Icarus 298 (2017) 98–110

Contents lists available at ScienceDirect

Icarus

journal homepage: www.elsevier.com/locate/icarus

Diviner lunar radiometer gridded brightness temperatures from geodesic binning of modeled fields of view

E. Sefton-Nash[a,*], J.-P. Williams[b], B.T. Greenhagen[c], K.-M. Aye[d], D.A. Paige[b]

[a] Planetary Science Institute, 1700 East Fort Lowell, Suite 106, Tucson, AZ 85719-2395, USA
[b] Department of Earth, Planetary and Space Sciences, University of California Los Angeles, 595 Charles Young Drive East, Box 951567, Los Angeles, CA 90095-1567, USA
[c] Johns Hopkins University Atmospheric Physics Laboratory, 11101 Johns Hopkins Road, Laurel, MD 20723, USA
[d] Laboratory for Atmospheric and Space Physics (LASP), 3665 Discovery Drive, Boulder, CO 80303, USA

ARTICLE INFO

Article history:
Received 30 June 2015
Revised 4 November 2016
Accepted 3 April 2017
Available online 12 April 2017

Keywords:
Effective field of view
Diviner
Remote sensing
Planetary mapping

ABSTRACT

An approach is presented to efficiently produce high quality gridded data records from the large, global point-based dataset returned by the Diviner Lunar Radiometer Experiment aboard NASA's Lunar Reconnaissance Orbiter. The need to minimize data volume and processing time in production of science-ready map products is increasingly important with the growth in data volume of planetary datasets. Diviner makes on average >1400 observations per second of radiance that is reflected and emitted from the lunar surface, using 189 detectors divided into 9 spectral channels. Data management and processing bottlenecks are amplified by modeling every observation as a probability distribution function over the field of view, which can increase the required processing time by 2–3 orders of magnitude. Geometric corrections, such as projection of data points onto a digital elevation model, are numerically intensive and therefore it is desirable to perform them only once. Our approach reduces bottlenecks through parallel binning and efficient storage of a pre-processed database of observations. Database construction is via subdivision of a geodesic icosahedral grid, with a spatial resolution that can be tailored to suit the field of view of the observing instrument. Global geodesic grids with high spatial resolution are normally impractically memory intensive. We therefore demonstrate a minimum storage and highly parallel method to bin very large numbers of data points onto such a grid. A database of the pre-processed and binned points is then used for production of mapped data products that is significantly faster than if unprocessed points were used. We explore quality controls in the production of gridded data records by conditional interpolation, allowed only where data density is sufficient. The resultant effects on the spatial continuity and uncertainty in maps of lunar brightness temperatures is illustrated. We identify four binning regimes based on trades between the spatial resolution of the grid, the size of the FOV and the on-target spacing of observations. Our approach may be applicable and beneficial for many existing and future point-based planetary datasets.

1. Introduction

We present a methodology to produce gridded data records of lunar surface temperatures from the Diviner Lunar Radiometer Experiment (Paige et al., 2010), a 9-channel filter radiometer on board NASA's Lunar Reconnaissance Orbiter, which since July 2009 has acquired approximately 1500 observations of the lunar surface per second, creating a database of records >60 TB in size. We build on the work of Teanby (2009) who presented an icosahedron-based method for binning of globally distributed remote sensing data. The geodesic grid used in binning does not suffer from bin size bias with latitude, which can result from binning onto grids defined in a cylindrical projection. Instead, bins are effectively equal in area. Typically the entire triangular grid is required to be constructed and stored in memory prior to binning. For grids with fine spatial scale, which are now required for high spatial resolution planetary datasets such as Diviner's, this approach can consume impractically large computer memory. Adaptation of the technique to cope with very large numbers of data points and/or very fine grid resolution is therefore required for large datasets.

* Corresponding author. Present address: European Space Research and Technology Centre, Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands.
E-mail address: elliot.sefton-nash@esa.int (E. Sefton-Nash).

http://dx.doi.org/10.1016/j.icarus.2017.04.007
0019-1035/

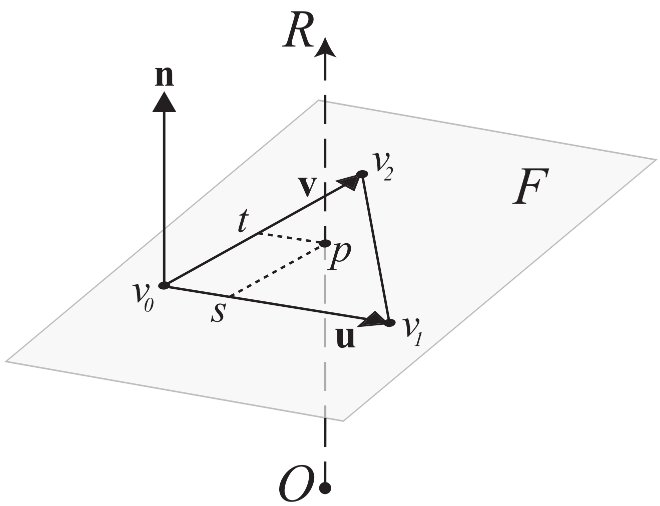

Fig. 1. Adapted from Sunday (2001). Method to test if a ray drawn from the origin to a point outside the unit sphere intersects with a triangle with vertices on the unit sphere. Triangle vertices define a plane, *F*. Parametric coordinates *t* and *s* are calculated as fractions of the unit vectors ***u*** and ***v***, which are parallel to the triangle sides v_0, v_1 and v_0, v_2 respectively.

We demonstrate the following specific developments:

1. Rather than calculating and storing the entire grid in memory prior to the start of the binning process, we implement a minimum storage, recursive scheme to bin data points onto only the required local sub-grid of the hierarchical triangular mesh. Each binning process is computationally independent, because the vertices of successive sub-grids are calculated as required when iterating toward the desired grid resolution. The overall procedure is therefore what is commonly referred to as 'embarrassingly parallel'.
2. To test whether a point lies in a bin or not we implement a fast ray-tracing algorithm (Sunday, 2001) originally developed for computer graphics (Möller and Trumbore, 1997).
3. We present an efficient methodology to store binned data points in a database.
4. Production of gridded data products from such a database proceeds using an algorithm that considers the spatial density of data points. We discuss the relationship between on-target observation spacing, field of view size and grid resolution, and how it may be used to tune parameters involved in production of gridded data.

2. Binning onto an icosahedral grid

To bin data points, a grid must first be defined in order to test the spatial intersection of the data point with each bin. The tetrahedron, octahedron and icosahedron are all acceptable starting grids that are constructed from triangular faces. Octahedral meshes have been favored by some approaches because their vertices can occupy cardinal points and edges align with the 90th, 180th and 270th meridians (Dutton, 1996). However, to demonstrate our method we choose an icosahedron as a starting grid because (i) an icosahedron more closely approximates a sphere than an octahedron, minimizing the deviation from equal-area of new triangles formed from bisecting triangle sides, and (ii) because this approach builds specifically on previous work that uses an icoshedron as a starting grid (Teanby, 2006).

A vertex v_n is defined as a position vector in three dimensions $v_n = [x_n, y_n, z_n]$ and each triangle is defined by 3 such vertices, e.g. as in Fig. 1 where the triangle t_0 is defined by the vertices [v_0, v_1, v_2]. The total number of triangles in an icosahedron-based grid is given by $20 \times n^{2l}$, where *l* is the level, or number of iterative subdivision of triangle sides and *n* is the number of segments a triangle side is subdivided into at each new level. *n* and *l* must be integers. Bisection of triangle side lengths is the simplest technique and we focus only on that in our example, but *n* and *l* may be tailored so that a desired triangle side length, *q* is reached. *n* may also vary as iteration proceeds so that a desired triangle side length may be reached that is not available through successive bisection alone. For $n = 2$ the number of triangles is equal to 20×4^l; at each new level the grid contains 4 times as many triangles as the previous grid level.

Table 1 lists the total number of triangles and memory that would be occupied at each level if global meshes were to be stored in their entirety. Storage of meshes is assumed to be two linked lists of vertices and faces, to minimize disk usage.

Coarse resolution icosahedral grids, i.e. with low values of *l*, may be sufficient to represent global trends where fine detail is not required. For example, global trends in the Moon's elevation are adequately resolved by binning altimetry data acquired by LRO's Lunar Orbiter Laser Altimeter (LOLA) onto an icosahedral grid with $n = 2$ and $l = 7$ (Fig. 2), giving a triangle side length of 17.91 km (Table 1). For broad summary products, the entire grid may be held in a relatively small amount of computer memory and, assuming implementation of the binning algorithm as a single process, the compute time depends primarily on the number of individual data points.

However, for many global point-based planetary datasets the target-projected field of view (FOV) can be very small relative to the target body. Spatial information is therefore lost when FOVs are binned onto grids with spacings that are much larger than FOV dimensions. To maximize preservation of spatial information when producing mapped data products from raw data, bin size should adequately sample the FOV. In addition to LRO Diviner, this approach is applicable to any point-based planetary dataset, such as returned by e.g. Mercury Laser Altimeter on board NASA's MESSENGER (Cavanaugh et al., 2007), the Lunar Exploration Neutron Detector (LEND) also on board LRO (Mitrofanov et al., 2010), or the NOMAD instrument on board ESA's 2016 Trace Gas Orbiter mission (Thomas et al., 2015).

2.1. Binning algorithm

Impractical volumes of computer memory are required to store meshes with levels that correspond to instrument fields of view smaller than a few hundred meters. In parallelizing the binning process we can reduce computation time in multi-CPU/core environments, which are now typical. The total processing time is reduced by approximately a factor of the number of concurrent computational processes assigned to binning data points into triangles.

Sunday (2001) presents an algorithm to test whether a ray intersects a triangle by calculating the parametric coordinates, *t*, *s* of the intersection point. As an improvement to the popular and efficient algorithm by Möller and Trumbore (1997), Sunday's approach is more efficient when triangle normals are pre-calculated, because it requires computation of only a single cross product, whereas Moller and Trumbore's calculates two regardless of whether the normal exists or not. *t* and *s* are positions on axes defined by the two of the triangle sides, represented by the unit vectors ***v*** and ***u***, respectively (Fig. 1). Geometrically, the procedure can be thought of as translating the triangle so that v_0 is at the origin and transforming it to a unit triangle in the plane *F*, with the ray direction aligned with *z*. The intersect point *p* is within the triangle when $s \geq 0$, $t \geq 0$ and $s + t \leq 1$. At vertex v_0, $t = 0$ and $s = 0$.

