

Intersection between spacecraft viewing vectors and digital elevation models[☆]

N.A. Teanby

*Atmospheric, Oceanic and Planetary Physics, Department of Physics, Clarendon Laboratory,
University of Oxford, Parks Road, Oxford OX1 3PU, UK*

Received 3 September 2007; received in revised form 14 November 2007; accepted 19 November 2007

Abstract

A method is presented for determining the latitude, longitude, and altitude of each pixel in a remotely sensed image of a planet for the case where there is significant surface topography. The method works by sequentially stepping along the line-of-sight of each pixel in an image until an intersection with the planet surface is detected. The position and altitude of each pixel on the planet can then be used for further analysis and allows comparison with other data sets. For pixels where no intersection occurs the altitude and location of the tangent point are determined. These pixels are important as they provide views of the planet limb and are useful for studying the vertical structure of the atmosphere. Provision is made for reference to an oblate spheroid when calculating the tangent point, which is required for atmospheric applications on oblate planets such as the Earth. The algorithm requires a digital elevation model, along with the viewing geometry and position of the instrument. Illustrative examples are given using the Martian MOLA topography data set for oblique and limb viewing cases.

© 2008 Elsevier Ltd. All rights reserved.

Keywords: Data analysis; Remote sensing; Topography; Geometry

1. Introduction

In remote sensing we often have spatially resolved measurements of a planet. In a given image, each pixel corresponds to either a point on the surface or a point off the limb of the planet. The horizon separates the two cases but is not always obvious from the image alone, especially if an atmosphere is present. For correct interpretation of such data it is necessary to know if the line-of-sight of each pixel

intersects with the surface and if so where the intersection occurs. Measurements can then be correlated with local surface features or other data sets. If high resolution surface imaging is the primary mission objective, observations are usually acquired with a near-nadir viewing geometry. In these cases distortions introduced by topography are minimal and the latitude and longitude of each pixel are simple to determine using analytical geometry by approximating the planet to a sphere or oblate spheroid.

However, topography can introduce significant distortion when images are taken obliquely, such as when performing limb scanning of a planet's

[☆] Code available from server at <http://www.iamg.org/CGEditor/index.htm>.

E-mail address: teanby@atm.ox.ac.uk

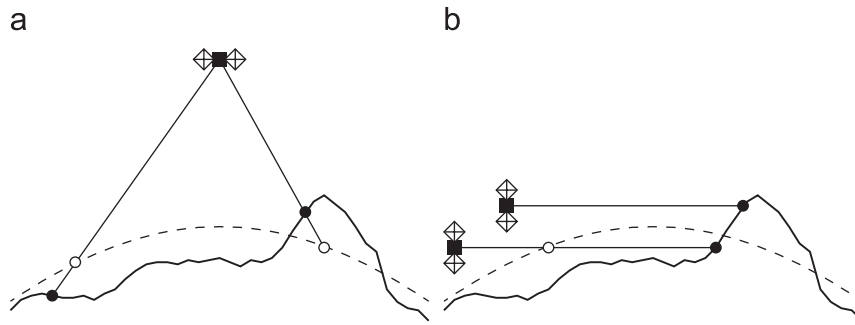


Fig. 1. Examples of how topography can influence the position of the intersection point between line-of-sight and surface. Symbols are: square with two diamonds, satellite; open circles, intersect points when planet is spherical (dashed line); solid circles, intersect points with illustrated topography (thick solid line). (a) Sub-nadir viewing geometry. Mountains cause intersect point to shift closer to spacecraft and depressions cause intersect point to move further away. (b) Shallow angle limb viewing geometry. Mountains can cause surface intersections with lines-of-sight that would otherwise be of the planet's limb. For shallow viewing angles the effect of topography on the intersect point can be very large.

atmosphere (e.g. Mars Climate Sounder (MCS) on Mars Reconnaissance Orbiter) or wide-angle imaging of small irregular bodies such as asteroids (e.g. the Near Earth Asteroid Rendezvous mission). If the planet/object has significant topography compared to the projected pixel size then viewing the surface at oblique angles significantly affects the location of the intersect point or even whether an intersect occurs at all. This effect is illustrated in Fig. 1. As the viewing angle increases and pixels get closer to the limb the effect of topography increases. For example, a ray that would just graze the surface of a spherical planet may intersect with a mountain that occurs before or after the tangent point (Fig. 1b). If these effects are not taken into account data could be misinterpreted.

Remote sensing observations are acquired for studying the surface, atmosphere, or both. For surface studies we would like to know whether the line-of-sight of each pixel in the detector array intersects with the surface, the latitude and longitude of the intersect point, the altitude of the local topography, and possibly the emission angle (angle between line-of-sight and zenith). Pixels that do not offer a surface view are generally not used for such studies.

For studies of the atmosphere we are also interested in pixels that do not intersect with the surface but instead offer a grazing incidence of the planet's limb. These data are extremely useful for determining the vertical structure of the atmosphere. Therefore, we need to know whether each pixel corresponds to a limb or surface view, the latitude and longitude of the tangent point, and the altitude of the tangent point above the local surface.

Here we consider the case where a gridded digital elevation model (DEM) is available and the spacecraft position and viewing vectors are known. DEMs are now a widely available and important resource for geological and geophysical applications. For the Earth, numerous topography data sets are available. For example, the USGS GTOPO30 data set,¹ which covers the entire globe at 30 arc sec (≈ 1 km) resolution or the recent very high resolution topography data from the Shuttle Radar Topography Mission (Farr et al., 2007) which has a resolution of about 1 arc sec (≈ 30 m). Topographic data sets are also becoming available for other Solar System bodies. The most widely used of which is the MOLA topography of Mars (Zuber et al., 1992; Smith et al., 2001). Examples of other DEMs are: Magellan topography of Venus (Ford and Pettengill, 1992); topographic models of Mars' moons (Thomas, 1993); optical shape models for small solar system bodies by Peter Thomas available from NASA's Planetary Data System²; and the topography of the asteroid EROS from the NEAR-Shoemaker mission (Zuber et al., 2000; Thomas et al., 2002).

The problem of determining the intersection of a ray with a gridded DEM has no analytical solution and a numerical technique must be employed. In this paper a method is presented for determining the geometric properties of each pixel in a remotely sensed image with respect to a DEM of the surface. The DEM is assumed to cover the entire globe, but the same method could easily be extended for use

¹<http://edc.usgs.gov/products/elevation/gtopo30.html>

²<http://www.psi.edu/pds/resource/oshape.html>

with local DEMs. When calculating the tangent point, the atmosphere of the planet is not assumed to be spherical, but instead made up of ellipsoidal shells. This extra complexity is required as most large planets with an atmosphere are best described by an oblate spheroid and the spherical assumption would introduce a bias in subsequent analysis.

Example applications are given for Mars using the MOLA DEM with both synthetic and real observations of the atmosphere and surface. The algorithm is implemented in FORTRAN and the source code is made freely available.

2. Method

The overall method involves incrementally stepping along the line-of-sight of each pixel until an intersection with the surface DEM is detected. If no intersection is detected the tangent point is determined analytically. By incrementally stepping along the line-of-sight we avoid having to re-project the entire DEM into the frame of the spacecraft, which would be computationally expensive given the large size of even modest resolution DEMs.

