DETERMINATION OF PLUTO'S RADIUS DURING THE *NEW HORIZONS* ENCOUNTER. C. J. Thomason[1*], F. Nimmo[1,] [1]Dept. Earth and Planetary Sciences, University of California Santa Cruz, 1156 High St, Santa Cruz, CA 95064, USA, (*cthomas1@es.ucsc.edu)

Introduction: Pluto's mean radius and shape are currently poorly constrained, but are important for understanding the interior structure [1]. To date the best estimates of Pluto's radius come from stellar occultation events. These measurements are limited because of Pluto's small angular size (<0.1 arcsec) and the small angular separation between Pluto and Charon (<1 arcsec)[1]. This is further complicated by the possibility of a transient atmosphere around Pluto. Uncertainties in the radius of Pluto tend to be ~10km. The *New Horizons* encounter will provide high resolution images from LOng-Range Reconnaissance Imager (LORRI) [2] permitting a much improved radius determination. Here we investigate the accuracy with which the radius will be recovered.

Method: We wrote a MATLAB GUI named LimbDetect. The basic methodology follows that of [3] and consists of four steps: initial guessing, finding the edge, removing bad picks, and fitting the ellipse.

Initial guessing. In contrast to [3], LimbDetect requires an initial ellipse. LimbDetect can either take a user input guess ellipse or can determine the guess automatically. The automatic method takes advantage of the fact that there is a sharp brightness gradient along the limb. The automatic guess ellipse is the ellipse fit to all points in the largest cluster of points where the image gradient is within 30% of the maximum value.

Picking limb points. LimbDetect refines its picks of the edge from the initial guess by scanning the image along both rows and columns. Along each scan a foreground (on-body) value is calculated. This value is calculated as the mean of points within a range set by the user in relation to the guess ellipse radius. An image background value is also calculated as the median of all points outside of the guess ellipse. For each scan, the limb location is determined by interpolation as the point where the DN value is some percentage of the difference between the foreground and background values. This percentage is nominally 50% but can be changed by the user.

Rejecting bad points. Because LimbDetect scans in both directions along each row and column there are always a large number of spurious picks. In addition to manual point removal, we have also implemented two automated tools that perform equally well.

The primary tool is cluster filtering. This works by grouping all of the points into clusters where each cluster is defined as the group of points each within 3 pixels of another cluster member. The user can then choose to keep the *N* clusters with the most data points. *N* is usually set to 1; however larger values may be used if the edge is a discontinuity along the edge (e.g. Saturn's rings transecting the limb).

A secondary filtering technique uses the DN gradient. When the location of the edge is determined, the gradient at that point is recorded. Because the flux should fall off towards the background, any points with a positive gradient are immediately discarded. After cluster filtering the user can remove the 5% of points with the smallest magnitude gradient. This process can be repeated as many times as required, and is useful for removing points along the terminator.

Ellipse fitting. Once the edge picks are made we have two methods for finding the best fit ellipse. The first is the analytical least-squares solution presented in [4]. When the two ellipse axes are close in magnitude the rotation angle becomes poorly constrained. Because of this we have also implemented a numerical least-squares algorithm that takes as input a rotation angle.

Ellipse Fitting Sensitivity: One test for the stability of the ellipse is to systematically remove data points from a full synthetic ellipse and see how it affects the fit ellipse parameters. Figure 1 shows the results of this test with 2000 initial data points and ±1 pixel of random noise in the x and y of each data point. We define the solution score σ as $\sigma=0.25*(|\Delta x_0|+|\Delta y_0|+|\Delta a|+|\Delta b|)$, where x_0, y_0 are the coordinates of the centre of the ellipse, a and b are the ellipse major and minor axes, and $|\Delta x_0|$ etc. denote the absolute difference between the two estimates. The analytic solution begins to degrade first which is to be expected since it is solving for an additional parameter (the rotation angle). The analytic solution begins to degrade when there is less than ~40% of the ellipse while the numerical solution is stable until less than a quarter of the ellipse is present. The reason the solutions begin to degrade is the solution starts to become non-unique. When fitting half an ellipse with noise, a tradeoff develops between x_0 and a and a corresponding tradeoff between y_0 and b. Depending on the orientation of the half ellipse, one of these pairs will be better constrained than the other. These tradeoffs will also occur with real images. As the number of points is reduced the fitting also becomes more sensitive to points near the edge. Because of these effects, small differences in the edge picks can be amplified in the ellipse fits.

