ELSEVIER

Computers & Geosciences 30 (2004) 59–72

COMPUTERS & GEOSCIENCES

www.elsevier.com/locate/cageo

Gridding Mars Orbiter Laser Altimeter data with GMT: effects of pixel size and interpolation methods on DEM integrity

Chris H. Okubo[a,*], Richard A. Schultz[a], Greggory S. Stefanelli[b]

[a] *Department of Geological Sciences/172, Geomechanics–Rock Fracture Group, Mackay School of Mines, University of Nevada, Reno, NV, USA*

[b] *DataWorks, University of Nevada, Reno Libraries, University of Nevada, Reno, NV, USA*

Received 4 March 2003; received in revised form 3 October 2003; accepted 11 October 2003

Abstract

High-resolution digital elevation models (DEMs) based on Mars orbiter laser altimeter (MOLA) data provide geospatial characterizations of Martian topography. MOLA range data are essentially two-dimensional topographic profiles. Transforming these profile data into three-dimensional DEMs requires the interpolation of a continuous surface between MOLA observations. To this end, we outline a method of generating MOLA-based DEMs using the generic mapping tools (GMT) software suite. The percentage of interpolated data within these DEMs is a function of the spatial density of the MOLA observations and is shown to vary inversely with the pixel size of the DEM. We test the relative accuracy of our DEMs by comparing interpolated elevation values against coincident MOLA observations. Tests are conducted on MOLA-based DEMs containing ~98% interpolated data at a resolution of 200 pixel/°. Our results yield average elevation differences and standard deviations for the interpolated surfaces that are comparable to the uncertainty of the original MOLA data. Based on these findings, we conclude that the GMT interpolation routines produce meaningful high-resolution MOLA-based DEMs.

Keywords: Digital elevation model; Topography; Continuous surface interpolation; Irregularly spaced data; Gridding algorithms

1. Introduction

High-resolution topographic data for Mars are now widely available from the Mars Orbiter Laser Altimeter (MOLA) instrument (Zuber et al., 1992) aboard the Mars Global Surveyor (MGS) spacecraft (Albee et al., 1998). MOLA acquired range measurements to the Martian surface every ~300 m along the MGS ground track, and the footprint of each range measurement is ~168 m in diameter (Smith et al., 2003a). MGS is in a polar orbit, thus the MOLA data tracks are densely overlapping at high latitudes and can be separated by as much as 10 km near the equator. Receiver specifications of Abshire et al. (2000) estimate vertical ranging errors of 0.23 m to ~10 m for target ground slopes of 1–30° respectively, for the MOLA channel 1 receiver, while 3 additional channels are reported to have smaller error estimates for steeper slopes. Additionally, crossover analysis of MOLA returns yields typical vertical precision of less than 1 m (Neumann et al., 2001). The geographic locations of each datum have uncertainties of approximately 30 m radially (Smith et al., 2003a).

Along-track spot elevations from MOLA are publicly distributed as precision experiment data records (PEDRs; Smith et al., 2003a) via NASA's Planetary Data System.[1] These PEDRs provide topographic profiles of the Martian surface along the predominantly north–south oriented MGS ground track. Analyses of

*Corresponding author. Fax: +1-775-784-1833.
E-mail address: chriso@mines.unr.edu (C.H. Okubo).

[1] NASA Planetary Data System's MOLA Data Product Archives. http://wufs.wustl.edu/missions/mgs/mola/index.html

0098-3004/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cageo.2003.10.004

geologic structures, however, commonly require construction of topographic profiles in a specific orientation, or require generation of a continuous surface terrain model, as demonstrated in current investigations using MOLA data (e.g. Tanaka et al., 2002; Wilkins et al., 2002; Montési and Zuber, 2003; Okubo and Schultz, 2003). The PEDRs have been gridded to continuous surface digital elevation models (DEMs) and are distributed as mission experiment gridded data records (MEGDRs; Smith et al., 2003b), which, at the time of writing, are available at resolutions up to 128 pixel/°, in geographic projection. Commonly, however, the end user may require a digital elevation model (DEM) that is constructed to a specific spatial resolution, one that is updated with revised or newly released MOLA data, or one that incorporates elevations from specific time intervals.

In this paper, we document a methodology for creating MOLA-based DEMs (Fig. 1) utilizing the generic mapping tools (GMT) software suite of programs (Wessel and Smith, 1998). Subsequently, we discuss key considerations for interpolating a continuous-surface DEM from the spatially discontinuous MOLA data. We first examine MOLA data density losses corresponding to increasing DEM pixel resolution. Then we compare the relative accuracies of the three surface interpolation methods available in GMT. We conclude that high-resolution MOLA-based DEMs are a viable 3D alternative to the 2D along-track MOLA profiles.

2. Procedure for data gridding

In this section, we present a procedure for generating DEMs using MOLA PEDRs. We utilize the freely available software packages GMT[2] and Perl.[3] Additionally, we use the Fortran program, *pedr2tab*, which is available from NASA's Planetary Data System (please see footnote 1). For this project, we use a networked workstation running under Microsoft Windows 2000 Server, which has a 2.2 GHz Pentium 4 processor with 2 GB of physical memory.

2.1. Extract PEDRs within ROI

Final release versions of the MOLA PEDRs (version L) are archived in a common directory and cataloged in a text file, pedr.idx, to facilitate data maintenance and retrieval. This archive contains PEDRs AP01578L through AP20327L and occupies 24 GB of disk space. This represents all of the MOLA Mapping Phase and Extended Mission data from February 1999 through June 2001. Data collection ceased on June 30, 2001 due to the failure of MOLA's oscillator, which it relies upon for range timing.

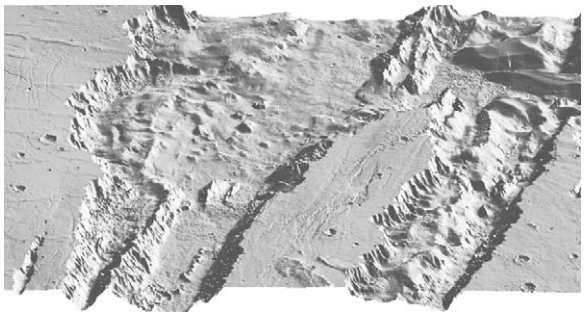

Fig. 1. West-directed perspective view of a MOLA-based DEM covering eastern Valles Marineris. Area spans 285°–295° longitude (foreground to background) and −15°—5° latitude (left to right). Ten degrees of latitude on Mars corresponds to ~592.7 km. Vertical exaggeration is ~3. Illumination from upper right.

Data extraction from the PEDR files is automated through a series of commands in Perl, which enables the use of a straightforward web interface to define a region of interest (ROI) and to execute data extraction routines. For this work, we specify two ROIs, one centered on Valles Marineris (Fig. 1) and one centered on the Pavonis Mons volcano (Fig. 2), both western equatorial Mars. These sites provide a wide range of ground slopes on which we will test the accuracies of the interpolated DEMs. For simplicity, we will next follow the steps used to generate the Pavonis Mons DEM, although the general workflow can be used for any ROI. A list of the GMT commands used here is listed in the appendix.

Based on the data supplied to the web interface, Perl writes the user's search parameters to a text file, pedr2tab.prm. Subsequently Perl executes *pedr2tab*, which utilizes the search parameters defined in pedr2tab.prm to sift through the archived PEDR files listed in pedr.idx. Each 20-shot MOLA data frame that overlaps the ROI is written into memory. Further, *pedr2tab* provides the option to incorporate crossover corrections in order to reduce track-to-track differences in coincident elevation measurements (cf. Neumann et al., 2001). We choose to apply these corrections and specify this option in pedr2tab.prm. The processed data are then written out to a text file, in our case pav.tmp. This file contains data processing and spacecraft engineering information, in addition to x, y, and z data (longitude, latitude, and elevation, respectively) for each MOLA range measurement. The x and y data are in decimal degrees and z are in meters (Smith et al., 2003a).