2.1. DEM format

The method requires a DEM that is evenly gridded in both latitude and longitude. Planetocentric latitude, longitude, and radius are used to define the DEM as they are equivalent to the spherical polar coordinate system, which simplifies the mathematics. If the DEM is in planetographic coordinates then it must first be converted into planetocentric coordinates and re-gridded onto an evenly spaced grid. Planetocentric coordinates are also more appropriate for irregular bodies such as small moons and asteroids.

The DEM information is stored in two files: (1) an ASCII file containing header information with file extension `.hdr` (see Table 1); and (2) a binary

file containing radii data with file extension `.dem`. The planet’s surface is split up into pixels and this file consists of an ordered list of radii from the planet centre to the surface at the centre of each pixel. Data are ordered by row then column, so all longitudes at a given latitude are specified first, then all longitudes at the next latitude, and so on. The resolution of the DEM is defined in terms of number of pixels per degree m_{pix} , which is assumed to be the same for both latitude and longitude coordinates. Therefore, the number of DEM pixels is $n_{\phi} = 360/m_{pix}$ in the longitude coordinate and $n_{\theta} = 180/m_{pix}$ in the latitude coordinate. The DEM format is shown in Fig. 2, and is similar to readily available DEMs such as GTOPO (Earth) and MOLA (Mars).

The use of a binary file saves disk space and allows individual records to be accessed directly without reading in the entire data set. This offers a time saving for cases with large, detailed, DEMs and sparsely measured data as radii can be read-in only if required. The record number k of the j th row (latitude) and i th column (longitude) is given by

$$k = (j - 1)n_{\phi} + i \tag{1}$$

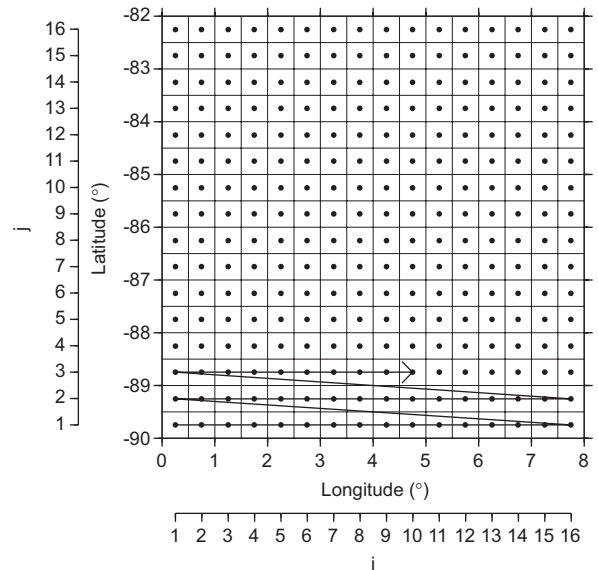


Fig. 2. Format of DEM. A rectangular area (in latitude–longitude space) is split into a fine grid of evenly spaced pixels. Pixel centres (spots) are each assigned a radius value. Longitudes are east-positive and latitudes are north-positive. Radii are stored in a binary `.dem` file ordered by longitude then latitude as indicated by the black winding arrow. Indices i and j refer to longitude and latitude grid positions, respectively. In this paper the DEM is assumed to have global coverage.

Table 1
Header information about DEM stored in `.hdr` file

Variable	Symbol	Type	Description
ppd	m_{pix}	Integer	Pixels per degree
rmax	r_{max}	Real	Maximum planet radius
f	f	Real	Oblateness of planet ^a
re	r_{eq}	Real	Equatorial radius ^a

^aThese quantities are only required if ellipsoidal tangent point is required, otherwise set $f = 0$ and $r_{eq} = 1$.

When k is known, the corresponding radius can be read directly from the binary .dem file as required.

2.2. Calculating the radius at arbitrary latitude and longitude

Due to the cyclicity of longitudes, we first define a function $\Delta(\phi_1, \phi_2)$, which is the eastward angle of ϕ_2 from ϕ_1 . For example, $\Delta(10^\circ, 350^\circ) = 340^\circ$ and $\Delta(350^\circ, 10^\circ) = 20^\circ$.

To calculate the interpolated radius r for a given test point with latitude θ and longitude ϕ , we first find the indices k of four radii values (r_1, r_2, r_3, r_4) that surround the point of interest (see Fig. 3):

$$r_1(i = i_1, j = j_1) \tag{2}$$

$$r_2(i = i_2, j = j_1) \tag{3}$$

$$r_3(i = i_1, j = j_2) \tag{4}$$

$$r_4(i = i_2, j = j_2) \tag{5}$$

The regular nature of the DEM grid allows i and j , and hence k , to be quickly calculated as follows:

$$i_1 = \lfloor \Delta(\phi_1, \phi) m_{\text{pix}} + 1 \rfloor \tag{6}$$

$$i_2 = i_1 + 1 \tag{7}$$

$$j_1 = \lfloor (\theta - \theta_1) m_{\text{pix}} + 1 \rfloor \tag{8}$$

$$j_2 = j_1 + 1 \tag{9}$$

where θ_1 and ϕ_1 are the minimum latitude and longitude in the DEM grid, and $\lfloor \rfloor$ indicates truncation to the nearest lowest integer. To allow

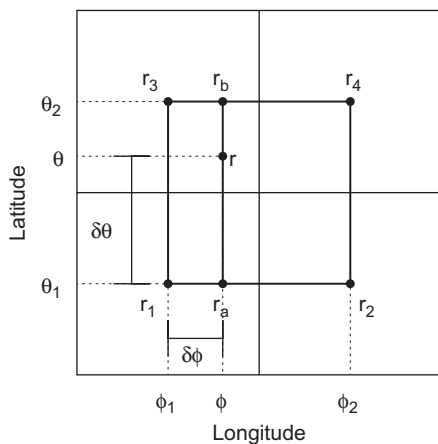


Fig. 3. Radius r at point (ϕ, θ) is determined from bilinear interpolation of radii (r_{1-4}) from the four surrounding pixels. Grid spacing is $1/m_{\text{pix}}$ in both latitude and longitude.

for longitude cyclicity, if $i_2 = n_\phi + 1$ then $i_2 = 1$. The record numbers of the four radii are calculated using Eq. (1).

Note that because the radii are defined for the centre of the DEM pixels there are no radii points exactly coincident with the poles. Therefore, if our test point is within half a pixel width of the pole the average radius at the pole is calculated and used for interpolation.

Therefore, $j_1 = 0$ corresponds to the south pole, so an average radius for the polar point is calculated from all points with $j = 1$ and assigned to r_1 and r_2 . Similarly, $j_2 = n_\theta + 1$ corresponds to the north pole and an average polar radius is calculated from all points with $j = n_\theta$ and assigned to r_3 and r_4 .

To determine the radius r of the test point bilinear interpolation (Press et al., 1992) is used, which is both fast and robust:

$$r_a = r_1 + m_{\text{pix}} \delta\phi (r_2 - r_1) \tag{10}$$

$$r_b = r_3 + m_{\text{pix}} \delta\phi (r_4 - r_3) \tag{11}$$

$$r = r_a + m_{\text{pix}} \delta\theta (r_b - r_a) \tag{12}$$

where r_a and r_b are the interpolated radii at constant latitude between $r_1 - r_2$ and $r_3 - r_4$, respectively.