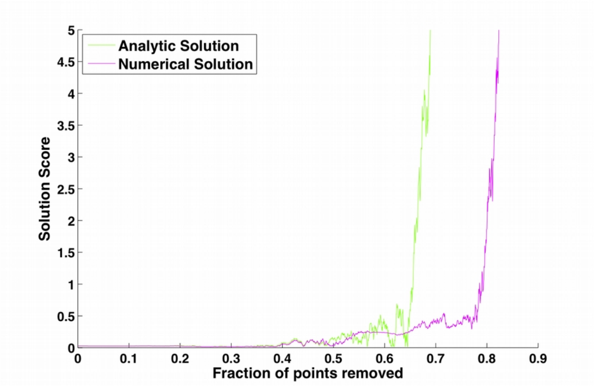

Figure 1: Quality of the ellipse fits as a function of the fraction of the ellipse used.

Pick Validation: To quantify the accuracy of our picks we compared them with independent picks of individual Cassini ISS images of Rhea [5]. Table 1 shows that the median distance between the two sets of picks was <0.25 pix with no manual removal of points, while the solution scores (see above) range from ~0.25 pixel up to 3.6 pixels. This wide range of in σ is due to the sensitivities of ellipse fitting described above. Figure 3 shows that both techniques pick up the same topographic features along the limb. Applying the LORRI point-spread function (Table1:N1514059120psf) had only a minor effect on our ability to determine the edge.

Image	Resolution (km/pix)	Median Misfit (pix)	σ (pix)
N1514059120	2.00	0.25	1.59
N1514059120psf		0.38	1.80
N1499997214	2.03	0.18	0.28
N1516199584	1.46	0.18	3.66
SYN Basic		0.26	2.75
SYN Smooth		0.27	2.09
SYN Rough		0.29	4.56

Table 1: Evaluation of our limb picks. Cassini images are compared to those done by Peter Thomas [6]. The limb picks shown used only cluster filtering (i.e. no user input in removing points). Synthetic images are denoted by SYN and are compared with the model limb.

As an additional test we created a series of synthetic Pluto images using a plausible photometric function [7] and point-spread function [8] and the planned *New Horizons* viewing geometry. Synthetic images included varying levels of simulated limb topography along with other complicating factors such as surface albedo variations, and image noise. The median distance between LimbDetect picks and the true limb locations was always less than half a pixel and was on average 0.3 pixels. The ellipse parameters were consistently correct to within 3 pixels, and within 1 pixel for the better-constrained of the two axes. Combining images from different geometries will reduce the ellipse uncertainties.

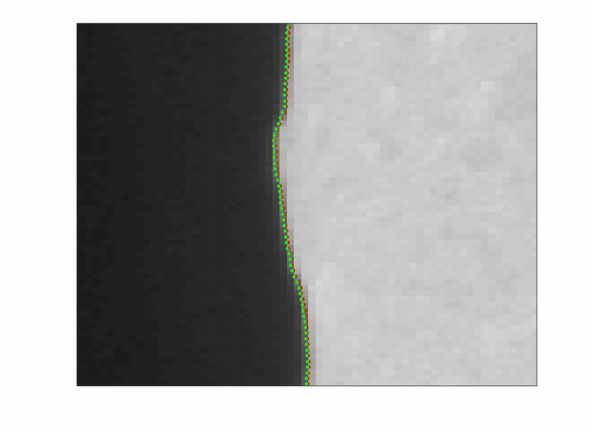

Figure 2: Comparison of limb picks done by Peter Thomas [6](red) and this study (green) for a portion of Cassini image N1499997214.

Conclusions: Through comparison with our models and published results we believe that LimbDetect can accurately determine the location of a planetary limb to within a quarter of a pixel. The ellipse parameters fit to that limb are much less certain and depend on the amount of surface topography as well as the illumination angle. Still we expect to be able to recover the well defined axis to within 1 pixel and the orthogonal axis to within 3 pixels.

Acknowledgments: Thanks to Peter Thomas and Jeff Moore for their comments and suggestions.

References: [1] Stern, A., & Tholen, D. J. (Eds.). (1997). *Pluto and Charon*. University of Arizona Press. [2] Cheng, A. F., et al. *Space Science Reviews* 140.1-4 (2008): 189-215.[3] Dermott, S. F., and Thomas, P. C. *Icarus* 73.1 (1988): 25-65. [4] Pilu, M et al. (1996) Pattern Analysis and Machine Intelligence, IEEE Transactions on, 21(5), 476-480. [5] Porco, C., et al. *Space Science Reviews* 115.1-4 (2004): 363-497. [6] Thomas, P. C. *Icarus* 208.1 (2010): 395-401.[7] Buratti, B.J., Icarus 59, 392-405, 1984. [8] Noble, M.W. et al., Inst. Methods Astrobiology Planetary Missions XII, SPIE Proc. 7441, 2009.