[2] GMT—The Generic Mapping Tools. http://gmt.soest.hawaii.edu

[3] Perl. http://www.perl.com

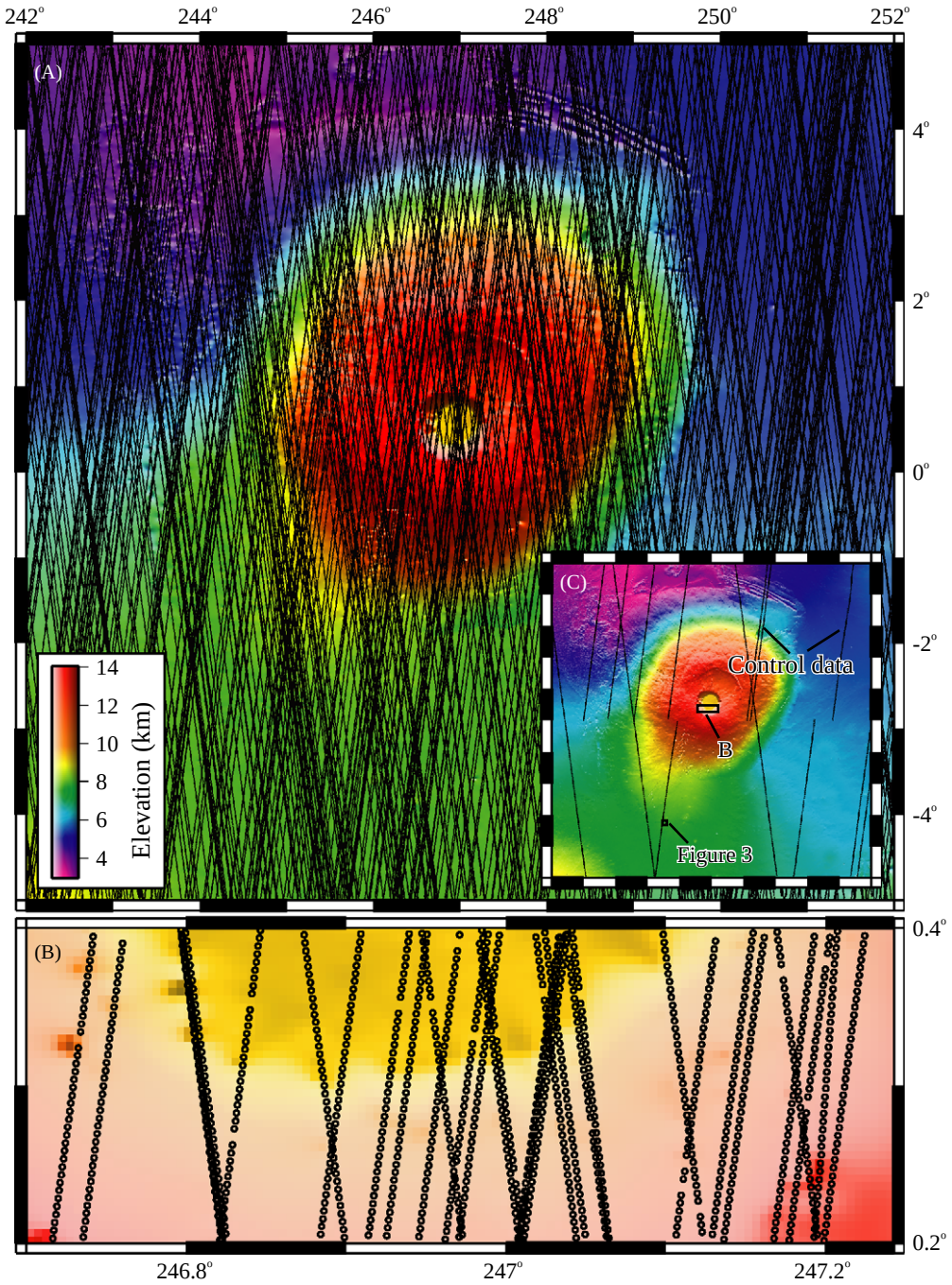

Fig. 2. MOLA-based DEM of Pavonis Mons overlaid with locations of MOLA observations. (A) Typical MOLA coverage within a DEM. Black lines are MOLA observations and colored areas are interpolated data. DEM resolution is 200 pixel/° and 296 m/pixel. (B) Summit region of Pavonis Mons. Open circles represent location and footprint size of individual MOLA ranges. Note that MOLA ranges are occasionally discontinuous along track. (C) Locations of control data and figures discussed in text. DEM resolution is 500 pixel/°, ~119 m/pixel.

The *pedr2tab* data file pav.tmp is then parsed in Perl and the x, y, z, and orbit number for each range is written to a final text file, pav.gmt.txt. The data are space-separated, with one record (one range measurement) per line. These data are now appropriately formatted for direct import into GMT.

2.2. Decimate PEDRs

MOLA data are dense along the MGS ground track and sparse across track (Fig. 2A and B). To avoid high-frequency along-track elevation changes within the final DEM, the MOLA range data in pav.gmt.txt are first decimated using a regularly spaced search node array.

Decimation is implemented in GMT through the *blockmean*, *blockmode* and *blockmedian* routines. Each binning routine uses either a grid-centered or pixel-centered search node array. Each node is assigned a data value corresponding to the mean, mode, or median of the data points within a user-specified search radius of the grid or pixel center. If no data fall within the search radius, the node is not assigned a value. The corresponding geographic coordinates of each mean, median, or modal elevation is set to either the coordinates of the center of the node, or the mean latitude and longitude values of the data points within that node.

We use the pixel-centered search array, which creates nodal positions equivalent to DEM pixel locations when the search radius is set to the DEM pixel half width. In order to reduce pixel-resampling degradation, the geographic extent and spatial intervals of each grid bin is set to the desired DEM pixel size and interval. We separately filter pav.gmt.txt using *blockmean*, *blockmode* and *blockmedian* to create three files, pav.mean.txt, pav.mode.txt and pav.median.txt respectively. We use these three files as the basis for subsequent DEM interpolation routines. This will allow us to evaluate the implications of choosing *blockmean*, *blockmode* or *blockmedian* on the relative accuracies of the resulting DEM.

At this point, pav.mean.txt, pav.mode.txt and pav.-median.txt contain data points registered to the desired DEM pixel centers. These data form a discontinuous mesh of elevation values defined by the mean, mode or median of the original MOLA observations within each pixel. The next step is to interpolate a topographic surface between these MOLA-based pixels.

2.3. Grid and visualize data

GMT provides three routines for interpolating irregularly spaced data to a regularly spaced grid. The *nearneighbor* routine searches for adjacent pixel values within a specified search radius (S) of a search node. The default quadrant search method will assign a nodal value only if each of the four quadrants, within a distance S, around that node contains at least one MOLA-based value. The assigned value is the distance-weighted (and optionally user-specified weighted) mean of the elevations within the search radius. A 'NaN' value (i.e. no value) is assigned to nodes (pixels) that fail the quadrant search test. Thus, the resulting DEMs may contain pixel values that are the weighted average of adjacent pixels, the weighted average of adjacent pixels and the initial nodal (pixel) elevation, or no elevation value. Therefore, the resulting DEM has a *discontinuous* surface and each pixel contains a MOLA-based value of distance-averaged elevations. To illustrate this point, Fig. 3A shows a DEM with a resolution of 1200 pixel/°, with a search radius of 1 km using *blockmean*-decimated data. Black dots show the locations of *blockmean*-filled pixels, which contain MOLA-based elevations. The larger black polygons are pixels that failed the quadrant search test, that is, these pixels are not within 1 km of at least one *nearneighbor*- or *blockmean*-filled pixel. This method is useful in limiting interpolation to within a distance S of MOLA-based pixels, but as a result may produce gaps in the DEM at small values of S.