2.3. Sequential ray stepping to intersect point

The basis of this algorithm is to incrementally step along the viewing vector (line-of-sight). At each increment the distance to the planet centre is calculated. When this distance is less than the interpolated radius from the DEM an intersection has occurred somewhere between the last two test points. The intersection point is refined using successive interpolation and iteration. This is explained in detail below.

From now on vectors are defined by bold lowercase letters (e.g. \mathbf{x}), vector magnitudes are defined by the corresponding italicised lowercase letters (e.g. x), and points in space are defined by uppercase letters (e.g. X).

2.3.1. Defining the viewing geometry

To define where the instrument is looking we require two vectors: one to describe the position of the instrument/spacecraft and another to describe which direction the instrument is pointed. The viewing geometry can be defined in either Cartesian or spherical polar coordinates with the origin coincident with the planet centre. A general

Cartesian position vector (x, y, z) in a right handed Cartesian frame is related to spherical polar coordinates (ϕ, θ, r) by

$$x = r \cos(\theta) \cos(\phi) \quad (13)$$

$$y = r \cos(\theta) \sin(\phi) \quad (14)$$

$$z = r \sin(\theta) \quad (15)$$

where ϕ is the eastward-positive longitude, θ is the latitude, and r is the distance from the planet centre. The x -axis corresponds to $(\phi = 0^\circ, \theta = 0^\circ)$, the y -axis to $(\phi = 90^\circ, \theta = 0^\circ)$, and the z -axis is aligned with the rotation axis pointing due north along the line $\theta = 90^\circ$. Therefore, the spacecraft position vector is defined by $\mathbf{s} = (\phi_s, \theta_s, r_s)$ or $\mathbf{s} = (x_s, y_s, z_s)$.

Now consider an instrument on the spacecraft with a detector element pointing in direction $\mathbf{l} = (x_l, y_l, z_l)$. Here, \mathbf{l} is a unit direction vector and can also be described in spherical polar coordinates as $\mathbf{l} = (\phi_l, \theta_l, 1)$ if it is translated to the origin. We refer to \mathbf{l} as the line-of-sight vector.

Unless we are dealing with an instrument with a single pixel it is also necessary to define the orientation of the detector array. This can be defined by another unit direction vector \mathbf{q} parallel to one side of the array. Each detector pixel is then assigned an individual line-of-sight vector so that we need not assume even pixel spacing in our instrument.

For example, consider a rectangular array with $n_h \times n_v$ elements and pixel spacing δh radians in the detector's horizontal h direction and δv radians in the detector's vertical v direction. If the centre of the field of view has line-of-sight vector \mathbf{l}_0 , and the unit direction vector of the detector's vertical side is \mathbf{q} . The unit direction vectors of the detector array's horizontal \mathbf{h} and vertical \mathbf{v} axes are

$$\mathbf{h} = \frac{\mathbf{l}_0 \wedge \mathbf{q}}{|\mathbf{l}_0 \wedge \mathbf{q}|} \quad (16)$$

$$\mathbf{v} = \mathbf{q} \quad (17)$$

where \wedge represents the vector cross-product and $||$ indicates taking the magnitude of a vector. The line-of-sight vector \mathbf{l}_{ij} of each pixel in the array is defined by

$$\mathbf{l}_{ij} = \mathbf{l}_0 + \sin\left(\left(i - \frac{n_h + 1}{2}\right)\delta h\right)\mathbf{h} + \sin\left(\left(j - \frac{n_v + 1}{2}\right)\delta v\right)\mathbf{v} \quad (18)$$

Note that \mathbf{l}_{ij} is not a unit vector and requires normalising. This expression is used in Section 4.1 to construct synthetic observations with a rectangular array.

2.3.2. Finding the intersection point

The viewing geometry is shown in Fig. 4. First we calculate the limb viewing angle ψ_0 using the dot product of the line-of-sight vector and spacecraft vector:

$$\cos \psi_0 = \frac{-\mathbf{s} \cdot \mathbf{l}}{s} \quad (19)$$

Note that if $\cos \psi_0 < 0$ (i.e. $\psi_0 > 90^\circ$) then the line-of-sight is pointing away from the planet and no tangent point exists in the direction that the instrument is looking.

Second the position vector of the tangent point T is calculated

$$\mathbf{t} = \mathbf{s} + s \cos \psi_0 \mathbf{l} \quad (20)$$

When stepping along the ray to look for an intersection, there is no need to step along the entire ray. This is because we know the maximum planet radius r_{\max} so we only need to step along the portion of the ray over which an intersection is possible. Therefore, we calculate the minimum d_0 and maximum d_m distance along the ray that we need to search:

$$d_0 = s \cos \psi_0 - \sqrt{r_{\max}^2 - t^2} \quad (21)$$

$$d_m = s \cos \psi_0 + \sqrt{r_{\max}^2 - t^2} \quad (22)$$

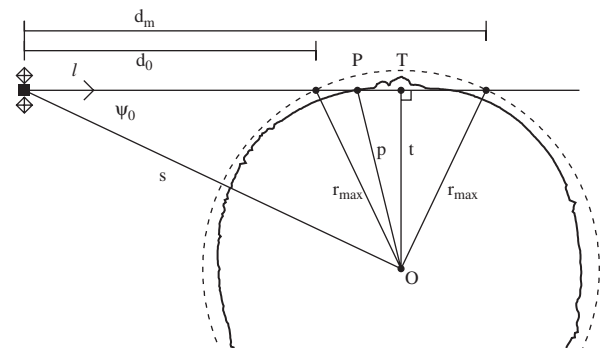


Fig. 4. Definition of viewing geometry parameters for a line-of-sight that intersects the surface: O , origin/centre of planet; r_{\max} , maximum radius of planet; dashed line, circle of radius r_{\max} ; solid undulating line, topography; s = distance from O to spacecraft; \mathbf{l} , line-of-sight vector; d_0 and d_m = minimum and maximum distances over which an intersection is possible; P , intersection point; and T , spherical tangent point.

The step size along the ray is chosen such that it subtends an angle $\delta\alpha$ at the planet centre equal to half the DEM grid spacing to ensure Nyquist sampling of the topography:

$$\delta\alpha = \frac{\cos\theta}{2m_{\text{pix}}} \quad (23)$$

The factor of $\cos\theta$ ensures that sampling is fine enough near the poles, where the great circle spacing of the longitude nodes scales as $\cos\theta$. Note that to prevent $\delta\alpha$ collapsing to zero at the poles the minimum and maximum allowable values of θ in this equation are θ_1 and θ_{n_0} . If smoothing has been applied to the DEM such that topography has a constant great circle angular resolution, the $\cos\theta$ factor can be omitted. This gives faster operation of the algorithm over the poles. Note that $\delta\alpha$ determines the number of steps that must be taken to reach the intersect point. Hence the computation time scales linearly with the grid resolution m_{pix} .

