (both bearing and range).

Both sets of the above techniques are powerful tools, but each falls short when it comes to considering bodies which are not well described by a tri-axial ellipsoid, such as the irregular bodies seen in asteroids and comets. The limb scanning techniques can be applied to these bodies, but they traditionally only solve for a bearing measurement, neglecting the valuable information about the range to the body that is available from the image. In addition, in the presence of large scale errors (either in the a priori range estimate to the body or in the shape model itself), the center-finding results quickly become degraded and biased. The ellipse matching techniques become less and less applicable the further a body’s shape deviates from a tri-axial ellipsoid, making them largely useless for many small bodies. In this paper we present a new technique that is capable of extracting 3DOF relative position estimates from both bodies that are well modeled by tri-axial ellipsoids and bodies that are not.
The rest of this paper is organized as follows. First, a brief review of current limb-based OpNav techniques is presented with a focus on some of the underlying theory that will be required to develop the new technique. Next, the new technique is discussed and described in detail. Finally, the new technique is applied to a range of sample imagery and shown to have strong performance.

**LIMB-BASED NAVIGATION REVIEW**

Prior work on limb-based navigation has fallen into two different approaches: limb scanning and ellipse matching. In this section we give a brief review of each of these styles of techniques while touching on the important concepts that will be required to describe the new technique.

**Limb Scanning**

Limb scanning has been in wide use since the early Voyager 2 encounter with Uranus. In this technique, limb locations in an image are determined by performing 1D correlations between predicted lines of intensities based on the *a priori* knowledge and extracted lines of intensities taken from the image of the body. The first step in this technique is to determine the theoretical surface points which will form the limb in the image based on the *a priori* knowledge of the scene.* Begin by assuming we know the tri-axial ellipsoid model of the target and the rotation matrix to get from the principal frame (where \( x, y, \) and \( z \) are aligned with the principal axes of the ellipsoid) to the camera frame. We can then show that any vector from the center of the ellipsoid falls on the surface of the ellipse if and only if

\[
\mathbf{p}_C^T A C \mathbf{p}_C = 1
\]  

where \( \mathbf{p}_C \) is a vector from the ellipse center to a point expressed in the camera frame, \( \mathbf{A}_C \) indicates a transpose, and

\[
\mathbf{A}_C = \mathbf{T}_C^P \begin{bmatrix} \frac{1}{\alpha^2} & 0 & 0 \\ 0 & \frac{1}{\beta^2} & 0 \\ 0 & 0 & \frac{1}{\gamma^2} \end{bmatrix} \mathbf{T}_C^P
\]  

with \( \alpha, \beta, \) and \( \gamma \) being the semi-axes lengths, \( \mathbf{T}_C^P \) being the rotation from the principal frame to the camera frame, and \( \mathbf{T}_C^P \) being the rotation from the camera frame to the principal frame. Further, we also know that any vector from the camera center to a limb point must be tangent to the surface, giving

\[
\mathbf{r}_C^T A_C \mathbf{p}_C = -1
\]

where \( \mathbf{r}_C \) is the vector from the camera center to the ellipsoid center expressed in the camera frame. Finally, we want the limb point that is on our scan plane defined by the scan center, \( \mathbf{c}_C \), and scan direction, \( \mathbf{s}_C \). From the equation of a plane, we know \( \mathbf{p}_C \) lies in the plane if and only if

\[
(\mathbf{c}_C \times \mathbf{s}_C)^T \mathbf{p}_C = -(\mathbf{c}_C \times \mathbf{s}_C)^T \mathbf{r}_C
\]

where \( \times \) indicates a cross product.

Equations (1), (3), and (4) give us three equations with three unknowns (the components of \( \mathbf{p}_C \)) which need to be solved for. Unfortunately, one of the equations is non-linear, but both analytic and numeric solutions exist, which we will not detail here. The end result is a predicted limb point \( \mathbf{p}_C \) in the camera frame.