The *triangulate* routine creates optimal Delaunay triangles between data points and calculates a continuous planar surface within each triangle. Elevations within these triangles define pixel values. Two triangulation algorithms are available. The algorithm of Watson (1982) is installed by default, but alternatively the algorithm of Shewchuck (1996) may also be used. Here, we use the default algorithm. The DEM generated by *triangulate* is a continuous surface that has a faceted appearance where data have been interpolated (Fig. 3B), with the vertices of each facet defined by MOLA-based pixels.

The *surface* routine uses the continuous spline in tension method of interpolating curved surfaces between data points (Smith and Wessel, 1990). This method is similar to physically stretching a thin rubber sheet through the MOLA-based pixels and results in a smooth continuous surface DEM (Fig. 3C). Specifying internal and boundary tension parameters (T) can adjust the curvature tightness of the surface. The model surface is iteratively solved for until a specified convergence limit (C) of the maximum absolute elevation change between successive iterations is attained, or until the maximum number of iterations (N) is reached. By default, T is set to 0, which gives a minimum curvature surface, and C is automatically calculated as 1% of the gradient of the input data.

GMT routines use a default Earth reference ellipsoid, and therefore the appropriate values for Mars must defined in an accompanying text file. The PEDRs use an aerocentric, center of mass reference system (Smith et al., 2003a). Here, a best-fit sphere to the IAU 2002 Mars ellipsoid (Seidelmann et al., 2002) is used to define a mean equatorial radius of 3369 km (Neumann et al., 2001, pers. comm.). Accordingly, all work presented here is referenced to this best-fit sphere.

2.4. Remove outliers

Some MOLA tracks have consistently higher or consistently lower elevations than neighboring tracks.

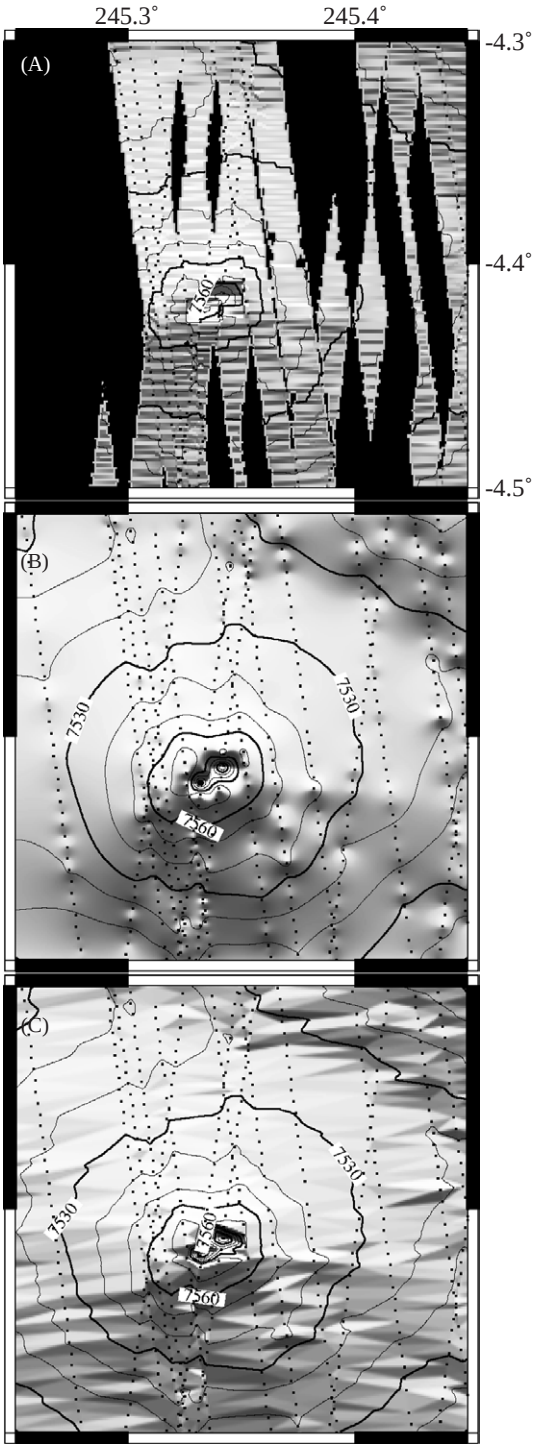

Fig. 3. Comparison between (A) *nearneighbor*- (B) *surface*- and (C) *triangulate*-interpolated DEMs. Points reflect location and size of individual *blockmean*-filled pixels. Illumination is from north. Black polygons in A represent pixels that failed search quadrant test, and thus have no elevation value. DEM resolution is 1200 pixel/°, ~49 m/pixel.

These offset tracks appear as latitudinal 'scratches' within the interpolated surface (Fig. 4A). Sources of these offset tracks may include excessive crossover corrections (Neumann et al., 2001), off-nadir pointing of the MOLA instrument, or a gap in spacecraft orientation data. Instructing *pedr2tab* to output these range correction values and off-nadir angles can help to identify some of these offset tracks. Through the use of a Perl script, records that have off-nadir angles greater than 1–2° or records that have excessive range corrections (on the order of 100's of meters vertical correction) can be filtered out.

Some offset tracks may still occur after automated filtering of questionable tracks. These remaining offset tracks are then removed manually. A preliminary DEM is first created as described above and visually inspected in map view using the postscript output of the *grdimage* or *grdview* GMT programs. Elevation contours can be applied to the DEM to accent offset tracks using *pscontour*. Additionally, *grdgradient* may be used to apply artificial illumination to the DEM. A lighting direction from the east or west can significantly enhance the visibility of offset tracks. Small $0.5^\circ \times 0.5^\circ$ maps are then created over a portion of each offset track, and locations of individual MOLA data points within each map are plotted using *psxy*. A label file is created from the pre-decimator pav.gmt.txt file using Perl, and each MOLA data point is labeled by orbit number using *pstext*. Offset tracks are then visually identified by orbit number and deleted in entirety from pav.gmt.txt using Perl. Finally, the corrected pav.gmt.txt is re-decimated and interpolated to create a refined DEM (Fig. 4B). Frequently, one or more offset tracks lie within a cluster of internally consistent tracks, and several iterations are required to fully identify and remove all of the offset tracks.

3. Evaluation of gridded models

3.1. Effect of grid interval

The number of pixels within a given DEM that contain interpolated data values is dependent on the spatial density of the original data, as well as the sampling frequency of the DEM (i.e. pixel size). We investigate this relationship for MOLA-based DEMs by determining the percentage of the DEM that is directly derived from MOLA data (i.e. the *blockmean*, *blockmode*, or *blockmedian*-filled pixels) at various pixel sizes.