Implementation of an optimized version of the algorithm written in the C language is detailed in Sunday (2001), and we here implement the same optimized algorithm in FORTRAN as the pro-

Table 1
Memory consumption of global icosahedral grids for levels 0–14, for $n = 2$. 3 point vectors of x, y, z coordinates are required to store the 3 vertices of each triangle, so that each triangle occupies 72 bytes when coordinates are stored as 8-byte floating point numbers. The memory required is only to store the total grid if it were to be calculated in its entirety prior to binning, which our method avoids by calculating and storing only the local sub-grid. Triangle dimensions are given for a general spherical case and when the radius of the target body is that of the lunar sphere, $R_M = 1737.4$ km.

Level, n	Num. triangles	Memory	General case		Lunar sphere, $R_M = 1737.4$ km	
			Tri. side (mrad)	Tri. solid angle (sr)	Tri. side length (km)	Tri. area (km^2)
0	20	1.4KiB	836.119	6.28×10^{-1}	1923.228	1601630
1	80	5.6KiB	560.904	1.57×10^{-1}	1091.452	515,834
2	320	22.5KiB	315.416	3.93×10^{-2}	566.931	139175
3	1280	90KiB	163.337	9.82×10^{-3}	286.333	35501
4	5120	360KiB	82.217	2.45×10^{-3}	143.166	8875
5	20,480	1.4MiB	41.201	6.14×10^{-4}	71.623	2221
6	81,920	5.6MiB	20.614	1.53×10^{-4}	35.819	555
7	327,680	22.5MiB	10.308	3.83×10^{-5}	17.910	139
8	1,310,720	90MiB	5.154	9.59×10^{-6}	8.955	34.7
9	5,242,880	360MiB	2.577	2.40×10^{-6}	4.478	8.68
10	20,971,520	1.4GiB	1.289	5.99×10^{-7}	2.239	2.17
11	83,886,080	5.6GiB	0.644	1.50×10^{-7}	1.119	0.543
12	335,544,320	22.5GiB	0.322	3.75×10^{-8}	0.560	0.136
13	1,342,177,280	90GiB	0.161	9.36×10^{-9}	0.280	0.0339
14	5,368,709,120	360GiB	0.081	2.34×10^{-9}	0.140	0.0085

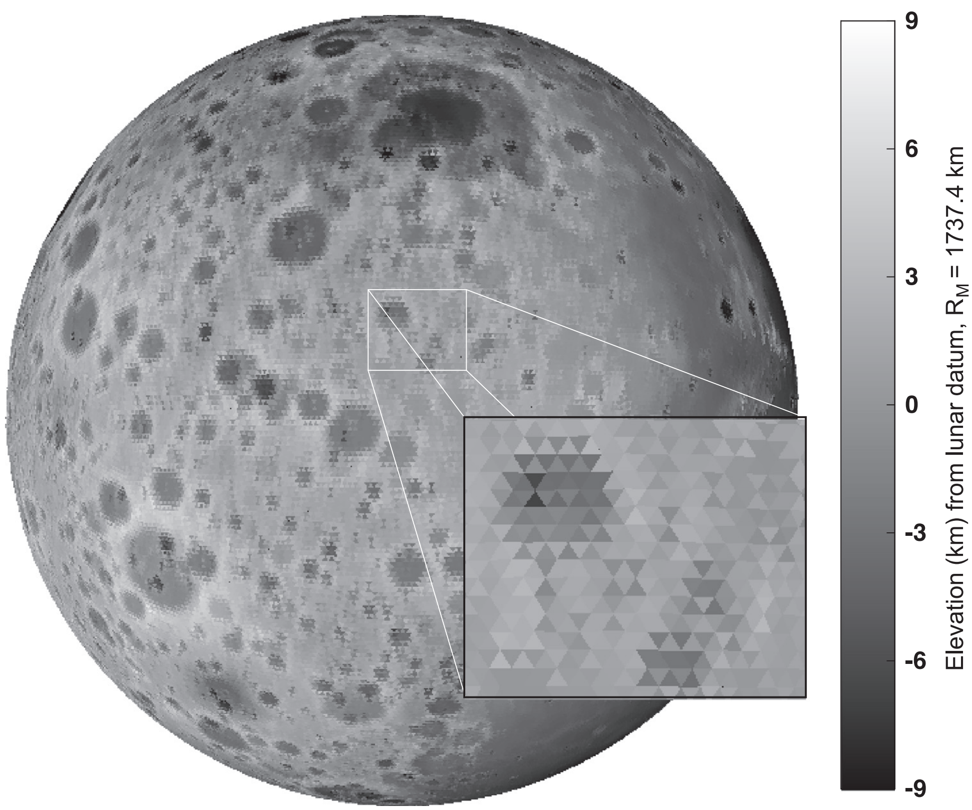

Fig. 2. Mean lunar elevation sourced from LRO's LOLA gridded products (Neumann, 2010) binned onto an icosahedral grid where $n = 2$ and $l = 7$. At this coarse spatial resolution global topographic trends are illustrated, but surface details smaller than the size of individual triangular bins are not resolved.

gram *ptribin* (supplementary materials). Sunday's algorithm contains a branch that culls of back-facing triangles. To optimize for speed, we implement only the branch that considers two-faced triangles. There is no risk of premature intersection with other triangles because R is defined outwards from O, triangle faces never overlap and triangle face normals are always in the direction of R.

To determine in which triangle a data point lies for a given level grid, we first define a starting icosahedron. Any data point must lie in one bin of a global grid, because grid cells do not overlap at their boundaries.

Our algorithm then iterates as follows:

1. A unit vector is defined as a ray (R) from the origin of the body (O) to a point defined by the latitude (ϕ) and longitude (θ) of the observation, converted into Cartesian coordinates. R must pass through one triangle in the grid.
2. The ray is tested for intersection with each triangular bin in the grid by calculating the parametric coordinates t and s using the algorithm detailed in (Sunday, 2001).
3. If the ray intersects, then (i) the number of the triangle in the grid is appended to an address for the data point, which defines its location in the grid, (ii) for the intersecting triangle each of the edges are divided by 2, making 4 new triangles (Fig. 3), (iii)

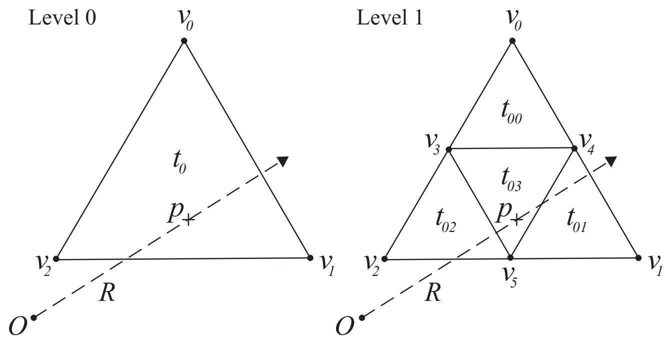

Fig. 3. Recursive procedure to subdivide triangular faces. When a ray R, drawn from the origin O through the point to be binned p, intersects with one of the faces in the current grid then the sides of that face are bisected, forming 3 new vertices, v_3, v_4 and v_5. These vertices are normalized to the unit sphere. The level 0 grid, an icosahedron of which t_0 is a member, is deallocated from memory and a new sub grid of 4 triangles (t_{00}, t_{01}, t_{02} and t_{03}) is formed from the new vertices plus the three vertices of t_0 (v_0, v_1 and v_2). The process iterates as intersection of R with each of the new faces is tested via the method of Möller and Trumbore (1997) (Fig. 1).

the level of the grid, n, is incremented by 1, and (iv) the 4 new triangles are assigned as the new grid in memory and the previous grid is discarded. R must intersect with one of the triangles in the new grid.

4. Steps 1–3 are iterated until the desired value of n is reached. When an intersection occurs in a grid of level n the total address of the data point is output. The address is a unique identifier for the bin in which the data point lies. In this case the first two digits of the address are the triangle number of the starting icosahedron and therefore are < 20. Subsequent digits are always < 4. With the starting icosahedron being a level 0 grid, the number of digits in the address is therefore given by $n + 2$.

For near-spherical bodies such as the Moon, which has an oblateness of 0.0014 (Smith et al., 1997), the effects of latitude on triangular bin area are negligible. Consequently lunar gridded map products are very often projected onto the widely used lunar reference sphere, where the lunar radius, $R_M = 1737.4$ km (Seidelmann et al., 2002). Consideration of non-spherical bodies in our application would therefore not produce a significant improvement to the spatial accuracy of mapped products and thus we assume an invariant lunar radius when defining the grid. Should it be desired, formulation of grids for more oblate bodies is detailed by Teanby (2006).

Due to the spherical nature of the initial triangles, i.e. triangle sides are arcs on great circles, the center triangle in each subdivision (Fig. 3, triangle $v_3v_4v_5$) has a slightly larger area than those at the corners. The effect is most pronounced where subdivided triangles are the largest, i.e. $l = 1$ and $n = 2$, where the curvature of a spherical triangle is greatest. In assessing the deviation from equal-area as a function of l for $n = 2$, we find that in the most extreme, but rare, cases spherical triangles that are always at the center of subdivisions can be up to $\sim$ 121% the size of an equivalent equal-area triangle at the same grid level, while those always at the edge can be just $\sim$ 61% of the area. The cumulative disproportion in area plateaus for $l > 4$ for center triangles and $l > 8$ for edge triangles, with the vast bulk of the difference introduced where triangles are largest relative to the sphere ($l < 4$).

An equal-area grid is not absolutely required by the technique we present, because the weight of each point is calculated based on the area it covers, and is propagated through the binning process. However, equal-area grids are necessary for datasets where the weight of data points in relation to the area they cover is not known, because unequally-sized bins would introduce binning artifacts and may misrepresent data. Despite not being necessary for our technique, equal-area grids remain more desirable because consistent spatial sampling is ensured. An equal-area correction for icosahedral meshes may be implemented using bubble meshing (Shimada and Gossard, 1995) or coordinate adjustment (Tegmark, 1996). We intend that our approach should include the coordinate adjustment procedure of Tegmark (1996). The procedure applies a series of rotations to any vertex on the icosahedron, and applies the adjustment as reduced to a single right triangle in x and y, before rotating the vertex back to its original position. We recommend inclusion of such a correction, but in this paper we consider a simple case that does not include the adjustment.