With the predicted limb point in the camera frame, we can now determine the predicted limb location in our image. This requires us to map the 3D point to the corresponding 2D point in the image, which is performed using a camera model. There are many different types of camera models ranging from simple to complex, but in the end, they all accept an input 3D point or direction and output the corresponding 2D point, giving

\[
\mathbf{x}_I = f (\mathbf{p}_C + \mathbf{r}_C)
\]  

*In this discussion, we largely follow the discussion in References 1 and 2 and we will assume that the target body is well modeled by a tri-axial ellipsoid in order to simplify the steps, but the majority of the steps are directly applicable to irregular bodies as well.
where \( f(\bullet) \) is the chosen camera model and \( x_I \) is the 2D image point expressed in units of pixels. For these techniques, the particular camera model chosen matters little as long as it provides an accurate projection of points in the camera frame onto the image plane.

With the predicted limb location in the image we can now determine the measured limb location in the image. This is done with the following steps:

1. Sample the image intensity at the predicted limb location and a number of points along the scan line to both sides of the predicted limb location. This results in a 1D extracted intensity profile, \( i_e \).
2. Take the sample image points and project them onto the surface if possible (which typically involves a ray tracing algorithm or something similar).
3. For each sample point that strikes the surface, determine if that surface point is illuminated by the sun.
4. For each illuminated surface point, compute the expected intensity based off of the geometry of the observation, giving a 1D predicted intensity profile, \( i_p \). This is typically done with a bidirectional reflectance distribution function (BRDF), which accepts the incidence, emission, and phase angles as input and outputs a brightness value.
5. Perform a 1D cross correlation between \( i_e \) and \( i_p \) to determine the observed location of the limb in the image.

After determining the observed limb locations in the image for each scan line, we are left with corresponding predicted and observed limb locations in the image. Given the predicted and observed limb locations, we can now minimize the residuals between them by shifting the predicted limb locations with a least squares estimation process as discussed in Reference 2. This gives us the observed location of the center-of-figure of the body in the image.

**Ellipse Matching**

Ellipse matching refers to a number of techniques that have been developed over the last few years.\(^3\)–\(^11\) While there are a number of different approaches that have been developed, fundamentally, each of these techniques makes use of the fact that tri-axial ellipsoids project to a well defined geometry on an image, and thus relate the projected geometry with the 3D model to extract a full 3DOF relative position estimate from a single image. In this subsection, we briefly review the technique described in Reference 11, as the concepts are most applicable to the rest of this work.

When a tri-axial ellipsoid is projected through a gnomic projection onto a 2D plane, the resulting outline on the plane forms a 2D ellipse. Using this information, it becomes simple to determine the direction and range to a body in an image. First, we need to identify the limb points in the image. This is typically done by using an image processing technique to extract sub-pixel edges in an image, such as those described in References 11 or 12, but it could also be done using the limb scanning techniques described in the previous section if desired. Once the sub-pixel edges have been extracted, any points that do not lie on the sub-pixel limb must be discarded. This can either be done by a human operator or by one of the autonomous techniques described in Reference 11. Given the set of limb points in the image, we now need to project them into the camera frame from the image plane using the inverse camera model

\[
\hat{x}_{C_i} = f^{-1}(x_{I_i})
\]  

(6)

where \( f^{-1} \) is the inverse camera model and \( \hat{x}_{C_i} \) is the \( i^{th} \) unit vector in the camera frame corresponding to the \( i^{th} \) limb point \( x_{I_i} \) in the image.