During the ray stepping we specify test points \mathbf{g} along the ray, at which to test for intersections with the surface. The position vector of the first test point \mathbf{g}_0 is defined by

$$\mathbf{g}_0 = \mathbf{s} + d_0\mathbf{l} \quad (24)$$

Subsequent test points are generated by using the previous test point and the angular step size $\delta\alpha$ to calculate the distance moved along the ray δg from \mathbf{g}_i to \mathbf{g}_{i+1} . From Fig. 5a it can be seen that

$$\psi_{i+1} = \psi_i + \delta\alpha \quad (25)$$

The step size δg along the ray is then found using the sine rule:

$$\delta g = \frac{\sin(\delta\alpha)g_i}{\sin(\pi - \psi_{i+1})} \quad (26)$$

Therefore

$$d_{i+1} = d_i + \delta g \quad (27)$$

and

$$\mathbf{g}_{i+1} = \mathbf{s} + d_{i+1}\mathbf{l} \quad (28)$$

Because \mathbf{g}_0 , ψ_0 , and d_0 are known we can find \mathbf{g}_i for any i by sequential stepping.

To determine if an intersection has occurred between test points \mathbf{g}_i and \mathbf{g}_{i+1} , we calculate the latitude and longitude of each test point

$$\theta_{g_i} = \sin^{-1}\left(\frac{g_{i_z}}{g_i}\right) \quad (29)$$

$$\phi_{g_i} = \tan^{-1}\left(\frac{g_{i_y}}{g_{i_x}}\right) \quad (30)$$

and calculate the radius r_i at this point from the DEM using the method in Section 2.2. We continue stepping along the ray until \mathbf{g}_{i+1} is below the surface (i.e. $g_{i+1} < r_{i+1}$) or we reach the maximum distance along the line-of-sight vector d_m , in which case no intersection has occurred and the observation is of the planet limb.

If an intersect does occur we assume the topography is linear between the test points and

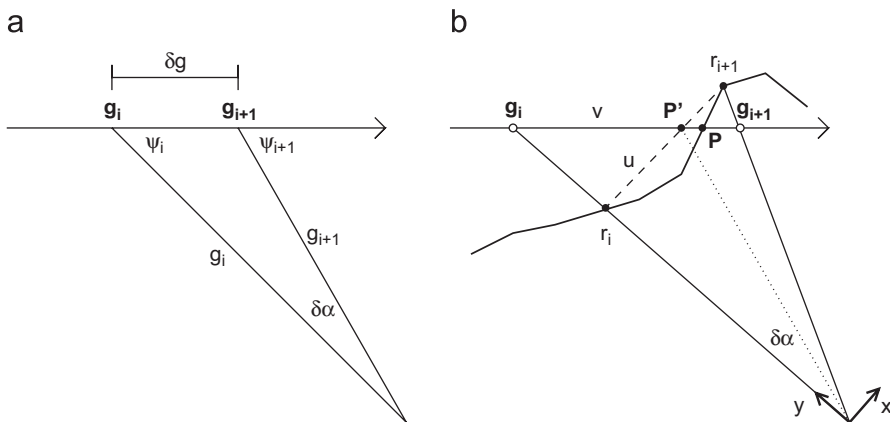


Fig. 5. Stepping along a ray to look for an intersection. (a) Definition of test points \mathbf{g} along line-of-sight and associated angles/distances. (b) An intersection has occurred between \mathbf{g}_i and \mathbf{g}_{i+1} . P' is interpolated estimate of intersect point, P is the actual intersect point and is found by iteration, v and u are the fractional distances to P' along $\mathbf{g}_i\mathbf{g}_{i+1}$ and $r_i\mathbf{r}_{i+1}$, respectively. x and y indicate perpendicular directions used in text when solving for v .

interpolate to find the predicted intersection point P' (see Fig. 5b). At P'

$$\mathbf{g}_i + v(\mathbf{g}_{i+1} - \mathbf{g}_i) = \mathbf{r}_i + u(\mathbf{r}_{i+1} - \mathbf{r}_i) \quad (31)$$

where v and u are the fractions moved along the vectors $\mathbf{g}_{i+1} - \mathbf{g}_i$ and $\mathbf{r}_{i+1} - \mathbf{r}_i$, respectively. Separating this vector equation into two perpendicular components (x and y shown in Fig. 5b) in the plane containing the ray and the origin gives

$$vg_{i+1} \sin \delta\alpha = ur_{i+1} \sin \delta\alpha \quad (32)$$

$$g_i + v(g_{i+1} \cos \delta\alpha - g_i) = r_i + u(r_{i+1} \cos \delta\alpha - r_i). \quad (33)$$

Solving these simultaneous equations for v gives

$$v = \frac{r_{i+1}(r_i - g_i)}{g_{i+1}r_i - g_i r_{i+1}} \quad (34)$$

Therefore, the predicted intersection point is

$$\mathbf{p}' = \mathbf{g}_i + v\delta g \mathbf{l} \quad (35)$$

The planet radius $r_{p'}$ at this point is then interpolated from the DEM and compared to p' .

If $r_{p'} > p'$ then the interpolated point is below the surface. In this case the interpolation is repeated with

$$\mathbf{g}_{i+1} = \mathbf{p}' \quad (36)$$

$$\mathbf{r}_{i+1} = \mathbf{r}_{p'} \quad (37)$$

$$\delta g = v\delta g \quad (38)$$

If $r_{p'} < p'$ then the interpolated point is above the surface. In this case the interpolation is repeated with

$$\mathbf{g}_i = \mathbf{p}' \quad (39)$$

$$\mathbf{r}_i = \mathbf{r}_{p'} \quad (40)$$

$$\delta g = (1 - v)\delta g \quad (41)$$

The interpolation and refinement of the test points is repeated until $r_{p'} = p'$ to within a specified tolerance. Then $\mathbf{p} = \mathbf{p}'$ and we have found the position vector of the intersection point. This process only takes one or two iterations because the angular step size is less than the DEM grid spacing so the assumption of linear topography between test points is accurate after one or two steps.

2.4. Properties of intersection point

From the position vector \mathbf{p} of the intersection point we can calculate all the required properties. The latitude θ_p , longitude ϕ_p , radius r_p at P are given by

$$\theta_p = \sin^{-1}\left(\frac{z_p}{p}\right) \quad (42)$$

$$\phi_p = \tan^{-1}\left(\frac{y_p}{x_p}\right) \quad (43)$$

$$r_p = p \quad (44)$$

For atmospheric studies the emission angle ε at the surface is important as it defines the path length through the atmosphere. The emission angle is defined by the angle between the line-of-sight vector and the normal \mathbf{n} to the surface of the best fitting oblate spheroid at (ϕ_p, θ_p)

$$\varepsilon = \cos^{-1}(-\mathbf{n} \cdot \mathbf{l}) \quad (45)$$

The normal \mathbf{n} is calculated using the equations for an ellipsoid. The surface of an ellipsoid with equatorial radius r_{eq} and polar radius r_{pol} is defined by

$$\frac{x^2}{r_{\text{eq}}^2} + \frac{y^2}{r_{\text{eq}}^2} + \frac{z^2}{r_{\text{pol}}^2} = 1 \quad (46)$$

The oblateness f and eccentricity e of an oblate spheroid are defined by

$$f = \frac{r_{\text{eq}} - r_{\text{pol}}}{r_{\text{eq}}} \quad (47)$$

$$e = \sqrt{\frac{r_{\text{eq}}^2 - r_{\text{pol}}^2}{r_{\text{eq}}^2}} \quad (48)$$

These quantities are related by $e^2 = f(2 - f)$. Rearranging Eq. (46) using $r_{\text{pol}}^2 = r_{\text{eq}}^2(1 - e^2)$ from Eq. (47) gives

$$x^2 + y^2 + \frac{z^2}{1 - e^2} = r_{\text{eq}}^2 \quad (49)$$

A general surface defined in 3D by function $F(x, y, z) = 0$ has surface normal $\mathbf{n} = \nabla F$ (Boas, 1983). Therefore, the normal \mathbf{n} to the ellipsoid surface at P is

$$\mathbf{n} = \left(x_p, y_p, \frac{z_p}{1 - e^2}\right) \quad (50)$$

Other quantities such as the solar azimuth can also be calculated from \mathbf{n} and \mathbf{p} if the position vector of the sun is known.