The ratio between the numbers of MOLA-based pixels to the total number of pixels in the DEM is here termed 'MOLA data density'. Higher MOLA data density correlates with a greater percentage of MOLA-based pixels within the DEM. The number of MOLA-based pixels is determined at the data decimation stage

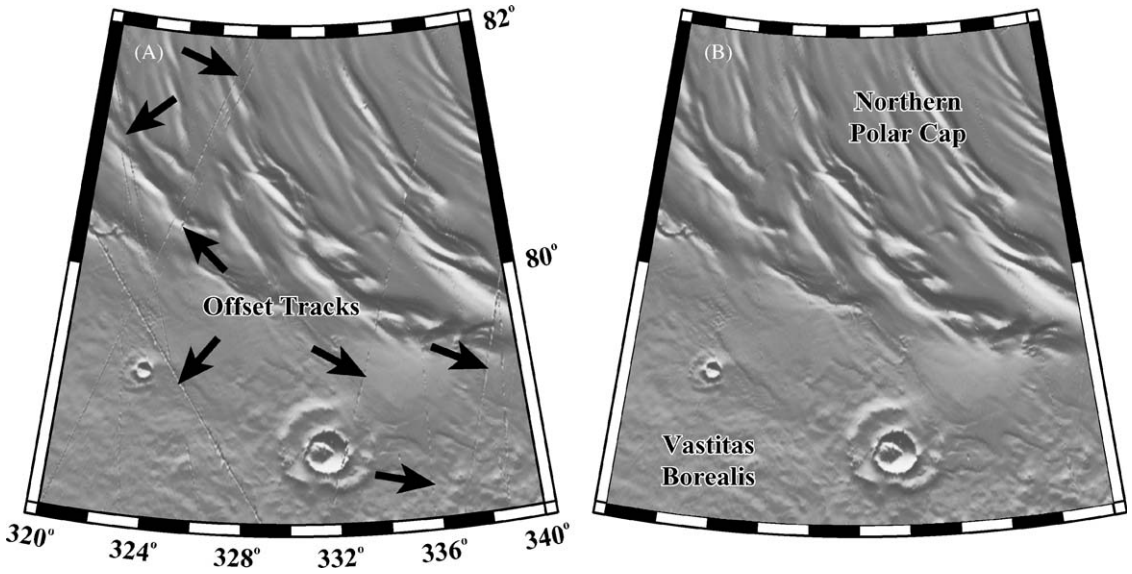

Fig. 4. Shaded relief images of an (A) uncorrected and (B) corrected DEM from edge of northern polar cap. Illumination is from north. DEM resolution is 200 pixel/°, ~51 m/pixel.

by *blockmean*, *blockmode*, or *blockmedian*, which, when run in verbose mode, returns the total number of pixels filled by mean or modal MOLA-based elevations. Since *blockmean*, *blockmode*, and *blockmedian* use the same grid size, grid interval, and input data, the number of MOLA-based pixels is the same for all three decimators. The total number of pixels within the DEM is calculated from its geographic extent and pixel size.

An independent variable in this analysis is the varying spatial coverage of MOLA data as a function of latitude. To address this variable, an equatorial, low data coverage, site centered on Pavonis Mons (Fig. 2) is compared to a high latitude, high data coverage site near the edge of the northern polar cap (Fig. 4).

Our results indicate that variations in spatial coverage at different latitudes contributes to <10% variation in MOLA data density and that pixel size is a much more influential factor (Fig. 5). At 30 pixel/° or less, MOLA data density is near 100%, with density rapidly decreasing at smaller pixel sizes (higher spatial resolution). Large pixel sizes clearly provide high data densities. In other words, the bulk of pixels within these low resolution DEMs are based on actual MOLA data and are interspersed with few interpolated pixel values. Conversely, high-resolution DEMs with more than ~110 pixel/° have less than 10% MOLA data density, that is, more than 90% of the values within these DEMs are interpolated. Fine scale DEMs have lower data densities because much of the data are interpolated. In the following section we quantitatively test and demonstrate that we can extract meaningful results from high-resolution DEMs.

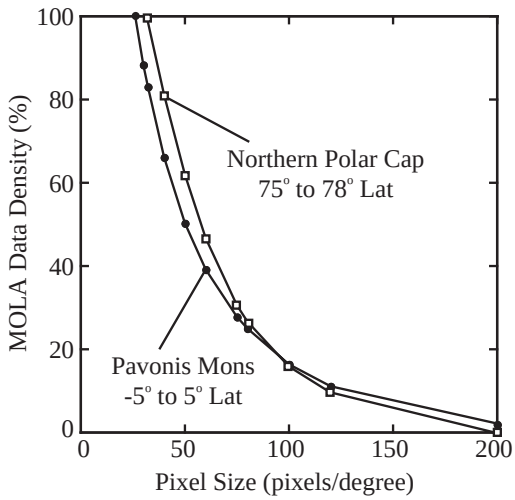

Fig. 5. MOLA data density vs. pixel size for MOLA-based DEMs. Curves are calculated for DEMs shown in Figs. 2 and 4.

3.2. Effect of interpolation routine

In this section, we test the precision of our interpolated DEMs against the original PEDR data for both the Valles Marineris and Pavonis Mons ROIs. This test is accomplished in two steps. First in Section 3.2.1, we compute the difference between the GMT interpolated elevations and the actual PEDR data, for both ROIs. Then in Section 3.2.2, we bin elevation differences for each ROI as a function of point to point ground slope

and compute average and standard deviation in each bin. We make separate linear regressions of the average difference and of the standard deviation as a function of slope in order to evaluate slope-dependent changes in the magnitude and distribution of these elevation differences. We evaluate the results of this analysis in Section 3.2.2, and conclude with a discussion of the amount of time required to complete each interpolation routine.

3.2.1. Compute elevation difference

In order to test the precision of the GMT-interpolated DEMs, we first use the *pedr2tab* output for each ROI. In the Pavonis Mons ROI example, this is the pav.txt file. Offset tracks are then filtered out by both automatic and visual means. Next we remove every other track from pav.txt until only 14 tracks remain. This provides a sampling of control data from the start of the Mapping Phase through the end of the Extended Mission. The spot elevations within these 14 tracks (Fig. 2C) represent 1.8% (15,620 individual spot elevations) of the total PEDR data contained in the *pedr2tab* output after filtering of the offset tracks. For control data in the Valles Marineris ROI, we extract 2.0% (18,555 individual spot elevations) of the total PEDR data. We next remove the control data from the *pedr2tab* output using Perl, and the remaining data (i.e. pav.tracksremoved.txt for Pavonis Mons) are used as our test data.

The subsetted test data for each ROI (e.g. pav.tracksremoved.txt) is decimated by separately using *blockmean*, *blockmode* and *blockmedian* to create three files containing mean, mode and median MOLA-based pixel values, respectively. These decimated files are used as the input for subsequent DEM interpolation in separate *nearneighbor*, *triangulate* and *surface* trials. For both ROIs, the test data are interpolated to DEMs with a resolution of 200 pixel/°, using varied interpolator-dependent options. In separate *nearneighbor* trials, S is varied between 5, 10 and 15 km. At 200 pixel/° (~297 m/pixel at the equator) these distances are approximately equal to search radii of 17, 34 and 51 pixels, respectively. Options for the alternate *triangulate* routine is limited to the choice of triangulation algorithms, for which we use the default algorithm of Watson (1982). In the *surface* routine, we vary T, with equal internal and boundary tensions, between 0 and 0.75. The value of N is set at the default 250, as well as at 500 and at 750. We use the default value for C, determined by *surface* to be 2.14 m for the Pavonis Mons *blockmean*, *blockmode* and *blockmedian*-decimated data, and 3.24 m for the corresponding Valles Marineris data. Additionally, we also use C limits of 1 and 0.5 m. Further, we monitor the *surface* processes to determine whether C or N is reached first. In comparison, the pre-gridded MOLA MEGDR products are created using *surface*, with $T = 0.5$, to grid PEDR data decimated with *blockmode* (Neumann et al., 2001, pers. comm.).

For both the Pavonis Mons and Valles Marineris ROIs, 12 different interpolator and option combinations are used to grid the *blockmean*, *blockmode* and the *blockmedian* test data, resulting in a total of 72 generated grids. In each ROI, GMT-interpolated DEM elevations that are coincident with the locations of the control data are extracted using *grdtrack*, with bilinear interpolation at the boundaries of the DEM. The difference between the GMT-interpolated elevations and PEDR control data are then calculated. Additionally, the point to point ground slope at each control point is determined from the elevations and offsets of the preceding and subsequent PEDR data points along the same track, enabling evaluation of the elevation difference as a function of the PEDR-defined point to point ground slope.