Now, notice that we can decompose equation 2 into

\[
A_C = \left( T_P^C \right)^T Q^T Q T_P^C
\]  

(7)
where
\[ Q = \begin{bmatrix} \frac{1}{a} & 0 & 0 \\ 0 & \frac{1}{b} & 0 \\ 0 & 0 & \frac{1}{c} \end{bmatrix} \] (8)

Using this factorization, we can transform the unit vectors from the limb points into a special factorized space using
\[ x'_{C_i} = QT^C_i x_{C_i} \] (9)
\[ \hat{x}'_{C_i} = \frac{x'_{C_i}}{||x'_{C_i}||} \] (10)

where \( ||\cdot|| \) is the vector norm. In this factorized space, it can be shown that the relative position of the object according to the observations is found by solving the least squares problem given by
\[ \begin{bmatrix} \hat{x}'_{C_1}^T \\ \hat{x}'_{C_2}^T \\ \vdots \\ \hat{x}'_{C_n}^T \end{bmatrix} n = 1_{m \times 1} \] (11)

for \( n \) where \( 1_{n \times 1} \) is a \( n \times 1 \) vector of ones. We can then get the optimal estimate of the relative position in the camera frame by using
\[ r_C = -(n^T n - 1)^{-\frac{1}{2}} T^C_i Q^{-1} n \] (12)

where \( Q^{-1} \) is the inverse of \( Q \). This results in the optimal estimate of the relative position between the camera and the body based on the observed limb points in the image. Again, note that this is a 3D estimate, which includes both the bearing to the center-of-figure in the camera frame, and the range from the camera center to the center-of-figure of the body.

**RELATIVE POSITION FROM LIMB SCANNING**

In the previous section, we discussed two different uses for limb-based navigation. In limb scanning, the measured and predicted limb points are used to estimate just the center of the body in the image. In ellipse matching, we assume that we have a good model of the target body and use this information combined with the measured limb points to solve for the full 3D relative position between the camera and the target body. It is possible to take the idea of trusting the model and using it to estimate the relative position to the technique of limb scanning, as well. In this section, we describe how we can use limb scanning to get a relative position estimate by trusting the model of the target body for both tri-axial ellipsoids and for irregular bodies.

**Relative Position Estimation for Tri-axial Ellipsoids**

Reconsider the limb scanning discussion from the previous section. While traditionally, the limb measurements have been used to estimate the center-of-figure of the body in the image, and occasionally the properties of the body (such as the semi-axes), it is also possible to directly estimate the relative position between the camera and the center-of-figure of the body. All that we need to compute is the change in the limb locations in the image with respect to a change in the vector from the camera center to the center-of-figure. Using the chain rule, we can see that this is given by
\[ \frac{\partial x_I}{\partial r_C} = \frac{\partial f(x_C)}{\partial x_C} \frac{\partial x_C}{\partial r_C} \] (13)

where \( x_C = p_C + r_C \) is the vector from the camera center to the limb point expressed in the camera frame, \( \partial f(x_C)/\partial x_C \) is the partial derivative of the camera model with respect to a change in the input camera frame vector, and \( \partial x_C/\partial r_C \) is the partial derivative of the vector from the camera center to the limb point with respect to the relative position of the target body. The camera model partial derivative is dependent on
the camera model used and thus outside of the scope of this work.* The limb vector partial derivative is given by

$$\frac{\partial \mathbf{x}_C}{\partial \mathbf{r}_C} = \frac{\partial \mathbf{p}_C}{\partial \mathbf{r}_C} + \mathbf{I}_{3 \times 3}$$  \hspace{1cm} (14)$$

where $\frac{\partial \mathbf{p}_C}{\partial \mathbf{r}_C}$ is the partial derivative of the limb surface point with respect to the relative position of the target body and $\mathbf{I}_{3 \times 3}$ is the $3 \times 3$ identity matrix.

The partial derivative of the limb surface point with respect to the relative position of the target body can be found by simultaneously differentiating equations (1), (3), and (4) which results in the following system of 9 equations and 9 unknowns

$$\begin{bmatrix} 2 \mathbf{p}_C^T \mathbf{A}_C \\ \mathbf{r}_C^T \mathbf{A}_C \\ (\mathbf{c}_C \times \mathbf{s}_C)^T \end{bmatrix} \frac{\partial \mathbf{p}_C}{\partial \mathbf{r}_C} = \begin{bmatrix} \mathbf{0}_{1 \times 3} \\ -\mathbf{p}_C^T \mathbf{A}_C \\ -(\mathbf{e}_C \times \mathbf{s}_C)^T \end{bmatrix}$$  \hspace{1cm} (15)$$

where $\mathbf{0}_{1 \times 3}$ is a $1 \times 3$ row vector of zeros. The left hand side of (15) will always be a full rank matrix for any realistic observing conditions making the system of equations solvable, as is discussed in the appendix.