2.4.1. Tangent point for an oblate spheroid

If the line-of-sight does not intersect the planet's surface then we have a view of the planet's limb. If we are only interested in the solid surface these data can be ignored. However, if we are interested in the planet's atmosphere these measurements are important and it is necessary to determine the latitude, longitude, and altitude of the tangent point. If the planet is an oblate spheroid, as most planets large enough to have an atmosphere are, then the tangent

point occurs at a different position to that for a sphere.

At the tangent point T_c of a ray to an oblate spheroid (or ellipsoid) the ray does not subtend an angle of 90° to the radius line. Instead it subtends an angle of 90° with the surface normal. This is illustrated in Fig. 6. In addition to T_c , we require the point on the planet's surface S_c directly below T_c .

With reference to Fig. 6a, the surface normal \mathbf{n}_c from S_c to T_c is given by

$$\mathbf{n}_c = -\mathbf{s}_c + \mathbf{s} + a\mathbf{l} \quad (51)$$

where a is the distance from the spacecraft to T_c along the line-of-sight. The surface normal and the line-of-sight are perpendicular so the dot product is zero:

$$\mathbf{l} \cdot (-\mathbf{s}_c + \mathbf{s} + a\mathbf{l}) = 0 \quad (52)$$

Therefore, rearranging and using $\mathbf{l} \cdot \mathbf{l} = 1$ gives

$$a = \mathbf{l} \cdot (\mathbf{s}_c - \mathbf{s}) \quad (53)$$

We now apply a linear stretch to the z -axis such that the ellipsoid is transformed into a sphere (Fig. 6b). The scale factor w is given by

$$w = \frac{r_{\text{eq}}}{r_{\text{pol}}} = \frac{1}{1-f} \quad (54)$$

All vectors have their z -components scaled by w and are indicated with a “dash”. Note that T_c is *not* stretched—instead T'_c is defined by the intersection between the projection of \mathbf{s}'_c and the stretched line-of-sight. The spherical tangent point T'_c is then

$$\mathbf{t}'_c = \mathbf{s}' + |\mathbf{s}' \cdot \hat{\mathbf{l}}| \hat{\mathbf{l}}' \quad (55)$$

where the hat on $\hat{\mathbf{l}}'$ indicates re-normalisation to unit length. The point directly below the tangent point in the stretched frame is

$$\mathbf{s}'_c = \mathbf{t}'_c \frac{r_{\text{eq}}}{|\mathbf{t}'_c|} \quad (56)$$

Scaling back to geographic coordinates by multiplying the z -components by $1/w$ gives \mathbf{s}_c and hence a by using Eq. (53). \mathbf{t}_c is then given by

$$\mathbf{t}_c = \mathbf{s} + a\mathbf{l} \quad (57)$$

and the perpendicular distance between the line-of-sight and the ellipsoid surface d is

$$d = |\mathbf{t}_c - \mathbf{s}_c| \quad (58)$$

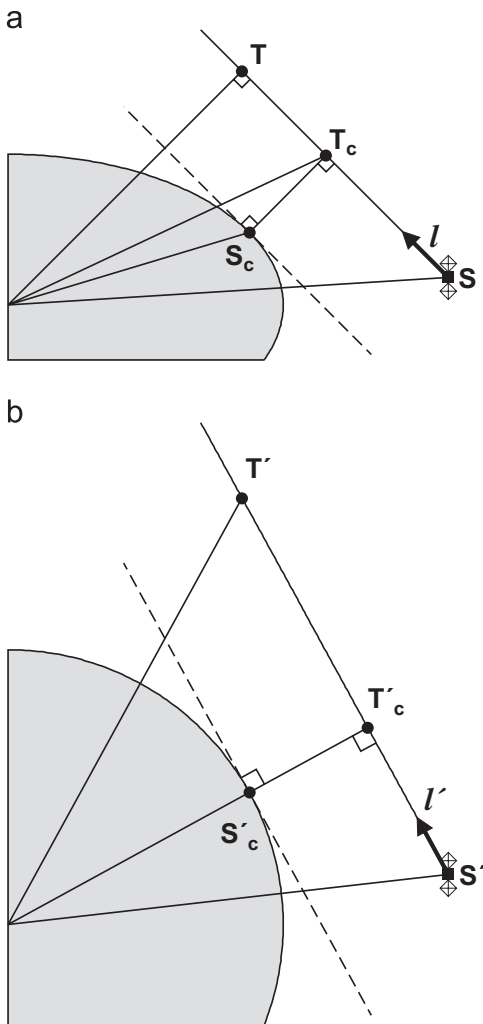


Fig. 6. (a) Tangent point of an oblate spheroidal planet (grey area). For a sphere the tangent point would occur at T , but for an oblate spheroid the tangent point occurs at point T_c such that line-of-sight vector is perpendicular to surface normal. S_c is point on surface directly below T_c . (b) z -axis has been scaled such that the ellipsoid is transformed into a sphere, which allows the scaled tangent point T'_c to be determined analytically.

The longitude and latitude of S_c (position vector $\mathbf{s}_c = (x_c, y_c, z_c)$) are

$$\phi_c = \sin^{-1} \left(\frac{z_c}{\sqrt{x_c^2 + y_c^2 + z_c^2}} \right) \quad (59)$$

$$\theta_c = \tan^{-1} \left(\frac{y_c}{x_c} \right) \quad (60)$$

To find the radius r_c of the spheroid at (ϕ_c, θ_c) substitute $x = r_c \cos \theta_c \cos \phi_c$, $y = r_c \cos \theta_c \sin \phi_c$, and $z = r_c \sin \theta_c$ into Eq. (46) to give

$$r_c = \sqrt{\frac{r_{\text{eq}}^2 (1 - e^2)}{1 - e^2 \cos^2 \theta_c}} \quad (61)$$

The altitude of the tangent point T_{calt} above the DEM surface is then given by the perpendicular height of the ellipsoid above the DEM plus the perpendicular distance between the line-of-sight and ellipsoid

$$T_{\text{calt}} = (r_c - r(\phi_c, \theta_c)) \cos \gamma + d \quad (62)$$

where $r(\phi_c, \theta_c)$ is the planet radius at (ϕ_c, θ_c) interpolated from the DEM, γ is the angle between \mathbf{n}_c and \mathbf{s}_c , and d is the distance between the line-of-sight and the oblate spheroid from Eq. (58). Using the dot product this can be rewritten as

$$T_{\text{calt}} = (r_c - r(\phi_c, \theta_c)) \frac{\mathbf{s}_c \cdot \mathbf{n}_c}{s_c n_c} + d \quad (63)$$

3. Code description

The algorithm described in the previous section has been implemented in FORTRAN77. The code has been compiled and run successfully using the Intel Fortran Compiler and the G77 (GNU) compiler.