3.2.2. Linear regressions of elevation differences

In order to evaluate any ground slope dependent scatter within the calculated elevation differences, the elevation difference values are binned as a function of the corresponding PEDR point to point slope in five-degree increments. The average value and standard deviation for elevation difference values within each bin are then calculated. The distribution of pixels in all cases is skewed toward ground slopes within the 0–5° bin, with a few outliers at steeper ground slopes (e.g. Fig. 6).

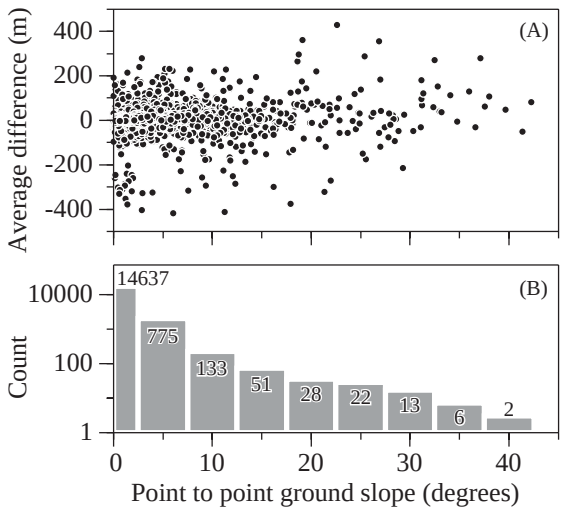

Fig. 6. Distributions of (A) elevation difference vs. point to point ground slope, and (B) ground slope bin count for a Pavonis Mons DEM created using *surface*, with default settings of $T = 0$, $N = 250$, and $C = 2.14$ m using *blockmode* as decimator. These distributions are typical of results obtained for both Valles Marineris and Pavonis Mons ROIs.

In order to compare the distribution of the average elevation differences and their standard deviations as a function of ground slope between different DEMs, we apply a pixel-weighted linear regression using the least absolute deviation method (Carr, 2002, pp. 45–46; Fig. 7 in this paper). The weight of each average difference and standard deviation for each bin is calculated as the count of values within each bin divided by the total value count across all bins (e.g. Fig. 6). The best-fit slope of the regression line (M) and the y-intercept of the regression line (B) are used as a relative measure of the quality of the corresponding DEM. In the average elevation difference regressions, M is interpreted to be a measure of the elevation difference as a function of point to point ground slope. B is interpreted to represent the slope-independent offset, or systematic bias, of the interpolated data. In the standard deviation regressions, M is interpreted to be a measure of the range of uncertainty as a function of point to point ground slope. B is interpreted to represent the systematic range of elevation differences for the interpolated data.

3.2.3. Results

Our tests on the two ROIs show that *surface* creates DEMs with comparable magnitudes of bias (y-intercept of the regression line) in average elevation difference (Table 1). In the Pavonis Mons ROI, the lowest absolute bias in average elevation difference, 0.1612 m, occurs in DEMs created by *surface*, using $T = 0$, $N = 250$, and $C = 2.14$ m (default), with *blockmode* as the decimator. Identical *surface* processes using *blockmean* and *blockmedian*-decimated data have slightly higher, but comparable magnitudes of bias. Interestingly, smaller convergence limits do not reduce the absolute bias. In both cases, interpolation processes stopped at the default number of iterations before converging on C. The *surface* routines that did reach convergence at limits $\leqslant 2.14$ m had *larger* magnitudes of bias. Conversely, in the Valles Marineris ROI, the lowest absolute bias, -0.0596 m, occurs in DEMs created by *nearneighbor*, using $S = 10$ km, with *blockmean* as the decimator. Again, identical *surface* processes using *blockmode* and *blockmedian*-decimated data have slightly higher, but comparable magnitudes of bias. Further, among the

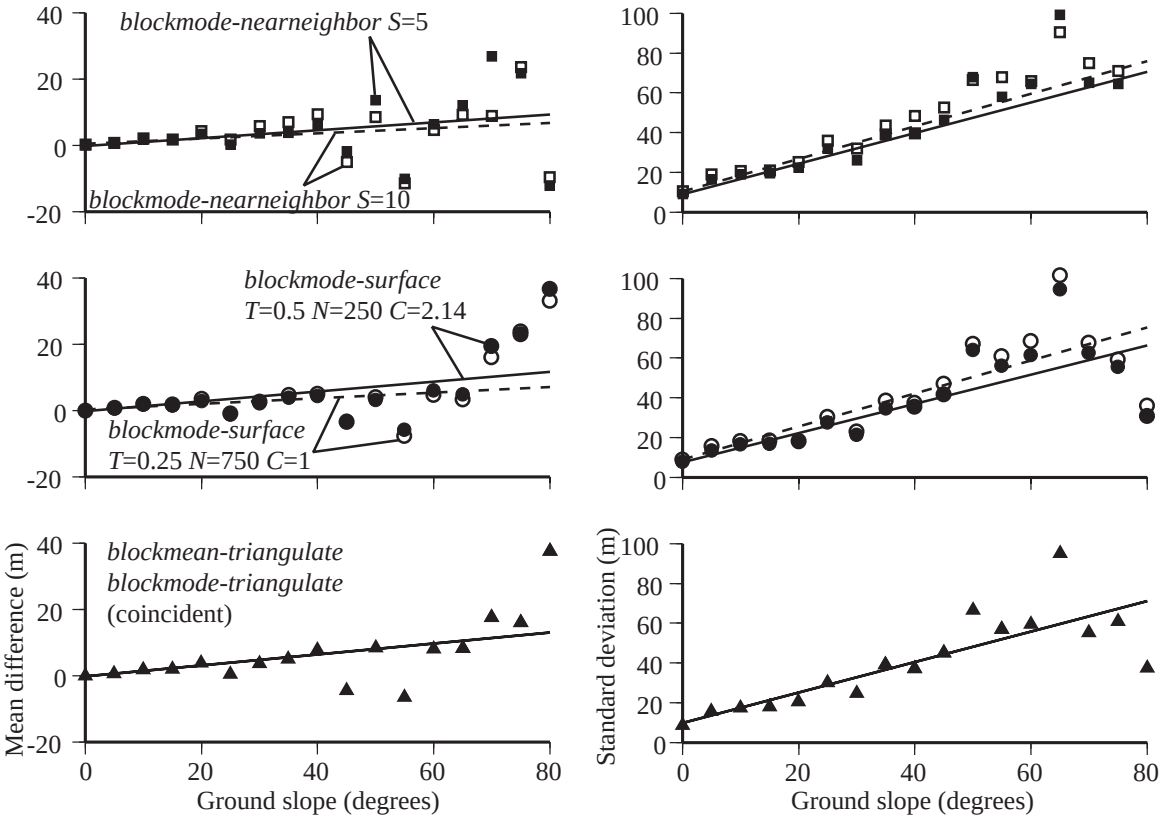

Fig. 7. Distributions of mean differences and standard deviations between interpolated elevations and control data for representative decimator-interpolator combinations from the Pavonis Mons ROI. Similar results are obtained with *blockmedian*-decimated data and with data from the Valles Marineris ROI (see Table 1). A weighted regression line is fit to each data set to facilitate comparisons of our test DEMs.