With the above partial derivatives in hand, it now becomes possible to estimate an update to the relative position vector using iterative least squares. For each iteration we must

1. Determine the observed and predicted limb locations based on the current best estimate of the relative position. This is the same set of steps described in the previous section.

2. Compute the Jacobian matrix defined in (13) using the current best estimate of the relative position for each limb point under consideration.

3. Update the relative position estimate using least squares estimation.

This means at each iteration we get new predicted limb locations, new refined observed limb locations, and a new relative position estimate between the camera and the target body.

**Relative Position Estimation for Irregular Bodies**

Let us now consider the case when the target body is not well modeled by a tri-axial ellipsoid and instead is represented by a triangular mesh (or any similar representation of a surface, such as spherical harmonics). The first change we need to make is a new way to determine the predicted limb points for each scan vector. Unfortunately, there is no simple analytic solution to determining the limb points for a body modeled with a triangular mesh. We can get a fairly quick numerical solution for the limb locations using a binary search algorithm coupled with ray tracing, as shown in Figure 1 and described below.\(^{13, 14}\)

1. Define an initial ray (position and direction), $\mathbf{r}_0$, in the camera frame that corresponds to the scan center vector, $\mathbf{c}_C$. This ray must be guaranteed to strike the surface of the target body at some point (which means that the scan center vector cannot be arbitrary).

2. Define a new ray, $\mathbf{r}_1$, in the camera frame that points in some direction different than the initial ray, but along the scan plane defined by $\mathbf{c}_C$ and $\mathbf{s}_C$. Choose this ray so that it is guaranteed to not strike the surface of the target body.

3. Define a third test ray, $\mathbf{r}_t$, halfway between $\mathbf{r}_0$ and $\mathbf{r}_1$ and test to see if it intersects the surface using ray tracing.

4. Update the rays as follows:

   - If $\mathbf{r}_t$ strikes the surface, then set $\mathbf{r}_0 = \mathbf{r}_t$ and leave $\mathbf{r}_1$ as is.

*We do note that this derivative is easily computed for most standard camera models.*
If $r_t$ does not strike the surface, then set $r_1 = r_t$ and leave $r_0$ as is.

5. Repeat steps 3 and 4 until convergence.

This technique will always be successful for any semi-regular body where the overall shape is convex. For more complicated bodies where portions of the overall shape are concave, there are some viewing geometries where there may be more than 1 limb along a scan line. In these cases, it generally suffices to adjust the initial $r_0$ and $r_1$ such that they enclose only one of the limbs, and then repeat the algorithm with $r_0$ and $r_1$ enclosing the other limb point. We discuss other degenerate cases in the upcoming subsection on limitations and workarounds.

![Figure 1](image.png)

**Figure 1**: For irregular bodies, potential limbs are identified using a binary ray trace search algorithm where the limb location is iteratively refined and defined as the point at which we transition from a ray which strikes the surface to a ray which does not strike the surface.