The area of a non-spherical equilateral triangle, Δ, is given by $\sqrt{3}q^2/4$, where q is triangle side length. Assuming flat, equal-area triangles we assess the relationship between Δ, grid level and the total number of triangles for an icosahedral grid the size of the lunar reference sphere where, $R_M = 1,737,400$ m (Fig. 5).

3. Choice of appropriate grid resolution

Telescope observations have a non-infinitesimal field of view (FOV) that is defined by instrument specifications and observation geometry. The goal of our approach is to bin observations onto a geodesic grid so as to store and retrieve them without loss of fine spatial detail in the highest resolution data. The size of the ground-projected FOV for an observation dictates the maximum useful spatial resolution of the data obtained. Consequently the ground-projected FOV size also dictates the minimum useful size of the triangular bins that is required to ensure that spatial information in observations is not needlessly oversampled at the expense of computer memory. Grid level, l, is therefore selected to match the smallest FOV that is possible for the combination of instrument specifications and spacecraft observation geometry.

Spatial information in raw data would be fully preserved if data were Nyquist sampled, where bin spacing is half the size of the smallest FOV. In this case some information is lost from observations with the smallest FOVs, which are undersampled. Conversely, those with the largest FOVs are oversampled. Such a compromise represents the balance between competing goals to minimize data volume and maximize preservation of spatial information in raw data. However, given sufficient computational resources, l may be set so that all FOVs are at least Nyquist sampled.

Thusfar we have referred only to the field-of-view in the general case, but this can refer to both the instantaneous field-of-view (IFOV) or the effective field-of-view (EFOV), which represents the cumulative signal that is incident on the detector over course of the observation. The IFOV for a single detector is represented by 2D probability function in the focal plane of the detector. It describes the relative response of the detector to incident radiation as a function of the angle between the incoming radiation and the normal to the focal plane. The mean IFOV of Diviner's 189 detectors (Paige et al., 2010) is plotted in Fig. 6. An instrument's IFOV may be altered by smearing in the in-track direction due to spacecraft motion over the integration time, or by a time-dependent response of the detector. An approach to quantify the effects of spacecraft motion and detector response time on the IFOV to calculate the EFOV is presented by Williams et al. (2016). We consider EFOVs in this study, but for simplicity and comprehension we refer to the general case, FOVs, unless explicitly stated. We do not consider uncertainties on the placement of the FOV, but note that these may be present due to uncertainties in spacecraft pointing or clock kernels. With the shape of the FOV constrained (Fig. 6 and Williams et al., 2016), the size of the projection of the FOV onto the target body is then dependent only on the observation geometry.

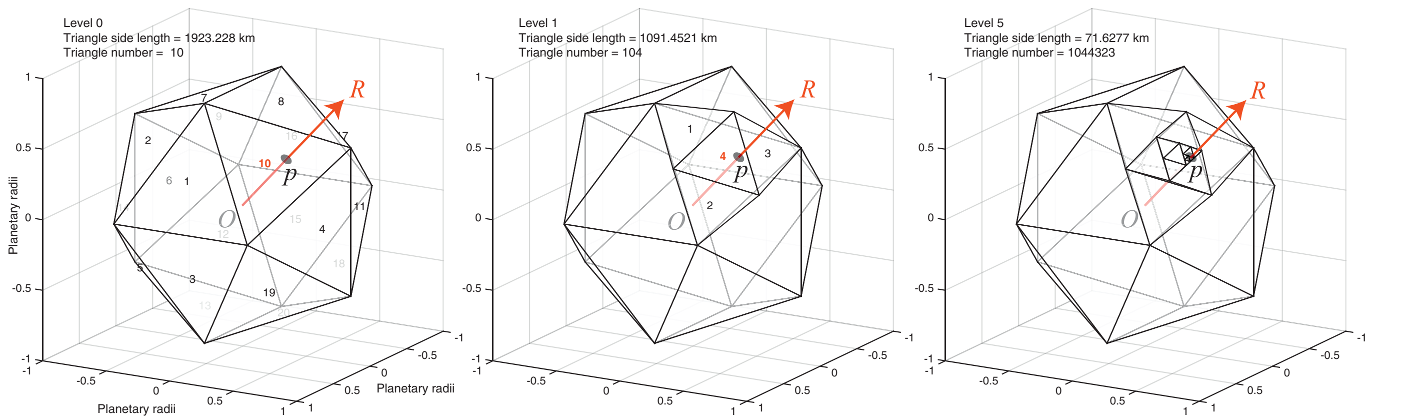

Fig. 4. Iterative subdivision of a Moon-sized icosahedron to determine the triangular bin that a point lies in. Left: An icosahedron is defined where vertices lie on a sphere of lunar radius (R_M = 1737.4 km). Triangular face numbers are labeled. Ray R is defined from the origin, O, through a point, p, on the surface that corresponds to the geographic coordinates of an observation. The ray is found to intersect with face 10 and the triangle number, or address, of the point is set to 10. Middle: Sides of triangle 10 are bisected, creating level 1 triangles 1–4. New vertices are normalized to the lunar radius, so that the level 1 subgrid more closely approximates the lunar sphere. The level 0 icosahedron is erased from memory. Each level 1 triangle is tested for intersection. R is found to intersect with triangle 4. The triangle number of p is appended with 4. Right: The process iterates until a level with the desired triangle side length is reached. Here, an observation at p has been binned into triangle 1044323 on a level 5 icosahedral grid where $n = 2$.

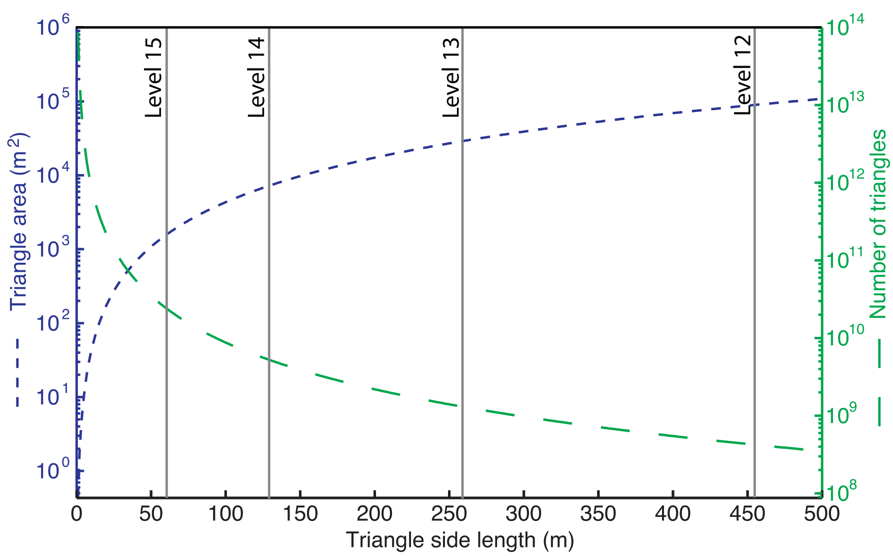

Fig. 5. Area and total number of triangles in an icosahedral grid (with $n = 2$) as a function of triangle side length, calculated for the lunar reference sphere (R_M = 1, 737, 400 m). Triangle areas are calculated for flat surfaces rather than those projected onto a sphere, i.e. triangle sides are linear.

In Diviner's nominal mapping mode the elevation actuator positions the boresight at nadir so that the emission angle for observations is ~ 0°. This configuration is desirable for systematic mapping of planetary surfaces because it minimizes elongation of the FOV and therefore maximizes the effective spatial resolution of observations. Assuming a flat body, changes in emission angle across the FOV due to curvature of the target are negligible in this case because the Moon is large compared to the ground-projected FOV. Therefore the dominant influence on the spatial resolution of the observation is spacecraft altitude. For circular orbits spacecraft altitude does not significantly vary, but for elliptical orbits the spatial resolution of observations is best at periapsis and worst at apoapsis.

We model an example of a ground-projected Diviner FOV as a function of spacecraft altitude (Fig. 7). Surfaces of constant probability are created by drawing contour lines on the FOV probability distribution shown in Fig. 6, then extruding them along the spacecraft altitude axis (Fig. 8). LRO's orbital altitude has varied between ~ 20–200 km over the commissioning, nominal, science and extended science mission phases (Fig. 7). Since December 2011 LRO has been in an elliptical orbit with its periapsis oriented over the lunar south pole, allowing the best spatial resolution data to be acquired of the south polar region.

At a common best-case altitude of ~ 30 km the majority of Diviner's FOV, when projected onto the lunar sphere, occupies an ellipse approximately 160 × 120 m in size, with its major axis oriented in the in-track direction. In selection of grid resolution we aim to strike a balance between maximizing preservation of spatial information in raw data while minimizing processing time and storage volume. Triangle spacing should therefore be between the Nyquist sample length for smallest surface-projected FOVs, but not so small as to grossly oversample the largest surface-projected FOVs. Consideration should also be given to the relative proportions of the dataset with different FOV sizes. For example, if small surface-projected FOVs are rare in the dataset, with the bulk of observations mostly having a consistently large FOV, then it may be more desirable to set l appropriately to sample large FOVs, accepting information loss in only a small fraction of the observations. In