Table 1
Weighted linear regressions for means and standard deviations of differences between modeled elevations and MOLA topography as a function of ground slope for (A) Pavonis Mons ROI and (B) Valles Marineris ROI

Decimator	Interpolator	Option			Ground slope vs. average difference		Ground slope vs. standard deviation		Process time (min)
(A)									
blockmean	*nearneighbor*	Search radius (S) (km)			M	B	M	B	
		5			−0.0138	0.4047	−0.1177	4.0477	21.46
		10			−0.0165	0.4523	−0.2467	7.5875	87.30
		15			−0.0204	0.5428	−0.1984	6.9747	190.52
	surface	Tension factor (T)	Maximum iterations (N)	Convergence limit (C) (m)					
	Reached N iterations	0	250	2.14	−0.0044	0.1616	−0.1870	5.6991	2.94
	Reached N iterations	0.25	250	2.14	−0.0058	0.2235	−0.2051	6.3881	2.00
	Reached N iterations	0.5	250	2.14	−0.0114	0.3518	−0.2216	7.0355	1.68
	Reached N iterations	0.75	250	2.14	−0.0124	0.3922	−0.2362	7.5932	1.39
	Converged	0.25	500	2.14	−0.0058	0.2235	−0.2051	6.3881	2.22
	Converged	0.25	750	2.14	−0.0058	0.2235	−0.2051	6.3881	1.59
	Converged	0.25	750	1	−0.0145	0.3829	−0.2045	6.3864	2.17
	Converged	0.25	750	0.5	−0.0145	0.3829	−0.2045	6.3864	4.35
	triangulate	Algorithm							
		Watson (1982)			−0.0153	0.4390	−0.2401	7.3217	214.16
blockmode	*nearneighbor*	Search radius (S) (km)							
		5			−0.0138	0.4012	−0.1177	4.0478	22.30
		10			−0.0171	0.4615	−0.2468	7.5881	82.31
		15			−0.0204	0.5416	−0.1984	6.9750	192.46
	surface	Tension factor (T)	Maximum iterations (N)	Convergence limit (C) (m)					
	Reached N iterations	0	250	2.14	−0.0044	0.1612	−0.1870	5.6994	2.76
	Reached N iterations	0.25	250	2.14	−0.0111	0.3156	−0.2052	6.3894	1.61
	Reached N iterations	0.5	250	2.14	−0.0124	0.3684	−0.2217	7.0364	1.85
	Reached N iterations	0.75	250	2.14	−0.0079	0.3122	−0.2362	7.5938	2.07
	Converged	0.25	500	2.14	−0.0111	0.3156	−0.2052	6.3894	1.57
	Converged	0.25	750	2.14	−0.0111	0.3156	−0.2052	6.3894	2.13
	Converged	0.25	750	1	−0.0144	0.3801	−0.2046	6.3875	2.90
	Converged	0.25	750	0.5	−0.0143	0.3801	−0.2046	6.3875	3.89
	triangulate	Algorithm							
		Watson (1982)			−0.0153	0.4378	−0.2401	7.3224	210.36

Table 1 (*continued*)

blockmedian	*nearneighbor*	Search radius (S) (km)							
		5			−0.0137	0.4001	−0.1579	4.7496	22.20
		10			−0.0204	0.5416	−0.1983	6.9723	80.74
		15			−0.0164	0.4496	−0.2466	7.5834	183.66
	surface	Tension factor (T)	Maximum iterations (N)	Convergence limit (C) (m)					
	Reached N Iterations	0	250	2.14	−0.0052	0.1737	−0.1869	5.6972	2.14
	Reached N Iterations	0.25	250	2.14	−0.0111	0.3159	−0.2051	6.3873	2.46
	Reached N Iterations	0.5	250	2.14	−0.0124	0.3688	−0.2216	7.0343	1.83
	Reached N Iterations	0.75	250	2.14	−0.0136	0.4122	−0.2364	7.5914	1.39
	Converged	0.25	500	2.14	−0.0111	0.3159	−0.2051	6.3873	2.20
	Converged	0.25	750	2.14	−0.0111	0.3159	0.2051	6.3873	1.39
	Converged	0.25	750	1	−0.0105	0.3129	−0.2045	6.3856	2.36
	Converged	0.25	750	0.5	−0.0105	0.3129	−0.2045	6.3856	3.61
	triangulate	Algorithm Watson (1982)			−0.0190	0.5019	−0.2401	7.3238	215.41
Decimator	interpolator	Option			Ground slope vs. average difference		Ground slope vs. standard deviation		Process time (min)
(*B*) *blockmean*	*nearneighbor*	Search radius (S) (km)			*M*	*B*	*M*	*B*	
		5			−0.0004	0.0692	−0.3831	17.4827	21.21
		10			0.0024	−0.0596	−0.4124	18.7687	83.48
		15			0.0061	−0.1845	−0.4679	20.4619	182.84
	surface	Tension factor (T)	Maximum iterations (N)	Convergence limit (C) (m)					
	Reached N iterations	0	250	3.24	−0.0065	0.3034	−0.5126	20.8912	2.34
	Reached N iterations	0.25	250	3.24	−0.0062	0.2999	−0.4681	19.5236	1.64
	Reached N iterations	0.5	250	3.24	−0.0059	0.2907	−0.5279	20.8731	1.42
	Reached N iterations	0.75	250	3.24	−0.0054	0.2729	−0.3849	17.7531	1.14
	Converged	0.25	500	3.24	−0.0062	0.2999	−0.4681	19.5236	2.23
	Converged	0.25	750	3.24	−0.0062	0.2999	−0.4681	19.5236	2.19
	Converged	0.25	750	1	−0.0071	0.3295	−0.4773	19.7671	2.21
	Converged	0.25	750	0.5	−0.0072	0.3379	−0.4860	19.9993	3.81
	triangulate	Algorithm Watson (1982)			−0.9012	41.2091	−15.0681	628.5478	205.45
blockmode	*nearneighbor*	Search radius (S) (km)							

		5			−0.0004	0.0652	−0.3832	17.4851	20.48
		10			0.0024	−0.0634	−0.4124	18.7673	78.15
		15			0.0062	−0.1908	−0.4684	20.4731	174.81
	surface	Tension factor (T)	Maximum iterations (N)	Convergence limit (C) (m)					
	Reached N iterations	0	250	3.24	−0.0062	0.2913	−0.5117	20.8565	2.36
	Reached N iterations	0.25	250	3.24	−0.0060	0.2915	−0.4684	19.5276	1.65
	Reached N iterations	0.5	250	3.24	−0.0058	0.2842	−0.5277	20.8670	1.47
	Reached N iterations	0.75	250	3.24	−0.0053	0.2681	−0.3850	17.7535	1.91
	Converged	0.25	500	3.24	−0.0060	0.2915	−0.4685	19.5276	1.94
	Converged	0.25	750	3.24	−0.0060	0.2915	−0.4684	19.5276	1.90
	Converged	0.25	750	1	−0.0074	0.3336	−0.4777	19.7698	2.01
	Converged	0.25	750	0.5	−0.0071	0.3319	−0.4863	20.0015	3.64
	triangulate	Algorithm							
		Watson (1982)			5.3398	−194.2607	−15.8241	661.7751	185.49
blockmedian	*nearneighbor*	Search radius (S) (km)							
		5			−0.0004	0.0664	−0.3829	17.4776	20.75
		10			0.0023	−0.0624	−0.4121	18.7621	78.20
		15			0.0061	−0.1865	−0.4679	20.4636	174.84
	surface	Tension factor (T)	Maximum iterations (N)	Convergence limit (C) (m)					
	Reached N iterations	0	250	3.24	−0.0062	0.2932	−0.5116	20.8516	3.19
	Reached N iterations	0.25	250	3.24	−0.0060	0.2922	−0.4669	19.4916	2.14
	Reached N iterations	0.5	250	3.24	−0.0057	0.2840	−0.5278	20.8679	2.03
	Reached N iterations	0.75	250	3.24	−0.0052	0.2669	−0.3849	17.7516	2.77
	Converged	0.25	500	3.24	−0.0060	0.2922	−0.4669	19.4916	2.34
	Converged	0.25	750	3.24	−0.0060	0.2922	−0.4669	19.4916	2.57
	Converged	0.25	750	1	−0.0067	0.3187	−0.4761	19.7341	2.53
	Converged	0.25	750	0.5	−0.0076	0.3452	−0.4847	19.9654	3.90
	triangulate	Algorithm							
		Watson (1982)			8.9154	−338.3153	−15.0279	627.4709	187.67

interpolators that create continuous-surface DEMs (*surface* and *triangulate*), the lowest absolute bias in average elevation difference, −0.2669 m, occurs in DEMs created by *surface*, using $T = 0.75$, $N = 250$, and $C = 3.24$ m (default), with *blockmedian* as the decimator. Identical *surface* processes using *blockmean* and *blockmode* data yield comparable biases.