With the predicted limb locations in hand, all we need now is a method to compute the Jacobian for the limb points with respect to the relative position between the camera and the target object’s center-of-figure. Once again, there is no easy analytic solution to this problem, but we have two choices to approximate the solution. First, for most bodies, we can roughly approximate the surface as a tri-axial ellipsoid by generating a best-fit ellipsoid to the vertex data. In this case, we compute the partial derivative according to Eqs. (13)–(15) using a best-fit ellipsoid, but we still use the more precise limb identification technique described in the previous paragraph. For other more complex bodies, where a large portion of the surface is concave (such as for 67P/Churyumov-Gerasimenko), we can still use a best-fit ellipsoid to the surface data, which will provide adequate performance for most cases. In a few degenerate cases, though, it may be necessary to approximate the surface with multiple ellipsoids and compute the derivative for each limb point based off of which ellipsoid it corresponds to. A geometric discussion of why this approach is valid is provided in the Appendix. The second option is to approximate the Jacobian using finite differencing. While this option can provide more accurate Jacobian representations, it is much slower and care must be taken to ensure appropriate steps are taken for the finite differences due to the piecewise nature of the surface. In the rest of this work, we proceed with the first option and show that it provides excellent results for a range of viewing conditions.

### Alternative Limb Detection Methods

In the previous two subsections, we assumed that the observed limb locations in the images are found using the 1D cross correlation technique that is typical of limb scanning algorithms. While this process generates accurate limb estimates, and is applicable to all body types, it can be relatively slow due to the rendering of the scan lines using the *a priori* estimate of the scene for each iteration. Luckily, we can replace this step with a simple sub-pixel edge detection method from image processing combined with a limb selection technique (where non-limb edge points are discarded) to extract the observed limbs in the image as is done in the ellipse-matching methods. This technique is generally much faster and provides just as accurate limb locations due to advancements in image processing.
We only need to make the following changes to the algorithms defined above to use this method to extract the limb locations in place of the limb scanning. First, instead of choosing scan lines based on the \textit{a priori} sun direction in the image, the scan lines are determined by the observed sub-pixel limb locations that we extract from the image and our scan center vector. Second, we do not need to recompute the observed limb locations at each iteration, and instead only need to recompute the predicted limb locations. Beyond making these two small changes, the techniques described are the same.

\textbf{Limitations and Workarounds}

Because the algorithms described in this paper rely on \textit{a priori} knowledge of both the shape model and the observation conditions for each image, there are a few limitations, particularly for very irregular bodies like the comet 67P/Churyumov-Gerasimenko. The primary limitation is that the scan center vector must be guaranteed to strike the target body and also must project onto a point that is inside the body’s projection in the image. In many cases, the \textit{a priori} knowledge of the relative position and orientation between the target body and the camera is not sufficient to guarantee both of these conditions. To overcome this limitation, one can use a simple moment algorithm to identify the center-of-brightness of the body in the image, and then adjust the \textit{a priori} scene so that the body lies along the unit vector through this point.

Another limitation to this technique is that the \textit{a priori} knowledge of the shape model and relative orientation between the target body and the camera must be sufficiently good. Because we do not estimate the relative orientation, or any of the parameters of the shape model, the technique assumes that the existing knowledge is close to truth. This means that errors in the shape model can contribute directly to errors in the resulting relative position estimate. In particular, if the shape model is improperly scaled, this will lead to direct errors in the range estimate from the camera to the center-of-figure of the body. In cases where the scale of the shape model is not well known (such as during the early stages of shape modeling), this technique can still serve as a useful tool to adjust the predicted range to the body to get a more accurate center-of-figure estimate than would be achievable with just cross-correlation or traditional limb scanning using the best range estimate available.\footnote{In early approach, when the range estimate to the body is very poor, or our knowledge of the scale of the body is very poor, traditional limb scanning and cross-correlation both tend to align the limbs of the model, which can cause large biases in the limb direction (either towards or away from the limb depending on whether the range/scale error is positive or negative). This leads to biased center-finding residuals which makes it difficult to resolve the scale or range error. Using this technique can allow you to get a good center-of-figure estimate even in instances of extreme scale/range errors, even if the resulting range estimate is off due to an incorrect scaling of the model.}

One final limitation to discuss is the potential for mis-identification of limb points in very irregular bodies. In some viewing conditions, the location of the predicted limb points can dramatically change with a small change in the relative position between the camera and the target body (demonstrated in Figure 2). In these degenerate cases, it becomes necessary to place a constraint that the \textit{a priori} knowledge of the relative position be close enough to ensure that the proper portion of the body is being considered for the limb.