The main subroutines are summarised in Table 2—the most important of which is `zcalc_planet_intersect.f`. The inputs for this routine are arrays of spacecraft positions, line-of-sight vectors, the location of the DEM, and the quantities in Table 1. For an image with N pixels taken while the spacecraft was at a single location, there would be N identical spacecraft position vectors and N line-of-sight vectors—one for each pixel. The outputs are the position vectors \mathbf{p} , \mathbf{t} , \mathbf{t}_c , \mathbf{s}_c , along with the emission angles.

Table 2
Summary of main subroutines

Routine name	Brief description
<code>zcalc_planet_intersect.f</code>	Main routine
<code>zdem_open.f</code>	Opens and checks DEM
<code>zdem_read.f</code>	Reads entire DEM (used if <code>dem_read_mode=0</code>)
<code>zdem_radius.f</code>	Interpolates DEM at given lat/lon (used if <code>dem_read_mode=0</code>)
<code>zdem_radius_increment.f</code>	Interpolates DEM at given lat/lon using incremental read (used if <code>dem_read_mode=1</code>)
<code>zcrossp.f</code>	Vector cross product
<code>zrotate_about_vector.f</code>	Rotate a vector about another vector by a specified angle
<code>ztangent_ellipsoid.f</code>	Calculated tangent point to an ellipsoid
<code>flondiff.f</code>	The function Δ
<code>fvsep.f</code>	Angle between two vectors

All subroutines are written in FORTRAN77.

A test program `test_planet_intersect.f` is included to illustrate how the subroutines are used.

When running the program there are two options that must be set:

dem_read_mode:

This sets the way that the DEM is read in and can be either 0 or 1:

0: The entire DEM is read in before performing calculations. This offers a run-time saving if there are many pixels or a dense coverage of measurements. This will not be possible if there is insufficient RAM to load the whole DEM at once.

1: The DEM points are read in one at a time as required. This offers a run-time saving if there are only a few measurements. It is also useful if RAM is limited or the DEM is very large.

step_mode:

This determines whether to include the factor of $\cos(\theta)$ in Eq. (23).

0: Omit factor of $\cos(\theta)$. This is appropriate if the DEM has been smoothed to have a constant great circle resolution. The algorithm will operate faster in this mode.

1: Include factor of $\cos(\theta)$. This is needed if the DEM has a non constant resolution and includes finer topographic features at the poles, where grid points bunch up.

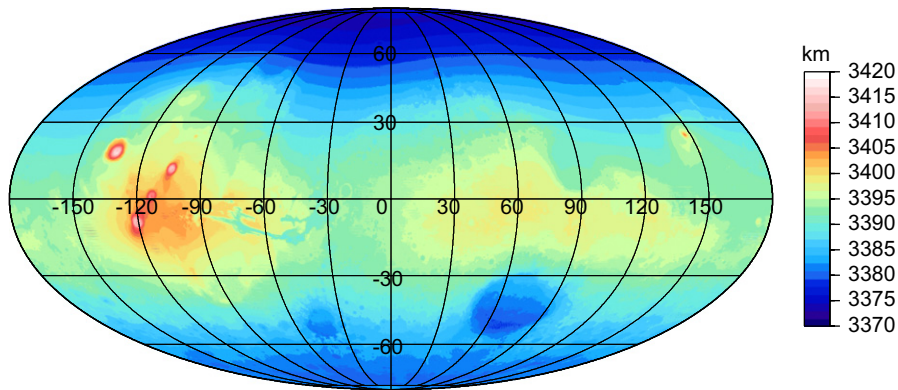


Fig. 7. MOLA DEM used for synthetic examples. Plot is centred on 0°E and 0°N. Scalebar indicates Mars' radius. Olympus Mons volcano used in example 1 is located at -133°E , 18°N .

It is also important to know whether we are observing a large planet or an irregular body. Planets large enough to retain a substantial atmosphere are well described by oblate spheroids. In these cases the equatorial radius r_{eq} and oblateness f must be set in the `.hdr` file to define the oblate spheroid so that T_c can be calculated. For small irregular bodies such as asteroids the calculation of an oblate spheroid tangent point is not relevant. In these cases the oblateness should be set to zero and r_{eq} to unity, which will give $T_c = T$.

4. Examples using MOLA topography data

In this section synthetic and real observations are used to illustrate how the routines could be used. Both test cases are for Mars and we use the topography data from the Mars Orbiter Laser Altimeter (MOLA) on board NASA's Mars Global Surveyor spacecraft (Zuber et al., 1992; Smith et al., 2001), which is available from NASA's PDS Geosciences Node.³ Fig. 7 shows a plot of the MOLA topography in Mollweide global projection (Bugayevskiy and Snyder, 1995). Contouring and plotting uses the Generic Mapping Tools software (Wessel and Smith, 1998).

4.1. Example 1: oblique surface view (synthetic data)

Consider a very simple case where Mars' surface has a temperature $T(z)$, which falls linearly from a

value of T_0 at the mean planetary surface level to $T(z)$ at altitude z above the mean surface level.

$$T(z) = T_0 - \Gamma z \quad (64)$$

where Γ is the "lapse rate" in K/km.

We observe a scene on Mars from orbit with an infrared radiometer, which measures the spectral radiance F (units: $\text{Wm}^{-2}\text{sr}^{-1}\text{m}^{-1}$) emitted by the planet at wavelength λ .

The spectral radiance $F(T, \lambda)$ emitted by a black body at temperature T and wavelength λ is given by the Planck function:

$$F(T, \lambda) = \frac{2hc^2}{\lambda^5(\exp(hc/kT\lambda) - 1)} \quad (65)$$

where h is the Planck constant, k is the Boltzmann constant, and c is the speed of light. This expression was used to generate synthetic radiances to which Gaussian random noise was added to simulate measurement error using the routine `gasdev.f` from Press et al. (1992). Eq. (65) was then rearranged to give T as the subject and used to convert the synthetic radiances into brightness temperatures by assuming that Mars emits as a black body. These brightness temperatures are plotted in Fig. 8a, which shows a scene containing the volcano Olympus Mons observed with a 101×101 pixel array.

One possible aim of analysing such a data set is to relate each of the measured brightness temperatures to locations on the planet's surface and determine the parameters Γ and T_0 from the observed data set. To do this we need to know the latitude, longitude, and surface altitude for each pixel.