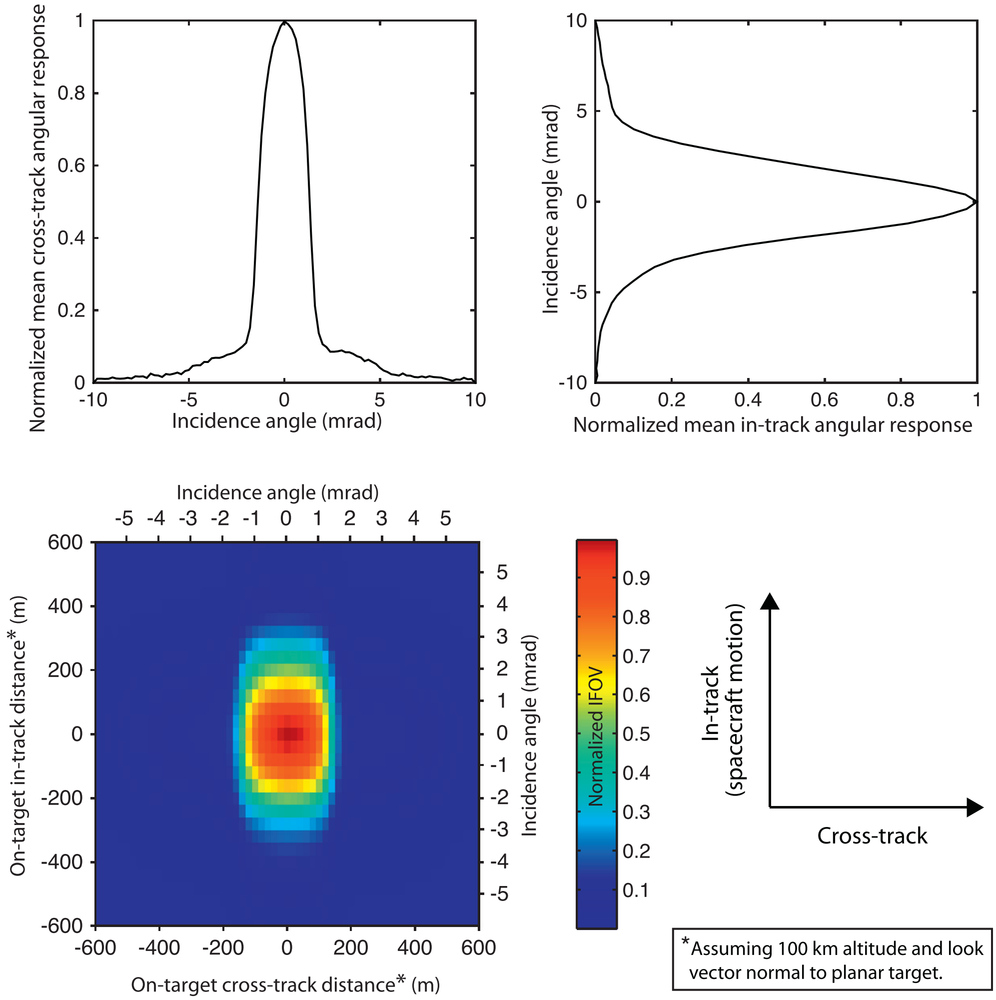

Fig. 6. The mean instantaneous field-of-view (IFOV) for LRO Diviner (Paige et al., 2010), calculated by averaging the in-track and cross-track angular responses for all detectors then convolving them to form a 2D probability distribution in the focal plane. For simplicity, here we consider only the channel-averaged IFOV, but this represents a key step in modeling an observation's 'effective field-of-view', or EFOV (Williams et al., 2016).

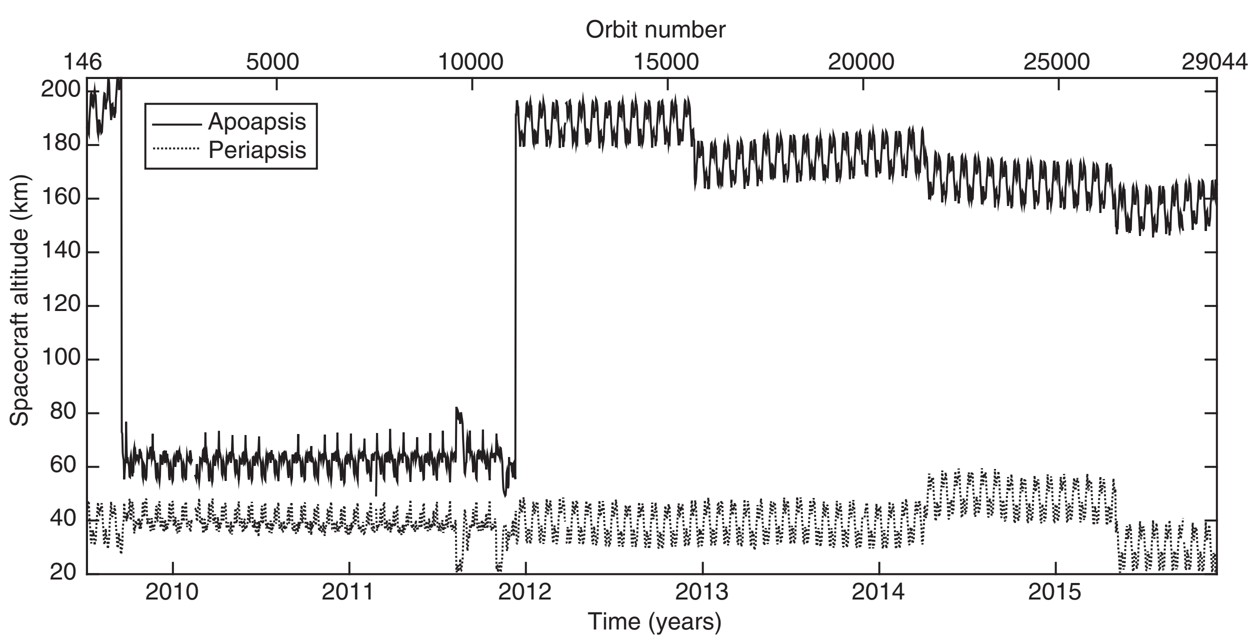

Fig. 7. Periapsis and apoapsis of the Lunar Reconnaissance Orbiter above the lunar reference sphere since Diviner's first orbital observation of the Moon. The distance from Diviner's focal plane to the lunar surface determines the ground-projected size of the FOV, as modeled in Fig. 8.

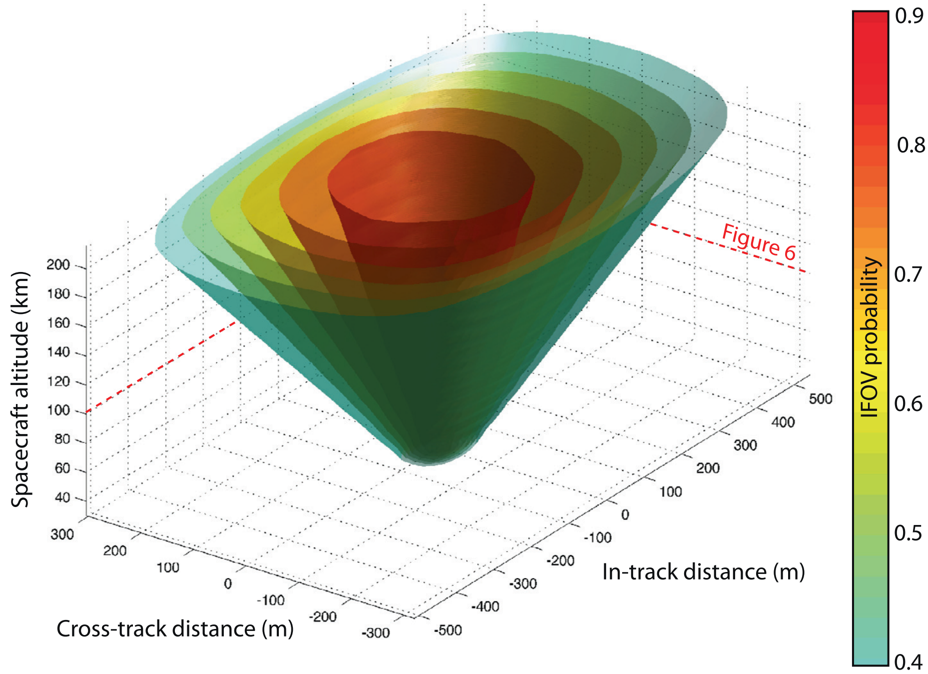

Fig. 8. Surfaces of constant probability created by drawing contour lines on the FOV probability distribution and extruding them along the spacecraft altitude axis. The plane in which Fig. 6 is calculated is defined by the dashed red line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

this case, a suitable grid may be constructed by only bisection of triangle sides down to l=14, where triangle sides are ∼ 140 m in length (Fig. 5).

4. Construction of a database

In addition to modeling the FOV, we also perform geometry calculations to project points onto a digital elevation model of the target surface, using the technique described in Teanby (2009). Both calculations are computationally expensive, and would normally be performed every time a gridded data record is constructed, causing unnecessary repeat processing for data that is used in multiple map products. It is therefore desirable to pre-process and store a database of observations that may then be directly gridded into map products without further processing. However, for even modest size datasets, several challenges regarding data volume must be overcome in order to store and query a database on such a scale. Here we describe our procedure to bin and store observations efficiently.

To store the spatial information contained in each observation, the FOV is modeled by populating the probability distribution with a cloud of n_{fov} points, whose spatial distribution represents the FOV (Fig. 9, panels a and b) (Williams et al., 2016). Point density is highest in locations that are the most significant in the observation. One cloud of modeled points (e.g. Fig. 9b represents a single Diviner observation and each point therefore represents some fraction of that observation. If the original observation has weight, w, then each of the n_{fov} points in the FOV represent w/n_{fov} of the observation. Each point carries the same brightness temperature value as the original because only a single observation was made.

Using the algorithm defined in our program *ptrinum*, each point in the FOV is binned into a triangle on the target grid (Fig. 9b). Our approach does not require the total grid to be pre-defined or stored in memory in its entirety, because the algorithm is iterative and operates only on the triangles in the current level's sub-grid, disregarding all others. A triangle is represented uniquely by a number that is built during iteration, with successive digits identifying triangles that the point was found to intersect with at each level (Fig. 4). Binning points onto the icosahedral grid is computationally parallel, because the triangle number of each point is independent of others. Our method therefore occupies very little memory and benefits from significant speedup in a multicore or cluster environment.

n_{fov} must be sufficiently large to ensure that all bins in the projected FOV are populated by the modeled point cloud, so that the spatial extent of the observation is completely resolved on the grid. This eliminates the occurrence of empty bins in areas where the surface contributed to an observation. Typically, n_{fov} is equal to a few hundred to a few thousand, with the modeled FOV tending to the actual FOV with increasing n_{fov}. However, the resulting data volume also becomes impractical to store or query with increasing n_{fov}. To store a representative approximation of the FOV the data volume must be reduced.

We therefore employ several techniques to compress data. Firstly, points that lie in the same triangular bin are gathered and their weight summed (Fig. 9c). This is accomplished computationally by the program *ptrgather*. Gathered points' positions and weights representatively sample the FOV as projected onto the grid. While this technique does result in some loss of spatial information by resampling onto the triangular grid, it allows for a reduction in data volume by a factor that is proportional to the mean number of points in each triangle.