\textbf{Figure 2}: In some viewing conditions, small changes in the relative position can lead to the limb point falling on an entirely different portion of the target body for very irregular bodies.
RESULTS

To demonstrate the capabilities of the technique described in this paper, we applied it to a number of synthetic images of the asteroid Bennu and the comet 67P/Churyumov-Gerasimenko. These images were generated using the Geomod geometry modelling tool and the Phillum physically-based stochastic ray tracer, both developed at GSFC as part of the Freespace simulation environment. This produced photo-realistic synthetic images for which we knew the truth relative positions to compare with. The shape model for Bennu was generated by the altimetry working group for OSIRIS-REx based on the radiometric shape model developed by Nolan. The shape model for 67P/Churyumov-Gerasimenko was generated from Rosetta’s OSIRIS imagery using Stereophotoclinometry by Robert Gaskell.

We applied the described algorithm to each image using the Goddard Image Analysis and Navigation Tool (GIANT) high-fidelity OpNav tool. In these results, we used a single best-fit ellipse for each body to approximate the change in the limb locations with respect to a change in the relative position. We also fed the algorithm the truth shape model and relative orientation between the body and camera, but we perturbed the initial relative position estimate by errors of 40 percent of the range to the body in a random direction.

The results are discussed for each target body in the following subsections. A further investigation of this technique for a wider range of viewing conditions is described in Reference 17.

Synthetic Images of Bennu

Bennu provides a good example of a body that can be considered semi-regular. On a large scale, it is entirely convex, and can easily be roughly approximated as a tri-axial ellipsoid.

Figure 3 shows the errors in the estimated relative position between the camera and Bennu as a function of range to the body. Figure 4 shows the errors in the estimated row/column location of Bennu in the image and the estimated range to Bennu from the camera center. As can be seen in both figures, the algorithm performs excellently, even in the face of very large initial errors. There is a slight bias in the range estimates. Currently, we believe this bias is due to slight differences in the way camera models are handled in Geomod, but we need to investigate further to come to a final conclusion. Figure 5 shows some sample images with the a priori and final limb locations.

Synthetic Images of 67P/Churyumov-Gerasimenko

67P provides a good stressing case for the algorithm with a very irregular shape. There are large portions of the body which are concave, and many potential degenerate cases.

Figure 6 shows the errors in the estimated relative position between the camera and 67P as a function of range to the body. Figure 7 shows the errors in the estimated row/column location of 67P in the image and the estimated range to Bennu from the camera center. As can be seen in both figures, the algorithm performs excellently in most cases, but there are a few outliers. These outliers are due to degenerate cases where it becomes difficult to correctly identify the limb in either the image or in the shape model. An example of one of these cases is shown in Figure 10. Clearly, the issue in this case is due to a mis-identification of the observed and predicted limb-points, and is not indicative of the overall performance of the algorithm and improvements in the robustness of the identification should cause these outliers to fall back into family with the other results. Figures 8 and 9 are the same plots as Figures 6 and 7 but cropped to show the performance without the outliers. As can be seen in the non-outlier cases, the performance is still very good, although slightly worse than the performance for Bennu. As was the case with Bennu, there is a slight bias in the range estimate. We believe this to be caused by the same issue with the simulation of the images. Figure 11 shows some sample images with the a priori and final limb locations.
Figure 3: Errors in the relative position between Bennu and the camera center expressed in the camera frame resulting from the limb scanning estimation. Note that the $z$–axis is aligned with the camera boresite.

Figure 4: Errors in the location of Bennu in the image and the range to the body resulting from the limb scanning estimation.
Figure 5: Sample figures with the *a priori* and final limb locations predicted by the model.
Figure 6: Errors in the relative position between 67P/Churyumov-Gerasimenko and the camera center expressed in the camera frame resulting from the limb scanning estimation. Note that the $z$–axis is aligned with the camera boresite.

