AUTOMATED BOULDER DETECTION AND MEASURING IN HIRISE IMAGES.

D.R. Hood[1] (dhood7@lsu.edu), S. Karunatillake[1], C.I. Fassett[2], S.F. Sholes[3] [1]Louisiana State University Geology and Geophysics (E 235 Howe-Russel-Kniffen Geoscience complex, Louisiana State University, Baton Rouge, LA 70803), [2] NASA Marshall Space Flight Center, Huntsville, Alabama, [3] Earth and Space Sciences & Astrobiology, University of Washington, Seattle, Washington.

Introduction: Images taken by the High Resolution Imaging Science Experiment (HiRISE) show that meter-scale boulders, as observed by landers and rovers, are present across the entire surface of Mars [1], [2]. Quantifiying estimates of these boulder populatuions, including their size and location, can adddress several outstanding questions including pedogenesis, surface weathering, impact processes, and mass wasting processes[3], [4]. However, manual measurement of boulder populations are time intensive, and can not be applied at large scales (e.g. more than a few square km of surface area). To facilitate this, we have developed a Python-based algorithm to automatically identify, locate, and measure boulders on the martian surface. This set of tools and programs is collected in a python library called the Martian Boulder Automatic Recognition System: MBARS.

This is designed as a publically available toolset to enable science and allow modification and improvement by the larger community. It is written in the Python language, which is free to use, and uses a number of common and accessible Python libraries (NumPy, SciPy, etc.). In addition, the program is designed to have simple command-line interaction, requiring little programming knowledge.

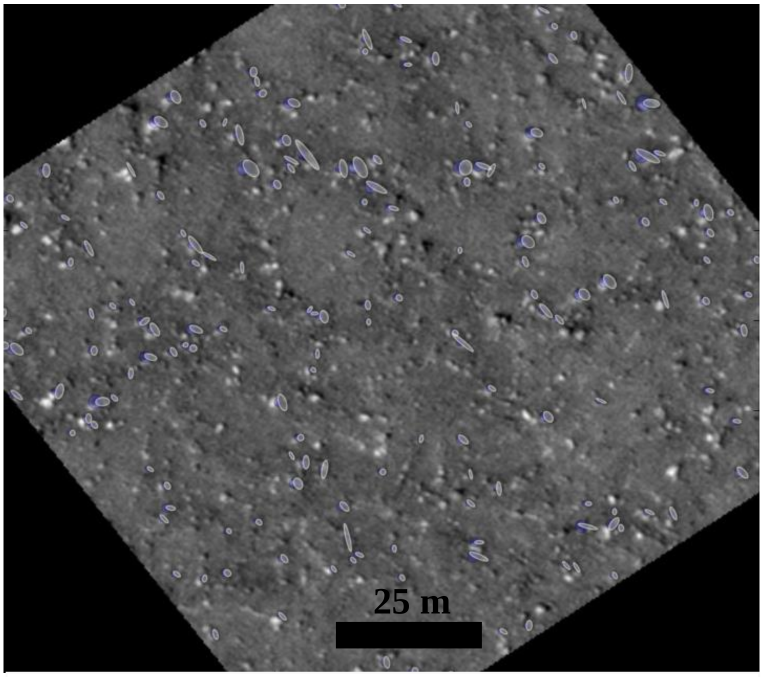

Figure 1: Portion of HiRISE image PSP_007718_2350 used in algorithm testing. White ellipses show shadow ellipses and blue circles show interpreted boulders. Algorithm catches many boulders, but misses some due to small shadows if settings are too conservative, shown here. Image is ~150m wide with the sun coming from the western (left) side. Image credit: NASA/JPL/University of Arizona

Methodology: MBARS relies primarily on the capabilities of the existing methodology described by Golombek in 2008 [1]. Our adaptation of Golombek's work relies on the isolation and segmentation of boulder shadows to determine their geometry. Before analysis from MBARS, the desired image must be paneled into sub images. This is easily achieved with GIS software and the user can control the level of paneling preferred. More individual panels will speed up the algorithm, but can introduce more edge effects. Fewer panels will reduce edge effects, but hardware limitations require a certain degree of paneling, as uncompressed HiRISE images are too large for general use computers to handle. The first step in processing is to identify pixels as either shadows or non-shadows, making a binary image. To achieve this, we fit a gaussian curve to the histogram of pixel intensity in the image, and select a shadow intensity boundary based on the limbs of the gaussian. This is one point in the algorithm that can be tuned to enhance performance. Once shadows are identified, the algorithm, uses the binary image to identify individual shadows (i.e. clusters of shadow pixels), and delineates the shadows. An ellipse is fitted to the boundary of these shadows, which is then used to determine the shape of the shadow-casting boulder. This step is one of the more computationally intensive, and uses an Orthogonal Distance Regression (ODR) package developed in Fortran accessed via the SciPy library [5]. The fitting parameters are the major and minor axes of the ellipse, the ellipse center and orientation. Initially, we predicted that the ellipse orientation could be fixed parallel to the sun direction, but this resulted in poor fits. The width of the shadow ellipse (i.e. the width perpendicular to the sun direc-