Using real observations, we demonstrate the FOV modeling and icosahedral binning process (Fig. 10). Diviner observations in channel 6 (responsive to 13–23 µm photons (Paige et al., 2010)) were selected in a geographic area that was observed during LRO orbit 502, acquired on August 5, 2009. The study area was selected to illustrate a common situation for Diviner data: FOVs contain a broad temperature distribution within their ground-projected footprints, the result of a SW-NE trending topographic feature and a solar incidence angle of ∼ 57° causing differential illumination conditions (observations in the scene were acquired over a period of ∼ 2.6 s and the local mean solar time was ∼ 15.785). An observation by LROC NAC (M1159678514RE) illustrates the study area under the

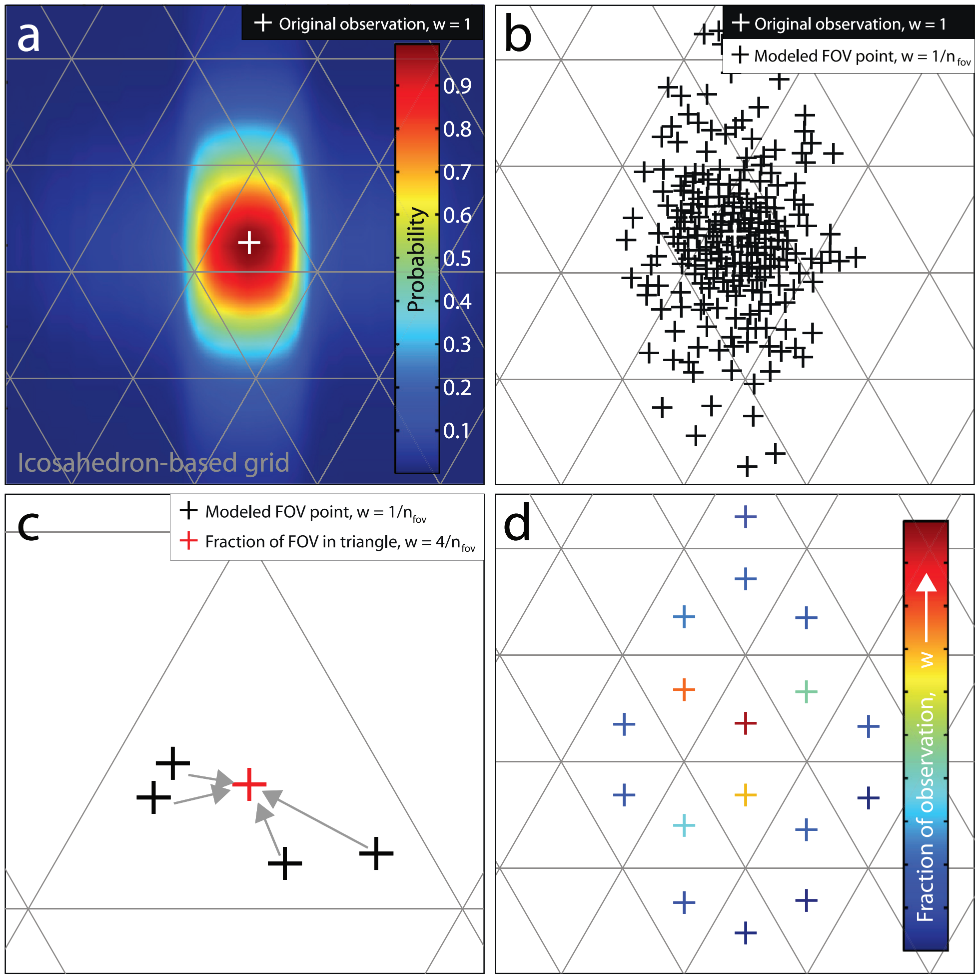

Fig. 9. Schematic pipeline for binning and gathering points in a modeled FOV onto a triangular grid (a) A single observation is a measured radiance at a discrete latitude and longitude on the target body (white cross), but is modeled as a probability distribution representing the FOV projected onto an icosahedron-based grid that approximates the target surface. (b) A Monte-Carlo population of n_{fov} points quantizes the FOV onto the grid. Point density reflects the relative contribution of radiance from the area observed. Each point represents a fraction $1/n_{fov}$ of the original observation and lies in one triangle on the grid. The triangle number is calculated for each point (e.g. Figs. 1, 3 and 4) using our program *ptrinum*. (c) Points are gathered in each bin to reduce their number by the program *ptrgather*. A gathered points' weight is equal to the sum of the ungathered points in the bin. (d) Gathered points' positions and weights representatively sample the FOV as projected onto the grid.

most similar available illumination conditions (Fig. 10a), and has a solar incidence angle of 50°.

Here we model the EFOV (Williams et al., 2016) of a single observation with a large number of Monte Carlo points ($n_{fov} = 10^4$) in order to well resolve its spatial extent (Fig. 10b). The EFOV is binned onto an icosahedral grid, where $l = 14$ and $n = 2$. The fraction of the observation that intersects with each triangle is represented in the database by weight, w (Fig. 10c). Here, the number of individual data points is reduced from 10^4 to 362, a reduction by a factor of $\sim$ 27.6, but an increase from the original single discrete point by a factor of 362. Additional methods are employed to further reduce data volume before storage in the database are described in Section 4.1.

To illustrate the process for data acquired in nadir pointing 'mapping' mode we also apply the procedure to the 140 total Diviner observations that are selected within the study area. 140 observations (Fig. 10d) are each modeled by 10^4 EFOV points (Fig. 10e). The resulting 1.4×10^6 EFOV points are then binned and gathered to 49,818 corresponding points on the icosahedral grid (Fig. 10f). Note that triangular bins are represented by points at their centroid and contain one point per triangle per intersecting EFOV. In this case there is significant spatial overlap of EFOVs, particularly in their low-probability edges. However, the central part of each EFOV remains the dominant contributor to the local signal.

4.1. Observations as linked lists

A Diviner observation comprises a single measurement of radiance at a geographic location and can therefore be represented by a single data record. Data fields in the record that are ancillary to the observation may include quantities such as emission angle, phase angle, Julian date or spectral channel. Diviner Reduced Data Records contain 33 such fields (Sullivan et al., 2013). Fields that relate to observation geometry such as emission, phase and incidence angle are specific to the precise latitude and longitude of the observation, whereas others, such as a timestamp or spectral channel number are invariant with location. Data records for modeled FOV and gathered points therefore only differ from the original observation record in those fields that relate to their location on the target surface. To further compress data volume we may therefore construct two linked lists. The first list contains one record per observation, with each record comprising fields whose values are independent of location. The second list contains one record per gathered point, with each record comprising fields whose values depend on location, and are unique for each gathered point in an observation. To reconstruct a full-length record for each gathered point, the row number of the original observation is stored as an additional column in the second list. Our program *ptrilink* accomplishes these tasks. At this stage the triangle number be-

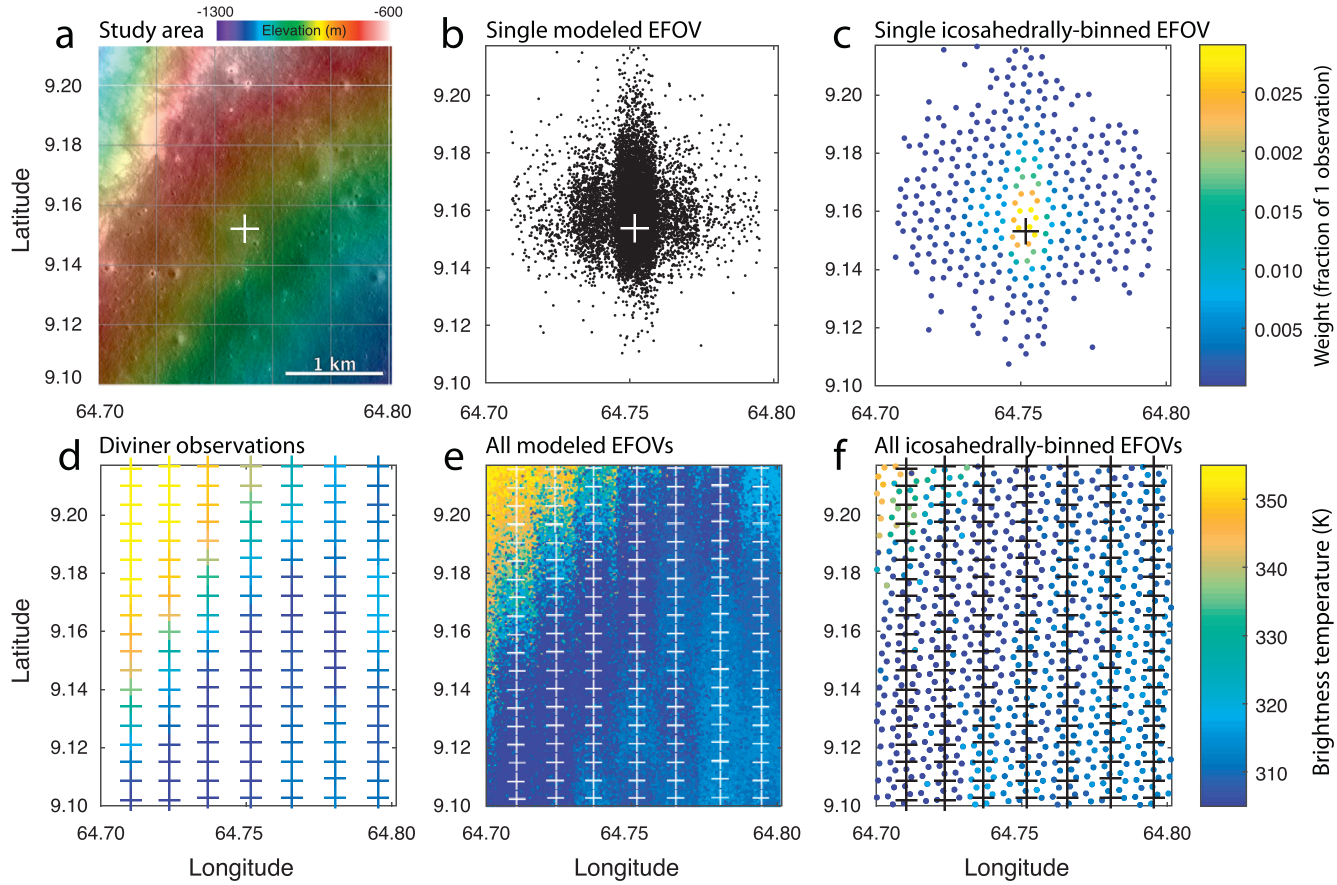

Fig. 10. Pipeline for binning and gathering points in a modeled EFOV onto a triangular grid as applied to real data. Data are brightness temperatures observed by Diviner channel 6 (~ 13–23 μm) during LRO orbit 502. Crosses mark the latitude and longitude recorded for each observation. (a) Study area: Portion of LROC NAC image M1159678514RE (NASA/GSFC/Arizona State University) with LOLA 128 ppd topography (Neumann, 2010) overlaid in color. A white cross marks the recorded location of a single Diviner observation shown in panels b and c. (b) Single Diviner observation is modeled as 10^4 discrete points via a Monte Carlo method described in Williams et al. (2016). (c) The modeled EFOV binned on an icosahedral grid ($l = 14$, $n = 2$) and gathered according to our method – one point per EFOV per triangle that is intersected, as in Fig. 9. Points are colored according to their weight, w, the fraction of the total observation that falls in each triangle. (d) 140 observations of brightness temperature in orbit 502. Colored crosses indicate brightness temperature. (e) EFOVs of the same observations modeled each as 10,000 points (colored dots). Locations recorded for each observation remain marked with white crosses. (f) EFOVs shown in panel e are binned and gathered onto the icosahedral grid, as demonstrated for the single observation in panels b and c. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

comes unnecessary and may be discarded to further save space, because each point's location is described by its latitude and longitude fields.

