Icarus 000 (2017) 1–13

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 xxx

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/

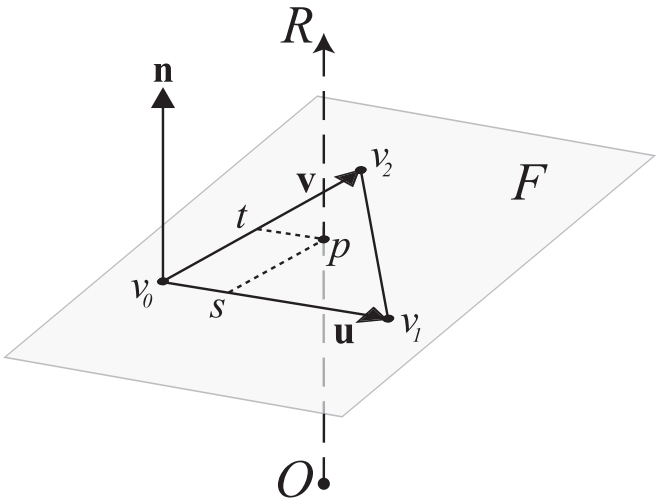

Fig. 1. Adapted from Sunday (2001). Method to test if a ray drawn from the origin to a point outside the unit sphere intersects with a triangle with vertices on the unit sphere. Triangle vertices define a plane, *F*. Parametric coordinates *t* and *s* are calculated as fractions of the unit vectors ***u***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

Please cite this article as: E. Sefton-Nash et al., Diviner lunar radiometer gridded brightness temperatures from geodesic binning of modeled fields of view, Icarus (2017), http://dx.doi.org/10.1016/j.icarus.2017.04.007