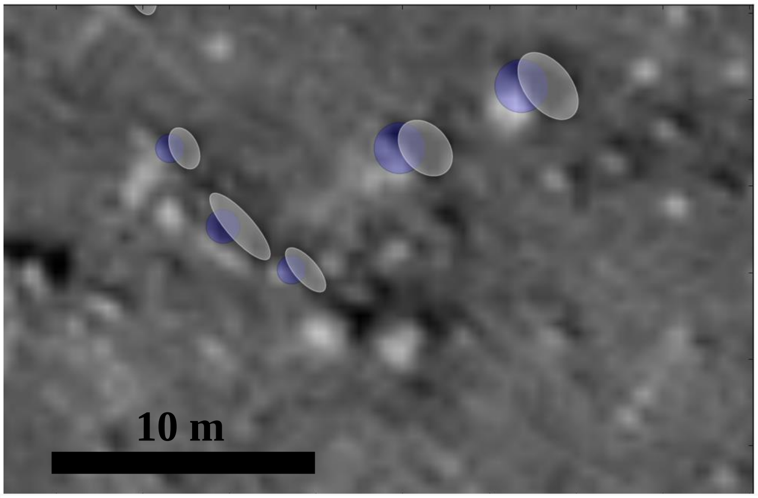

Figure 2: Closeup of figure 1 showing match between shadows and boulders. As before, shadow ellipses are shown in white and interpreted boulders are shown in blue. The boulder is placed at the sunward end of the shadow ellipse. Scene is approximately 30 m wide.

tion) is taken as the width of the boulder, and the height is determined based on the length of the shadow and the angle of the sun. These final parameters of width, location, and height as well as many of the intermediate parameters (fit parameters, object flag, etc.) are stored as python objects that maintain the association between measured boulder and cast shadow. Built into MBARS are several tools to acess and analyze these python objects.

In application, this method has been shown to be statistically robust, typically having errors approaching 1-2 pixels for the diameter and location of detected objects [1].

Caveats and Challenges: While this algorithm excels at identifying boulders, there are a number of other features that are seen on the martian surface that could be mistaken for boulders. Large features such as cliff faces and carter rims can cast shadows which could possibly be mis-identified. In addition, multiple adjacent boulders often have a contiguous shadow, which the algorithm must either split, or flag for manual splitting. In scenarios where the shadow is too large, as would be the case for shadow casting topography, the program ignores the shadow entirely. In some cases, predictions for the boulder width and height result in axial ratios that are not reasonable. This can come from boulders with merged shadows, creating the appearance of a wide but short boulder. In addition, this can arise from structures like linear ridges, which cast long narrow shadows. In these cases, the program flags the objects as problematic and defers to manual adjustment. In both these cases, the size and axial ratio boundary for flagging and removal can be adjusted by the user depending on expectations for the terrain

Performance: A key attribute of this program was speed and accessibility. This program is designed to be used on common computer equipment, and does not require advanced computational facilities. The speed of processing is dependent on the boulder populations in the images, rock-poor regions will require less processing and regions with abundant rocks will require more. On a server with 16GB of RAM and a 4-core, 3.6 GHz processor, a boulder-dense HiRISE image can be completed in a few hours. This rapidly outpaces manual counting methods by a hundred-fold.

Algorithm Testing and Verification: In order to verify and refine our results from MBARS, we rely on existing datasets to “ground truth” our measurements. Previous measurements of boulder populations from other algorithms [1], [2] as well as manual measurements provide data to test against MBARS’ ability to assess populations. Height of measured objects can only be verified in limited cases where we use particularly large, individual rocks [6] seen by landers as well as the landers and rovers themselves as test cases for height. In particular, manual boulder measurements from several sections within HiRISE image PSP_007718_2350 have been our main test case. This image has a high rock density and low relief, making it essentially the ideal target for MBARS. We use three key plots to test consistency with manual measurements: Boulder density, cumulative fractional area, and size-frequency distributions. Boulder density plots focus solely on the location and identification of boulders, testing that manual and automatic measurements find boulders in the same location. The latter two plots, cumulative fractional area and size-frequency distributions, capture the size estimate of boulders by MBARS compared to manual methods.

References: [1] M. P. Golombek et. al, *JGR*, vol. 113, p. E00A09, Jul. 2008. [2] M. P. Golombek et. al, *Int. J. Mars Sci. Explor.*, vol. 2, 2012.
[3] T. de Haas,et. al *GRL*, vol. 40, no. 14, 2013.
[4] T. C. Orloff et. al, *JGR*, vol. 116, no. E11, Nov.2011. [5] P. T. Boggs, et. al, *U.S. Dep. Commer.*, June, p. 99, 1992. [6] H. J. Moore and J. M. Keller, *A Bibliogr. Planet. Geol. Geophys. Princ. Investig. their Assoc. 1990-1991*, 1991.