Linked lists are stored as tables in HDF5 files with 'blosc' compression (Alted and Vilata, 2002). For large datasets, files may be organized by commonly-queried fields such as latitude, channel number or date. Retrieval of data is performed using a query tool. To avoid long query times a simple indexing scheme is employed, where the maximum and minimum of each field is stored in a separate table. Upon a user query the index is consulted to determine which files contain points that meet the query constraints, and should therefore be decompressed and returned.

5. Production of gridded data records

Points returned from an icosahedral database query require no further processing and are ready to be binned onto Cartesian grids to create science-ready map products.

5.1. Relationship between grid size and data density

Data density in gridded data records is governed by the relationship between the on-target spacing between observations, l_o, and the spatial resolution of the grid onto which data are binned, l_g. For a set of typical Diviner data, l_o is effectively invariant, because observation spacing is pre-determined by the spacecraft orbit and observation geometry. For our application l_g is also fixed, because maps with consistent spatial resolutions must be produced en masse for release to a planetary data archive (NASA's Planetary Data System). Our goal is to produce maps with continuous data coverage within the cumulative EFOVs of many observations. For illustrative purposes we assert that the most significant part of a field of view is that within the full-width half maximum (FWHM). The FWHM may be as arbitrary as any isosurface of probability (e.g. as in Fig. 8), but provides a consistent definition: It is the part of the FOV where the signal from an observation is dominant in a gridded product built from observations acquired by a pushbroom instrument (i.e. the low probability edges of FOVs overlap, but ideally the FWHMs do not). In considering the relationship between l_o, l_g and the *FWHM* (Fig. 11) we find that four distinct regimes exist, each uniquely affecting the data density and uncertainty in gridded products.

The most effective transfer of information from observations to gridded product occurs when $l_o \approx l_g \approx FWHM$, or in other words, when there is roughly one observation per bin and when the significant portion of each observation is the same size as a bin. However, the idealized situation where observations are mapped one-to-one into pixels, is rare.

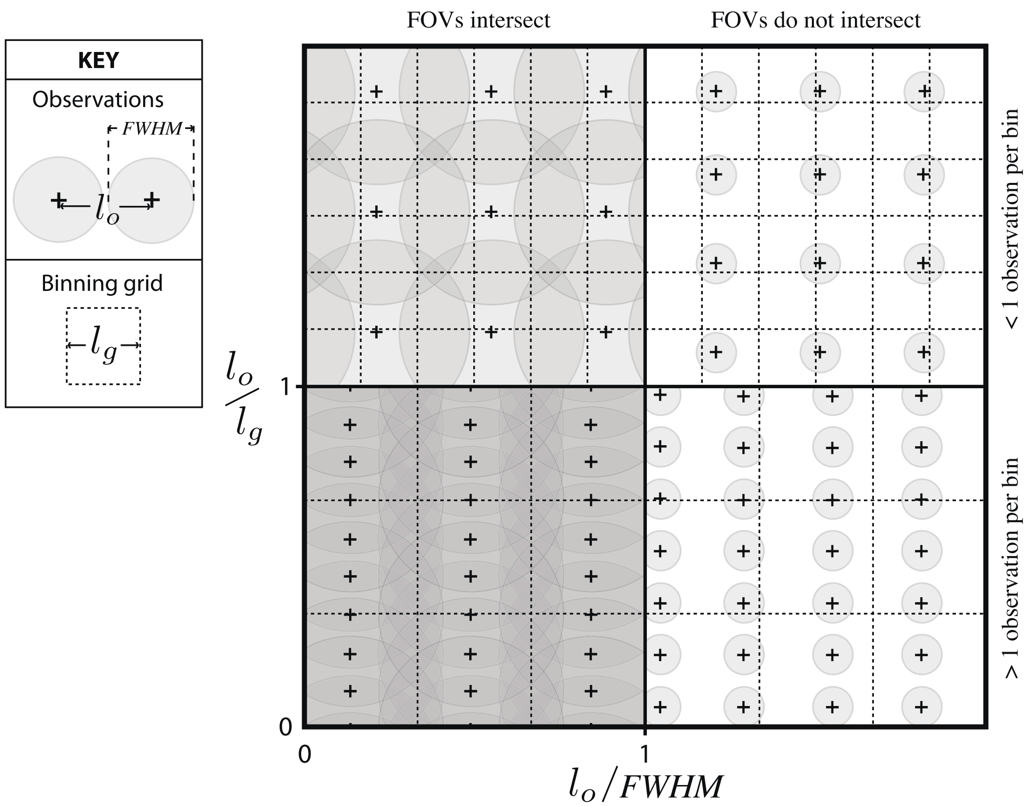

Fig. 11. Schematic of 4 binning regimes identified using ratios of observation spacing over grid spacing (l_o/l_g) and observation spacing over the most significant part of an observation, which for the purpose of illustration we arbitrarily define as the FWHM of the FOV. The in-track direction is aligned with the vertical axis. Observation spacing is consistent with Diviner's pushbroom mode of operation, where linear arrays of 21 detectors are oriented approximately in the cross-track axis and observe every 0.128 ms (Paige et al., 2010). Note that l_o refers to mean observation spacing for the general case – in the lower quadrants they are more closely vertically spaced than laterally to adequately represent the situation for Diviner observations.

5.2. Effect of varying n_{fov}

Prior to this method, Diviner GDRs have been produced by modeling EFOVs and binning them onto the target grid in a single step, which negated the need to store the large data volume produced by EFOV modeling. However, the value of n_{fov} was limited for computational practicality, i.e. to allow faster processing times that did not delay the regular data release schedule. Ideally, n_{fov} would be set as large as practically allowed, to maximize the fidelity of EFOVs. The construction of a database built with large n_{fov} is of high scientific value. In our example shown here we set $n_{fov} = 10^4$, a factor of 100 higher than the 10^2 points previously used to model each observation. It is more CPU-intensive to construct the database in this manner, but after construction it enables bespoke map products to be built quickly, and with better-resolved EFOVs. The greatest advantages occur when the same observations are used in multiple GDRs, as observations would have previously been processed for each GDR.

To illustrate the benefits of using a higher value of n_{fov}, we demonstrate the difference between 128 ppd maps built using EFOVs with $n_{fov} = 10^2$ and $n_{fov} = 10^4$ (Fig. 12). The observations included, and the study area, are the same as for Fig. 9. Maps of brightness temperature (Fig. 12A) and the total number of observations in each bin are produced from: raw observations, EFOVs resolved with $n_{fov} = 10^2$, $n_{fov} = 10^4$ and finally, those resolved with $n_{fov} = 10^4$ that have been retrieved from the icosahedral database. Brightness temperature maps appear relatively similar in all cases where the EFOV is modeled (i.e. excluding the map built with observations represented as single points), but there are clear differences in the number of observations per bin (Fig. 12B). Slight differences in the spatial distribution of each observations' EFOV are introduced by the disparity in EFOV fidelity between $EFOV_{100}$ and $EFOV_{10,000}$ (Fig. 12C, left). Differences are also introduced by the icosahedral binning and gathering process, which we illustrate as the difference between $EFOV_{10,000}$ and $ICOS_{10,000}$ (Fig. 12C, right).

However, differences between the brightness temperature maps (Fig. 12C) indicate that disagreements of up to 5 K in $EFOV_{100} - ICOS_{10,000}$ are due to the inability of the low-fidelity $EFOV_{100}$ to adequately sample the ground projected footprint of each observation, which in this case may contain a broad range of temperatures. Differences are largest where temperature heterogeneity in the EFOV is largest. The cumulative $EFOV_{100} - ICOS_{10,000}$ across the study area is 148.22 K, whereas a much smaller value of 59.05 K is calculated for $EFOV_{10,000} - ICOS_{10,000}$, indicating that increasing n_{fov} benefits data quality, and that the effect of icosahedral binning on a well-resolved EFOV does not appear to cause significant information loss.

To construct brightness temperature maps from the icosahedral database, we use the weighted mean of the distribution of fractional observations that fall in each bin. Each point retrieved from the database has a corresponding weight that is calculated as the fraction of the original observation that falls in a triangle on the icosahedral grid.