This is achieved by using the ray stepping technique described in Section 2.3. Fig. 8b shows

³<http://pds-geosciences.wustl.edu/missions/mgs/mola.html>

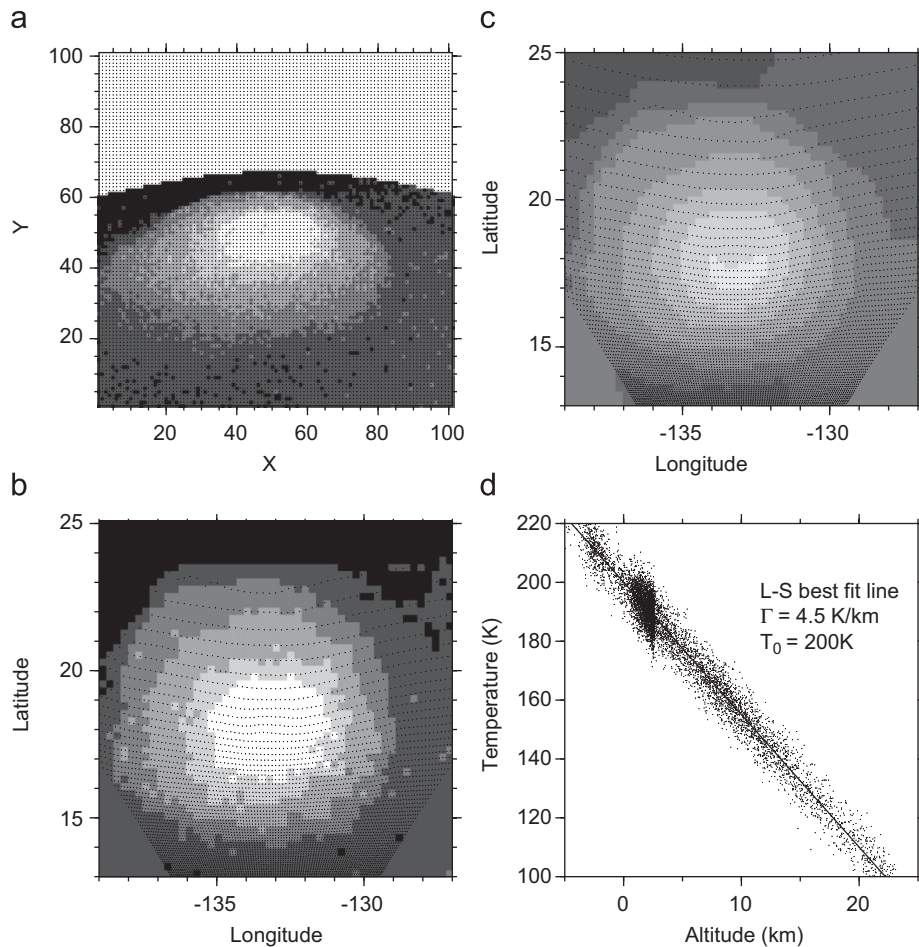


Fig. 8. Synthetic observation of an oblique view of Olympus Mons. (a) Shows brightness temperature measured by a 101×101 pixel detector array with angular pixel size 0.015 rad from a satellite due south of the volcano at an altitude of 325 km above the mean surface. Colder temperatures (lighter colours) occur at greater elevations. Dots show pixel positions. (b) Shows data after re-projection into latitude and longitude coordinates using the ray-stepping technique. This map view is easier to interpret and compare to other data sets. (c) Shows a map view of topography interpolated from the DEM for pixels that intersect with the surface (lighter colours indicate higher topography). (d) Shows basic analysis of transformed data set in order to find lapse rate and T_0 . Determining latitude, longitude, and altitude for each pixel greatly simplifies such analysis.

the measured data converted into a map view in latitude–longitude coordinates. Each pixel also has an altitude assigned to it based on the DEM (Fig. 8c).

Once the pixel positions and altitude have been mapped it becomes possible to perform further analysis on the data. The altitude is plotted against brightness temperature in Fig. 8d, which can be used to determine Γ and T_0 from the gradient and intercept of a least-squares straight line fit. Conversion to a map view also facilitates comparison with additional data sets taken from different view points.

4.2. Example 2: limb view of atmosphere (MCS)

Here we consider observations of Mars' limb made by MCS (McCleese et al., 2007) on-board NASA's Mars Reconnaissance Orbiter (MRO). MRO is in a polar orbit and the MCS instrument observes Mars primarily in a forward limb scanning mode with nine linear arrays of 21 pixels, each covering a different wavelength band. The linear pixel arrays are aligned perpendicular to the horizon and overlap with the atmosphere and surface so that atmospheric properties can be determined as a function of altitude. Fig. 9a shows MOLA