Figure 7: Errors in the location of 67P/Churyumov-Gerasimenko in the image and the range to the body resulting from the limb scanning estimation.
Figure 8: Errors in the relative position between 67P/Churyumov-Gerasimenko and the camera center expressed in the camera frame resulting from the limb scanning estimation. Note that the $z-$axis is aligned with the camera boresite. This is the same plot as Figure 6 but cropped to show the performance excluding the outliers.

Figure 9: Errors in the location of 67P/Churyumov-Gerasimenko in the image and the range to the body resulting from the limb scanning estimation. This is the same plot as Figure 9 but cropped to show the performance excluding the outliers.
**Figure 10:** An example of one of the degenerate cases that caused the limb scanning to fail. In this case, the extracted and predicted limb points were clearly misidentified along the larger lobe (left) leading to a poor relative position estimate. Cleaning up this identification will remove the outliers and allow the algorithm to still generate a good fit.

**Figure 11:** Sample figures with the *a priori* and final limb locations predicted by the model.
CONCLUSION

This paper describes a new technique for extracting both bearing and range measurements from monocular images of regular and irregular bodies using only the illuminated limb points in the image. The new technique is based on limb scanning and uses an iterative least squares estimation routine to estimate the full 3D relative position between the camera and the target body in the camera frame. Because the technique only considers limb-points, it is in general much faster than cross-correlation algorithms and shows similar center-finding performance. The paper describes all of the required steps to perform the algorithm, from limb extraction from the image to the estimation process. The algorithm is applied to synthetic images of the asteroid Bennu and the comet 67P/Churyumov-Gerasimenko and is shown to have good performance with the exception of a few outlier cases.

Future work includes investigating a more robust technique for pairing the predicted and extracted limb points. In addition, analysis needs to be performed to identify how the algorithm performs with degraded knowledge of both the target shape model and the relative orientation between the target body and the camera frame. Finally, the bias in the estimated range values needs further investigation to identify if it is due to a deficiency in the algorithm or a deficiency in the simulation of the synthetic images.

ACKNOWLEDGMENTS

The author would like to thank Olivier Barnouin, Eric Palmer, Michael Daly, Robert Gaskell, and Catherine Johnson of the ALTWG for the excellent work they performed in generating the surface models used to simulate the images of the asteroid Bennu. The author would also like to thank Robert Gaskell for producing the 67P/Churyumov-Gerasimenko shape model. Finally, thanks to Kenneth Getzandanner of NASA Goddard and Natalie Liounis for their valuable feedback.

REFERENCES

APPENDIX

Invertability of the Coefficient Matrix for the Limb Jacobian

Consider the left hand side of Eq. (15). To show that this matrix will be full rank for any realistic observing
conditions, we can simply show that each row of the matrix is linearly independent.

First, consider the first and second rows which we can prove are linearly independent by contradiction. Let
us assume that the rows are linearly dependent. By definition then

\[ A_C r_C = 2 \alpha A_C p_C \tag{16} \]

where \( \alpha \) is some constant multiplier, and we have made use of the fact that \( A_C = A_C^T \) due to the symmetric
nature of \( A_C \). Also, since \( A_C \) will always be invertible, then we can simplify this to

\[ r_C = \alpha p_C \tag{17} \]

where we have absorbed the coefficient of 2 into the term \( \alpha \). Now, use this equation to substitute into Eq. 3
which gives

\[ \alpha p_C^T A_C p_C = -1 \tag{18} \]

Using the relationship from Eq. (1), then we get

\[ \alpha = -1 \tag{19} \]

Returning to Eq. (17), we now have

\[ r_C = -p_C \tag{20} \]

which implies that \( r_C \) must be on the surface, which is clearly not a valid viewing geometry, and thus, for
any valid viewing geometry, the first two rows will be linearly independent.∗