The weighted mean, μ^* and its variance, σ_w^2 are given by:

$$\mu^* = \frac{\Sigma_{i=1}^N w_i x_i}{\Sigma_{i=1}^N w_i} \quad (1)$$

and

$$\sigma_w^2 = \frac{\Sigma_{i=1}^N w_i (x_i - \mu^*)^2}{\Sigma_{i=1}^N w_i} \quad (2)$$

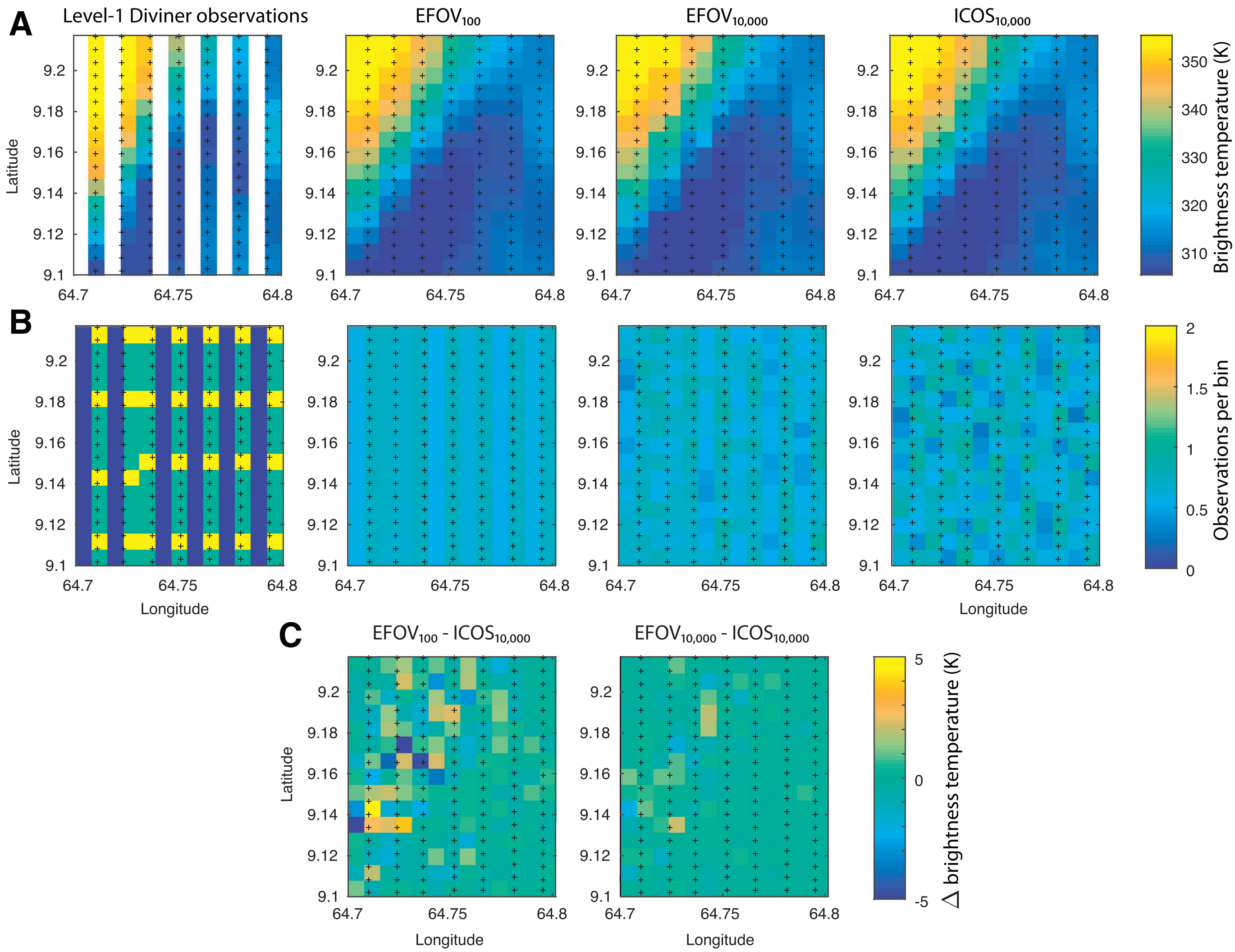

Fig. 12. Differences between binning EFOVs modeled with different numbers of Monte Carlo points, and storing them in the icosahedral database in order to produce 128 ppd gridded data records. Data and study area are as for Fig. 10. (A) Gridded brightness temperatures. Comparison between 128 ppd maps produced with, [left] 140 Diviner observations (EFOVs are not modeled), 100 [center-left] and 10,000 [center-right] Monte Carlo points, and [right] with 10,000 EFOV points retrieved from an icosahedral database (with $l = 14$, $n = 2$). (B) Total number of observations per bin in each brightness temperature map. For raw (level 1) Diviner observations, this can only be an integer. For EFOVs this is the sum of the total fractional observations in each bin, where the minimum value is $1/n_{fov}$. For icosahedrally-binned EFOVs it is the sum of the weights of fractional observations in each bin. (C) Left: Brightness temperature difference between maps built from EFOVs with $n_{fov} = 100$ and $n_{fov} = 10,000$. Right: Brightness temperature difference between maps built with and without icosahedral binning of EFOVs with $n_{fov} = 10,000$.

respectively, where x_i is each data point, N is the total number of points to be binned, and w_i is the weight of each data point. In our binning algorithm, two passes of the point cloud to be binned is therefore required, the first to calculate μ^*, and the second to use μ^* to calculate the variance on the weighted mean, σ_w^2. This most straightforward method is sufficient when the number of points per bin is small enough not to cause overflow or other errors caused by the limitations of floating point arithmetic. Chan et al. (1983) discuss the selection of appropriate algorithms to mitigate issues arising due to computational precision when the number of points per bin is very large. This would be important when e.g. calculating the mean of temperatures observed over a long time period, using large numbers of repeat observations.

5.3. Data density and interpolation

In general, FOV modeling increases the spatial continuity of map products, but it does not eliminate the occurrence of empty data bins on the target grid. When $l_g > l_o$ (Fig. 11 lower quadrants, where $l_o/l_g < 1$) there is on average more than one observation per bin, and therefore a high data density and low probability of empty bins. Additionally, large FOVs can reduce the occurrence of empty bins. However, when FOVs are small and observations are widely spaced (Fig. 11 top-right), empty bins are more likely to occur. Empty bins are possible even when FOVs are large and intersecting (e.g. Fig. 11 top-left), but are inadequately sampled. Undersampling of FOVs can occur if n_{fov} is set too low when building the database, or if the triangle size in the database approaches or exceeds l_g.

In general we would not recommend interpolating between Diviner observations on a grid, because it adds no information and risks misrepresenting the data. However, if the spatial density of observations offers sufficient confidence on the temperature field within the bin then an empirical approach may be applied to interpolate over empty bins. We illustrate a possible approach by binning data at 128 and 256 pixels per degree (Fig. 13). Data include those from Fig. 12, but cover a wider area to show an entire orbit swath. In this case, a 128 ppd grid resolution (Fig. 13 – top row) allows for $l_o/l_g \approx 1$, because the number of observations per bin is

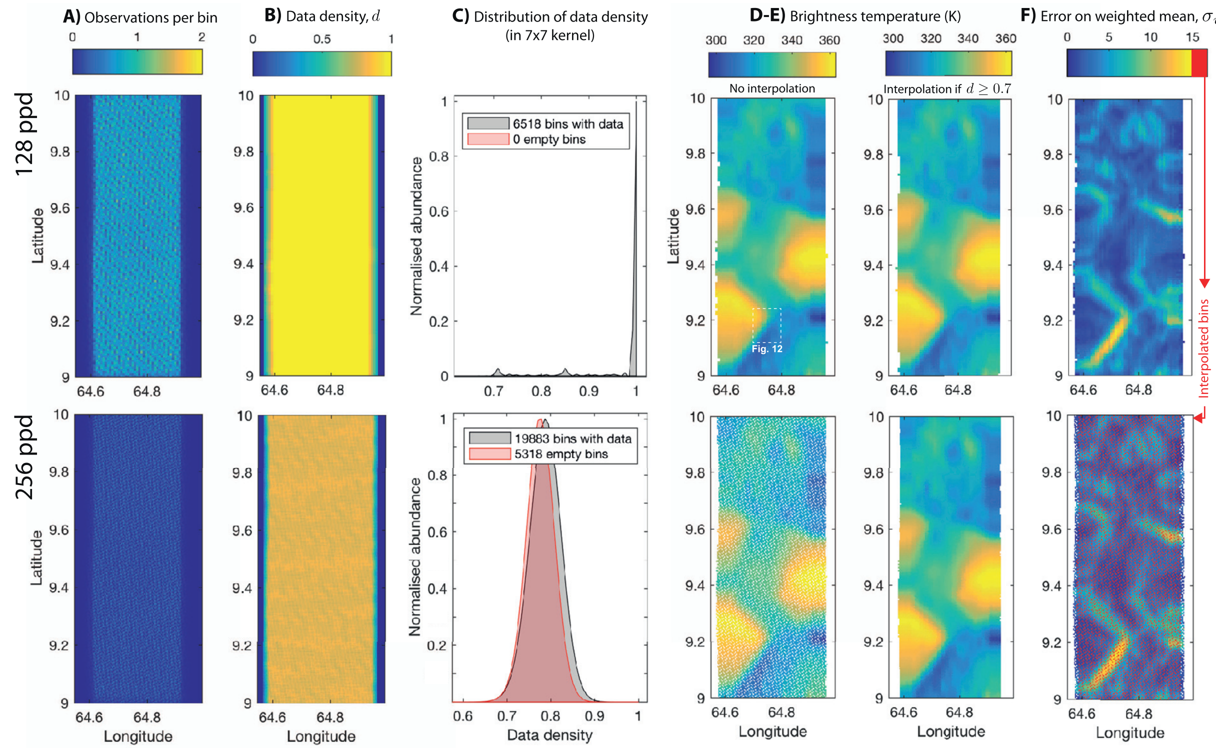

Fig. 13. The effect of, and empirical remedy to, cases where the FOVs are not sufficiently resolved to fill all bins that they intersect with. Data are binned onto 128 ppd (top row) and 256 ppd (bottom row) grids and include those from Fig. 12, but cover a wider area to show an entire orbit swath. (A) The total number of observations per bin, approximated by the sum of the weights of all database points in each bin. (B) Data density, d, as calculated in a 7 × 7 kernel using Eq. (3). (C) Distribution of data density in bins where there is data – if FOVs are well-sampled, there are no empty bins in the swath, as for the 128 ppd map, but not for the 256 ppd map. (D and E) Brightness temperature maps built by calculating the weighted mean, μ^*, of the brightness temperature distribution in each bin. The boundary of Fig. 12 is outlined by the dashed white box in the 128 ppd map in column D. For maps in column E, interpolation is performed for bins where data density is equal to or exceeds a threshold, $d \geq 0.7$, selection of which is informed by the distribution of data density in C. In both D & E, pixels are nulled where data density is below the threshold. This removes scattered and isolated data points at edges of the swath, where cumulative FOV weight is low. (F) Error on the weighted mean, σ_w. A special value is assigned to bins where interpolation has occurred (red), to indicate that no data was present and so it is not possible to calculate uncertainties for these bins. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

close to 1, whereas the value for a 256 ppd grid (Fig. 13 – bottom row) is lower by approximately a factor of $1/2^2$.