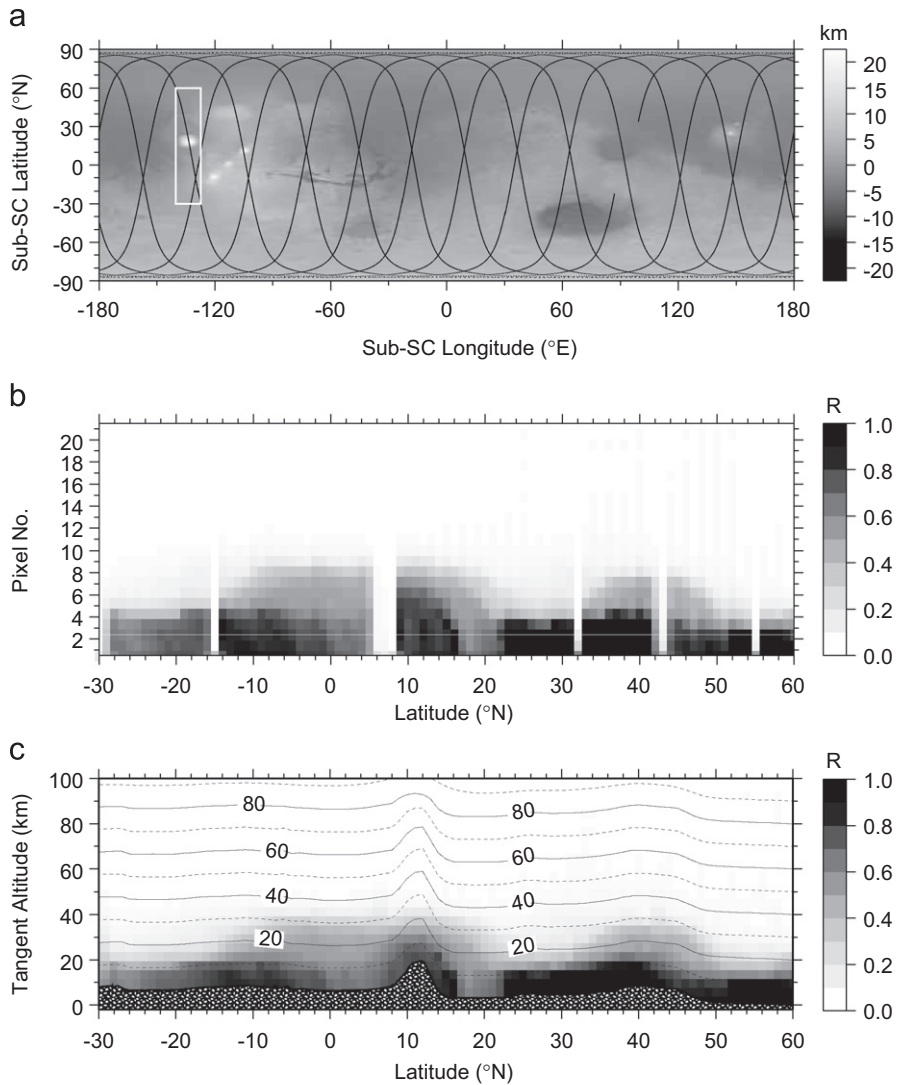


Fig. 9. Observations of Mars’ limb taken by Mars Climate Sounder instrument. (a) Martian topography overlain with orbital tracks of Mars Reconnaissance Orbiter. White rectangle shows orbital segment used in this example, where MRO passed directly over Olympus Mons volcano. (b) Measured normalised radiance R as a function of latitude and pixel number. No pointing geometry has been applied so it is not clear if features are due to pointing, topography, or atmosphere. (c) Ray stepping algorithm has been applied and radiances are now plotted against tangent altitude above the Mars reference ellipsoid. Contours are altitude of oblate tangent point above local surface in kilometres. Pixels that intersect with surface are not plotted. Patterned area shows along-track topography variation for comparison. The information obtained about each pixel from the ray stepping technique is essential for further analysis such as measuring atmospheric temperature as a function of altitude.

topography overlaid with the orbital tracks of MRO for a 24 h period.

The measured (normalised) radiances are shown in Fig. 9b as a function of latitude and pixel number. In this raw data it is difficult to assess whether the observed features in the radiance are due to pointing, topography, or the atmosphere. In Fig. 9c the ray stepping technique has been used to select pixels that only sample the atmosphere and

determine the altitude of the tangent point above the local surface for each of these pixels. The tangent altitude information can be used to assign altitudes to atmospheric properties derived from the measured radiances—resulting in vertical profiles of temperature and cloud density for example. This technique was used by Teanby et al. (2007) to obtain vertical profiles of temperature and dust in Mars’ atmosphere.

5. Discussion

A ray stepping technique is presented for determining the intersection between a planet's surface and the line-of-sight of pixels in a remotely sensed image. The method requires a DEM for the planet being observed and the instrument viewing geometry.

For lines-of-sight that intersect with the surface the latitude, longitude, altitude, and emission angle at the intersect point are determined. For purely atmospheric paths (i.e. views of the planet limb) the location of the atmospheric tangent point of each image pixel is returned. When calculating the tangent point and emission angle, provision is made for planets that are oblate spheroids. Therefore the method is appropriate for both surface and atmospheric remote sensing applications. However, it is most useful for limb studies of the atmosphere and images of the surface taken at oblique angles, such as whole-body or wide-angle views of irregular asteroids and moons.

Example applications using the MOLA Mars DEM are given to illustrate use of the method for studies of the surface and atmosphere. For surface applications, obliquely viewed images can be transformed into latitude/longitude maps and an altitude and emission angle assigned to each pixel. For atmospheric limb studies, the tangent point altitude of each pixel along with whether or not an intersection with the surface occurs is returned. This greatly aids interpretation of such data sets and permits further analysis.

Processing time scales linearly with the DEM grid resolution so that very high resolution grids do not result in very long run times. A suite a FORTRAN subroutines and test programs to perform these transformations are available from the Computers and Geosciences web site.

Acknowledgements

This research was funded by the UK Science and Technology Facilities Council. The author gratefully acknowledges Charles Backus and an anonymous reviewer for insightfully comments that helped improve the manuscript.

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at doi:[10.1016/j.cageo.2007.11.007](https://doi.org/10.1016/j.cageo.2007.11.007).

References

- Boas, M.L., 1983. *Mathematical Methods in the Physical Sciences*, second ed. Wiley, New York, 793pp.
- Bugayevskiy, L.M., Snyder, J.P., 1995. *Map Projections: A Reference Manual*. Taylor & Francis Ltd., London, 248pp.
- Farr, T.G., Rosen, P.A., Caro, E., Crippen, R., Duren, R., Hensley, S., Kobrick, M., Paller, M., Rodriguez, E., Roth, L., Seal, D., Shaffer, S., Shimada, J., Umland, J., Werner, M., Oskin, M., Burbank, D., Alsdorf, D., 2007. The shuttle radar topography mission. *Reviews of Geophysics* 45, RG2004.
- Ford, P.G., Pettengill, G.H., 1992. Venus topography and kilometer scale slopes. *Journal of Geophysical Research* 97 (E8), 13103–13114.
- McCleese, D.J., Schofield, J.T., Taylor, F.W., Calcutt, S.B., Foote, M.C., Kass, D.M., Leovy, C.B., Paige, D.A., Read, P.L., Zurek, R.W., 2007. Mars Climate Sounder: an investigation of thermal and water vapor structure, dust and condensate distributions in the atmosphere, and energy balance of the polar regions. *Journal of Geophysical Research* 112 (E5), E05S06.
- Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., 1992. *Numerical Recipes*, second ed. Cambridge University Press, Cambridge, UK, 992pp.
- Smith, D.E., Zuber, M.T., Frey, H.V., Garvin, J.B., Head, J.W., Muhleman, D.O., Pettengill, G.H., Phillips, R.J., Solomon, S.C., Zwally, H.J., Banerdt, W.B., Duxbury, T.C., Golombek, M.P., Lemoine, F.G., Neumann, G.A., Rowlands, D.D., Aharonson, O., Ford, P.G., Ivanov, A.B., Johnson, C.L., McGovern, P.J., Abshire, J.B., Afzal, R.S., Sun, X.L., 2001. Mars Orbiter Laser Altimeter: experiment summary after the first year of global mapping of Mars. *Journal of Geophysical Research* 106 (E10), 23689–23722.
- Teanby, N.A., Irwin, P.G.J., Howett, C., Calcutt, S.B., Lolachi, R., Bowles, N.E., Taylor, F.W., Schofield, J.T., Kleinboehl, A., McCleese, D.J., 2007. Vertical profiles of temperature and dust derived from Mars Climate Sounder. *Bulletin of the American Astronomical Society* 39 (3), 492.
- Thomas, P.C., 1993. Gravity, tides, and topography on small satellites and asteroids—application to surface-features of the Martian satellites. *Icarus* 105, 326–344.
- Thomas, P.C., Joseph, J., Carcich, B., Veverka, J., Clark, B.E., Bell, J.F., Byrd, A.W., Chomko, R., Robinson, M., Murchie, S., Prockter, L., Cheng, A., Izenberg, N., Malin, M., Chapman, C., McFadden, L.A., Kirk, R., Gaffey, M., Lucey, P.G., 2002. Eros: shape, topography, and slope processes. *Icarus* 155, 18–37.
- Wessel, P., Smith, W.H.F., 1998. New, improved version of the Generic Mapping Tools released. *EOS Transactions, American Geophysical Union* 79, 579.
- Zuber, M.T., Smith, D.E., 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 (E5), 7781–7797.
- Zuber, M.T., Smith, D.E., Cheng, A.F., Garvin, J.B., Aharonson, O., Cole, T.D., Dunn, P.J., Guo, Y.P., Lemoine, F.G., Neumann, G.A., Rowlands, D.D., Torrence, M.H., 2000. The shape of 433 Eros from the NEAR-Shoemaker laser rangefinder. *Science* 289, 2097–2101.