Now consider the second and third rows which we can also prove are linearly independent for all valid
viewing geometries using contradiction. Begin by assuming that the rows are linearly dependent, which
gives by definition

\[ n_C = \alpha A_C r_C \tag{21} \]

where \( n_C = c_C \times s_C \) is the normal vector of the scan plane. Substituting this into Eq. (4) gives

\[ \alpha r_C^T A_C p_C = -\alpha r_C^T A_C r_C \tag{22} \]

Now, substitute in the relationship from Eq. (3) and simplify to get

\[ 1 = r_C^T A_C r_C \tag{23} \]

∗The negative sign represents the fact that \( r_C \) is defined from the camera to the center of the body, while \( p_C \) is defined from
the center of the body to the surface.
This equation is the same as Eq. (1) which implies that \( r_C \) must be a point on the surface and thus the second and third rows are only linearly dependent when the viewing location is located on the limb. Therefore, again for all valid viewing conditions, the second and third rows will be linearly independent.

Finally, consider the first and third rows. Assume that the rows are linearly dependent

\[
e_C \times s_C = \alpha A_C p_C
\]  

(24)

Now consider the geometric implications of this equation. The left hand side is the normal vector of the scan plane. The right hand side is the surface normal vector at point \( p_C \) scaled by some constant. Therefore, this implies that the normal vector of the scan plane and the normal vector of the surface are the same. The only way this can happen is if the scan plane is actually tangent to the surface at point \( p_C \) (and thus only touches the surface at point \( p_C \)). In addition, since our scan center vector must lie in the scan plane and must intersect the surface, then we know that our scan center vector is actually directly pointed at the surface point \( p_C \). Therefore, as long as we have chosen the scan center vector so it strikes the middle of the body, and not the limb itself, then the first and third rows will be linearly independent.

**On the Validity of Approximating the Limb Jacobian Using a Tri-Axial Ellipsoid**

When using iterative linearized least squares, the Jacobian represents a first-order linear approximation of how the residuals will change with respect to a change in the estimated parameters. Since this process relies on an approximation to begin with in definition, the accuracy of the Jacobian does not need to be perfect. In particular, the magnitude of the Jacobian is not entirely crucial, but the general direction of the Jacobian is more important.

To demonstrate that a tri-axial ellipsoid serves as an adequate representation of the limb Jacobian, we need to use geometric intuition. For a semi-regular body like Bennu, it is clear that a tri-axial ellipsoid provides a good approximation of the overall limb Jacobian as demonstrated in Figure 12. When the relative position is shifted, the change in the limb points for a tri-axial ellipsoid clearly mimic the change in limb points for the actual shape of Bennu.

![Figure 12](image)

**Figure 12:** A tri-axial ellipsoid clearly provides a good approximation for the limb Jacobian for semi-regular bodies.

For an irregular shape like 67P/Churyumov-Gerasimenko, the approximation is less apparent. Consider Figure 13. As long as the surface is largely convex in the direction of the limb point, then the tri-axial ellipsoid will do a fair job of representing the change in the limb point with respect to a change in the relative position. For most natural bodies, the limb direction will always be largely convex (ignoring small scale terrain like craters), meaning that the approximation will be sufficient. This is demonstrated for 67P/Churyumov-Gerasimenko in Figure 14. This approximation will break down, however, if the convex argument does not hold. One instance where this can be the case is along the rim of a bowl-shaped object. Clearly, for limb points on the rim of the bowl-shaped object, the gross shape is concave and a tri-axial ellipsoid would not approximate the change in the limb point well, as shown in Figure 15. Luckily, the vast majority of natural bodies do not have features like this.
**Figure 13:** A tri-axial ellipsoid can provide a good approximation for even irregular bodies as long as the shape model is largely convex in the direction from the camera to the limb point.

**Figure 14:** While there are portions of 67P that are concave, the bulk shape in the direction of the limb vectors will always be largely convex, which means a tri-axial ellipsoid can be used to represent the Jacobian of the limb point.

**Figure 15:** A theoretical bowl shaped object would cause the tri-axial ellipsoid approximation to break down along the rim of the bowl. Luckily most objects are not shaped like this.