Data density, d, can be determined with a 2D kernel density function. Valid quantities to describe data density within the kernel include the fraction of non-empty bins, the mean number of data points per bin, or the mean observation weight per bin. For simplicity, we here use the kernel density of non-empty bins because it is independent of observation weight, and therefore does not vary as a function of orbital altitude. If the probability of a bin at grid coordinates (x', y') containing data is equal to either 0 or 1 then the density of data in a square kernel, with an odd integer side length, n, and center bin at (x, y), is given by:

$$\frac{1}{n} \sum_{x\prime=x-n}^{x+n} \sum_{y\prime=y-n}^{y+n} d(x\prime, y\prime) \tag{3}$$

d is therefore the fraction of non-empty bins in the neighborhood of each pixel (Fig. 13B). Interpolation can proceed in empty bins where the local data density exceeds some threshold, i.e. $d > d_t$. The distribution of data density (Fig. 13C) may inform the selection of the thresholds to be used to determine where to interpolate over empty bins. In this example we select $d_t = 0.7$, because it allows interpolation in the majority of the 5318 empty bins in the 256 ppd map (Fig. 13C and D – lower row), but excludes those with the lowest significance, which are likely to fall on swath edges. We use a Delaunay triangulation (Delaunay, 1934), though we have not assessed the relative merits of other methods (e.g. bilinear, cubic, spline). In addition to determining areas of sufficient confidence, data density can identify outlying, isolated bins that result from scattered, low probability fringes of EFOVs. This commonly occurs near the edges of orbit swaths. In these locations there is insufficient confidence in the temperature field to produce a spatially continuous map and bins may be nulled where $d < d_t$. After interpolation bins that contain data, but that are in areas of insufficient data density ($d < d_t$), are nulled. This is similar to the bin count cutoff applied in Williams et al. (2016). Interpolated bins do not contain real data and therefore contribute no statistical significance to the map. We record this in the uncertainty map by assigning those bins a reserved and identifying value, to ensure that science data are not misinterpreted (Fig. 13F – red bins).

The resulting interpolated 256 ppd map compares well to the non-interpolated 128 ppd version (Fig. 13E) in terms of continuity and apparent consistency. We present an extreme case where a significant number of bins are interpolated, but ideally, grid resolution would be selected such that data are not unnecessarily oversampled, so that $l_o/l_g \approx 1$.

5.4. Separation of orbits by swath

Our example deals with data from a single orbit, but the vast majority of Diviner mapped data products contain data from multiple orbits that may have spatially intersecting swaths. Averaging apparent brightness temperatures within overlapping swaths/FOVs that were acquired in the same orbit is an acceptable representation of the radiative state of the lunar surface, because subsequent observations are closely spaced in time, ∼ 128 ms (Paige et al., 2010). However, caution should be exercised when binning observations acquired at significantly different local times, because differences in illumination conditions between observations would cause gridded products to be less meaningful; they would average together multiple radiative states of the surface from different times of day. This becomes increasingly important for daytime observations with high solar incidence angles, because shadows may quickly sweep across terrains during LRO's approximately 2 h orbital period. We therefore adopt the approach demonstrated by (Hayne et al., 2015) who correlate thermal extremes with UV albedo to show evidence for surface H_2O at the lunar south pole. In their study, minimum, maximum and mean temperature maps are constructed by compounding bolometric brightness temperature grids binned on a per orbit basis.

We present a function *bin2dwbyorbit* and associated functions, that apply the binning method illustrated in Fig. 13, but also combine swaths from multiple orbits to produce broader scale map products. The weighted mean, data density and uncertainty of the brightness temperature field within each orbit swath is first calculated. Optionally, low data density bins are then interpolated and low probability swath fringes may be nulled. Individual orbit tracks are then combined such that in regions of overlap the unweighted mean of the temperature field is calculated, but total observation weight and uncertainty are propagated to the final map products. We assert that this approach adequately represents observations of the time-varying temperature field acquired in subsequent orbits.

6. Conclusions

A pipeline is presented to bin, store and retrieve globally distributed modeled fields of view for the production of gridded brightness temperatures from LRO's Diviner Lunar Radiometer Experiment. Our approach can be implemented in a highly parallel way and the construction of a database avoids repeat processing of data points. In optimizing the quality of science-ready data products, we have identified four binning regimes based on quantification of the spatial resolution of the grid, the size of the FOV and the on-target spacing of observations. We propose that calibration of the data density threshold, d_t should be performed based on a per-map basis, informed by our parameterization of the spatial characteristics of the data and grid. We have implemented this approach in the production of gridded data records from the Diviner Lunar Radiometer Experiment, but note that it is applicable to a wide range of existing and future point-based planetary datasets.

Acknowledgments

We thank the LRO and Diviner operations teams at NASA Goddard Space Flight Center, Jet Propulsion Laboratory, and the University of California Los Angeles for their support. Elliot Sefton-Nash, Jean-Pierre Williams, Benjamin Greenhagen, K.-Michael Aye and David Paige are supported by the Lunar Reconnaissance Orbiter program. We thank three anonymous reviewers for constructive feedback that helped to improve the manuscript.

Appendix A

Supplementary code and data associated with this article is available online in two online repositories. FORTRAN code to perform icosahedral binning is available at https://github.com/elliotsn/icosbin. Matlab code to construct map products by filtering FOVs and interpolating according to data density on a per orbit basis is available at https://github.com/elliotsn/bin2dbyorbit.

References

Alted, F., Vilata, I., 2002. Pytables: Hierarchical datasets in python. URL http://www.pytables.org.

Cavanaugh, J.F., Smith, J.C., Sun, X., Bartels, A.E., Ramos-Izquierdo, L., Krebs, D.J., McGarrym, J.F., Trunzo, R., Novo-Gradac, A.M., Britt, J.L., Karsh, J., Katz, R.B., Lukemire, A.T., Szymkiewicz, R., Berry, D.L., Swinski, J.P., Neumann, G.A., Zuber, M.T., Smith, D.E., 2007. The mercury laser altimeter instrument for the messenger mission. Space Sci. Rev. 131, 451–479.

Chan, T.F., Golub, G.H., LeVeque, R.J., 1983. Algorithms for computing the sample variance: analysis and recommendations. Am. Stat. 37 (3), 242–247.

Delaunay, B., 1934. Sur la sphère vide. Otdelenie Matematicheskikh i. Estestvennykh Nauk 7, 793–800.

Dutton, G., 1996. Encoding and handling geospatial data with hierarchical triangular meshes. In: Kraak, M.J., Molenaar, M. (Eds.), Advances in GIS Research II. Taylor and Francis, London, pp. 505–518.

Hayne, P.O., Hendrix, A., Sefton-Nash, E., Siegler, M.A., Lucey, P.G., Retherford, K.D., Williams, J.-P., Greenhagen, B.T., Paige, D.A., 2015. Evidence for exposed water ice in the moon's south polar regions from lunar reconnaissance orbiter ultraviolet albedo and temperature measurements. Icarus 255, 58–69.

Mitrofanov, I.G., Bartels, A.E., Bobrovnitsky, Y.I., Boynton, W.V., Chin, G., Enos, H., Evans, L., Floyd, S., Garvin, J., Golovin, D.V., Grebennikov, A.S., Harshman, K., Kazakov, L.L., Keller, J., Konovalov, A.A., Kozyrev, A.S., Krylov, A.R., Litvak, M.L., Malakhov, A.V., McClanahan, T., Milikh, G.M., Mokrousov, M.I., Ponomareva, S., Sagdeev, R.Z., Sanin, A.B., Shevchenko, V.V., Shvetsov, V.N., Starr, R., Timoshenko, G.N., Tomilina, T.M., Tretyakov, V.I., Trombka, J., Troshin, V.S., Uvarov, V.N., Varennikov, A.B., Vostrukhin, A.A., 2010. Lunar exploration neutron detector for the NASA reconnaissance orbiter. Space Sci. Rev. 150, 183–207.

Möller, T., Trumbore, B., 1997. Fast, minimum storage ray-triangle intersection. J. Graphics Tools 2 (1), 21–28.

Neumann, G.A., 2010. Lunar orbiter laser altimeter raw data set, LRO-L-LOLA-4-GDR-V1.0. NASA planetary data system. NASA Planetary Data System

Paige, D.A., Foote, M.C., Greenhagen, B.T., Schofield, J.T., Calcutt, S.B., Vasavada, A.R., Preston, D.J., Taylor, F.W., Allen, C.C., Snook, K.J., Jakosky, B.M., Murray, B.C., Soderblom, L.A., Jau, B., Loring, S., Bulharowski, J., Bowles, N.E., Thomas, I.R., Sullivan, M.T., Avis, C., De Jong, E.M., Hartford, W., McCleese, D.J., 2010. The lunar reconnaissance orbiter diviner lunar radiometer experiment. Space Sci. Rev. doi:10.1007/s11214-009-9529-2.

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 Mech. Dyn. Astron. 82 (1), 83–111.

Shimada, K., Gossard, D.C., 1995. Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing. ACM Third Symposium of Solid Modeling and Applications.

Smith, D.E., Zuber, M.T., Neumann, G.A., Lemoine, F.G., 1997. Topography of the moon from the clementine lidar. J. Geophys. Res. 102 (E1), 1591–1611.

Sullivan, M.T., Paige, D.A., Arvidson, R.E., Grayzeck, E., 2013. Lunar Reconnaissance Orbiter Diviner Lunar Radiometer Experiment: Reduced Data Record and Derived Products Software Interface Specification. Technical Report, Version 1.13. PDS Geosciences Node.

Sunday, D., 2001. Intersection of a ray/segment with a triangle. URL http://geomalgorithms.com/.

Teanby, N.A., 2006. An icosahedron-based method for even binning of globally distributed remote sensing data. Comput. Geosci. 32, 1442–1450.

Teanby, N.A., 2009. Intersection between spacecraft viewing vectors and digital elevation models. Copmut. Geosci. 35, 566–578.

Tegmark, M., 1996. An icosahedron-based method for pixelizing the celestial sphere. Astrophys. J. 470, L81–L84.

Thomas, I., Vandaele, A.C., Lopez-Moreno, J.J., Patel, M.R., Bellucci, G., Drummond, R., Neefs, E., Depiesse, C., Daerden, F., Rodriguez-Gomez, J., Neary, L., Robert, S., Willame, Y., Mahieux, A., the NOMAD Team, 2015. The exomars trace gas orbiter nomad spectrometer suite for nadir and solar occultation observations of mars' atmosphere. EGU General Assembly 2015. Geophysical Research Abstracts, vol. 17.

Williams, J.-P., Sefton-Nash, E., Paige, D.A., 2016. The temperatures of Giordano Bruno crater observed by the diviner lunar radiometer experiment: application of an effective field of view model for a point-based data set. Icarus 273, 205–213.