The magnitude of average elevation differences in all cases remains generally constant with increases in point to point ground slope. The smallest values of M in the ground slope vs. average difference regressions (a measure of ground slope dependent changes in elevation difference), occur within the *surface* and *nearneighbor*-generated DEMs in both ROIs regardless of the decimator used. The triangulate interpolator generally creates DEMs with the largest magnitudes of M (and larger values of B).

The lowest bias in the slope vs. standard deviation regressions occurs in DEMs created by *nearneighbor* for the Valles Marineris ROI and in DEMs created by *surface* in the Pavonis Mons ROI. In Valles Marineris, *surface* produces DEMs with standard deviation biases comparable to the *surface*-generated DEMs of Pavonis Mons. The dependency of standard deviation on ground slope also follows a similar distribution.

3.2.4. Process time

For all test DEMs, the length of time that each interpolation process requires to generate a grid is shown in Table 1. For both ROIs, the size of the *nearneighbor*-interpolated (pixel-centered) grid is 2000×2000 pixels *surface* and *triangulate* generate grid-centered DEMs, at 2001×2001 pixels, that are subsequently resampled to a pixel-centered registration. By far, the *triangulate* routines require the longest processing times, at ~3.1–3.6 h. In contrast, the fastest routines of *surface* typically require less than 3–4 min, for the tested values of convergence limits and maximum iterations. Increasing the convergence limit, with a large enough value of N, will increase *surface's* processing time. However the benefit of doing this may be negligible given that the average elevation differences and standard deviations for the interpolated surfaces, at $C > 0.5$ m, are similar to or less than the magnitudes of uncertainty predicted for the original PEDR data by Abshire et al. (2000).

4. Summary and conclusions

We have outlined a method for gridding the irregularly spaced Mars orbiter laser altimeter data using the freely available GMT software. Our analysis of the resulting DEMs shows that at resolutions greater than 100 pixel/°, less than 10% of the pixel values are based on MOLA data. The majority of the DEM elevations are interpolated. Tests of 200 pixel/° DEMs created using a systematically varied set of decimating and interpolating routines show that the differences between the modeled elevations and the actual MOLA data are similar to the magnitudes of the uncertainties in the MOLA data itself. Further, the process of decimation and interpolation does not produce significant bias in the DEM.

We find that the *surface* routine produces continuous surface DEMs with the lowest average elevation differences and standard deviations, and that the choice of decimator has negligible effect on the average elevation differences and standard deviations of the final DEMs, for grid intervals of 200 pixel/°. In both tested ROIs, *surface* interpolates elevation values that are comparable to the original PEDR data, in terms of overall average elevation difference and standard deviation, regardless of the *surface*-specific interpolation options used. *surface* is also ten to a hundred times faster than *nearneighbor* or *triangulate* for the same interpolator options. We therefore conclude that overall *surface* is our preferred interpolation routine due to its efficiency and accuracy. Further, we suggest that high-resolution *surface*-generated MOLA-based DEMs are reliable when compared to the inherent uncertainty of the irregularly spaced PEDR data.

In this paper, we focused on interpolators supported by GMT because this software package is freely available and commonly used in the planetary science and geoscience communities. Similar comparisons of predicted elevations against the PEDR data have also been used to test the precision of photogrammetrically generated DEMs of Mars (Baratoux et al., 2001). In future work, DEMs produced by commercial software packages (e.g. ArcGIS;[4] ENVI[5]) should also be tested against the PEDR data, if these alternative methods are to be relied upon.

Acknowledgements

Insightful reviews and discussion with James Carr and Laurent Montési are gratefully acknowledged. Comments on an earlier version of the manuscript by Gregory Neumann, Scott Wilkins and Paul Wessel helped to focus the scope of the final paper. This work was supported by NASA's Planetary Geology and Geophysics Program and the W.M. Keck Foundation.

[4] ESRI ArcGIS. http://www.esri.com
[5] RSI ENVI. http://www.rsinc.com

Appendix A

The following DOS batch commands used to create the grids (DEMs) discussed in Section 3.

```
rem Decimate the pedr2tab output file pav.tracksremoved.txt:
blockmean –R242/252/-5/5 -I0.005 -V pav.tracksremoved.txt > pav.mean.xyz
blockmode –R242/252/-5/5 -I0.005 -V pav.tracksremoved.txt > pav.mode.xyz

rem Run nearneighbor on blockmean- and blockmode-decimated data:
nearneighbor pav.mean.xyz -R242/252/-5/5 -I0.005 -V -F -N4 -S5k

-Gpav.mean.nearneighbor_S5.grd
nearneighbor pav.mean.xyz -R242/252/-5/5 -I0.005 -V -F -N4 -S10k

-Gpav.mean.nearneighbor_S10.grd
nearneighbor pav.mean.xyz -R242/252/-5/5 -I0.005 -V -F -N4 -S15k

-Gpav.mean.nearneighbor_S15.grd
nearneighbor pav.mode.xyz -R242/252/-5/5 -I0.005 -V -F -N4 -S5k

-Gpav.mode.nearneighbor_S5.grd
nearneighbor pav.mode.xyz -R242/252/-5/5 -I0.005 -V -F -N4 -S10k

-Gpav.mode.nearneighbor_S10.grd
nearneighbor pav.mode.xyz -R242/252/-5/5 -I0.005 -V -F -N4 -S15k

-Gpav.mode.nearneighbor_S15.grd

rem Run surface on blockmean- and blockmode-decimated data:
surface pav.mean.xyz -R242/252/-5/5 -I0.005 -V -T0

-Gpav.mean.surface_T0.0.tmp
surface pav.mean.xyz –R242/252/-5/5 -I0.005 -V -T0.25

-Gpav.mean.surface_T0.25.tmp
surface pav.mean.xyz -R242/252/-5/5 -I0.005 -V -T0.5

-Gpav.mean.surface_T0.5.tmp
surface pav.mean.xyz -R242/252/-5/5 -I0.005 -V -T0.75

-Gpav.mean.surface_T0.75.tmp
surface pav.mean.xyz -R242/252/-5/5 -I0.005 -V -T0.25 -N500

-Gpav.mean.surface_T0.25N500.tmp
surface pav.mean.xyz -R242/252/-5/5 -I0.005 -V -T0.25 -N750

-Gpav.mean.surface_T0.25N750.tmp
surface pav.mean.xyz -R242/252/-5/5 -I0.005 -V -T0.25 -C0.5 -N750

-Gpav.mean.surface_T0.25N750C0.5.tmp
surface pav.mode.xyz -R242/252/-5/5 -I0.005 -V -T0

-Gpav.mode.surface_T0.0.tmp
surface pav.mode.xyz -R242/252/-5/5 -I0.005 -V -T0.25

-Gpav.mode.surface_T0.25.tmp
surface pav.mode.xyz -R242/252/-5/5 -I0.005 -V -T0.5

-Gpav.mode.surface_T0.5.tmp
surface pav.mode.xyz -R242/252/-5/5 -I0.005 -V -T0.75

-Gpav.mode.surface_T0.75.tmp
surface pav.mode.xyz -R242/252/-5/5 -I0.005 -V -T0.25 -N500

-Gpav.mode.surface_T0.25N500.tmp
surface pav.mode.xyz -R242/252/-5/5 -I0.005 -V -T0.25 -N750

-Gpav.mode.surface_T0.25N750.tmp
surface pav.mode.xyz -R242/252/-5/5 -I0.005 -V -T0.25 -C0.5 -N750

-Gpav.mode.surface_T0.25N750C0.5.tmp

rem then convert surface output to a pixel-centered grid:
grdsample pav.mean.surface_T0.0.tmp -T -Gpav.mean.surface_T0.0.grd
del pav.mean.surface_T0.0.tmp
grdsample pav.mean.surface_T0.25.tmp -T -Gpav.mean.surface_T0.25.grd
del pav.mean.surface_T0.25.tmp
grdsample pav.mean.surface_T0.5.tmp -T -Gpav.mean.surface_T0.5.grd
del pav.mean.surface_T0.5.tmp
grdsample pav.mean.surface_T0.75.tmp -T -Gpav.mean.surface_T0.75.grd
del pav.mean.surface_T0.75.tmp
grdsample pav.mean.surface_T0.25N500.tmp -T -Gpav.mean.surface_T0.25N500.grd
del pav.mean.surface_T0.25N750.tmp
grdsample pav.mean.surface_T0.25N750.tmp -T -Gpav.mean.surface_T0.25N750.grd
del pav.mean.surface_T0.25N750.tmp
grdsample pav.mean.surface_T0.25N750c1.tmp -T -Gpav.mean.surface_T0.25N750c0.5.grd
del pav.mean.surface_T0.25N750c0.5.tmp
surface pav.mean.xyz -R242/252/-5/5 -I0.005 -V -T0.25 -C1 -N750
-Gpav.mean.surface_T0.25N750C1.tmp
grdsample pav.mean.surface_T0.25N750C1.tmp -T -Gpav.mean.surface_T0.25N750c1.grd
del pav.mean.surface_T0.25N750C1.tmp
grdsample pav.mode.surface_T0.0.tmp -T -Gpav.mode.surface_T0.0.grd
del pav.mode.surface_T0.0.tmp
grdsample pav.mode.surface_T0.25.tmp -T -Gpav.mode.surface_T0.25.grd
```

```
del pav.mode.surface_T0.25.tmp
grdsample pav.mode.surface_T0.5.tmp -T -Gpav.mode.
surface_T0.5.grd
del pav.mode.surface_T0.5.tmp
grdsample pav.mode.surface_T0.75.tmp -T -Gpav.mode.
surface_T0.75.grd
del pav.mode.surface_T0.75.tmp
grdsample pav.mode.surface_T0.25N500.tmp -T -Gpav.
mode.surface_T0.25N500.grd
del pav.mode.surface_T0.25N500.tmp
grdsample pav.mode.surface_T0.25N750.tmp -T -Gpav.
mode.surface_T0.25N750.grd
del pav.mode.surface_T0.25N750.tmp
grdsample pav.mode.surface_T0.25N750C1.tmp -T
-Gpav.mode.surface_T0.25N750C0.5.grd
del pav.mode.surface_T0.25N750C0.5.tmp
surface pav.mode.xyz -R242/252/-5/5 -I0.005 -V -T0.25
-C1 -N750
-Gpav.mode.surface_T0.25N750C1.tmp
grdsample pav.mode.surface_T0.25N750c1.tmp -T
-Gpav.mode.surface_T0.25N750C1.grd
del pav.mode.surface_0.25N750C1.tmp

rem Run triangulate on blockmean- and blockmode-
decimated data:
triangulate pav.mean.xyz -R242/252/-5/5 -I0.005 -V
-Gpav.mean.triangulate.tmp > ijk.tmp
triangulate pav.mode.xyz -R242/252/-5/5 -I0.005 -V
-Gpav.mode.triangulate.tmp > ijk.tmp

rem then convert triangulate output to a pixel-centered
grid:
grdsample pav.mean.triangulate.tmp -T -Gpav.mean.
triangulate.grd
del pav.mean.triangulate.tmp
grdsample pav.mode.triangulate.tmp -T -Gpav.mode.
triangulate.grd
del pav.mode.triangulate.tmp
```

References

Abshire, J.B., Sun, X., Afzal, R.S., 2000. Mars orbiter laser altimeter: receiver model and performance analysis. Applied Optics 39, 2449–2460.

Albee, A.A., Palluconi, F.D., Arvidson, R.E., 1998. Mars global surveyor mission: overview and status. Science 279, 1671–1672.

Baratoux, D., Delacourt, C., Allemand, P., 2001. High-resolution digital elevation models derived from Viking orbiter images; method and comparison with Mars orbiter laser altimeter data. Journal of Geophysical Research 106, 32927–32941.

Carr, J.R., 2002. Data Visualization in the Geosciences. Prentice-Hall, Englewood Cliffs, NJ, 267pp.

Montési, L.G., Zuber, M.T., 2003. Clues to the lithospheric structure of Mars from wrinkle ridge sets and localization instability. Journal of Geophysical Research 108, 5048 (doi:10.1029/2002JE001974).

Montési, L.G., Zuber, M.T., 2003. Clues to the lithospheric structure of Mars from wrinkle ridge sets and localization instability. Journal of Geophysical Research 108, 5048 (doi:10.1029/2002JE001974).

Neumann, G.A., Rowlands, D.D., Lemoine, F.G., Smith, D.E., Zuber, M.T., 2001. Crossover analysis of mars orbiter laser altimeter data. Journal of Geophysical Research 106, 23,753–23,768.

Okubo, C., Schultz, R., 2003. Mechanical stratigraphy in the western equatorial region of Mars based on thrust fault-related fold topography & implications for near-surface volatile reservoirs. Geological Society of America Bulletin 2003, in press.

Shewchuck, J.R., 1996. Triangle: engineering a 2D quality mesh generator and delaunay triangulator. In: First Workshop on Applied Computational Geometry, Philadelphia. Association for Computing Machinery, Pennsylvania, pp. 124–133.

Seidelmann, P.K., Abalakin, V.K., Bursa, M., Davies, M.E., de Bergh, C., Lieske, J.H., Oberst, J., Simon, J.L., Standish, E.M., Stooke, P., Thomas, P.C., 2002. Report of the IAU/IAG working group on cartographic coordinates and rotational elements of the planets and satellites: 2000. Celestial Mechanics and Dynamical Astronomy 82, 83–110.

Smith, W.H.F., Wessel, P., 1990. Gridding with continuous curvature splines in tension. Geophysics 55, 293–305.

Smith, D.E., Neumann G. A., Ford, P.G., Arvidson, R.E., Guinness, E.A., Slavney, S., 2003a. Mars Global Surveyor Laser Altimeter Precision Experiment Data Record. MGS-M-MOLA-3-PEDR-L1A-V1.0, NASA Planetary Data System.

Smith, D.E., Neumann G. A., Ford, P.G., Arvidson, R.E., Guinness, E.A., Slavney, S., 2003b. Mars Global Surveyor Laser Altimeter Precision Experiment Data Record. MGS-M-MOLA-5-MEGDR-L3-V1.0, NASA Planetary Data System.

Tanaka, K.L., Kargel, J.S., MacKinnon, D.J., Hare, T.M., Hoffman, N., 2002. Catastrophic erosion of Hellas basin rim on Mars induced by magmatic intrusion into volatile-rich rocks. Geophysical Research Letters 29, doi:10.1029/2001GL013885.

Watson, D.F., 1982. Acord: automatic contouring of raw data. Computers & Geosciences 8, 97–101.

Wessel, P., Smith, W.H.F., 1998. New, improved version of generic mapping tools released. EOS Transactions of the American Geophysical Union 79, 579.

Wilkins, S.J., Schultz, R.A., Anderson, R.C., Dohm, J.M., Dawers, N.C., 2002. Deformation rates from faulting at the Tempe Terra extensional province, Mars. Geophysical Research Letters 29, doi:10.1029/2002GL015391.

Zuber, M.T., Smith, D.H., Solomon, S.C., Muhleman, D.O., Head, J.W., Garvin, J.B., Abshire, J.B., Bufton, J.L., 1992. The Mars observer laser altimeter investigation. Journal of Geophysical Research 97, 7781–7797.